Economic Impacts of Information Technology

Kevin J. Stiroh
Federal Reserve Bank of New York[1]

GLOSSARY

capital deepening An increase in the amount of productive capital available per unit of labor.
computer productivity paradox Co-existence of slow measured productivity growth and the computer revolution in the United States in the 1980s and early 1990s.
constant-quality price index A price series for a consistently defined set of production characteristics over time; also called a quality-adjusted price deflator.
growth accounting Methodology for analyzing the primary sources of economic growth (capital, labor, and total factor productivity growth).
hedonic function Relationship between the price of a good or service and the quantity of characteristics that product embodies.
information technology (IT) investment Purchases of computer hardware, computer software, and telecommunications equipment by firms to be used as a production input.
labor productivity growth Increases in output per hour worked.
production function Economic relationship between inputs, technology, and output for a firm, industry, or economy.
total factor productivity growth (TFP) Increases in output per all production inputs.

One of the defining features of the United States economy in the 1990s is the massive expenditure on information technology equipment and software. With the United States economy surging over the same period, this raises an obvious and important question: What is the economic impact of information technology? Not surprisingly, there is a large and growing literature on the question and this essay reviews the empirical research done by economists and business analysts. The essay begins with a discussion of how economists think about information technology and measure it using the hedonic theory of prices. The next section reviews the empirical estimates of the impact of information technology, both at the macroeconomic level of the entire United States economy and then at the more microeconomic level of individual firms or industries. In both cases, the evidence is building that information technology is having a substantial economic impact. The final section of the essay discusses some of the conclusions about information technology and the potential impact on the United States economy in the future.

I. INTRODUCTION

Over the last two decades United States firms have pumped more than $3 trillion into information technology (IT) investments, defined here to include purchases of computer hardware, computer software, and telecommunications equipment by United States firms. In 1999 alone, United States investments in these assets totaled $373 billion, which accounted for nearly one-quarter of all United States nominal investment in fixed assets. Clearly, United States firms are making a

[1]The views in this paper are those of the author only and do not necessarily represent those of the Federal Reserve System or the Federal Reserve Bank of New York.

major commitment to IT, presumably in the hope of improving their performance and profitability.

This steady adoption of IT reflects the unprecedented decline in the relative price of computing power and the explosion of IT capability—measured in terms of the gigantic leaps in processing speed, memory, transmission capacity, and storage space, etc. The official United States price data for IT investment goods, for example, show the quality-adjusted price of computer hardware, which measures the price of a fixed amount of computing power, falling nearly 18% per year since 1965—a cumulative decline of 99.8%. In recent years, the declines are even more dramatic with the quality-adjusted prices of computer hardware falling nearly 28% per year from 1995–1999.

Enormous price declines like these are familiar to technologists and economists alike, and reflect the fundamental technical progress in the production of computers, semiconductors, and other high-tech gear—Moore's Law in action. This leads directly to the rapid accumulation of IT investment goods as firms and households respond to relative price changes, substitute between production inputs, and invest heavily in IT assets that are much cheaper now than in the past. Real investment in computer hardware, for example, grew more than 37% per year from 1995–1999, while real investment in software and telecommunications equipment increased 21 and 14%, respectively. These growth rates far outpace those of other investment assets and gross domestic product (GDP), and show the increasing importance of IT to United States businesses and the United States economy.

What is the impact of this technological progress and accumulation of IT assets on the United States economy? Not surprisingly, there has been a sizable amount of economic and business research that addresses this question. The early evidence was typically disappointing and led to a great deal of interest in the "computer productivity paradox," which can be summed up by Nobel Laureate Robert Solow's famous quote: "You can see the computer age everywhere but in the productivity statistics." This research is summarized by a number of excellent reviews that discuss the early evidence and potential explanations by Brynjolfsson and Yang (1996), Sichel (1997), and Triplett (1999). In recent years, however, the United States economy has performed much better and many economists have begun to believe that IT is now having a substantial impact on the United States economy.

Beginning in the mid-1990s, the United States economy experienced a surge in labor productivity (defined as output per hour) growth, a critical measure of the success of an economy that helps determine long-run economic growth and living standards. For the years 1996–1999, labor productivity for the United States nonfarm business sector grew 2.45% per year, compared to only 1.45% for 1990–1995 and 1.38% for 1980–1989. While a one percentage point increase in labor productivity might seem small at first glance, it is quite significant and, if sustainable, reflects a profound change for the United States economy. Sustained productivity growth at a higher rate raises the potential growth rate for the economy, increases per capita income at a faster pace, and contributes to higher standards of living.

Figure 1 shows the two trends of rapid acceleration of IT investment and the surge in labor productivity in the late 1990s. The simple observation of strong IT investment, falling IT prices, and accelerating productivity growth, of course, does not identify IT as the causal factor in the success of the United States economy.

In 2000, Jorgenson and Stiroh, Oliner and Sichel, and Whelan, however, all concluded that IT is indeed playing a major role in the United States productivity revival. After examining the impact of both the *production* and the *use* of IT, this evidence suggests that IT is having a substantial impact through both channels, a conclusion that stands in sharp contrast to earlier research that typically found only a modest IT impact at the aggregate level.

In terms of IT production the industries that make IT goods, particularly computers and semiconductors, are experiencing fundamental technological progress that is driving down the relative prices of these goods and allowing them to greatly expand their production. Economists often measure this type of technological progress as "total factor productivity growth," which represents the ability to produce an increasing amount of output from the same inputs. The IT-using industries, in turn, respond to these falling prices and are rapidly accumulating IT through the rapid investment mentioned above. This "capital deepening" contributes to labor productivity growth as workers in these industries have more and better capital equipment with which to work.

Both the technological progress in the production of IT and the capital deepening from the use of IT have contributed to improved labor productivity for the United States economy as a whole. While estimates vary and some debate remains about the specific mechanism, a consensus is emerging that IT has played an important role in the recent success of the United States economy. Moreover, IT-related components, notably processing chips, are increasingly incorporated into a wide variety of other business, government, and

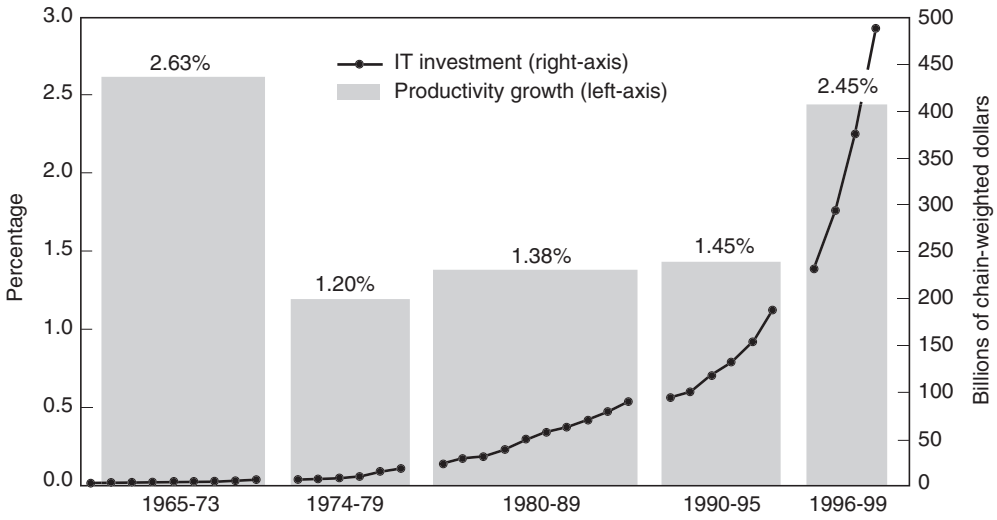


Figure 1 U.S. labor productivity growth and real IT investment, 1965–1999. [Productivity is annual average growth rate for the nonfarm business sector from the Bureau of Labor Statistics (BLS), August 2000. IT investment is from the Bureau of Economic Analysis (BEA) National Income and Product Accounts (NIPA), August 2000.]

consumer goods that directly increase the economic impact and consumer welfare effects of IT.

There is also a large body of microeconomic evidence that examines the impact of IT on the individual firms and industries that are making these massive investments. A consensus is also building there that suggests IT is having a substantial impact on the productivity and performance of the IT-using firms. In 2000, Brynjolfsson and Hitt show this is particularly true when the firms make complementary innovations, e.g., retraining the labor force or reorganizing the production structure, that allow them to fully realize the gains from IT. Some argue that these complementary investments are the critical part of the successful implementation of IT-related production techniques.

The strong recent performance of the United States economy and apparent success of some IT-related firms and industries has also led some observers to argue that IT is changing the way businesses operate in more fundamental ways that go beyond traditional economic analyses. By pointing out the potential effects of network externalities, production spillovers, and the benefits of lower information costs, these proponents claim there is a "new economy" driven in large part by IT. One must be careful, however, not to overstate the economic impact of IT. In many ways, old economic principles still apply and the recent behavior of the United States economy is readily explained by traditional methods. Technological progress, relative price changes, and capital deepening effects have always played a large part in determining the success and growth of an economy, and IT is now another channel through which these familiar forces apply.

II. HOW ECONOMISTS MEASURE IT

A critical first step in understanding the economic impact of IT is to successfully quantify and measure IT in a way that is consistent with both economic theory and the available data. The appropriate theory is a production function that relates output to various production inputs, e.g., labor, physical capital, and purchased materials, and the level of technology. In this framework, IT represents both the *output* of those firms and industries that produce IT goods and a capital *input* into the production process of other firms and industries that use IT. In both cases, the U.S. National Income and Product Accounts (NIPA), which measure the United States GDP, provide the necessary data to begin an analysis of the economic impact of IT at the aggregate level.

This essay focuses on IT investment goods, defined here to include computer hardware, computer software, and telecommunications equipment. All three IT assets are investment goods that are produced by one firm and sold to other firms as a long-lasting production asset; they are a component of the investment category of the NIPA. This captures a large portion of IT-related assets, but not everything that might be considered IT. Some IT-related goods like semiconductors are primarily sold as intermediate goods and

incorporated into a wide range of other products, e.g., the processing chip in an automobile or machine tool. While these goods embody much of the same technology and form an important part of the broader IT revolution, this review focuses on the three investment assets typically designated as IT by economists.

As a second caveat, IT goods are also becoming an increasingly important type of consumption good for United States consumers and investment good for the government sector. For consumers, purchases of home personal computers, personal digital assistants, and wireless telephony are clearly important aspects of the IT revolution. Similarly, the United States government devotes considerable resources to IT investment to provide improved government services. This review focuses on the private United States business economy, so the impact of IT on consumers and the government is not discussed in detail.

A. IT in the National Accounts

The official statistics for the United States economy, the NIPA maintained by the Bureau of Economic Analysis (BEA) of the Department of Commerce, contain data on United States private business investment in a variety of equipment asset and structures. The NIPA investment series are published quarterly and include data for computer hardware, computer software, and telecommunications equipment. These three IT classes are all actually aggregates of more detailed investment data; computer hardware, for example, is an aggregate of several subcomponents (personal computers, mainframes, storage devices, printers, and monitors, and other computer peripheral equipment, not elsewhere classified) that comprise the "computer and peripheral equipment" category in the NIPA. Most economic studies use the level of IT investment aggregation easily available in the NIPA data.

For each investment asset, the NIPA report information on the dollar value of expenditures (called nominal or current dollars), the price (called the price index or price deflator), and the quantity of investment (called real, constant, or chain-weighted dollars). As a matter of definition, real investment equals nominal investment divided by the corresponding price index for individual assets.

The nominal value of investment in IT assets is collected by BEA from various surveys and is relatively straightforward to obtain and analyze. These data for investment goods typically come from surveys by the U.S. Census Bureau, but also reflect a variety of other sources. For IT investment goods, in particular, some data sources are a bit different. For example, the current dollar investment for software are also estimated from the input-output tables maintained by BEA, the Bureau of Labor Statistics (BLS) employment data, and other sources.

The construction of a consistent time series of prices and real investment is much more difficult from the economic perspective. A dollar spent today on computer hardware, for example, provides considerably more real computing power than a dollar spent last year, so a simple comparison of dollar expenditures on IT over time would be quite misleading. This type of comparison also implies that real computing power is becoming much cheaper over time since a given dollar of expenditure yields more computing power today than in the past.

To deal with this difficulty, the NIPA and economists employ "constant-quality price indexes," which measure the price of a common set of productive characteristics over time and translate observed nominal dollar expenditures into comparable estimates of real investment, e.g., real computing power for computer hardware. This effectively captures the enormous quality improvements across successive generations of IT assets and treats these quality gains as a reduction in the price of IT. For example, by dividing the observed series of nominal hardware investment by the corresponding constant-quality price index, one obtains a consistent time series that represents real investment in computing power, measured in "constant-quality efficiency units," each year. As a concrete example, if nominal investment increases 10% between two years while the constant-quality price index falls 20%, the NIPA treat this as a 30% increase in real investment.

These steady quality gains largely reflect familiar technological forces like Moore's Law—the doubling of the power of a computer chip every 18 months or so—that are such a dominant feature of IT. Since the performance and capabilities of recent models of computers easily surpass those considered state-of-the-art just a few years earlier, this type of adjustment is critical to developing an accurate assessment of the economic impact of IT. Moreover, since IT investment is an increasingly important part of the United States GDP, accurate measurement of IT improves the statistical system and helps us to better understand the production capabilities of the United States economy.

B. Measuring Constant-Quality Prices

The BEA first introduced quality-adjusted price indexes for computers and peripherals in December

1985, drawing on joint work it had done with IBM by Cole et al. (1986). In 1996, BEA introduced the "chain-weighting" concept for measuring real output, which reduced the bias resulting from prices that rapidly change relative to their base-year values. This helps to more accurately gauge the contribution of any investment asset to output growth. More recently, BEA incorporated quality-adjusted price information from the Producer Price Index (PPI) program at the BLS for various types of computer hardware and software investment goods.

The constant-quality price indexes were originally implemented based on a hedonic function, defined as "a relation between prices of varieties or models of heterogeneous goods—or services—and the quantities of characteristics contained in them" (Triplett, 1989, pg. 128)." The hedonic theory of prices provides one way to effectively account for the enormous quality change over time and to construct a constant-quality price deflator by measuring the price change associated with a particular set of productive characteristics. For example, if a personal computer sold for the same price in two adjacent years but contained more memory and a larger hard drive in the second year, then a constant-quality price index would measure this as a decline in the price of real computing power.

A simple specification for the hedonic approach would be a regression equation like:

$$\ln P_{i,t} = \sum_{j} \beta_j c_{j,i,t} + \sum_{t} \delta_t D_t + \epsilon_t \qquad (1)$$

where $P_{i,t}$ is the price of computer i at time t, $c_{j,i,t}$ are the j relevant characteristics of that computer observation, and D_t are a set of dummy variables for each period in the sample.

Intuitively, the hedonic approach does not focus on the price of an individual computer unit like a personal computer or a mainframe system ($P_{i,t}$), but rather on the implicit prices (β_j) of the component characteristics ($c_{j,i,t}$) that users demand and value. The hedonic function for desktop computers employed by the BLS in 1998, for example, included productive characteristics like main processor speed (CPU in megaherz), hard-drive size (in gigabytes), amount of Sdram (in megabytes), and dummy variables for the inclusion of alternative storage devices, digital video disks, speakers, fax modem, etc.

By comparing changes in the observed prices of computers and the quantity of various productive characteristics over time, one can construct a constant-quality price index for a fixed set of characteristics from the implicit prices of the included characteristics. More specifically, a constant-quality price index can be calculated by taking the antilogs of the estimated δ coefficients in Eq. (1) to estimate how the price of a fixed set of characteristics varies over time.

Alternatively, one could use the implicit prices to adjust for the different amount of characteristics across models and impute prices for those not produced in a given year. This latter approach is similar to that actually employed by the BLS PPI program, which uses hedonic estimates to quality-adjust observed prices when the producer indicates a change in the technological characteristics of the product. The basic goal of adjusting for quality, however, is the same and the end result is a constant-quality price index that recognizes changes in the productive characteristics of a given asset over time.

C. Current Methodology for Estimating IT Prices

The U.S. NIPA currently incorporate constant-quality price indexes for several types of IT investment goods. For computer hardware, the price indexes in the NIPA are currently based on PPIs for mainframes, personal computers, storage devices, terminals, and other peripheral equipment that incorporate various quality adjustments. Details on the methodology and source data used to develop price estimates of each type of computer hardware vary across the particular hardware component and time period.

For computer software, several different estimation techniques and data sources are used by BEA for various types of computer software in the NIPA. These different software categories include prepackaged software (nonspecialized use software that is sold or licensed in standardized form), custom software (software that is tailored to the specifications of a business or government unit), and own-account software (in-house expenditures for new or upgraded software for internal use). For prepackaged software, BEA uses a combination of quality-adjusted estimates that are based on the price index for computer hardware, a hedonic index for applications, a matched model index for selected prepackage software, and the PPI for applications software. For own-account software, the price index is based on the input-cost indexes that reflect the changing costs of computer programmers, systems analysts, and intermediate inputs. This approach assumes no change in the productivity of computer programmers and shows considerably less price decline than the quality-adjusted prepackaged indexes. For custom software, the prices index is

defined as the weighted average of the prepackaged and own-account indexes.

Finally, for telecommunications equipment, quality-adjusted price indexes based on hedonic estimates are incorporated only for telephone switching equipment and cellular phones. Conventional price deflators are employed for transmission gear and other components.

D. Data on IT Prices and Quantities

The next step is to examine the data on investment in the three IT assets: computer hardware, software, and communications equipment. All data discussed here come directly from the U.S. NIPA and are available from the BEA web site at http://www.bea.doc.gov. Figure 2 shows the official investment deflators for each IT asset, while Fig. 3 shows the real quantity of investment, measured in chain-weighted 1996 dollars. Both figures report data from 1965–1999.

Figure 2 clearly shows the enormous decline in the quality-adjusted price of computer hardware. From 1965–1999, for example, the price of computer hardware fell 17.9% per year; the more recent period, 1995–1999, is even more remarkable with an average annual price decline of 27.7% per year. As a comparison, the GDP deflator, the price of all GDP output, rose 1.6% per year from 1995–1999. This dramatic relative price change reflects the massive technical progress in the production of computers and the steady increase in computing power that is captured by the constant-quality price methodology currently employed by BEA.

The prices of computer software and telecommunications equipment, in contrast, show a much more modest annual price increase of 1.1 and 2.1% for 1965–1999, respectively. For 1995–1999, prices declined by 1.6 and 1.8% per year. These relatively slow price declines may reflect differences in the underlying rate of technical progress in the production of software and telecommunications equipment, or it may reflect practical difficulties in the construction of the price indexes for these assets. As discussed above, only portions of the nominal investment in software and telecommunications equipment are currently associated with constant-quality deflators. Some have argued that these methodological differences are quite important and that the official price indices may understate the true decline in cost of purchasing software and communications power.

Figure 3 shows real investment in the three IT assets, measured in 1996 chain-weighted dollars and highlights the dramatic acceleration in real IT investment during the 1990s. From 1995–1999, real investment in computer hardware, software, and telecommunications equipment grew 37.1, 20.8, and 14.3%, respectively, compared to 4.1% for United States GDP.

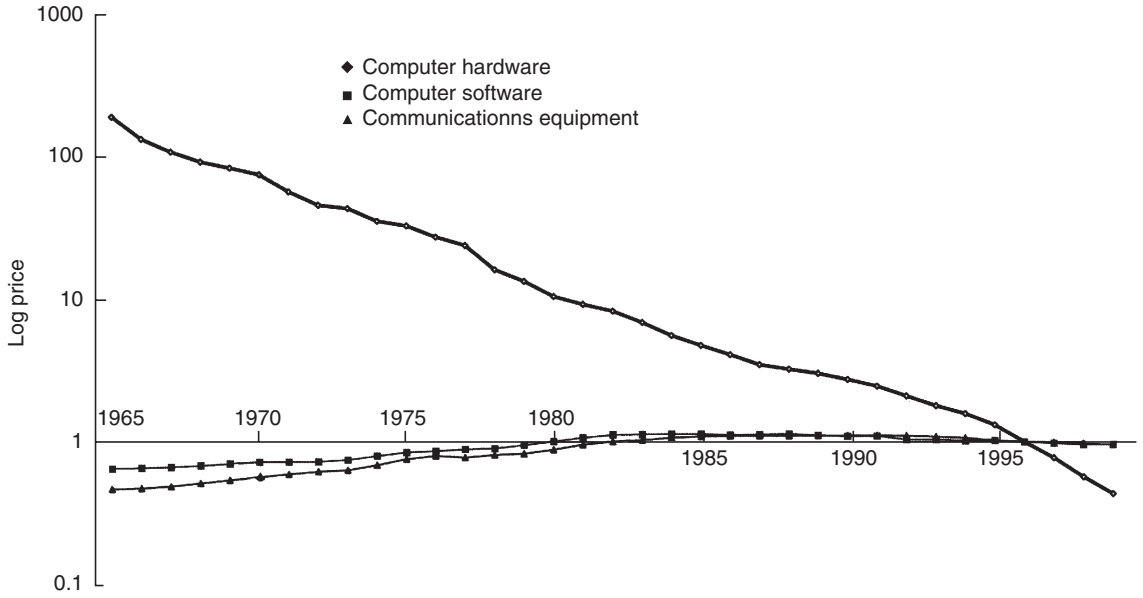


Figure 2 Prices of IT investment, 1965–1999. [From the Bureau of Economic Analysis (BEA) National Income and Product Accounts (NIPA), August 2000.]

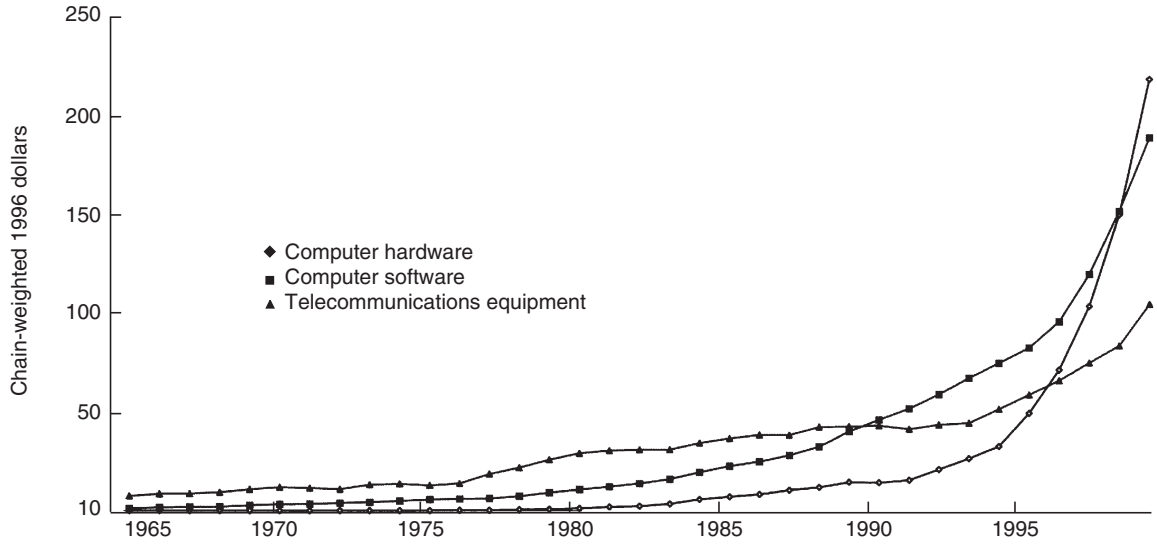


Figure 3 Real IT investment, 1965–1999. [From BEA NIPA, August 2000.]

Moreover, if the official prices are in fact understating the quality change and relative price declines of software and communications equipment, these real investment series are also understated. Simulations by Jorgenson and Stiroh that incorporate alternative deflators with larger quality adjustments for software and telecommunications equipment suggest this could have a sizable impact on measured output growth.

This brief review discusses the methodology currently employed by BEA to measure IT investment and reports current estimates on the price and quantity of IT investment. Since investment is an important component of GDP, this type of analysis is crucial to understanding the role of IT in the United States economy. Moreover, IT investment data provide the critical building block needed to measure the economic impact of IT in a production function analysis.

III. QUANTIFYING THE ECONOMIC IMPACT OF IT

This section turns the discussion to the measurement of the impact of IT on the United States economy. As mentioned above, IT has distinct roles as an output and an input. The economic impact through both channels can be measured with a standard "production function" analysis. This framework can be used at the firm, industry, or aggregate level, and has a long and successful history in economic analysis as a tool for studying the sources of economic growth and productivity.

A. IT and the Aggregate Production Function

An appropriate starting point is a brief discussion of an "aggregate production function," which relates the amount of output an economy produces to the amount of inputs available for production and the level of technology. This approach has long been a tool for understanding the historical sources of economic growth and for projecting the potential growth of an economy in the future.

The basic production function models a relationship between an economy's outputs, Y, and the primary factors of production that include capital, K, and labor, L, and the level of technology A:

$$Y = A \cdot f(K,L) \tag{2}$$

where Y is a concept similar to GDP that measures the production of goods and services in the portion of the economy being examined; K is the service flow from the productive capital stock of equipment and structures, as well as land and inventories; and L is a measure of labor input that represents both the quality of the labor force and the number of hours worked. A is often referred to as total factor productivity (TFP), and measures how effectively inputs are transformed into outputs. An increase in TFP allows more output to be produced from the same inputs.

To quantify the impact of IT, one must quantify the two roles for IT identified above. First, IT investment is an output of the firms and industries that produce IT and thus is a part of Y, which can be decomposed

into an IT portion and a non-IT portion. Second, IT investment is an input that adds directly to the productive capital of firms and industries that use IT. Therefore, K can also be decomposed into an IT portion and a non-IT portion. More specifically, Eq. (2) can be rewritten in an extended form to highlight the role of IT:

$$Y(Y_n, I_c, I_s, I_m) = A \cdot f(K_n, K_c, K_s, K_m, L) \quad (3)$$

where Y_n is real non-IT output, I_c is real computer hardware investment, I_s is real software investment, I_m is real telecommunications equipment investment, K_n is non-IT capital services, K_c is computer hardware capital services, K_s is computer software capital services, and K_m is telecommunications equipment capital services. L and A are defined as in Eq. (2).

Under standard economic assumptions about labor, capital, and output markets, theory suggests that Eq. (3) can be transformed into a growth rate version where the weighted growth rates of various outputs equal the weighted growth rates of inputs plus the growth rate of TFP:

Weighted Growth of Outputs =
Weighted Growth of Inputs + Growth in TFP
or

$$\begin{aligned} w_{Y_n} \cdot \Delta\ln Y_n + w_{I_c} \cdot \Delta\ln I_c + w_{I_s} \cdot \Delta\ln I_s + w_{I_m} \cdot \Delta\ln I_m \\ = v_{K_n} \Delta\ln K_n + v_{K_c} \cdot \Delta\ln K_c + v_{K_s} \cdot \Delta\ln K_s \\ + v_{K_m} \cdot \Delta\ln K_m + v_L \cdot \Delta lnL + \Delta lnA \end{aligned} \quad (4)$$

where w refers to the nominal share of the subscripted output in total output, v refers to the nominal share of the subscripted input in total output, Δ refers to a change or first difference, and the following equalities must hold $w_{Y_n} + w_{I_c} + w_{I_s} + w_{I_m} = v_{K_n} + v_{K_c} + v_{K_s} + v_{K_m} + v_L = 1.0$. The share-weighted growth of each variable is referred to as the "contribution" of that variable.

Equation (4) is a standard "growth accounting" equation, which can identify the contribution of IT outputs and inputs to economic growth. The left-hand side decomposes output growth into the contribution from non-IT output and the various IT components, while the right-hand side identifies the contributions of IT capital inputs, non-IT capital inputs, labor, and TFP growth. Note that TFP growth is calculated as a residual, defined to balance Eq. (4).

At this point it is useful to be precise about what economists mean by technology in the aggregate production function framework. In this model, technology represents the ability to produce more output from the same inputs, measured as TFP growth, $\Delta\ln A$ in Eq. (4). While measured TFP growth is often viewed as a pure indicator of technological progress, it more accurately represents the joint impact of technological change and other important factors like omitted variables such as research and development spending, increasing returns to scale, or output reallocations. Thus, measured TFP growth is best viewed as a proxy for technological progress.

It should also be emphasized that in the production function framework, IT investment is not considered "technology" per se. Rather, IT investment is a capital input that contributes to production as firms make IT-related investments and accumulate capital. Technology, as defined by economists and measured as total factor productivity, however, does factor directly into the production of IT as will be discussed in detail below.

B. Data and Measurement for Production Functions

Successful implementation of this production function approach to measuring the impact of IT requires a substantial amount of IT-related data, most of which can be obtained from the basic NIPA investment data described above. On the output side, one needs the annual growth rate of each type of IT investment and the corresponding share of IT investment in total output. The growth rate of investment can be calculated directly from the chain-weighted investment series in the NIPA that are presented in Fig. 3. The nominal share of IT investment can be easily calculated as the nominal value of IT investment divided by the value of total output produced, essentially nominal GDP.

On the input side, things are a bit more complicated, since one must transform investment into a productive capital stock and estimate the value of the productive services from that stock. Briefly, capital stocks are typically estimated on a net basis by cumulating past investment and subtracting deterioration and retirement as older goods become less useful and are eventually scrapped. That is, the capital stock grows as firms make new investments in equipment and structures, but falls as older goods become less productive or are retired. The growth rate of IT capital is calculated from the time series of this net stock. The value of IT capital services is estimated from economic theory and reflects a variety of factors including depreciation, capital gains (or losses in the case of IT due to falling prices), the opportunity cost of investing, and tax parameters. The value of IT capital services can then be used to calculate the nominal input share of IT by dividing by the value of output.

C. The Macroeconomic Evidence

Before reviewing the empirical results, it is worthwhile to discuss the expected impact of IT in this standard economic model. Since IT is both an output and an input, there should be at least two observable impacts.

On the output side, more rapid IT investment by United States firms suggests more rapid production of IT goods, so one should expect to see a growing contribution of IT as an output in Eq. (4). Moreover, since it is fundamental technological change in the production of IT that is driving down the relative prices, one should also expect to see somewhat faster aggregate TFP growth due to more effective IT production. On the input side, rapid investment leads directly to growth in the capital stock and, therefore, a larger contribution of IT capital inputs in Eq. (4).

Note, however, that the use of IT through investment does not directly affect measured TFP growth in the standard model. If IT is measured correctly, then the productive benefits from using IT will be directly captured by the contribution of IT capital inputs. That is, IT investment may allow firms to produce more output, but it is precisely because they have more productive capital goods, not because they have technological progress in the economic sense described above. If rapid investment is sufficient so capital growth outpaces growth in the labor force, this increases the amount of capital available per worker, "capital deepening," and should contribute directly to rising labor productivity.

Table I reports estimates for the sources of growth analysis described by Eq. (4). This provides information on the rate of economic growth and the contributions of inputs and outputs, and highlights several well-known trends. The United States economy shows relatively slow economic growth during much of the 1970s, 1980s, and early 1990s, followed by a strong pickup in the late 1990s. This pickup reflects both an increase in the contribution of capital inputs from the recent investment boom and faster TFP growth. As discussed next, both trends can be traced in part to IT.

On the output side, the rapid acceleration in the production of IT investment goods can be clearly seen from the growing contribution of each IT asset. When the early 1990s are compared to the late 1990s, for example, the total contribution of the three IT assets nearly doubles from 0.38 percentage points to 0.72 percentage points so more production of IT assets is an important contributor to the faster output growth in recent years. On the input side, the story is similar. The three IT capital inputs show a steady increase in their contribution to United States economic growth. For the most recent period, they contributed 0.75

Table I The Sources of U.S. Economic Growth, 1959–1999

	1959–1998	1959–1973	1973–1990	1990–1995	1995–1998	Preliminary 1995–1999
Growth in output (Y)	3.63	4.33	3.13	2.74	4.73	4.76
Contribution of selected output components						
Other output (Y_n)	3.35	4.18	2.83	2.36	4.01	4.04
Computer investment (I_c)	0.15	0.07	0.16	0.20	0.39	0.39
Software investment (I_s)	0.07	0.03	0.08	0.13	0.21	0.21
Telecommunications investment (I_m)	0.06	0.05	0.06	0.05	0.12	0.13
Contribution of capital services (K)	1.77	2.07	1.62	1.20	2.17	2.33
Other capital (K_n)	1.45	1.89	1.27	0.80	1.42	1.53
Computer capital (K_c)	0.18	0.09	0.20	0.19	0.46	0.49
Software capital (K_s)	0.08	0.03	0.07	0.15	0.19	0.21
Telecommunications capital (K_m)	0.07	0.06	0.08	0.06	0.10	0.11
Contribution of labor (L)	1.23	1.25	1.17	1.18	1.57	1.44
Aggregate total factor productivity (TFP)	0.63	1.01	0.33	0.36	0.99	0.99
Average labor productivity (ALP)	2.04	2.95	1.44	1.37	2.37	2.58

Notes: A contribution of an output and an input is defined as the share-weighted, real growth rate. Other output includes the remaining types of investment goods plus consumption goods. Other capital includes the remaining types of capital input plus consumer durable services.

Adapted from Jorgenson, D. W., and Stiroh, K. J. (2000). Raising the speed limit: U.S. economic growth in the Information Age. *Brookings Papers on Economic Activity 1.* pp. 125–211.

percentage points, which accounted for nearly one-third of the total contribution of all capital assets. Compared to earlier periods, this is a dramatic acceleration in the contribution of IT, particularly for computer hardware, which more than doubled from 0.19 percentage points for the early 1990s to 0.46 for the late 1990s.

The estimates also show that TFP growth jumped from about 0.35% per year from 1973–1995 to 0.99% for 1995–1998. This is a remarkable acceleration and reflects, in part, improved technology and efficiency in the production of goods and services, particularly IT goods.

These specific estimates are taken from Jorgenson and Stiroh, but other studies show similar magnitudes. Oliner and Sichel (2000) employed a similar methodology and estimated that both the contribution of IT capital and TFP growth increased substantially in the late 1990s, and Whelan estimated an even larger IT contribution. While methodological differences make exact comparisons difficult, these three studies reach a similar conclusion—IT has played an important role in the resurgence of United States economic growth in the late 1990s.

D. Industry-Level Evidence

One can also directly consider how IT-producing industries contributed to the acceleration of aggregate TFP growth in the 1990s by examining TFP growth in these particular industries. The BLS produces estimates of TFP growth for 20 manufacturing industries that include two IT-producing industries. Computer hardware, for example, is included in the "industrial and commercial machinery" industry, while telecommunications equipment and semiconductor production are included in the "electrical and electronic machinery" industry, defined by the Standard Industrial Classifications (SIC) codes as SIC #35 and #36, respectively.

Table II reports the BLS estimates of total factor productivity growth for the manufacturing industries

Table II Industry Total Factor Productivity Growth, 1949–1998

Industry	1949–1973	1973–1979	1979–1990	1990–1995	1995–1998
Manufacturing	1.5	−0.6	1.1	1.1	2.5
Food and kindred products	0.7	0.1	0.4	0.7	−0.5
Tobacco manufactures	na	na	na	na	na
Textile mill products	2.3	3.3	2.1	1.5	1.1
Apparel and related products	0.7	1.9	0.6	0.5	1.4
Paper and allied products	1.6	−1.2	0.0	0.2	1.1
Printing and publishing	0.5	−0.7	−0.9	−1.2	−0.3
Chemicals and allied products	2.5	−2.6	0.7	−0.3	1.0
Petroleum refining	0.8	−0.6	−0.1	0.3	1.2
Rubber and miscellaneous plastic products	1.0	−1.9	1.4	1.1	1.6
Leather and leather products	na	na	na	na	na
Lumber and wood products	1.7	0.4	2.5	−1.8	−0.8
Furniture and fixtures	0.6	0.4	0.6	0.6	1.6
Stone, clay, glass, and concrete products	1.1	−1.3	1.4	0.7	2.1
Primary metals industries	0.4	−2.3	0.3	0.7	1.6
Fabricated metals products	0.5	−1.0	0.5	0.4	−0.1
Industrial & commercial machinery	0.7	0.1	3.2	3.0	7.6
Electrical & electronic machinery	2.1	1.0	3.1	5.5	7.2
Transportation equipment	1.5	−0.6	0.2	0.4	1.2
Instruments	1.8	1.2	1.4	0.0	0.8
Miscellaneous manufacturing	1.5	−1.1	1.1	−0.4	0.2

Notes: All growth rates are average, annual rates. IT-producing industries are shaded.
From Bureau of Labor Statistics (2000). Multifactor productivity trends, 1998. USDL 00–267. http://www.bls.gov/mprhome.htm.

from 1949 to 1998. For the most recent period 1995–1998, the electrical and electronic machinery industry shows TFP growth of 7.2% per year, while the industrial and commercial machinery industry shows growth of 7.6%. Both estimates are considerably faster than the 2.5% annual growth for manufacturing as a whole, which suggests that it is indeed fundamental technical progress that is driving output growth and generating falling relative prices in the production of high-tech investment goods.

To be more precise about the contribution of the IT-producing industries to aggregate TFP growth, Oliner and Sichel isolated the contribution to aggregate TFP from component industries, e.g., computer-producing sector, semiconductor-producing sector, and the remainder of the economy. They also estimated aggregate TFP growth for the nonfarm business sector, which they report increased from 0.48% per year for 1991–1995 to 1.16% for 1996–1999. This is similar to the aggregate estimates reported in Table I, but methodological and definitional differences cause some divergence.

Table III reports Oliner and Sichel's decomposition estimates and clearly shows that accelerating TFP growth in the production of computers and semiconductors made a substantial contribution to faster aggregate TFP growth for the United States economy in the late 1990s. For the early 1990s, the vertically integrated computer sector that includes the production of embedded semiconductors contributed just 0.23 percentage points to aggregate TFP growth. For 1996–1999, the contribution of this broad sector more than doubled to 0.49 percentage points per year. Given the still relatively small size of the computer and semiconductor sectors, this contribution is substantial and highlights the important impact that IT is having on the United States economy.

What is responsible for this strong measured TFP growth in the production of computers, telecommunications equipment, and semiconductors? At a practical level, it largely reflects the accelerating declines in the quality-adjusted prices of computers and semiconductors. As discussed earlier, the NIPA price index for computer hardware showed a dramatic change in the mid-1990s when price declines accelerated from the 20% per year range to the 30% per year range. This translates into faster real output growth.

At a deeper and more fundamental level, this suggests more rapid technical progress (measured as total factor productivity growth) in the production of these high-tech assets. As technology improves, high-tech firms are able to produce better, more powerful IT goods at lower and lower prices. Some have attributed this development to the shortening of the product cycle for semiconductors and more intense competitive pressures. As more powerful models are brought to market more rapidly, this yields falling quality-adjusted prices and suggests more rapid technical progress in the production of high-tech assets.

E. Microeconomic Evidence

The evidence discussed above primarily deals with the impact of IT on the aggregate United States economy and component industries. There is also a great deal of empirical research that moves beneath this economy-wide data to examine the impact on the individual firms and narrowly defined industries that use IT.

The theoretical impact of IT investment and use is relatively clear—IT investment should lead to better performance and labor productivity gains for the firms and industries that invest in the latest IT equipment. The earliest evidence reviewed by Brynjolfsson and Yang (1996), however, was somewhat mixed with some studies finding a substantial positive impact from computer use, and others reporting evidence of a negative or insignificant impact.

Table III Contribution of IT to Aggregate TFP Growth, 1974–1999

	1974–1990	1991–1995	1996-1999
Nonfarm business TFP	0.33	0.48	1.16
Computer sector	0.12	0.16	0.26
Semiconductor sector	0.08	0.12	0.39
Other nonfarm business	0.13	0.20	0.50
Computer plus computer-related semiconductor sector	0.17	0.23	0.49

Notes: All numbers are annual, average growth rates.

Adapted from Oliner, D., and Sichel, D. E. (Fall 2000). The resurgence of growth in the late 1990s: Is information technology the story? *Journal of Economic Perspectives,* Vol. 14, No. 4, 3–22.

More recently a consensus points to a substantial positive impact from IT and they conclude "taken collectively, these studies suggest that information technology is associated with substantial increases in output and productivity." IT can also affect firms in ways that go beyond output and productivity gains. There is some evidence, for example, that IT can lead to higher market valuations, reduce the needed level of inventories, increase the level of intangible assets, and contribute significantly to enhanced consumer welfare.

A related branch of research looks at how IT affects the structure of firms that use it most intensively. These studies conclude that a critical part of the successful implementation of IT is the need for firms and managers to undertake complementary investments and pursue innovations in their production processes and work practices. That is, investment in IT hardware and software is not enough; firms must also reengineer themselves to reap the potential benefits.

There are many types of innovations that may be necessary. For example, the lower costs of IT may allow more flexible production that is not restricted to simple, standardized products. Other types of complementary innovations that might be needed include workforce reskilling, increased worker authority, new procedures for interacting with suppliers like electronic supply chains, business-to-business applications, and specialized pricing in recognition of the heterogeneity of customers. Moreover, the magnitude of these complementary investments may be much greater than the cost of the IT system itself, so it may be difficult to isolate and identify the precise impact of IT on the performance or profitability of a given firm.

IV. ALTERNATIVE VIEWS OF THE IMPACT OF IT

The evidence discussed so far suggests an important role for IT in the recent success of the United States economy. Both in terms of output and input effects, the aggregate and industry data show that IT is making a clear and growing contribution to the growth and productivity of the United States economy. Not all economists agree, however, and there are alternative points of view that range along the optimism scale from quite pessimistic to exuberantly optimistic.

A. IT Pessimism

Critics of the optimistic IT picture argue that some economists ignore many obvious factors that affect how IT actually affects the performance of the individual firms that use these high-tech assets. For example, there are complementary costs like continuous software upgrades, system compatibility problems and downtime, large training and staffing costs, and well-publicized failures of IT systems that simply do not work as planned. All of these have economic effects that combine to reduce the impact of IT.

Gordon outlined his skeptical view in 2000 and argued that the use of IT is having relatively little impact on the United States economy, and that the IT revolution pales in comparison to the inventions from earlier ages. His study examines the performance of the United States economy and concludes that there has been little pickup in the long-run trend growth rate of productivity, which is what economists really care about, in the bulk of the economy not involved in the actual production of IT assets. Rather, he finds that the recent productivity gains for the United States economy can largely be attributed to cyclical factors, and to the enormous productivity gains in the relatively narrow portion of the economy that actually produces computers and other durable goods.

For the United States economy as a whole, Gordon reported that labor productivity increased by 1.33 percentage points when 1995–1999 is compared to the trend for 1972–1995. Of this increase, he estimates that only a 0.83 percentage point reflects a rise in the underlying productivity trend, which is attributed primarily to methodological changes in price calculations, labor composition changes, capital deepening, and the contribution of faster TFP growth in the production of computers, semiconductors and other durables. This leaves very little trend productivity growth elsewhere in the economy. Gordon attributed the remaining one-half of a percentage point, 1.33 less 0.83 percentage points, to an unsustainable cyclical effect that may fade as the economy slows. That is, productivity growth naturally accelerates during output expansions and output growth has been quite rapid in recent years. When output growth slows, productivity growth may also slow. The exact magnitude of this cyclical effect remains uncertain, however, since we have not completed the business cycle, so this remains an important question.

Gordon also concludes that computers and the Internet fall short when compared to the great inventions and innovations of earlier periods like electricity, the internal combustion engine, indoor plumbing, chemicals, and plastics, etc. This argument is supported by the observation that the continued decline in the price of computers now allows them to be used for less beneficial and productive pursuits, an argu-

ment about diminishing marginal benefits of IT. Other factors that may limit the impact of IT on the United States include limitations imposed by human beings that operate computers, the need for face-to-face contact in some industries that preclude computers from replacing labor, and the still relatively small share of computers in the United States economy.

B. IT Optimism

A second alternative perspective comes from the optimistic end of the spectrum. Some "new economy" proponents argue that IT is unique, a fundamentally different type of innovation with much larger actual and potential gains. In this view, IT is a "transcendent technology" that changes the way business operates by boosting productivity, reducing costs, and improving the flexibility of the firm and its labor force. Kelly, for example, argued in 1997 that IT may yield additional benefits from increasing returns, productive externalities, and network economies where scale and scope lead to exponential growth once a critical mass is reached.

The optimistic view of IT is supported by the idea that economists may be unable to measure much of what IT actually does. This measurement problem view is often mentioned by the new economy proponents who claim that the impact of IT is understated in the economic statistics, so there may be even larger benefits than economists can readily observe. Examples include improved consumer convenience like automated teller machines (ATMs), increased product quality and variety from better inventory management systems, and new financial products, all of which are directly related to the IT revolution but hard for economists to measure. As evidence, one can point to the high concentration of IT investment in service sectors where output and productivity are notoriously hard to measure. In 1998, Stiroh reported that approximately three-fourths of IT investment in the 1980s occurred in service related industries like trade, finance, insurance, and real estate (FIRE), and other services.

Mismeasurement of IT-related output can obscure the economic impact in two ways. One, IT-intensive service sectors are steadily growing as a share of the economy, so any existing measurement problems now lead to a larger understatement of aggregate output and productivity. Sichel (1997b) finds this channel to be quantitatively small, however. Two, measurement errors may be becoming worse within IT-intensive sectors as IT facilitates a host of new products and quality improvements. Indeed, the BLS recognizes this to be a severe problem for productivity analysts, as discussed by Dean in 1999. The question of whether IT is making economic measurement problems more severe is an important, and unanswered, research question.

Finally, the optimistic view of IT gains some support from the "delay hypothesis," which argues that the largest economic impact of IT may still lie ahead. In 1990, David argued that the productive impact of electricity took several decades to permeate through the economy as firms learned how to best implement this new technology. One might expect the same delay for IT; once a critical mass is reached and sufficient operational knowledge is accumulated, there could be massive gains from IT. This delay hypothesis, however, is countered by the argument of diminishing returns of Gordon, who suggests that the most productive uses of IT have already been exploited by the early adopters and by the fact that it is now more than four decades since the first commercial sale of a mainframe computer.

V. CONCLUSIONS

This article examines the impact of information technology on the United States economy from a variety of economic perspectives. In sharp contrast to earlier research from the 1980s and early 1990s that led to the "computer productivity paradox," a consensus is now building that information technology has indeed played a major role in the recent revival of the United States economy. Economic growth is strong, and labor productivity is rising for an extended period at rates not seen since the 1960s. While some questions remain unresolved and economists will undoubtedly continue to debate the evidence, the record is becoming increasingly clear—IT has been an important economic force in the 1990s.

Looking to the future, it is less clear what one should expect from IT, and economists have made reasonable arguments with quite different conclusions. Some argue that the best may be behind us. If the most attractive information technology applications like automated reservation systems or electronic inventory management systems have already been exploited, then future information technology investment may be employed in increasingly unattractive pursuits. This decidedly pessimistic view is opposed by the "new economy" proponents, who argue that we are just now achieving the critical mass of IT, complementary innovations, and knowledge that will spark a continued wave of productivity and value-enhancing uses.

A third, less extreme, view is that as long as the price of computing power continues to decline at the

rates seen over the last few decades, United States firms and households will continue their investment and substitution toward IT. In this scenario, familiar economic forces will operate and the rapid accumulation of IT will likely remain an important contributor to the growth and productivity of the United States economy. Since these falling prices are the direct result of the fundamental technological advances driving the information technology revolution, their likelihood is a question best suited for the engineer and computer scientist, and not the economist.

SEE ALSO THE FOLLOWING ARTICLES

Accounting • National and Regional Economic Impacts of the Silicon Valley • Productivity

BIBLIOGRAPHY

Brynjolfsson, E., and Hitt, L. M. (Fall 2000). Beyond computation: Information technology, organizational transformation and business practices. *Journal of Economic Perspectives,* Vol. 14, No. 4, 23–48.

Brynjolfsson, E., and Yang, S. (February 1996). Information technology and productivity: A review of the literature. *Advances in Computers.* Vol. 43, 179–214.

Bureau of Labor Statistics (September 2000). Multifactor productivity trends, 1998. USDL 00–267. http://www.bls.gov/mprhome.htm.

Cole, R., Chen, Y. C., Barquin-Stolleman, J. A., Dulberger, E., Helvacian, H., and Hodge, J. H. (January, 1986). Quality-adjusted price indexes for computer processors and selected peripheral equipment. *Survey of Current Business.* 66, 41–50.

Dean, E. R. (February 1999). The accuracy of the BLS productivity measures. *Monthly Labor Review.* 24–34.

Gordon, R. J. (Fall 2000). Does the "new economy" measure up to the great inventions of the past? *Journal of Economic Perspectives,* Vol. 14, No. 4, 49–74.

Jorgenson, D. W., and Stiroh, K. J. (2000). Raising the speed limit: U.S. economic growth in the information age. *Brookings Papers on Economic Activity 1.* pp. 125–211.

Oliner, S. D., and Sichel, D. E. (Fall 2000). The resurgence of growth in the late 1990s: Is information technology the story? *Journal of Economic Perspectives,* Vol. 14, No. 4, 3–22.

Parker, R., and Grimm, B. (April 2000). Software prices and real output: Recent developments at the bureau of economic analysis. Bureau of Economic Analysis. http://www.bea.doc.gov/bea/papers.htm.

Sichel, D. E. (1997a). *The computer revolution: An economic perspective.* Washington, DC: The Brookings Institution.

Sichel, D. E. (1997b). The productivity slowdown: Is a growing unmeasurable sector the culprit? *Review of Economics and Statistics.* 79(3), 367–370.

Stiroh, K. J. (April 1998). Computers, productivity, and input substitution. *Economic Inquiry.* Vol. XXXVI. No. 2, 175–191.

Triplett, J. E. (1989). Price and technological change in a capital good: A survey of research on computers. *Technology and Capital Formation* (Dale W. Jorgenson and Ralph Landau, eds.). Cambridge, MA: The MIT Press.

Triplett, J. E. (1999). The Solow productivity paradox: What do computers do to productivity? *Canadian Journal of Economics.* Vol. 32. No. 2, 309–334.

Wasshausen, D. (April 2000). Computer prices in the national accounts. Bureau of Economic Analysis.

Whelan, K. (January 2000). Computers, obsolescence, and productivity. Federal Reserve Board, Finance and Economics Discussion Series, pp. 2000–06.

Electronic Commerce

Hossein Bidgoli
California State University, Bakersfield

GLOSSARY

business-to-business (B2B) e-commerce Involves electronic transactions among and between businesses.
business-to-consumer (B2C) e-commerce Businesses sell goods and services directly to consumers.
buyer-controlled marketplace A buyer or a group of buyers open an electronic marketplace and invites sellers to bid on the announced products or request for quotations (RFQs).
consumer-to-business (C2B) e-commerce Involves individuals selling goods and services to businesses.
consumer-to-consumer (C2C) e-commerce Involves business transactions among individuals using the Internet and web technologies.
electronic business (e-business) Any electronic transaction (e.g., information exchange), which subsumes e-commerce. E-business encompasses all the activities that a firm performs for selling and buying services and products using computers and communications technologies.
electronic data interchange (EDI) An application-to-application transfer of business documents between computers using industry-defined standards.
electronic funds transfer (EFT) Electronic transfer of money by financial institutions.
e-commerce business models Different methods and procedures of doing business used by e-businesses to generate revenue and stay viable using the web and web technologies.
e-procurement Conducting procurement functions (buying goods and services) over the Web.
extranet A secure network that uses Internet and Web technologies to connect two or more intranets of business partners, enabling business-to-business communications.
intranet A network within the organization that uses Web technologies (TCP/IP, HTTP, FTP, simple mail transport protocol [SMPT], HTML, and XML) for collecting, storing, and disseminating useful information throughout the organization. This information supports e-commerce activities such as sales, customer service, and marketing.
nonbusiness and government e-commerce Using e-commerce functions and technologies for improving productivity by government agencies (local, state, federal) and by nonprofit organizations.
organizational (intrabusiness) e-commerce Using e-commerce functions and technologies for improving productivity within an organization.
seller-controlled marketplace Businesses and consumers use the seller's product catalog and order products and services on-line.
third-party controlled marketplace This B2B model is not controlled either by sellers or buyers, rather by a third party. The marketplace generates its revenue from the fees generated by matching buyers and sellers.
trading partner agreements A B2B e-commerce model automates the processes for negotiating and enforcing contracts between participating businesses.

value chain Consists of a series of activities designed to satisfy a business need by adding value (or cost) in each phase of the process.
voice-based e-commerce Conducting e-commerce activities using voice (natural language) and communications technologies.
wireless e-commerce Conducting e-commerce activities using mobile and wireless networks.

I. INTRODUCTION

This article first defines value chain analysis, explains the e-commerce role in the value chain process, and reviews popular e-commerce business models that are being used by successful e-businesses. It then provides a detail definition of e-commerce and e-business and reviews the major components of e-commerce. This article also compares e-commerce with traditional commerce and then discusses the major categories of e-commerce including business-to-consumer (B2C), business-to-business (B2B), consumer-to-consumer (C2C), organizational (intrabusiness), consumer-to-business (C2B), and nonbusiness and government. It further explores the advantages and disadvantages of e-commerce and then explains major activities involved in a B2C e-commerce life cycle. This article provides a discussion of various B2B e-commerce models including seller-controlled marketplace, buyer-controlled marketplace, third-party controlled marketplace, and trading partner agreements. It then concludes with an overview of wireless and voice-based e-commerce as the most promising growth areas in the e-commerce environment.

II. VALUE CHAIN AND E-COMMERCE

One way to look at e-commerce and its role in the business world is through value-chain analysis. Michael Porter introduced the value-chain concept in 1985. It consists of a series of activities designed to satisfy a business need by adding value (or cost) in each phase of the process. A typical business organization (or a division within a business organization) designs, produces, markets, delivers, and supports its product(s) or service(s). Each of these activities adds cost and value to the product or service that is eventually delivered to the customer. For example, in a furniture-manufacturing firm, the firm buys raw materials (wood) from a logging company and then converts these raw materials (wood) into finished product (chair), chairs are shipped to retailers, distributors, or customers. The firm markets and services these products (chairs). In addition to these primary activities that result in a final product or service, Porter also includes supporting activities in this process. These supporting activities include the following:

- Obtaining various inputs for each primary activity
- Developing technology to keep the business competitive
- Managing human resources
- Managing company infrastructure

In the above example, the value chain may continue after delivering chairs to the furniture store. The store, by offering other products and services and mixing and matching this product with other products, may add additional value to the chair. Also, value-chain analysis may highlight the opportunity for the furniture manufacturer to manufacture chairs directly. This means it may enter in the logging business directly or through partnership with others. In any industry, an enterprise is located in a value chain when it buys goods and services from suppliers, adds value, and sells them to customers.

E-commerce, its applications and its supporting technologies, provides the business platform for realizing Porter's visions. The Internet can increase the speed and accuracy of communications between suppliers, distributors, and customers. Moreover, the Internet's low cost means that companies of any size will be able to take advantage of value-chain integration. E-commerce may enhance value chain by identifying new opportunities for cost reduction. The following are some examples:

- Reducing cost. Using e-mail to notify customers versus using regular mail.
- Revenue improvement or generation. Selling to remote customers using the company web site. These sales would not have been materialized otherwise, or selling digital products such as songs or computer software or distributing software through the Web.
- Product or service improvement. Offering on-line customer service or new sales channel identification.

Many companies have taken advantage of the Web and e-commerce to reduce cost, improve revenue, and increase customer service. For example, Microsoft Direct is a web site that provides on-line order-management and customer-assistance services to shoppers acquiring Microsoft products. It operates using on-line storefronts that hand shoppers over to Microsoft Direct as soon as they have selected the prod-

ucts they want to order. Dell Computer generates a large portion of its revenue through the Web by eliminating the middleman. Cisco Systems sells much of its networking hardware and software over the Web, improving revenue and reducing cost. United Parcel Service (UPS) and Federal Express use the Internet to track packages that result in enhanced customer service.

According to Jupiter Media Metrix, a New York Internet research company, on-line retail sales will reach $104 billion in 2005 and $130 billion by 2006, up from $34 billion in 2001. This data indicates the Internet growth and its applications in brick-and-mortar companies as well as in on-line retailers for generating additional revenue.

III. E-COMMERCE BUSINESS MODELS

Similar to traditional businesses, the ultimate goal of an e-business is to generate revenue and make a profit. It is true that the Internet has improved productivity for almost all the organizations that are using it. However, the bottom line is that productivity must be converted to profitability. The fall of many "dotcom" companies in 2000 and 2001 is a clear indication of this phenomenon. The survivors are clearly those businesses with a sound business model of how they plan to make a profit and sustain a business for future growth.

To achieve profitability as the ultimate goal, different e-businesses or e-commerce sites position themselves in different parts of the value-chain model discussed in the last section. To generate revenue, an e-business either sells products or services or shortens the link between the suppliers and consumers. Many B2B models try to eliminate the middleman by using the Web to deliver products and services directly to their customers. By doing this they may be able to offer cheaper products and better customer service to their customers. The end result would be a differentiation between them and their competitors, increased market share, and increased customer loyalty.

Products sold by e-businesses could be either traditional products, such as books and apparel or digital products, such as songs, computer software, or electronic books. E-commerce models are either an extension or revision of traditional business models, such as advertising and auction models, or a new type of business model that is suitable for the Web implementation, such as informediary, selling information collected over the Web about individuals and businesses to other businesses. The most popular e-commerce models are the following:

- Merchant
- Brokerage
- Advertising
- Mixed
- Informediary
- Subscription

The merchant model is basically the transferring of an old retail model to the e-commerce world by using the Internet. There are different types of *merchant* models. The most common type of merchant model is similar to a traditional business model that sells goods and services over the Web. Amazon.com is a good example of this type of business model. An e-business similar to amazon.com utilizes the services and technologies offered by the Web to sell products and services directly to the consumers. By offering good customer service and reasonable prices, these companies establish a brand on the Web. The merchant model is also used by many traditional businesses to sell goods and services over the Internet. Dell, Cisco Systems, Gateway, Staples, Micro Warehouse, Nordstrom, and Compaq are popular examples.

These companies eliminate the middleman by generating a portion of their total sale over the Web and by accessing difficult to reach customers.

Using the brokerage model the e-business brings the sellers and buyers together on the Web and collects a commission on the transactions. The best example of this type is an on-line auction site such as eBay, which generates additional revenue by selling banner advertisement on their sites. Other examples of the brokerage model are on-line stockbrokers, such as NDB.com, E*TRADE.com, and Schwab.com, which generate revenue through collecting commissions from both buyers and sellers of securities.

The advertising model is an extension of traditional advertising media, such as radio and television. Search engines and directories such as AltaVista and Yahoo! provide contents (similar to radio and TV) and allow the users to access this content for free. By creating significant traffic, these e-businesses are able to charge advertisers for putting banner ads or leasing spots on their sites.

The mixed model generates revenue both from advertising and subscription. Internet service providers (ISPs) such as America Online (AOL) Time Warner generate revenue from advertising and their customers' subscription fees for Internet access.

E-businesses that use the informediary model collect information on consumers and businesses and then sell this information to interested parties for marketing purposes. For example, NetZero.com provides free Internet access; in return it collects information

related to the buying habits and surfing behavior of customers. This information is later sold to advertisers for direct marketing. eMachines.com offers free PCs to its customers for the same purpose. E-businesses such as BizRate.com® collect information related to the performance of other sites and sell this information to advertisers.

Using a subscription model an e-business might sell digital products and services to its customers. *The Wall Street Journal* and *Consumer Reports* are two examples. Street.com is another example of this model that sells business news and analysis based on subscription.

Table II Business Uses of the Internet

Buying and selling products/services
Collaborating with others
Communicating within organizations
Gathering general information
Gathering information on competitors
Providing customer service
Providing software update and patches
Providing vendor support
Publishing and disseminating information

IV. DEFINING E-COMMERCE

An e-business encompasses all activities that a firm performs for selling and buying products and services using computers and communications technologies. In broad terms, an e-business includes a host of related activities, such as on-line shopping, sales force automation, supply-chain management, electronic payment systems, web advertising, and order management. E-commerce is buying and selling goods and services over the Internet. Based on this definition, e-commerce is part of e-business. However, in many cases the two are used interchangeably.

E-business, a major contributor to the popularity of global information systems, is a system that includes not only those transactions that center on buying and selling goods and services to generate revenue, but also those transactions that support revenue generation. These activities include generating demand for goods and services, offering sales support and customer service, or facilitating communications between business partners.

E-commerce builds on traditional commerce by adding the flexibility offered by computer networks and the availability of the Internet. By generating and delivering timely and relevant information through computer networks, e-commerce creates new opportunities for conducting commercial activities on-line, and thus it fosters easier cooperation between different groups: branches of a multinational company sharing information for a major marketing campaign; companies working to design and build new products or offer new services; or businesses sharing information to improve customer relations.

Table I lists some of the major beneficiaries of e-commerce. A close examination of these businesses and entities reveals the potential for e-commerce to generate revenue and reduce costs. For example, banks use the Web for diverse business practices and customer service. The entertainment industry utilizes the Web extensively for offering diverse products and services.

Different branches of governments using e-commerce have experienced major cost savings. For ex-

Table I Major Beneficiaries of E-Commerce

Banks
Brokerage firms
Entertainment companies
Government agencies
Insurance companies
Marketing firms
Publishing firms
Retailers
Training organizations
Travel industries
Universities

Table III Popular Products and Services Purchased On-Line

Airline tickets and travel
Apparel and footwear
Banking services
Books and music
Computer hardware, software, and other electronics
Flowers and gifts
Stock brokerage services
Entertainment
Information for conducting research and evaluating competition

Table IV Examples of Companies Using E-Commerce

Amazon.com provides access to several million books electronically. It also sells music CDs, electronics, software, toys, video games, prescription drugs, and much more
Drugstore.com, and CVS.com refill and sell new drugs and vitamins and other health and beauty products on-line
American Express successfully uses e-commerce for credit card transactions
Apple Computer sells computers on-line (Apple.com)
autobytel sells cars over the Web
Charles Schwab, National Discount Brokers, and E*TRADE have successfully used e-commerce for on-line security transactions
Cisco Systems sells data communications components over the Web
Dell Computer and Gateway sell computers through their web sites and allow customers to configure their systems on the Web and then purchase them
Epicurious.com sells exotic foods over the Web
Peapod.com sells groceries over the Web
Proctor & Gamble and IBM conduct order placements electronically
Virtual Vineyards sells expensive wines from small vineyards over the Web

ample, the United States federal government uses electronic data interchange (EDI) for requests for quotes (RFQs), quotes, award notices, purchase orders, and invoices.

Table II lists business uses of the Internet. These services and capabilities are a core part of a successful e-commerce program. They are either parts of a value chain or are included as supporting activities discussed earlier in the article.

Table III lists some popular products and services that can be purchased on-line. Close examination of these products and services reveals their suitability for e-commerce transactions. Several successful e-businesses including amazon.com have established their business models around selling these products and services.

Table IV lists companies using e-commerce, highlighting the products and services that are most suitable for web transactions. Table V lists the top 10 countries with the highest sales volume in e-commerce operations. This table is a guide for the investigation and implementation of e-commerce on the international scene. As this table shows, e-commerce is estimated in the United States to generate approximately $3.2 trillion in revenue in 2004.

Table VI lists the top on-line retail sites based on sales volume in August 2000. Again this table high-

Table V Top 10 E-Commerce Countries

Country	Total sales (%)	E-commerce sales (in $ millions)	
	2004	2000	2004 (estimates)
1. United States	13.3	488.7	3189.0
2. Japan	8.4	31.9	880.3
3. Germany	6.5	20.6	386.5
4. United Kingdom	7.1	17.2	288.8
5. Australia	16.4	5.6	207.6
6. France	5.0	9.9	206.4
7. South Korea	16.4	5.6	205.7
8. Taiwan	16.4	4.1	175.8
9. Canada	9.2	17.4	160.3
10. Italy	4.3	7.2	142.4

[Adapted from Forester Research and INFOWORLD, May 15, 2000, p. 20.]

Table VI Top On-Line Retail Sites Based on Sales Volume in 2000

Amazon.com
Ticketmaster.com
Buy.com
CDnow.com
Sears.com
Barnesandnoble.com
JCPenney.com
Real.com
Pets.com

[Adapted from INFOWORLD, September 25, 2000, p. 22.]

lights those e-businesses that have been able to generate the highest sales in 2000. A close investigation of the business model used by these companies may serve as a guide for others.

V. E-COMMERCE VERSUS TRADITIONAL COMMERCE

Although the goals and objectives of both e-commerce and traditional commerce are the same, selling products and services to generate profits, they do it quite differently. In e-commerce, the Web and telecommunications technologies play a major role. In e-commerce there may be no physical store, and in most cases the buyer and seller do not see each other. Table VII compares and contrasts traditional commerce and e-commerce. However, it is important to notice that currently many companies operate with a mix of traditional and e-commerce. Just about all medium and large organizations have some kind of e-commerce presence. The Gap, Toys-R-Us, Office Depot, Wal-Mart Stores, and Sears are a few examples.

VI. MAJOR CATEGORIES OF E-COMMERCE

The several categories of e-commerce in use today are classified based on the nature of transactions, including B2C, B2B, C2C, C2B, organizational (intrabusiness), and nonbusiness and government. In the following paragraphs we define these categories.

A. Business-to-Consumer

In business-to-consumer (B2C) e-commerce, businesses sell directly to consumers. Amazon.com, barnes-andnoble.com and onsale.com are three good examples of this category. Amazon.com and its business partners sell a diverse group of products and services to their customers, including books, videos, DVDs, prescription drugs, on-line auctions, and much more. In addition to pure B2C e-commerce players such as amazon.com, other traditional businesses have entered the virtual marketplace by establishing comprehensive web sites and virtual storefronts. Wal-Mart Stores, the Gap, and Staples are examples of companies that are very active in B2C e-commerce. In these cases, e-commerce supplements the traditional commerce by offering products and services through electronic channels. Some experts believe that, in the long term, these types of businesses should be more successful than pure e-commerce businesses. Some of the advantages of these e-commerce sites and companies include availability of physical space (customers can physically visit the store), availability of returns

Table VII E-Commerce versus Traditional Commerce

Activity	Traditional commerce	E-commerce
Product information	Magazines, flyers	Web sites and on-line catalogs
Business communications	Regular mail, phone	E-mail
Check product availability	Phone, fax, letter	E-mail, web sites, and extranets[a]
Order generation	Printed forms	E-mail, web sites
Product acknowledgments	Phone, fax	E-mail, web sites, and EDI[b]
Invoice generation	Printed forms	Web sites

[a] Extranets are the connection of two or more intranets. Intranets are internal networks that use web technologies.
[b] Electronic data interchange.

(customers can return a purchased item to the physical store), and availability of customer service in these physical stores.

B. Business-to-Business

Business-to-business (B2B) involves electronic transactions among and between businesses. This technology has been around for many years through electronic data interchange (EDI) and electronic funds transfer (EFT). In recent years the Internet has significantly increased B2B transactions and has made B2B the fastest growing segment within the e-commerce environment. In recent years extranets have been effectively used for B2B operations. The reliance of all businesses upon other companies for supplies, utilities, and services has enhanced the popularity of B2B e-commerce. An example of B2B is an auto exchange formed by Ford, DaimlerChrysler, and General Motors called covisint (http://www.covisint.com). This system offers services in areas of procurement, supply-chain management, and collaborative product development. Partners achieve build-to-order capability through connectivity among the key lines of business and throughout an individual company's supply chain. Companies using systems such as covisint report millions of dollars in savings by increasing the speed, reducing errors, and eliminating many manual activities. Wal-Mart Stores are another major player in B2B e-commerce. Wal-Mart's major suppliers (e.g., Proctor & Gamble, Johnson & Johnson, and others) sell to Wal-Mart Stores electronically; all the paperwork is handled electronically. These suppliers can access online the inventory status in each store and replenish needed products in a timely manner. In a B2B environment, purchase orders, invoices, inventory status, shipping logistics, and business contracts handled directly through the network result in increased speed, reduced errors, and cost savings.

C. Consumer-to-Consumer

The consumer-to-consumer (C2C) category involves business transactions among individuals using the Internet and web technologies. Using C2C, consumers sell directly to other consumers. For example, through classified ads or by advertising, individuals sell services or products on the Web or through auction sites such eBay.com. Using the Web, consumers are able to sell a wide variety of products to each other. A typical C2C e-commerce offers catalogs, auctions, and escrow services. Consumers are also able to advertise their products and services in organizational intranets (discussed later in the article) and sell them to other employees.

D. Consumer-to-Business

Consumer-to-business (C2B) e-commerce involves individuals selling to businesses. This may include a service or product that a consumer is willing to sell. In other cases an individual may seek sellers of a product and service. Companies such as priceline.com and mobshop.com for travel arrangements are examples of C2B. Individuals offer certain prices for specific products and services.

E. Organizational (Intrabusiness)

Organizational or intrabusiness e-commerce involves all the e-commerce-related activities that take place within the organization. The organization intranets provide the right platform for these activities. These activities may include exchange of goods, services, or information among the employees of an organization. This may include selling organization products and services to the employees, conducting training programs, offering human resources services, and much more. Although they are not direct selling and buying, some of these activities provide support for a successful e-commerce program in human resources management, finance, and marketing.

F. Nonbusiness and Government

The e-commerce applications in government and many nonbusiness organizations are on the rise. Several government agencies in the United Sates have been using e-commerce applications for several years, including the Department of Defense, Internal Revenue Service, and the Department of Treasury. Universities are using e-commerce applications extensively for delivering their educational products and services on a global scale. Not-for-profit, political, and social organizations also use e-commerce applications for various activities, such as fundraising and political forums. These organizations also use e-commerce for purchasing (to reduce cost and improve speed) and for customer service.

Experts predict that various types of e-commerce involving government, businesses, and consumers will grow significantly. For example, government agencies

will purchase goods ranging from paper clips to military helicopters using the Web. The market for on-line government purchases is substantial and potentially profitable for all the involved parties. According to a report from technology research firm Gartner federal, state, and local governments' e-commerce spending will grow to more than $6.2 billion by 2005 from $1.5 billion in 2000. The following are the various categories of e-commerce involving government.

- Business to government (B2G): sale of goods or services to a branch of the government
- Government to business (G2B): sale of goods or services to a business, e.g., building permits
- Government to consumers (G2C): sale of goods or services to consumers, e.g., driving licenses
- Consumers to government to (C2G): on-line transactions between consumers and government, e.g., tax payment, issuance of certificates or other documents
- Government to government (G2G): on-line transactions between and among different branches of government or selling goods and services from one state government to another state

VII. ADVANTAGES AND DISADVANTAGES OF E-COMMERCE

Similar to traditional businesses, e-commerce presents many advantages and disadvantages. If the e-commerce is established based on the correct business model, the advantages of e-commerce significantly outweigh its disadvantages. Table VIII highlights some of the advantages of e-commerce.

In the e-commerce world, doing business around the globe 7 days a week, 24 hours a day is a reality. Customers in any part of the world with an Internet connection can log onto the e-commerce site and order a product or service. Holidays, weekends, after hours, and differences in time zones do not pose any problem.

Using various tools such as cookies, e-mail, and the company web site, the e-commerce site is able to gain additional knowledge about potential customers. This knowledge could be effectively used to better market products and services. For example, the e-business would know the customer preferences, shopping habits, gender, age group, etc.

In the e-commerce environment, customer involvement could be significantly improved. For example, the customer can provide an on-line review of a book that he or she has recently purchased from the e-commerce site, or the customer may participate in various open forums, chat groups, and discussions.

Table VIII Selected Possible Advantages of E-Commerce

Doing business around the globe 7 days a week, 24 hours a day
Gaining additional knowledge about potential customers
Improved customer involvement
Improved customer service
Improved relationships with suppliers
Improved relationships with the financial community
Increased flexibility and ease of shopping
Increased number of customers
Increased return on capital and investment, since no inventory is needed
Personalized service
Product and service customization

An e-commerce site, by using tools such as an on-line help desk, company web site, and e-mail is able to improve customer service. Many of the customers' questions and concerns are answered using these tools with minimum cost. Printing forms on-line, downloading software patches, and reviewing frequently asked questions (FAQs) are other examples of customer service.

A B2B e-commerce site can improve its relationships with suppliers. E-commerce technologies enable these businesses to exchange relevant information on a timely basis with minimum cost. Using B2B e-commerce assists businesses in managing a comprehensive inventory management system. An e-commerce site can improve its relationships with the financial community through the timely transfer of business transactions and a better understanding of the business partner's financial status.

Increased flexibility and ease of shopping is a significant advantage of e-commerce. The customer does not need to leave his or her home or office and commute to purchase an item. The customer does not need to look for parking in a shopping mall during holidays, nor risk losing the supervision of his or her small children or elderly relatives (for even a short period). Shopping tasks can be done from the privacy of the home with a few clicks of mouse.

An e-business or a traditional business with an e-commerce presence could increase its potential customers. Customers from remote locations and those outside of the business geographical boundaries can purchase products and services from the e-commerce site.

In many cases an e-commerce site should be able to increase return on capital and investment since no

inventory is needed. An effective e-commerce program is able to operate with no inventory or with minimum inventory. In some cases an e-commerce site serves as middleman, taking orders from customers, routing orders to suppliers and making a profit. In other cases an e-commerce site is able to maintain minimal inventory and fill customers' orders through a just-in-time (JIT) inventory system. By having no or minimal inventory, the e-commerce site could avoid devaluation in inventory due to the release of a new product, change in fashion, season, etc.

In many cases an e-commerce site by using various web technologies is able to offer personalized service to its customers and at the same time customize a product or service that best suits a particular customer. By collecting relevant information on different customers, a particular product or service could be tailor-made to customer taste and preference. In some cases, the customer may pick and choose, as in sites that allow the customer to create his or her own CD, travel plan, PC, automobile, etc.

Many of the disadvantages of e-commerce are related to technology and business practices. Most of these disadvantages should be resolved in the near future. Table IX lists some of the disadvantages of e-commerce. In the following paragraphs I provide a brief description of these disadvantages.

Possible capacity and bandwidth problems could be a serious problem, however, several projects are underway to resolve this issue in the near future.

Security and privacy issues are major concerns for many e-businesses and consumers. Security risks to companies involved in e-commerce posed by hackers or e-terrorist attacks are varied and must be considered. Denial of service attacks that brought several popular e-commerce sites into a temporary halt in 2000 and 2001 are major problems and must be carefully analyzed. However, security issues and measures are expected to improve in coming years, through the use of media other than credit cards on the Web, such as e-wallet, e-cash, and other electronic payment systems. Also, the acceptance of digital signatures, more widespread application and acceptance of encryption and biometric technologies, and greater awareness and understanding of customers' concerns may resolve some of the security and privacy issues.

Table IX Some Disadvantages of E-Commerce

Possible capacity and bandwidth problems
Security and privacy issues
Accessibility (not everybody is connected to the Web yet)
Acceptance (not everybody accepts this technology)
A lack of understanding of business strategy and goals
Integration with traditional legacy systems

The accessibility of customers issue will certainly become more manageable, as the number of Internet users increases daily. Also, the reduction in cost of PCs, handheld, and other Internet appliances should further increase Internet applications and result in further accessibility of e-commerce.

Similar to other technologies acceptance of e-commerce by the majority of people will take time. However, the growth of the Internet and on-line shopping points to further acceptance of e-commerce applications in the near future. When the technology is fully accepted, a company's e-business, strategies, and goals should also become better understood.

The failure of communications companies to meet demand for DSL (digital subscriber line) and other high bandwidth technologies, and difficulties of integrating new technology with companies' legacy or incompatible propriety technologies are among other disadvantages of e-commerce that must be considered.

VIII. A BUSINESS-TO-CONSUMER E-COMMERCE CYCLE

There are five major activities involved in conducting B2C e-commerce:

1. *Information sharing.* A B2C e-commerce model may use some or all of the following applications and technologies to share information with customers:
   - Company web site
   - On-line catalogs
   - E-mail
   - On-line advertisements
   - Multiparty conferencing
   - Bulletin board systems
   - Message board systems
   - Newsgroups and discussion groups
2. *Ordering.* A customer may use electronic forms (similar to paper forms, available on the company's web site) or e-mail to order a product from a B2C site. A mouse click sends the necessary information relating to the requested item(s) to the B2C site.
3. *Payment.* The customer has a variety of options for paying for the goods or services. Credit cards, electronic checks, and digital cash are among the popular options.
4. *Fulfillment.* The fulfillment function could be very complex depending upon the delivery of physical products (books, videos, CDs) or digital products (software, music, electronic documents). It also

depends on whether the e-business handles its own fulfillment operations or outsources this function to third parties. In any case, fulfillment is responsible for physically delivering the product or service from the merchant to the customer. In case of physical products, the filled order can be sent to the customer using regular mail, Federal Express, or UPS. The customer usually has the option to choose from these various delivery systems. Naturally for faster delivery, the customer has to pay additional money. In case of digital products, the e-business uses digital certificates to assure security, integrity, and confidentiality of the product. It may also include delivery address verification and digital warehousing. Digital warehousing stores digital products on a computer until they are delivered. Several third-party companies handle the fulfillment functions for an e-business with moderate costs.

5. *Service and support.* Service and support are even more important in e-commerce than traditional businesses because e-commerce companies lack a traditional physical presence and need other ways to maintain current customers. It is much cheaper to maintain current customers than to attract new customers. For this reason, e-businesses should do whatever they can in order to provide timely, high-quality service and support to their customers. The following are some examples of technologies and applications used for providing service and support:
   - E-mail confirmation
   - Periodic news flash
   - Online surveys
   - Help desk
   - Guaranteed secure transactions
   - Guaranteed online auctions

E-mail confirmation, periodic news flash, and on-line surveys may also be used as marketing tools. E-mail confirmation assures the customer that a particular order has been processed and that the customer should receive the product or service by a certain date. In most cases, the e-mail confirmation provides the customer with a confirmation number that the customer can use to trace the product or service.

Periodic news flash is used to provide customers with the latest information on the company or on a particular product or offering. Although on-line surveys are mostly used as a marketing tool, their results can assist the e-commerce site to provide better services and support to its customers based on what has been collected in the survey.

Help desks in the e-commerce environment are used for the same purpose as in traditional businesses. They provide answers to common problems or provide advice for using products or services.

Guaranteed secure transactions and guaranteed on-line auctions assure customers that the e-commerce site covers all the security and privacy issues. These services are extremely important because as mentioned earlier, many customers still do not feel comfortable conducting on-line business.

The B2B e-commerce model uses a similar cycle, as discussed earlier; however, businesses use the following four additional technologies extensively:

- Intranets
- Extranets
- Electronic data interchange (EDI)
- Electronic funds transfer (EFT)

IX. BUSINESS-TO-BUSINESS E-COMMERCE: A SECOND LOOK

B2B is the fastest growing segment of e-commerce applications. The B2B e-commerce creates dynamic interaction among the business partners; this represents a fundamental shift in how business will be conducted in the 21st century.

According to Jupiter Communications, Inc., an Internet research company, B2B on-line trade will rise to $6.3 trillion by 2005. The B2B e-commerce reduces cycle time, inventory, and prices and enables business partners to share relevant, accurate, and timely information. The end result is improved supply-chain management among business partners. Table X summarizes the advantages of B2B e-commerce. The following paragraph provides brief descriptions of these advantages.

A B2B e-commerce lowers production cost by eliminating many labor-intensive tasks. More timely information is achieved by the creation of a direct on-line connection in the supply chain. Accuracy is improved

Table X Advantages of B2B E-Commerce

Lower production cost
More timely information
Increased accuracy
Improved cycle time
Increased communications
Improved inventory management

because fewer manual steps are involved. Cycle time improves because flow of information and products between business partners is made simpler. In other words raw materials are received faster and information related to customer demands is also more quickly transferred. Naturally this close communication between the business partners improves overall communication, which results in improved inventory management and control. Most of the disadvantages of e-commerce outlined in Table IX, also apply to B2B e-commerce.

X. MAJOR MODELS OF BUSINESS-TO-BUSINESS E-COMMERCE

The three major types of B2B e-commerce models are determined by who controls the marketplace: seller, buyer, or intermediary (third party). As a result, the following three marketplaces have been created:

- Seller-controlled
- Buyer-controlled
- Third-party exchanges

A relatively new model, called trading partner agreements, facilitates contracts and negotiations among business partners and is gaining popularity. Each model has specific characteristics and is suitable for a specific business. The following paragraphs provide a description and examples of each.

A. Seller-Controlled Marketplace

The most popular type of B2B model for both consumers and businesses is the seller-controlled marketplace. Businesses and consumers use the seller's product catalog to order products and services on-line. In this model the sellers who cater to fragmented markets such as chemicals, electronics, and auto components come together to create a common trading place for the buyers. While the sellers aggregate their market power, it simplifies the buyers search for alternative sources.

One popular application of this model is e-procurement, which is radically changing the buying process by allowing employees throughout the organization to order and receive supplies and services from their desktop with just a few mouse clicks. E-procurement significantly streamlines the traditional procurement process by using the Internet and web technologies. This results in major cost savings and improves the timeliness of procurement processes and the strategic alliances between suppliers and participating organizations. It also offers all of the benefits and advantages outlined in Table X.

Using e-procurement, the business logistics and processes reside on the side of the purchasing company (the receiving partner). The procurement application often has workflow procedures for the purchasing-approval process, allows connection to only company approved e-catalogs, and provides the employee with prenegotiated pricing. The main objective of e-procurement is to prevent buying from suppliers other than the preapproved list of sellers, which many companies will have for their normal procurement activities, and also to eliminate processing costs of purchases. Not following this process can be costly to the receiving company because it may result in paying higher prices for needed supplies.

By using ongoing purchases, e-procurement may qualify customers for volume discounts or special offers. E-procurement software may make it possible to automate some buying and selling, resulting in reduced costs and improved processing speeds. The participating companies expect to be able to control inventories more effectively, reduce purchasing agent overhead, and improve manufacturing cycles. E-procurement is expected to be integrated into standard business systems with the trend toward computerized supply-chain management. Using e-procurement, buyers will have local catalogs with negotiated prices and local approvals.

B. Buyer-Controlled Marketplace

Large corporations (e.g., General Electric or Boeing) with significant buying power or a consortium of several large companies use this model. In this case a buyer or a group of buyers opens an electronic marketplace and invites sellers to bid on the announced products or RFQs (request for quotation). The consortium among DaimlerChrysler, Ford, and General Motors (to which Toyota recently joined) is a good example of this model. Using this model the buyers are looking to efficiently manage the procurement process, lower administrative cost, and exercise uniform pricing.

Companies are making investments in a buyer-controlled marketplace with the goal of establishing new sales channels that increase market presence and lower the cost of each sale. By participating in a buyer-controlled marketplace a seller could do the following:

- Conduct presales marketing
- Conduct sales transactions

- Automate the order management process
- Conduct postsales analysis
- Automate the fulfillment process
- Improve understanding of buying behaviors
- Provide an alternate sales channel
- Reduce order placement and delivery cycle time

C. Third-Party-Controlled Marketplace

A third-party-controlled marketplace model is not controlled by sellers or buyers, but rather by a third party. The marketplace generates revenue from the fees generated by matching buyers and sellers. These marketplaces are usually active either in a vertical or horizontal market. A *vertical market* concentrates on a specific industry or market. The following are some examples of this type:

- Altra Energy (energy)
- Cattle Offering Worldwide (beef & dairy)
- Neoforma (hospital product supplies)
- PaperExchange.com (supplies for publishers)
- PlasticsNet.com (raw materials and equipment)
- SciQuest.com (laboratory products)
- Verticalnet.com (provides end-to-end e-commerce solutions that are targeted at distinct business segments through three strategic business units VerticalNet Markets, VerticalNet Exchanges, and VerticalNet Solutions.)

A *horizontal market* concentrates on a specific function or business process. It provides the same function or automates the same business process across different industries. The following are some examples of this type:

- iMARK.com (capital equipment)
- Employee.com (employee benefits administration)
- Adauction.com (media buying)
- Youtilities.com (corporate energy management and analysis)
- BidCom.com (risk and project management services)

A third-party-controlled marketplace model offers suppliers a direct channel of communication to buyers through online storefronts. The interactive procedures within the marketplace contain features like product catalogs, request for information (RFI), rebates and promotions, broker contacts, and product sample requests.

D. Trading Partner Agreements: An Emerging Application

The main objectives of the *trading partner agreements* B2B e-commerce model are to automate the processes for negotiating and enforcing contracts between participating businesses. This model is expected to become more common as extensible markup language (XML) and the e-business XML initiative (ebXML) become more accepted. This worldwide project is attempting to standardize the exchange of e-business data via XML, including electronic contracts and trading partner agreements. Using this model enables customers to submit electronic documents that previously required hard copy signatures via the Internet. An act passed by the United States Congress (in October 2000) gives digital signatures the same legal validity as handwritten signatures. By electronically "clicking" on the "button" entitled "I Accept," the sender inherently agrees to all of the terms and conditions outlined in the agreement. Using this model, business partners can send and receive bids, contracts, and other information required in offering and purchase of products and services.

The agreement ensures that bids, signatures, and other documents related to transactions are genuine when received electronically over the Internet. The agreement is a substitute for all the hard copy forms, ensuring that all obligations created are legally binding for all trading partners. It binds the parties to all the previously agreed upon requirements of the documents and regulations.

The XML, a subset of the standard generalized markup language (SGML), is a recent and flexible technology for creating common information formats that share both the format and the information on the e-commerce infrastructure. The content in terms of what information is being described and transmitted is described by XML. For example, a <BCONTRACT> could indicate that the information transmitted was a business contract. In this case, an XML file is processed purely as information by a program, stored with similar information on another web site, or, similar to an HTML document, displayed using a browser. Using this XML-based model, contracts are transmitted electronically, and many processes between trading partners are performed electronically, including inventory status, shipping logistics,

purchase orders, reservation systems, and electronic payments.

The main advantage of XML over hypertext markup language (HTML) is that it can assign data type definitions to all the data included in a page. This allows the Internet browser to select only the data requested in any given search, leading to ease of data transfer and readability because only the suitable data are transmitted. This may be particularly useful in m-commerce (mobile commerce); XML loads only needed data to the browser, resulting in more efficient and effective searches. This would significantly lower traffic on the Internet and speed up delay times during peak hours. At present, the technology for trading partner agreements is mostly based on EDI technology, either Web-based or proprietary. More and more proprietary EDI applications are being replaced with Web-based EDI. This will enhance ease of use, lower cost, and increase the availability of this technology for smaller and medium-sized corporations.

XI. WIRELESS AND VOICE-BASED E-COMMERCE

Wireless e-commerce based on the wireless application protocol (WAP) has been around for many years. European countries have been using wireless devices for various e-commerce applications for several years. Many telecommunications companies including Nokia have been offering Web-ready cellular phones. Microsoft is offering a wireless version of its Internet Explorer called Mobile Explorer. Motorola, Nokia, and Ericsson are in partnership with Phone.com to offer wireless browsers. Phone.com has a full product line that allows complete information services to be developed and deployed for wireless devices. Major e-commerce companies are developing the simple, text-based interfaces required by today's screen-limited digital phones. Already, amazon.com has made it possible to purchase various products using these wireless devices. On-line brokerage firms such as Charles Schwab offer stock trading using wireless devices. Delta Air Lines is testing a system to offer all flight information through wireless devices. The next step in this revolution is voice-based e-commerce.

Just imagine picking up a phone and accessing a web site and ordering a product. This application already exists. At the core of these new services are voice recognition and text-to-speech technologies that have improved significantly during the past decade. Customers will be able to speak the name of the web site or service they want to access, and the system will recognize the command and respond with spoken words. By using voice commands, consumers would be able to search a database by product name and locate the merchant with the most competitive prices. At present, voice-based e-commerce will be suitable for applications such as the following:

- Placing a stock trade
- Receiving sports scores
- Reserving tickets for local movies
- Buying a book
- Finding directions to a new restaurant

One method to conduct voice-based e-commerce is to use digital wallets (e-wallets) on-line. In addition to financial information these wallets include other related information, such as the customer's address, billing information, driver's license, etc. This information can be conveniently transferred on-line. Digital wallets are created through the customer's PCs and used for voice-based e-commerce transactions. Security features for voice-based e-commerce are expected to include the following:

- Call recognition, so that calls have to be placed from specific mobile devices
- Voice recognition, so that authorizations have to match a specific voice
- Shipping to a set address that cannot be changed by voice

There are already several voice portals on the market. The following are among the most popular:

- BeVocal.com
- InternetSpeech.com
- Talk2.com
- Tellme.com

SEE ALSO THE FOLLOWING ARTICLES

Advertising and Marketing in Electronic Commerce • Business-to-Business Electronic Commerce • Computer History • Electronic Commerce, Infrastructure for • Enterprise Computing • Intranets • Marketing • Mobile and Wireless Networks • Operating Systems • Sales • Service Industries, Electronic Commerce for • Value Chain Analysis

BIBLIOGRAPHY

Afuah, A., and Christopher, L. T. (2000). *Internet business models and strategies.* Boston, MA: McGraw-Hill-Irwin.

Anonymous. (March 2000). Voice-based e-commerce looms large. *E-Commerce Times* available at: http://www.ecommercetimes.com/news/articles2000/000328-1.shtml.

Banham, R. (July 2000). The B-to-B. *Journal of Accountancy*. pp. 26–30.

Bidgoli, H. (2002). *Electronic commerce: Principles and practice*. Academic Press, San Diego, CA: Academic Press.

Blankenhorn, D. (May 1997). GE's e-commerce network opens up to other marketers. NetMarketing available at:http://www.tpnregister.com/tpnr/nw_ra.htm.

Greenberg, P. A. (December 1999). Get ready for wireless e-commerce. *E-Commerce Times*.

Kobielus, J. G. (2001). *BizTalk: implementing business-to-business e-commerce*. Upper Saddle River, NJ: Prentice Hall.

Ovans, A. (May–June, 2000). E-procurement at Schlumberger. *Harvard Business Review,* pp. 21–22.

Porter, M. E. (1985). *Competitive advantage: Creating and sustaining superior performance*. New York: Free Press.

Purchasing online available at: http://www.manufacturing.net/magazine/purchasing/archives/2000/pur0323.00/032isupp.htm.

Electronic Commerce, Infrastructure for

Manish Agrawal
University of South Florida

Chun-Jen Kuo
State University of New York, Buffalo

Kichan Nam
Sogang University, Korea

H. R. Rao
State University of New York, Buffalo

GLOSSARY

application service providers Third-party entities that manage and distribute software-based services and solutions to customers across a wide area network from a central data center.
electronic commerce Conducting business on the Internet and through EDI. Buying and selling products with digital cash is also part of e-commerce.
hyperText transfer protocol The protocol used by the World Wide Web that defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
public key infrastructure A system of digital certification that allows verification and authentiction of each party involved in an Internet transaction.
server A computer or device on a network that manages network resources. Web servers and database servers are very important in e-commerce.
uniform resource locator The global address of documents and other resources on the World Wide Web.
Web browser A software application used to locate and display Web pages, Netscape and Internet Explorer are prime examples.
World Wide Web A network of loosely related Internet servers that support hypertext documents and follow a standard format called HTML (HyperText Markup Language) that supports links to documents, graphics, audio, and video files.

I. INTRODUCTION

A. What Is Electronic Commerce?

1. Defining Electronic Commerce

Broadly defined, electronic commerce (e-commerce) is a modern business methodology that addresses the needs of organizations, merchants, and consumers to cut costs while improving the quality of goods and services and increasing the speed of service delivery. The term particularly applies in the context of the use of computer networks to search and retrieve information in support of human and corporate decision making. In other words, all business activities fulfilled via networks can be attributed as part of e-commerce. These activities include information search, business transactions, funds transfers, etc. The networks mentioned here include the Internet, intranet, and extranet. Each of these networks demonstrates the "Network Effect" to some extent, which states that the utility of a device on a network increases in proportion to the square of the number of users of the network.

E-commerce has a history that dates back about 25 years to electronic data interchange (EDI) systems. EDI can be considered as a very early stage of e-commerce. The Internet was a big force in moving e-commerce beyond the boundaries of proprietary EDI systems to open systems and consumers and businesses all over the world. The next generation of e-commerce includes the use of mobile computing devices (handheld computers, for

example); however, a discussion of wireless (mobile) e-commerce, sometimes called m-commerce, is outside the scope of this article.

2. Electronic Data Interchange (EDI)

In EDI, the electronic equivalents of common business documents, such as request for quotes, purchase orders, and invoices, are transmitted electronically between the EDI-capable companies. These electronic documents are given standardized electronic formats and numbers (referred to as ANSI X12 standard), so everyone involved can correctly interpret the information that is sent to them. Value-added networks (VANs), provided by companies similar to long-distance phone companies or clearinghouses, provide connectivity between EDI-capable companies.

Since EDI runs over proprietary networks, it is typically well regarded in terms of reliability, security and performance. However, EDI deployment and ongoing VAN subscription premiums have proven too onerous for most companies and EDI is being less widely used today.

3. State of the Art—Less Electronic Data Interchange (EDI) and More World Wide Web (WWW) and Internet

We believe the major principle of EDI—reducing the process costs of intercompany trade—will live on and EDI will continue to exist. However, EDI is expensive, complicated, and not user-friendly. In contrast, more and more companies are using World Wide Web-based applications and the Internet for their e-commerce solutions. We believe that as businesses continue to get more comfortable about the security, reliability, and performance of the Internet, the use of EDI and the ever-expensive, proprietary VANs will give way to Internet-facilitated transactions.

Table I is a brief comparison table of EDI and WWW-based applications. (Note: Only the differences between EDI and WWW-based applications are listed.)

B. Why We Need Electronic Commerce

E-business offers value. On the surface it provides a new means of reaching and serving customers, partners, employees, and investors. But beneath the surface, it is transformational. It breaks down barriers between departments, companies and countries. It enables business activities to be partitioned and distributed in limitless ways. It often changes economic fundamentals and challenges us to rethink the way we work, play, and communicate.

Here are some explicit advantages of e-commerce:

- *Eliminate data entry errors.* Using EDI or WWW-based e-commerce applications eliminate, possible human errors and retyping tasks from conventional paper orders or faxes.
- *Cost efficiency.* Data and orders are transferred electronically, thus involving less manpower and reducing paper consumption.
- *Quick response, easy to access.* Networks can carry text, pictures, and video, which improve response time to customers and suppliers. Also Web browsers are familiar to almost everyone and are available everywhere at low cost or even free of charge.
- *Increase business territory and revenue.* Using worldwide Internet virtually expands geographic markets, thus creating more revenues.

Table I Comparison of EDI and WWW-Based Applications

	Advantage	Disadvantage
EDI	1. More secure due to usage of private networks and restricted format and protocol	1. More expensive due to more hardware expenses and EDI agent costs 2. Proprietary user interface, more training time needed 3. Higher up-front cost in initialing capital investment
WWW-based application	1. Browser, as the user interface, decreases end-user training cost 2. Lower up-front installation costs 3. Browser enables easy access of information everywhere	1. More security concerns due to using public Internet

- *Order configuration check.* Many orders need appropriate associated accessories in order to operate properly. On-line configurators can guarantee the right configuration. For example, Cisco's e-business Web site is able to check whether procurement of modules or memories fit a router before submitting the order.
- *Better customer satisfaction.* E-commerce enables customers to do business on-line around the world, around the clock. An on-line order-tracking system enables customers to track their orders 24 hours a day, 7 days a week.

C. Different Electronic Commerce Categories

Generally, e-commerce can be divided into four categories: business-to-business (B2B), business-to-consumer (B2C), consumer-to-consumer (C2C), and consumer-to-business (C2B).

1. Business-to-Business (B2B)

B2B indicates the business activities among companies that use computer technology and the Internet to achieve results. EDI, quick response systems, electronic forms, and on-line customer service are some examples of the technology.

In the category of B2B e-commerce, Internet trading exchanges play a very important role. They are aggregation points that bring buyers and sellers together to create markets for exchanging goods and services. Their job is analogous to the role of Cisco routers for bits on the networks—switching, and routing, exchanges do the same thing for commercial transactions.

The key processes and technologies required to maintain markets include:

1. *Requisition routing and approval.* The purchasing enterprise typically has an internal approval process for orders of different sizes. Procurement software implements the approval process by routing orders to appropriate managers for approval.
2. *Supplier sourcing.* An exchange has to source suppliers to sell through its network, which is part of the value. Much like a distributor, the exchange does the legwork to find the suppliers and get them registered in the marketplace.
3. *Order matching.*
   - *Catalog order.* The buyer browses a catalog to identify a fixed-price item. This is the most popular order-matching technique.
   - *Dynamic pricing.* This is used mostly for products that trade frequently with volatile pricing. The exchange matches the order in real time as bids and quotes come into the marketplace. The volatile pricing might occur from changes in capacity, supply, or demand.
   - *Auction.* This usually involves infrequently traded or unique items that can significantly vary in value depending on the buyer. Equipment disposals are the main market in this category.
   - *Request for proposal.* This technique facilitates complex requisitions in time. It is appropriate for project-oriented work, e.g., system integration and construction.
4. *Fulfillment.* Fulfillment is the most complicated, costly step, but it is also the step with potentially huge cost savings. Fulfillment gets complicated because of exceptions such as backorders, partial shipments, returns, substitute products, incorrect orders, and changed SKUs. Moving the fulfillment on-line would lower the number of exceptions since the buyer or technology will be able to solve many of the issues in real time. On-line fulfillment needs to provide the buyer the ability to retrieve real-time product availability information and to reserve products by serial and bin number before they hit the buy button.
5. *Settlement.* Exchanges largely rely on P-cards (procurement cards which are similar to debit cards) and credit cards for financial settlement of orders. However, credit cards are designed for consumer credit and usually the purchase sizes are small. E-commerce transaction needs larger credit lines and a different fee structure. Now, more sophisticated payment systems are emerging that are more attuned to business commerce. These systems may also have to accommodate barter transactions. Companies like eCredit are building B2B payment networks with fee structures that reflect the lower credit risk of corporate customers.
6. *Content management.* Displaying merchandise for sale through an online catalog is a fundamental requirement without which an exchange has a tough time existing. Catalog management is much more complicated than it sounds. Questions that need to be resolved include where to host the catalog and how often does one need to update information, real-time or periodically? Which option does the customer like most?

2. Business-to-Consumer (B2C)

These are the general business activities between companies and consumers, including on-line shopping, auctioning, retailing, etc. Consumers in B2C are the end users; individuals are provided an efficient, convenient, and low-cost shopping environment that is created by electronic transmitting technology. People buy or bid on things that they want on the Internet. Amazon.com is the representative example of this type of business.

3. Consumer-to-Consumer (C2C)

C2C represents the direct transactions among consumers, i.e., some consumers are sellers while others are buyers. In this category, C2C Web sites play the role of broker that provides a marketplace for seller-consumers to post the items they want to sell and for buyer-consumers to submit orders they want to buy. Generally, buyers submit bid prices and compete with other bidders. EBay.com is the typical example, which allows individuals to sell their personal collections, used items, and so on. The line between B2C and C2C is getting finer as companies use these systems to dispose of excess and used inventory.

4. Consumer-to-Business (C2B)

The market size of C2B is considered the smallest in these four categories of e-commerce. C2B is like a reverse transaction of B2C; consumers go onto the Web site and ask for some specific products, and then the C2B market maker will try to find suppliers who can provide the specific products and match the transaction. SwapIt.com is an example of C2B. The company has partnered with some logistics companies to help facilitate the swapping of unwanted CDs and games for anything in SwapIt's inventory of 50,000 used CDs and games. There is a $2 transaction fee for swaps, and SwapIt tests the CDs for quality control before distribution.

II. COMPONENTS OF ENHANCED E-COMMERCE

As e-commerce systems move from simple informational front ends to add enhanced transaction-oriented features, technology-enabled business components such as methods of e-payment, search engines, intelligent agents, and portal sites have become important in various applications such as customer relationship management initiatives.

A. Electronic Payments

- *Micropayment.* Micropayment generally refers to transactions less than $1, such as the fee for downloading files from the Internet or for reading the contents of paid Web pages. Using conventional processes, banks have to spend $1 to process a 5¢ transaction. Because of the advantages of low-cost cash flow and the fact that conventional banks do not have an efficient mechanism to process small-amount transactions, micropayments emerged and some companies are contributing to provide *electronic tokens* that can be used on the Internet.
- *Electronic Tokens.* An electronic token is the digital form of various payment methods supported by banks and financial institutions. Electronic tokens are used like tokens of currency and can be exchanged between buyers and sellers. Transactions are fulfilled via electronic token transfer. There are two basic types: prepaid electronic tokens in which users need to have credits before using, and post-paid electronic tokens in which users pay the bill after the transaction. Digital cash, debit cards, and electronic cases are prepaid electronic tokens and electronic check and credit card are examples of postpaid electronic tokens.
- *Digital (electronic) cash.* Based on digital signatures (which will be discussed later), digital cash is designed for on-line transactions. Before a purchase, users buy digital cash from on-line currency servers or banks. After digital cash has been purchased, buyers can use it to pay for transactions on-line and be protected by digital signature verification.
- *Electronic check.* Similar to conventional checks, the owner of an electronic check account can issue an electronic document including payer's name, paying bank, check number, payee's title, and the dollar amount. The difference is that an electronic check is sent electronically, is endorsed via digital signature; and needs digital certificates to certify the payer, paying bank, and account. After collection by banks, electronic checks are settled via some kind of automated clearinghouse.
- *Credit and debit card systems.* Card owners of credit cards or debit cards also can use their credit cards or debit cards to make the payments. The transactions which use credit cards or debit cards on-line are secured by Secure Electronic Transaction (SET), which was introduced by VISA Corp., MasterCard Corp., Microsoft, IBM, etc. (we will discuss SET shortly).

- *Smart Card.* Smart card is the same size as a credit card and has a microchip embedded to store personal information or digital cash. It can be used as an identifier, digital wallet, prepaid phone card, or debit card. There are two kinds of smart cards: a touched smart card that has to be read via a card reader and an untouched smart card that can be read via radio wave without touching the card reader.

B. Search Engines

A search engine is a service that searches documents for specified keywords and returns a list of the documents where the keywords were found. Although a search engine is really a general class of programs, the term is often used to specifically describe systems such as AltaVista and Excite that enable users to search for documents on the WWW and USENET newsgroups.

Many search engines, including Yahoo!, started by manually classifying Web sites under different categories to facilitate search. Most search engines that classify Web sites under different categories continue to depend upon manual classification. However, search engines such as Yahoo! have Web sites that have been classified by category as well as sites that do not fall under any classification. The latter are typically located by sending out a spider or crawler to fetch as many documents as possible. The spider retrieves a document and then recursively retrieves all documents linked to that particular document. Another program, called an indexer, then reads these documents and creates an index based on the words contained in each document. Each search engine uses a proprietary algorithm to create its indices such that, ideally, only meaningful results are returned for each query.

C. Portal Sites

A portal site is a Web site or service that offers a broad array of resources and services, such as e-mail, forums, search engines, and on-line shopping malls. Integrators combine content along with a search interface to help the searcher navigate to specific information. The first Web portals were on-line services such as AOL that provided access to the Web, but by now most of the traditional search engines have transformed themselves into Web portals to attract and keep a larger audience.

Portals may be classified as vertical and horizontal. Horizontal portals such as Netscape offer basic services that meet common needs of most users. These include daily news, stock quotes, etc. Vertical portals are highly focused on specific industries and offer specialized services oriented toward meeting the specific needs of participants in an industry. VerticalNet is a company that is attempting to create vertical portals in a number (57 at last count) of industries.

D. Intelligent Agents

Promising developments in the use of the Web are intelligent agents. An agent is software that is capable of hosting itself on other computers and making autonomous decisions on behalf of its creator. Their utility can be as simple as fetching price quotes and information search and as complicated as negotiating purchases on behalf of their creator based on specific criteria. Agents not only collect data, they also classify data. Several kinds of intelligent agents are commonplace today: e-mail agents, news agents, and personal shopping agents, for example. E-mail agents can act as filters to block unwanted mail, or they can be used to prioritize e-mail. News agents search the Web for news that is interesting to a specific consumer. Personal shopping agents scan the Web for the lowest price of a given product. The shopping agent consists of a database with merchants on the Web and how to access their databases. Once customers decide on a product, they are able to key in the product name, part number, or keyword relating to the product and the shopping agent starts checking all the merchants for the product.

E. Consumer Relationship Management

Each of the above components can be utilized to provide immediate, consistent and personalized service on-line. E-commerce can potentially capture the loyalty of today's sophisticated customers. A robust customer information infrastructure can provide many benefits to companies and their customers. Customers can receive the information they need to make better purchasing decisions and effectively resolve problems. As a result, we are likely to see:

- Increased sales
- Improved customer satisfaction
- Higher customer retention
- Improved help desk morale and reduced turnover

(by giving employees the tools they need to do their jobs better)
- Reduced overall cost of customer support (because help desk staff will be able to independently handle more difficult questions and problems)

III. TECHNOLOGY INFRASTRUCTURE COMPONENTS

A. Internet Infrastructure

E-commerce is trading on the Internet; therefore, Internet infrastructure is the first determining factor for successful e-commerce. From 1991 when the Internet first became available for commercial use, both private- and government-sponsored projects started to build a faster and broader Internet.

1. Internet Infrastructure in North America, Europe, and Pacific Asia Areas

In 1993, when the U.S. Government proposed the National Information Infrastructure (NII) project, the Internet became the hottest topic of modern informational society. Governments of North America, Europe, and Pacific Asia areas, followed the concept of NII and projected similar plans within their countries to enforce their network infrastructure. Now, the Internet infrastructure in North America, Europe, and Pacific Asia areas is considered the highest deployed and developed in the world. E-commerce in these regions comprised 94% of all worldwide transactions in the year 2000, according to studies, and the market is expected to grow from $2.9 trillion in 2000 to $9.5 trillion by 2003. Furthermore, by 2003, B2C transactions in these regions will account for 99.9% of all B2C worldwide owing to the better Internet infrastructure.

B. Internal Infrastructure to Support Electronic Commerce

A firm that has built up an e-commerce system after significant expenditures is expected to conduct a significant fraction of its total business on-line. Yet if the site is difficult to access or sends back error messages, the firm has lost an opportunity to build up revenues. Furthermore, the promotional dollars spent to attract consumers to the site may actually send them to a competitor's site because some of the promotional activities affect the entire industry and not just one firm. The keys to building loyalty for a site are simple: it must be ensured that the site is always available and ready to take customer orders. Scalability and availability are two main keys to keeping a site available around the clock.

- *Scalability.* Directing business order traffic to local and geographically distributed mirror servers can dramatically shorten customers' waiting time traversing on the Internet. Also scalability reduces the expenses to sustain a costly, high-speed Internet connection that satisfies heavy peak-time traffic, but has a very low utilization rate during nonpeak times.
- *Availability.* How can an e-commerce site provide around-the-clock uptime to the customer? The answer is to deploy both redundant and load balancing machines in the infrastructure. Redundancy (of servers, network devices, or connections) removes single points of failure. Load balancing, like an air traffic controller, involves redirecting incoming traffic equally to each server and network segment to provide optimal performance for the business site. High availability guarantees that servers and devices will always be available for the customers.

C. Enterprise Middleware

Middleware is the software that sits between network/platform operation systems and the business aware application systems. In other words, middleware is business unaware software that connects applications and users, allowing them to communicate with one another across a network.

Middleware also automates business operations, tying together a company's back-end and front-end operations. It works like glue that connects disparate applications such as Web-based applications and older mainframe-based systems. It also lets companies continue to benefit from their investments in legacy systems, while allowing them to connect with newer systems and the latest developments that drive newer applications. Figure 1 describes the position of middleware in relation to other information technology blocks in an e-commerce infrastructure.

The main types of middleware are:

- *Primitive services middleware*—terminal emulation, e-mail, etc.

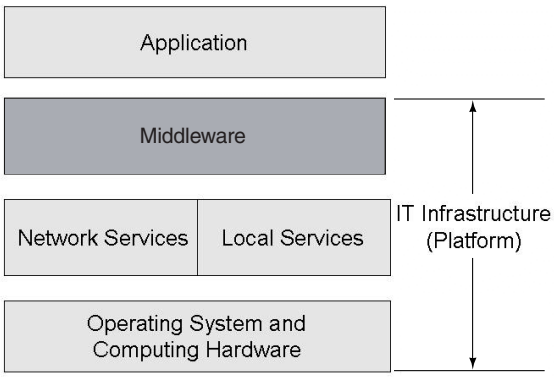


Figure 1 Information technology (IT) building blocks.

- *Basic client/server middleware*—remote procedure call (RPC); remote data access (RDA); message-oriented middleware (MOM), security, directory, and time services in DCE (Distributes Computing Environment); etc.
- *Web middleware*—Web servers, Web browsers, search engines, HTML and scripting languages, database gateways, etc.
- *Distributed data management middleware*—allows users to store their data in a set of computers but not have to know how and where data are stored
- *Distributed transaction processing middleware*—sits between a requesting client program and databases, ensuring that all databases are updated properly
- *Distributed object middleware*—extends the scope of local OO software to distributed objects management. For example, from Smalltalk and C++ (running on local platforms) to CORBA (Common Object Request Broker Architecture) on distributed platforms.

D. Virtual Private Networks (VPNs)

1. How Do Virtual Private Networks Help E-Commerce?

Security is always a major concern for companies doing business. With VPNs, companies can save a lot of infrastructure cost of leased lines connected to their partners directly to ensure data flow would not be eavesdropped on the halfway. Furthermore, a VPN can fasten the speed of acquiring business partners by simply adding some equipment and software and increasing the scalability of expanding new mirror sites or branch offices globally. Figure 2 is an example of a VPN between a central office and two branch sites. Two tunnels (which act like two leased lines) are created from two branches to the central site via Internet or commercial network provided by the network service provider (NSP). Using the same architecture, either business partners or branches can be connected privately under a public network by simply adding more tunnels.

2. Virtual Private Network Technology

A VPN is a connection that has the appearance and many of the advantages of a dedicated link, but occurs over a shared network. Using a technique called tunneling, data packets are transmitted across a public routed network, such as the Internet or other commercially available network, in a private tunnel that simulates a point-to-point connection. A VPN enables

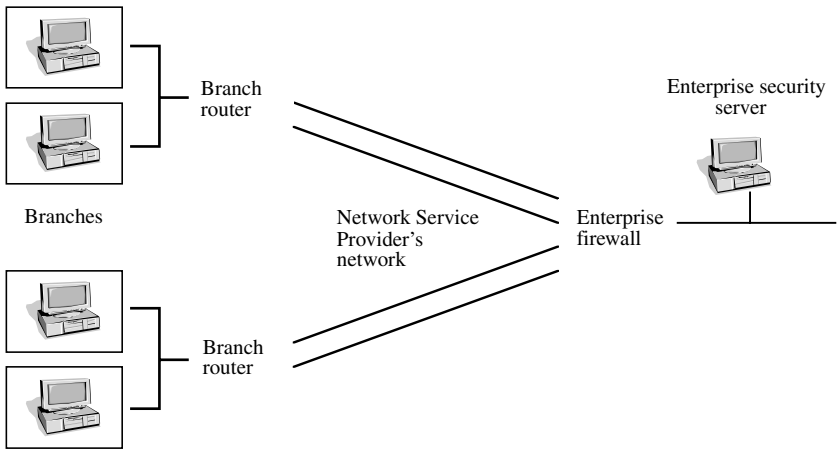


Figure 2 A VPN example between the central office and two branch sites.

network traffic from many sources to travel by separate tunnels across the same infrastructure. It allows network protocols to traverse incompatible infrastructures. It also enables traffic from many sources to be differentiated, so it is directed to specific destinations and receives specific levels of service. VPN capabilities can be added to existing networking equipment through a software or board-level upgrade.

VPNs are based on familiar networking technology and protocols. In the case of a remote access VPN, for example, the remote access client is still sending a stream of point-to-point protocol (PPP) packets to a remote access server. Similarly, in the case of LAN-to-LAN virtual leased lines, a router on one LAN is still sending PPP packets to a router on another LAN. Instead of going across a dedicated line, the PPP packets are going across a tunnel over a shared network.

The most widely accepted method of creating industry-standard VPN tunnels is by encapsulating network protocols (IP, IPX, AppleTalk, and more) inside the PPP and then encapsulating the entire package inside a tunneling protocol, which is typically IP but could also be ATM or frame relay. This approach is called layer 2 tunneling because the passenger is a layer 2 protocol packet.

IV. ELECTRONIC COMMERCE DEVELOPMENT SOFTWARE

E-commerce development software is composed of database software, high-level programming software, as well as software to develop Web interfaces

A. Database Management System (DBMS)

DBMSs are a collection of programs that enable users to store, modify, and extract information from a database. There are many different types of DBMSs, ranging from small systems that run on personal computers to huge systems that run on mainframes. The following are examples of database applications:

- Computerized library systems
- Automated teller machines
- Flight reservation systems
- Computerized parts inventory systems

From a technical standpoint, DBMSs can differ widely. The terms *relational, network, flat,* and *hierarchical* all refer to the way a DBMS organizes information internally. The internal organization can affect how quickly and flexibly you can extract information.

Requests for information from a database are made in the form of a *query,* which is a stylized question. For example, the query

```
SELECT ALL WHERE LASTNAME = "SMITH"
AND AGE > 30
```

requests all records in which the `LASTNAME` field is `SMITH` and the `AGE` field is greater than 30. The set of rules for constructing queries is known as *query language.* Different DBMSs support different query languages, although there is a standardized database query language called *structured query language* (SQL). Sophisticated languages for managing database systems are called *fourth-generation languages,* or 4GLs for short.

The information from a database can be presented in a variety of formats. Most DBMSs include a *report writer program* that enables the user to output data in the form of a report. Many DBMSs also include a graphics component that enables the user to output information in the form of graphs and charts.

B. High-Level Programming Software

A number of development environments are available for developing e-commerce systems. The most popular environments are those developed by Sun Microsystems and Microsoft.

1. Java, Jini, and JavaBeans

a. Java

Java is a high-level programming language developed by Sun Microsystems. Java was originally called *OAK* and was designed for handheld devices and set-top boxes. *OAK* was unsuccessful so in 1995 Sun Microsystems changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web. Java is an object-oriented language similar to C++, but simplified to eliminate language features that cause common programming errors. Java source code files (files with a *.java* extension) are compiled into a format called *bytecode* (files with a *.class* extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and run-time environments, known as *Java virtual machines* (VMs), exist for most operating systems, including UNIX, the Macintosh OS, and Windows. Bytecode can also be converted directly into machine language instructions by a just-in-time compiler (JIT). Java is a general pur-

pose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on a personal computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

b. Jini

Jini (pronounced GEE-nee; loosely derived from the Arabic for magician) is a kind of software from Sun Microsystems that seeks to simplify the connection and sharing of devices, such as printers and disk drives, on a network.

Built on the Java standard, Jini works by passing snippets of programs, called applets, back and forth among devices. Any computer that can run Java (Java run-time environment, JRE) will be able to access the code and data that passes among devices. Jini creates a network consisting of all types of digital devices without extensive planning, installation or human intervention. Thus, an impromptu community is created when multiple devices are networked and share services. They do not need to have prior knowledge of each other to participate. Jini allows the simplified delivery of products and services over a network.

c. JavaBeans

JavaBeans is a specification developed by Sun Microsystems that defines how Java objects interact. An object that conforms to this specification is called a *JavaBean,* and is similar to an ActiveX control. It can be used by any application that understands the JavaBeans format. The principal difference between ActiveX controls and JavaBeans is that ActiveX controls can be developed in any programming language but executed only on a Windows platform, whereas JavaBeans can be developed only in Java but can run on any platform. The JavaBeans component model enables developers to write reusable components and helps reduce the time to develop applications.

2. Active Server Pages

Active Server Pages *(asp)* are a specification for dynamically created Web pages with an *.asp* extension that utilizes ActiveX scripting—usually VB Script or Jscript code. When a browser requests an *asp* page, the Web server generates a page with hypertext markup language (HTML) code and sends it back to the browser. So asp pages are similar to CGI scripts, but they enable Visual Basic programmers to work with familiar tools.

C. Technologies for Developing Web Interfaces

1. Browsers

A browser allows people to view pages of information on the World Wide Web (a term first coined by Tim Berners-Lee of CERN, Geneva). The first browser, called Mosaic, was developed in 1993 at the University of Illinois by Marc Andreessen, co-founder of Netscape Communications Corp. and others. This development resulted in an explosion of the popularity of the Web, and as interest mounted, other software developers created more advanced browsers. Instead of reading text, people using the latest browsers can see animation, watch video, and listen to audio. Browsers are fast becoming enablers of e-commerce across the globe.

2. Hypertext Markup Language (HTML)

The browser is in essence client side software that provides a graphical user interface (GUI) to the user and presents incoming HTML information in a user comprehensible format. Information can be presented to the user in various formats ranging from text, graphics, audio, video, etc. A browser handles most of the details of document access and display. Consequently, a browser contains several large software components that work. Each browser must contain an HTML interpreter to display documents. This corresponds to the presentation layer that renders pages on the screen for users to enjoy. Most of the efforts in upgrading Web browsers have focused on this layer, introducing many options for fancy layout ranging from the annoying (animated GIFs) to the useful style sheets.

Input to an HTML interpreter consists of a document that conforms to HTML syntax. The interpreter translates HTML specs into commands that are appropriate for the users' screen. For example, if it encounters a heading tag in the document, the interpreter changes the size of text used to display the heading. Similarly, if it encounters a break tag, the interpreter begins a new line of output. One of the most important functions in an HTML interpreter involves selectable items. The interpreter stores information about the relationship between positions on the display and anchored items in the HTML document. When the user selects an item with the mouse, the browser uses the current cursor position and the stored position information to determine which item the user has selected.

3. Javascript

Javascript was a scripting language developed by Netscape to enable Web authors to design interactive sites. Although it shares many of the features and structures of the full Java language, it was developed independently. Javascript can interact with HTML source code, enabling Web authors to spice up their sites with dynamic content. JavaScript is endorsed by a number of software companies and is an open language that anyone can use without purchasing a license. It is supported by recent browsers from Netscape and Microsoft, though Internet Explorer supports only a subset, which Microsoft calls Jscript.

4. Dynamic Web

In the early years of e-commerce, most organizations developed static Web pages for information dissemination. The only action that a Web server had to do was locate the static Web page on the hard disk and pass it on to the Web browser. However, this was soon found to be too limited, and firms wanted to tie the Web pages to databases, where information could be dynamically changed. Dynamic Web pages can be built using cascading style sheets (CSS), the document object model, and Javascript, for instance. CSSs are templates that contain a set of rules that specify the rendering of various HTML elements. In the document object model (DOM) a document contains objects that can be manipulated. The DOM can be used to remove, alter, or add an element to a given document. The DOM can be utilized to get for example a list of all the "H2" elements in a document. Javascript is a scripting language that can be used for modifying objects in the DOM model. The combination of HTML, CSS, DOM, and Javascript allows the creation of dynamic and interactive Web pages, and is known as dynamic HTML.

5. eXtensible Markup Language (XML)

With the advent of e-commerce requiring the exchange of critical business transaction information among transacting parties, the simplicity of HTML has prompted the industry to consider a more useful and extensible language for the Web. A new standard called the extensible markup language (XML) is being developed under the aegis of the W3C to accommodate the diverse needs of data format and information presentation to satisfy business transactions. XML, like HTML, is a subset of the standard generalized markup language (SGML). XML is a simplified subset of SGML that allows content providers, programmers, and integrators to define customized tags and document types to support virtually any kind of data as and when needed. For example, once a table is generated in HTML format using data obtained from a database, without a great deal of effort it is difficult to import that data back into another database as the database schema or structure is lost during the HTML conversion process. This represents a severe limitation of HTML, specifically for exchanging business data among transacting parties. The XML technology standard basically promises the design of Web-enabled systems that enable effortless exchange of data across the Internet, intranets, and extranets using the simple browser-based technology. The XML standard is written using a subset of the document style semantics and specification language (DSSSL). XML hyperlinking is a subset of HyTime, an ISO standard for hypertext, hypermedia, and time-based multimedia. XML offers improvements such as bidirectional links that can be specified and managed outside the document. The healthcare industry has accepted XML as the solution to making complex patient information available and portable among concerned parties. XML essentially allows delivery of information in a form that can be manipulated without further interaction with the server or the network. Additionally, with the ability to customize tags to manipulate data in a more meaningful way to be presented to the user, XML promises to provide the much needed flexibility in business transactions. For example, spreadsheet type data can be downloaded on the client machine with its schema intact, thereby allowing the creation of different views of the data locally to satisfy the information needs of the user. Microsoft has developed its channel definition format (CDF) for its push technology based on XML. At the same time, Netscape has developed its resource definition framework (RDF) on XML as well

6. Plug-ins

Some software programs called plug-ins or add-ons extend the multimedia capabilities of browsers. They enable Web page developers to add rich graphics, motion video, and synchronized audio on their pages. Unfortunately, there is yet no agreement in video and audio standards, and a number of software packages are available in the market. To increase their popularity, the client side software is generally available for free download from the company's Web site, while the server side is sold to developers.

V. INTERNET SECURITY

With the stupendous amount of money involved in the transactions on a global scale, security becomes a very important aspect of e-commerce transactions. Firewalls, cryptographic tools, and virus-fighting tools, for instance, would each make an impact on a worry-free transaction for the consumer.

Security exists both at the server level as well as at the client or user level. Digital certificates are the most commonly used method for ensuring security at the client or user level. A digital certificate is a file that is encrypted and password protected. This file may contain information about the owner of the certificate, both publicly known information such as an e-mail address and private information such as a credit card number. Digital certificates are used to secure the communication between browsers and servers, or customers and merchants using secure socket layer encryption or SET encryption. These digital certificates are issued by a trusted third party called the Certification authority. Different levels of certification exist, for example Classes 1, 2, and 3. Class 1 certificates are the easiest to obtain (only a valid e-mail is required), while Classes 2 and 3 require higher levels of identification.

Security at the server level is most commonly implemented by installing firewalls. Firewalls implement access control policies which allow users to access certain resources on other networks. They are placed in the data path between internal and external clients and servers. Firewalls work at the transmission control protocol/Internet protocol (TCP/IP) level, allowing ports to be opened and closed.

A. Firewalls

A firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software or in a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially *intranets*. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.

The basic firewall example of single layer architecture is shown in Fig. 3.

There are several types of firewall techniques:

- *Packet filter.* Looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.
- *Application gateway.* Applies security mechanisms to specific applications, such as FTP and Telnet servers. This is very effective, but it can impose a performance degradation.
- *Circuit-level gateway.* Applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.
- *Proxy server.* Intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses.

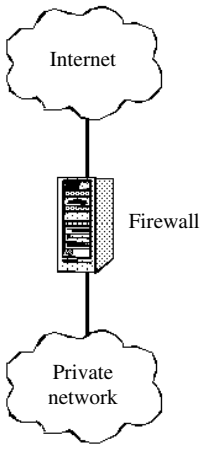


Figure 3 A simple firewall example of single layer architecture.

Let us look at the simplest case. As technique 4 shows, a proxy service requires two components: a proxy server and a proxy client. In this situation, the *proxy server* runs on the dual-homed host (Fig. 4). A *proxy client* is a special version of a normal client program (i.e., a Telnet or FTP client) that talks to the proxy server rather than to the "real" server out on the Internet. The proxy server evaluates requests from the proxy client and decides which to approve and which to deny. If a request is approved, the proxy server contacts the real server on behalf of the client (thus the term "proxy") and proceeds to relay requests from the proxy client to the real server, and responses from the real server to the proxy client.

In practice, many firewalls use two or more of these techniques in concert. A firewall is considered the first line of defense in protecting private information. For greater security, data can be encrypted.

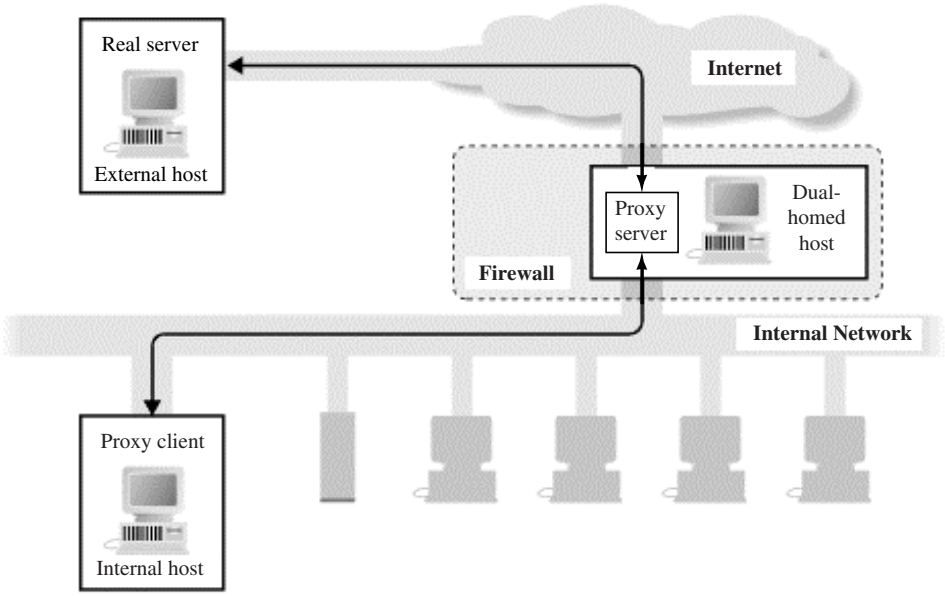


Figure 4 Firewall example of proxy server with a dual-homed host.

1. Cryptographic Tools—Secure Electronic Transaction

Secure electronic transaction (SET) is a standard that will enable secure credit card transactions on the Internet. Virtually all the major players in the e- commerce arena, including Microsoft, Netscape, Visa, and MasterCard, have endorsed SET. By employing *digital signatures,* SET will enable merchants to verify that buyers are who they claim to be.

Digital signatures are explained in greater detail later. The SET protocol is an attempt to eliminate some of the security concerns about using credit cards on the Web. It operates by hiding credit card information from the merchants and directly connects to the credit card company that conveys to the merchant whether or not the card is valid. If the merchant receives an authorization from the card company, the merchant goes ahead and fulfils the order and receives the payment from the credit card. Since the card information is not visible to anyone but the user and the card vendor, it is expected to lead to safer transactions on the Web.

Figure 5 shows the communications and operations taking place between the various actors in electronic trading.

The elements of Fig. 5 are explained below:

- *Customer.* User who uses a card supplied by his or her bank. SET guarantees the confidentiality of the card data in communications between customer and merchant.
- *Merchant.* Organization offering goods or services against payment. SET enables the merchant to offer electronic interactions which are secure for customers' use.
- *Acquirer.* Financial institution which opens an account with a merchant and processes card authorizations and payments.
- *Passway.* Device operated by an acquirer or a third party which is used to process payment messages from merchants (including customers' payment instructions). The chief mission of this system is to enable financial institutions to accept electronic transactions from merchants operating on open networks (e.g., Internet) by controlling access without disrupting their present host systems.
- *Certification Authority.* Agent of one or more card brands which creates and distributes electronic certificates to customers, merchants, and passways.

As Fig. 5 shows, the customer interacts directly with the merchant's systems. This interface supports the customer segment of the payment protocol and enables the customer to initiate payment and to receive the status and confirmation of the order. The customer also has an indirect interface with the acquirer through the merchant. This interface supports the data which are sent in code through the merchant and can only be decrypted by the passway.

At the same time the merchant interacts with the acquirer through the passway using the payment protocol to receive electronic payment authorizations

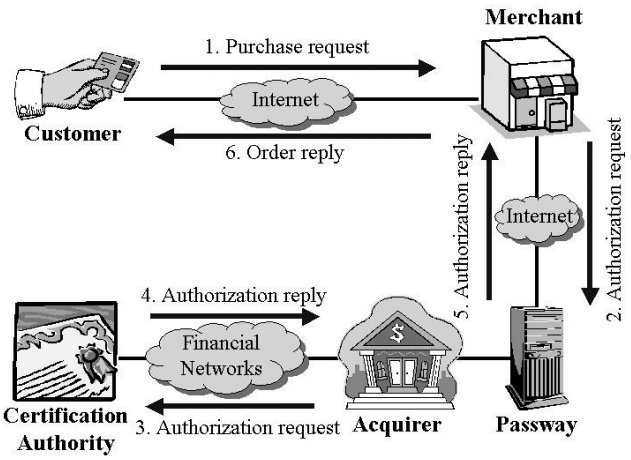


Figure 5 Elements of electronic trading according to SET.

and capture services. The number of passways with which a merchant interacts depends on the specific electronic trading scenario concerned. In the simplest example, the merchant will interact with a single passway which processes all card brands. However, a merchant may be connected to several acquirers, each processing different card brands, and one acquirer may operate in several different passways. Moreover, SET provides an option whereby the acquirer sends the customer's payment information to the merchant, encrypted with the merchant's key.

In the scenario depicted in Fig. 5, the SET protocol is used in the flow of messages between customer, merchant, and passway. Once the messages have been processed by the passway, interbank communications follow conventional payment authorization and capture procedures.

2. Public-Key Cryptography

A *public-key cryptosystem* is one in which messages encrypted with one key can only be decrypted with a second key, and vice versa. A strong public-key system is one in which possession of both the algorithm and the one key gives no useful information about the other key and thus no clues as to how to decrypt the message. The system gets its name from the idea that the user will publish one key (the public key), but keep the other one secret (the private key). The world can use the public key to send messages that only the private key owner can read; the private key can be used to send messages that could only have been sent by the key owner.

With the aid of public-key cryptography it is possible to establish a secure line of communication with anyone who is using a compatible decryption program or other device. The sender and the receiver no longer need a secure way to agree on a shared key.

A third-party registry of public keys is required in this schema. A certification authority (CA) often represents the third party to accept the registration and store the public keys. However, there is still the problem of identifying the registered information as correct or not. Unless the registry also certifies the accuracy of the information it contains (e.g., check the information via the Social Security Bureau), a probability of public key fraud is still possible.

3. Virus Fighting

In May 2000, a virus called "I love you" appeared as an e-mail attachment to many computer users. It modified the system registry of the computers and hid or rewrote many files on the computers and caused hundreds of millions of dollars in damages in just a few hours before it was detected. Such incidents demonstrate the necessity of protecting computer systems against malicious computer programs called viruses that can irrevocably damage data and software on computers. The importance of computer virus protection may be gauged from the fact that the National Infrastructure Protection Center routinely provides warnings of such attacks and monitors them.

To protect against viruses, companies use software programs such as those distributed by McAfee or Symantec.

VI. TRUST

Although estimates vary, it is widely agreed that e-commerce over distributed networks, such as the Internet, is set for explosive growth in the new millennium. E-commerce via the Internet presents challenges that are not encountered by traditional transactions such as face-to-face sales and telephone sales. One of the major challenges in this arena is trust.

A. Problems Encountered in Electronic Commerce Transactions

Generally, there are some problems that e-commerce would present.

1. Basic Transactional Issues

- How to move value.
- How to ensure that communications are secure from eavesdroppers.

2. Merchant's Desires

- *Authentication.* Knowing the buyer's identity before making the sale may assist in proof of order and guarantee of payment. The merchant also may wish to build up a database of customers and their buying profiles.
- *Certification.* The merchant may need proof that the buyer possesses an attribute required to authorize the sale. For example, some goods may only be sold to those licensed to use them; other goods require that the purchaser be over 18 years old. Some products cannot be sold in some parts of the country, or others cannot be exported.
- *Confirmation.* The merchant needs to be able to prove to any third party involved in the transaction (such as a credit card company) that the customer did indeed authorize the payment.
- *Nonrepudiation.* The merchant wants protection against the customer's unjustified denial that he or she placed the order or that the goods were not delivered.
- *Payment.* The merchant needs assurance that payment will be made. This can be achieved by making the payment before sale, at the time of sale, or by provision of a payment guarantee. A credit reference by a trusted third party provides a lesser form of assurance, but it at least demonstrates that the buyer is capable of making the payment.
- *Anonymity.* In some cases, the merchant may want to control the amount of transactional information disclosed to the customer.

3. Buyer's Desires

- *Authentication.* Confirming the seller's identity prior to purchase helps ensure that goods will be genuine and that service or warranties will be provided as advertised.
- *Integrity.* Protection against unauthorized payments.
- *Recourse.* Comfort that there is recourse if the seller fails to perform or deliver.
- *Confirmation.* A receipt.
- *Privacy.* Control over the amount of buyer/transactional information disclosed to third parties.
- *Anonymity.* Control over the amount of transactional information disclosed to the merchant.

Solving such problems frequently requires the intervention of a trusted third party who is a certificate-issuing CA. Issuing certificates entails the creation of new entities, new businesses, and new relationships for which the duties and liabilities are currently uncertain.

B. Transaction Issues: Moving Value and Authentication

To illustrate concepts in this section, we use the oft-used story of two individuals, Alice and Bob. If Alice has no hardware available to her other than her computer, she can choose to move value to Bob across the Internet with a debit card, a credit card, or electronic cash.

- *Debit cards and credit cards.* Today, the simplest way for Alice to pay Bob across the Internet is to use a debit card or credit card. This payment mechanism has the great virtue of familiarity. It uses established mechanisms to apportion risk of nonpayment and repudiation. Although it is vulnerable to eavesdropping, the risk may be smaller than commonly believed.

  Though it is not easy to obtain credit card numbers by monitoring large volumes of Internet traffic, if Alice sends out unencrypted credit card information on the Internet she takes a chance that a third party will intercept the information.

  If Alice wants greater security, she can encrypt her credit card data before sending it. Similarly, Bob may want assurances that Alice is who she purports to be. Bob may want Alice to send her order encrypted with her private key, thus uniquely identifying the order as emanating from her. For a greater level of security, Alice and Bob may require that identifying certificates from a reputable CA accompany the exchange of public keys.
- *Electronic Cash.* Electronic cash implementations vary. While generalizations are hazardous, most true digital cash systems that are entirely software based (for example, do not rely on a smart card or other physical token to provide authentication or to store value) use some variation of the digital coin. A digital coin is a sequence of bits, perhaps signed with an issuing financial institution's private key, which represents a claim of value.

  Software-based digital coins are potentially suitable for small transactions, such as charging a penny or less to view a Web page, where credit cards would be prohibitively expensive. Unfortunately, since bits are easy to copy, digital coin schemes require fairly elaborate mechanisms

to prevent a coin from being spent more than once. One method of preventing double spending is to require that coins be cleared in real time. If Alice offers a coin to Bob, Bob immediately accesses the issuing bank to make sure that the coin is valid and has not previously been spent. A necessary consequence of this protocol is that if Alice uses a digital coin to pay Bob, Bob cannot spend it directly. Instead, Bob must either deposit the coin in an account at the issuer or turn it in for another digital coin or conventional money. An on-line clearing system can be configured to ensure that the bank does not know who gave Bob the coin (payer anonymity), but the bank will know that Bob received the coin (no payee anonymity).

1. Confirmation Issues: Nonrepudiation, Receipt, and Resource

All that Alice needs in order to prove that Bob made a promise to buy or to pay is a message including the promise signed with Bob's digital signature. The issue of proving the promise is separate from whether a digital signature is a signature for legal rules that require a writing bear a signature. Alice will find it less cumbersome to prove Bob's promise if she has access to a certificate, valid at the time of Bob's promise, that links Bob to the signature appearing on the message. However, a certificate may not be strictly necessary depending on the payment mechanism and the nature of the transaction.

Debit and credit cards leave an information trail that can assist Alice in finding Bob, and vice versa. Because a payer might have an anonymous or pseudonymous debit/credit card, or because a payee might have disappeared in the time since the transaction was recorded, the trail is not perfect. However, the trail of information is significant and not much different from what would likely be in a certificate, so it is likely to make the certificate somewhat redundant.

Digital cash can be designed to protect the anonymity of the payer who does not double spend. A prudent payee who is tendered digital cash with this anonymizing feature may seek an identifying certificate from the payer if the transaction makes it important to know him or her. As most digital cash schemes do not protect the anonymity of the payee, the payer will request an identifying certificate only if the cost of the certificate is less than the expected value of the cost of persuading the bank to release the payee's identity on an occasion where this might be needed, adjusted for the danger that the payee will get away before being identified. The cheaper and quicker it is to use a certificate, the more likely it will be used.

2. More Than Security

Most e-commerce businesses claim they are using the most up-to-date cryptographic technology such as SET, public and private keys, and digital signature to ensure the transaction secure. However, how many people do transact on-line? According to the eCommerce Trust Study in 1999 from Cheskin Research, "only 10% of the respondents perceived little or no risk when purchasing on the Web. For the rest, issues of trust, particularly about security of personal information, were mentioned important concerns. . . . 23% of respondents felt threatened by hackers and 16% were concerned about people or firms obtaining and abusing their personal information." (Cheskin Research, Redwood Shores, CA).

Why don't people trust online transactions, or the Internet? By now it is well known that the Internet is a global, but insecure, network. It is also increasingly well understood that cryptography can contribute greatly to the transactional security that Internet commerce so obviously lacks. What is less well understood is that cryptography is only part of the security story. Many cryptographic protocols for secure electronic transactions require at least one trusted third party to the transaction, such as a bank or a CA. These partly cryptographic, partly social, protocols require new entities, or new relationships with existing entities, but the duties and liabilities of those entities are uncertain. Until these uncertainties are resolved, they risk inhibiting the spread of the most interesting forms of e-commerce and causing unnecessary litigation.

3. More Than Privacy

Many Web sites in Internet do not post their policies governing privacy and use of garnered information. *Business Week* examined the 100 top Web sites and discovered that only 43% posted privacy policies. Some of these policies were not only difficult to find, but were inconsistent in their explanations of how information is tracked and utilized.

Consumer trusts issues are not limited to privacy. Users are also worried that hack attacks could compromise their credit card data, and poor inventory management could prevent timely delivery of purchased merchandise. On-line merchants of all stripes express a need to convince shoppers that they are just as trustworthy as their off-line counterparts.

In fact, some surveys suggest that privacy is not always a top consumer concern. Take the recent Amazon.com controversy. Despite a stated policy that it would never sell customer data, the e-commerce giant said it might sell such data if the company were acquired. This happened just as Toysmart.com came under fire for actively seeking to sell its customer data as part of its liquidation, even after it had vowed never to do so. However, though this was a big story in the media, consumers didn't care much about it They were more worried about if somebody could hack into Amazon and steal their credit card information.

4. The Essential Role of Trusted Third Parties

Persons who are not previously acquainted, but wish to transact with one another via computer networks such as the Internet will need a means of identifying or authenticating each other. One means of achieving this is to introduce a trusted third party into the bilateral relationship. This third party, a CA, can vouch for a party by issuing a certificate identifying him or her or attesting that he or she possesses a necessary qualification or attribute. CAs may become essential to much, but not all, e- commerce. Along with the rapid growth of e-commerce, the demand for CA's services should grow rapidly as well. Now there are some third party sites such as TrustE and VeriSign that are well known, and people are becoming willing to trusted the Web sites that are certified by these third parties.

VII. PERSONNEL

A. Information Technology Staff: In-House Building or Outsourcing?

Outsourcing has traditionally been considered as a cost-cutting measure, a solution to the perennial shortage of IT workers and an efficient mechanism to clear off noncritical activities. However, increasingly, companies are looking at outsourcing to handle core functions in their organizations. This is even more so in the arena of e-commerce outsourcing.

Since e-commerce involves many fundamental changes in business and IT practices coupled with extreme time pressures, it is difficult for organizations to develop significant levels of competence in e-commerce systems development using internal resources alone. The time to deploy an application has a very significant impact on how quickly the organization can benefit from it. Unfortunately, large and complex implementations can take months or years to implement e-commerce systems, especially when IT professionals are in short supply, users are located around the world, and disparate new business units must be brought on-line. Organizations are finding that they cannot upgrade infrastructures or adopt technology fast enough to keep pace with current technology development. Also, it is increasingly hard to find and retain top IT talent in today's market. Software developers, engineers, and consultants are in short supply, and companies are continually faced with the prospect of losing their most talented people to competitors.

Under this dynamic environment, application service providers (ASPs) emerge as an alternative solution for managing an IT infrastructure. An ASP is an agent or broker that assembles functionality needed by enterprises and packages it with outsourced maintenance and other services. It differs from data center or systems outsourcing where the provider runs the data center and the applications for a specific customer. It is estimated that the size of the application outsourcing market and its growth potential ranges from $150 million to $2.7 billion in 1999 to $2 billion to $30 billion or more in 2003. The wide range in estimates is due to a lack of consensus on the definition of an ASP.

From the end-user's perspective, one of the benefits of the ASP model is that it allows businesses to leverage the technologies, processes, and expertise of the leading providers of enterprise applications without having to make investment in technology. Offerings from ASPs are currently centered on enterprise applications for small to midsize enterprises, though increasingly, Fortune 1000 organizations are adopting the model too. The early customers of the model were dot.com firms and firms that did not have to integrate the hosted applications with many other systems. In the latter half of 2000, firms began to look at ASP vendors as a part of the enterprise resource planning (ERP) software selection process, and e-commerce applications were also popular choices for small to midsized companies to outsource. ASP-based e-commerce applications enable companies to concentrate on their sales and marketing strategies, rather than devoting scarce resources to an IT staff that will subsequently be faced with challenges of rapid technological obsolescence and shrinking software release cycles.

B. Laws and Regulations

The rapid developments in e-commerce have necessitated changes in the law to support on-line transactions. On June 30, 2000, the Electronic Signatures in

Global and National Commerce (e-sign) Act was passed by the president of the United States to give electronic signatures the same legal status as those written in ink on paper. It took effect on October 1, 2000. This is expected to make it easier and faster to conduct business electronically. A sample application of the e-sign act is that on-line brokerages can allow customers to open accounts without mailing or faxing their signatures to the firm. It is estimated that in a number of industries, as many as 35% of the customers interested in opening online accounts do not do so because they do not send back their signed forms.

The e-sign act defines an electronic signature as anything that the two parties to a contract agree upon, including electronic sounds and symbols that are attached to or logically associated with a contract and executed by a person with the intent to sign the accord. In fact, even passwords created by users could serve as electronic signatures.

An electronic signature, therefore, could simply be the e-mail signature attached by many users to their e-mails. For a document to be legally binding in court, someone needs to be able to authenticate that the document was indeed signed by the person who claims to have signed it. More sophisticated systems create digital signatures by applying mathematical algorithms to a document to generate "digital signatures" that can help determine whether the document has been modified after the signature was created. The rapid development of technologies implies that consumers and businesses alike will very soon be able to use software to generate and use digital signatures.

A number of vendors offer digital signature technology, including Approvelt, Cyber-Sign, and PenOp. Others such as Entrust, RSA Technologies, Verisign, and Xcert act as CAs to establish that the parties are whom they say they are.

VIII. CONCLUSION

The previous sections give an outline of the infrastructure of e-commerce. When a business decides to move onto the Internet and take advantage of e-commerce, issues related to hardware, software, and applications and to human resources need to be considered.

A. Hardware

Hardware such as servers, firewall, network infrastructure, etc. are fundamental elements for an e-commerce implementation. However, there is a variety of hardware to choose from, for example, servers range from high-performance, high-cost systems such as the Ultra 2 workstation (Sun Microsystems) to a Pentium III PC with Linux operating system that offers relatively lower performance but at a much lower cost.The same applies to firewalls, network infrastructure and other related hardware. The final choice depends on the business plan and future requirements. A professional consulting company can be helpful, though they usually can be expensive.

B. Software and Application

A new choice is becoming available with regard to software applications. ASPs are third-party entities that manage and distribute software-based services and solutions to customers across a wide area network from a central data center. In essence, ASPs are a way for companies to outsource some or almost all aspects of their information technology needs. Software solutions may also be obtained from e-commerce consulting companies, who can also provide intangible services such as consulting, product installation and maintenance. The rapid adoption of Internet technology has opened a whole new competitive environment that enables even the smallest companies to compete effectively against larger competitors. In addition, the Internet has opened new avenues for reaching trading partners of all sizes cost effectively and with minimal or no integration efforts. All of these achievements can be simply reached by deploying off-the-shelf software developed by some innovated companies dedicated in the e-commerce territory.

E-commerce software solutions such as transactions management; purchasing management; catalog management; logistics management; warehouse management; customer relationship management; Web hosting; collaboration management between buyers, suppliers, and carriers; etc. all can be acquired directly via e-commerce solution providers and adopted into existing computing environment with relatively few modifications and integration problems.

C. Human Resources

The availability of skilled personnel to develop and maintain systems is a critical factor for the success of deploying e-commerce applications. Hardware and software are relatively infrequent purchases; however, appropriately skilled personnel are essential to keep all of these working. The shortage of skilled personnel is one of the key drivers toward the trend to outsourcing mission-critical applications in addition to noncore activities.

For the development of e-commerce information systems, the overall technology architecture should meet the real business needs that the company has defined. It involves determining individual integration between the application and data sources, the application and back-end software, and the diverse back-end systems. Every decision should be made with an attention toward not only the functionality of the application, but also the ability of the platforms to scale up in the future.

In designing an e-commerce infrastructure, it is required to determine the business and technology components that will make up the final platform. However, it is important to distinguish between the description of this type of logical architecture and the actual planning of the physical architecture of the application. Logical architecture issues include how to distribute the application to take advantage of networked resources, the network infrastructure to employ, and the location and type of the data resources that the application requires. This stage includes settling the final ownership issues for the application, data, business knowledge, hardware, and human resources necessary to implement and support the application. Considering the rapid changes in IT, outsourcing is expected to be a popular way of building e-commerce systems. Firms should be prepared to manage outsourcing partners in order to manage e-commerce systems effectively.

SEE ALSO THE FOLLOWING ARTICLES

Advertising and Marketing in Electronic Commerce • Business-to-Business Electronic Commerce • Digital Goods: An Economic Perspective • Electronic Data Interchange • Electronic Payment Systems • End-User Computing Concepts • Internet, Overview • Marketing

BIBLIOGRAPHY

Amor, D. (2000). *The e-Business Revolution—Living and Working in an Interconnected World.* Upper Saddle River, NJ: Prentice Hall PTR.

Birznieks, G., and Sol, S. (1997). *CGI for Commerce: A Complete Web-Based Selling Solution.* New York: M&T Books.

Black, U. (1999). *Advanced Internet Technologies.* Upper Saddle River, NJ: Prentice Hall PTR.

Blodget, H., and McCabe, E. (2000). *The B2B Market Maker Book.* Merrill Lynch, New York.

Brown, S. (1999). *Implementing Virtual Private Networks.* New York: McGraw-Hill.

Cintron, D. (1999). *Fast Track Web Programming.* New York: Wiley Computer Publishing.

Cronin, M.J. (1996). *The Internet Strategy Handbook.* Boston, MA: Harvard Business School Press.

Fournier, R. (1999). *A Methodology for Client/Server and Web Application Development.* Upper Saddle River, NJ: Yourdon Press.

Goncalves, M. (1997). *Protecting Tour Website with Firewalls.* Upper Saddle River, NJ: Prentice Hall PTR.

Greenstein, M., and Feinman, T.M. (2000). *Electronic Commerce: Security, Risk Management and Control.* Boston: Irwin McGraw-Hill.

Holincheck, J. (1999). *Application Outsourcing and ASPs: Definition and Key Evaluation Criteria.* p. 1199–2007. Cambridge, MA: Giga Information Group.

Kalakota, R., and Whinston, A. B. (1997). *Readings in Electronic Commerce.* Reading, MA: Addison-Wesley.

Kalakota, R., and Whinston, A.B. (1996). *Frontiers of Electronic Commerce.* Reading, MA: Addison-Wesley.

Keen, P., *et al.* (2000). *Electronic Commerce Relationships: Trust by Design.* Upper Saddle River, NJ: Prentice Hall.

Kosiur, D. (1997) *Understanding Electronic Commerce.* Redmond, WA: Microsoft Press.

Lynch, D.C., and Lundquist, L. (1996). *Digital Money: The New Era of Internet Commerce.* New York: Wiley.

Maddox, K., and Blankenhorn, D. (1998). *Web Commerce—Building a Digital Business.* New York: Wiley.

McGrath, S. (1998) *XML by Example.* New York: Prentice Hall.

Nam, K., *et al.* (1996). A two-level investigation of information systems outsourcing. Communications of the ACM. 39(7): 36–44.

Petrovsky, M. (1998). *Dynamic HTML in Action.* New York: Osborne McGraw-Hill.

PriceWaterhouseCoopers. (2000). *Technology Forecast: 2000.* New York.

Rao, H.R., Agrawal, M., and Salam, A.F. (1998). *Internet Browsers.* Encyclopedia of Electrical Engineering. New York: Wiley.

Seybold, P.B., and Marshak, R.T. (1998). *Customers.com: How to Create a Profitable Business Strategy for the Internet and Beyond.* New York: Randon House.

Shapiro, C., and Varian, H.R. (1999). *Information Rules.* Boston, MA: Harvard Business School Press.

Umar, A. (1997). *Application Reengineering: Building Web Based Applications and Dealing with Legacies.* Englewood Cliffs, NJ: Prentice Hall.

Wendland, R. (1999). *Application Service Providers: A Report by Durlacher Research.* London: Durlacher.

Yeager, N.J., and McGrath, R.E. (1996). *Web Server Technology.* San Fransisco, CA: Morgan Kaufmann.

Electronic Data Interchange

Izak Benbasat, Paul Chwelos, Albert S. Dexter, and Clive D. Wrigley
University of British Columbia

GLOSSARY

ANSI X12 EDI standard developed by the American National Standards Institute, primarily used in North America.

authentication the ability to verify the identity of the sender and receiver of an electronic message.

ebXML Electronic business XML (eXtensible Markup Language), an evolving standard designed to support exchange of standard business documents using XML that has the sponsorship of international standards bodies.

EDIFACT (EDI) standard developed by the United Nations, primarily used in Europe. Acronym for EDI for Administration Commerce, and Trade.

Internet-based EDI An EDI system that uses the Internet, rather than a value-added network, as the medium for document transmission; typically uses independent application systems across trading partners.

nonrepudiation The capability to track an electronic message across a network, and to verify when it has been sent and received; therefore, neither the sender nor the recipient can deny having sent or received the message, respectively.

privacy The documents are not observed in transit.

validation The ability to determine that the business document has not been changed in transit. This ability assures the integrity of the document.

value-added network (VAN) An on-line network that provides the communications capability between EDI partners as well as "value-added" services such as authentication, auditing capability, security, nonrepudiation, and message translation.

Web-based EDI An EDI system that provides a Web-based interface to one trading partner, allowing the partner to complete an on-line form and then have it translated into an EDI message. Typically, only one trading partner is running an EDI system, and the other logs into it via the Web.

XML eXtensible Markup Language is an extension of HTML (hypertext markup language), the syntax underlying the World Wide Web. Unlike HTML, XML is a general purpose language that provides a flexible system of tags that allows for the embedding of metadata within documents. It describes the structure of a document, procedural information and supports links to multiple documents, allowing data to be machine processable.

ELECTRONIC DATA INTERCHANGE (EDI) is the interorganizational exchange of business documents in structured, machine-readable form, typically between independent application systems. EDI is not a single technology, rather it is a set of standardized syntax and protocols that govern the structure and sequence of electronic documents, which may be delivered in a variety of ways: using value-added networks (VAN), a direct dial-up connection between independent systems, over the Internet, or even on physical media such as tapes or disks. The standards describing EDI documents are publicly available, and are published by standards bodies such as the American National

Standards Institute (ANSI) or the United Nations. EDI originated in the 1970s through efforts in the transportation, retail, and grocery industries. Today, EDI is used in the majority of Fortune 1000 and Global 2000 firms, but the penetration of small- to medium-sized enterprises (SME) remains very low. Nonetheless, EDI remains an important technology, with approximately a half-trillion dollars of transactions per annum in the United States alone. EDI can be thought of as an early form of electronic commerce (EC) that predates commercial use of the Internet by more than twenty years, and appears to be firmly entrenched in many organizations.

I. HISTORY OF EDI

Organizations have long had the goals of improving the efficiency and effectiveness of their interactions with trading partners, including suppliers, customers, banks, customs brokers, transportation carriers, and others. During the last few decades there has been a convergence of communication and computer technologies that enables the achievement of this goal. Initially, organizations worked to integrate their information systems with proprietary technology. These business-to-business relationships became known as Interorganizational Systems (IOS). Kaufman, in 1966, introduced IOS to the information systems community when he asked general managers to think beyond their own firms in order to network computers and share information processing across organizational boundaries. EDI followed as a means of creating interorganizational communications using open, standardized formats. More recently many firm-to-firm transactions have surfaced using Internet technology, typically referred to as electronic or digital commerce.

EDI emerged as a result of efforts to define communications standards in the transportation industry in the late 1960s. Groups in the grocery, retail, and other industries followed in the 1970s, each developing their own standards. ANSI created a committee, labeled X12, to develop an EDI standard general enough to be used across all industries. The ANSI X12 committee continues to revise and extend the standards on the basis of voluntary industry participation and open public comment of proposed standards.[1] In 1986, the United Nations Economic Commission for Europe began working on an international standard for EDI. The result was EDIFACT, which has become the primary EDI standard in Europe.

II. EDI PROTOCOLS

Each of the protocols describing EDI achieves the same function: providing a format to structure business information in a machine-readable and -processable form. Each standard outlines the types of business documents (or "transaction sets") supported, the information required for each document, and the syntax of this information: sequence, encoding, and data definition.

A typical example of the use of EDI in a business context is the purchase order (PO) process. EDI is suitable for this business process due to the structured nature of the transactions and the need for a high level of accuracy in communication. The process can be readily traced as follows: the customer determines a need for a good or service, and initiates a customer inquiry to the supplier. The supplier typically responds with data concerning availability and pricing. The customer responds with a PO, and the supplier responds with a PO acknowledgment. Shipping data, receipt of goods, billing notices, and payments complete the cycle. The EDI transaction set provides structured data to buyer and seller, and to both firms' financial institutions. A simplified illustration showing the described transaction set appears in Fig. 1. The numbers in parentheses indicate ANSI X12 document types.

Once the two partners have agreed upon standards and the specific implementation for their business transactions, all parts of the transaction are automated, with each document having its own EDI code. The table below summarizes the purchase ordering process described above in our example. The actual codes and brief descriptions for the various components of the customary and established business processes are shown using ANSI X12.

819 Invoice[1]: billing information sent from the supplier to the buyer for goods and services provided.

820 Payment Order/Remittance Advice: used in the settlement phase (1) to order a financial institution to make payment to payee(s) on behalf of the sending party, (2) to report the completion of a payment to payee(s) by a financial institu-

[1]This document illustrates some of the transaction sets and standards developed by the committee, the value of which will be readily apparent to the reader.

[1]The Functional Acknowledgment (997) acknowledges the syntactical correctness/incorrectness of each document received, according to the defined control structure. The 997 is sent back to the sender for each document in the transaction cycle.

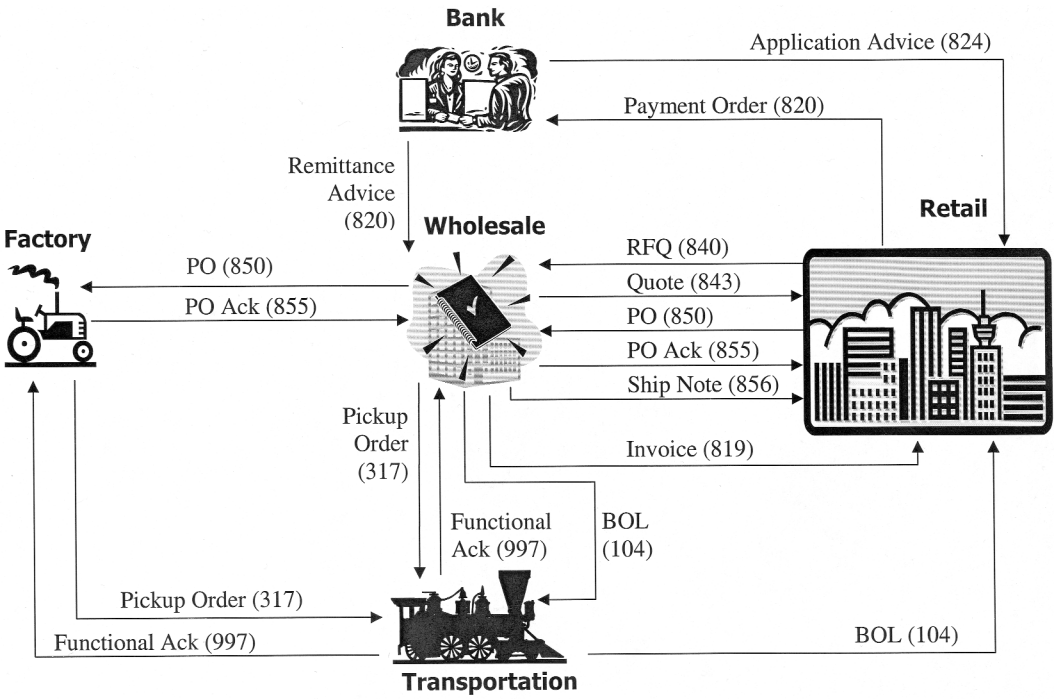


Figure 1 Transaction flows.

tion, and (3) to provide advice to the payee by the payor about payments made to them. A single payment order may require multiple copies to be distributed among the buyer, supplier and their respective financial institutions.

824 Application Advice: provides the originator of the 820 with acceptance, rejection, or acceptance with change from the financial institution. With acceptance, the financial institution agrees to act on the 820 instructions of the payor. With rejection, the financial institution is unable to process the 820 as submitted.

840 Request for Quotation: solicits price, delivery schedule, and other items from supply chain trading partners.

843 Response to Request for Quotation: provides buyer with requested information from the 840. It may be considered a contractual offer depending upon the trading partner agreement in force.

850 Purchase Order: provides contractual information for the purchase of goods or services. It may be considered a contractual offer or acceptance of an 843, depending upon the trading partner agreement in force.

855 Purchase Order Acknowledgment: provides the seller's acknowledgment of a buyer's 850 and its ability to supply the ordered goods or services.

856 Shipping Notice/Manifest: provides the buyer and other related parties with shipping information resulting from the complete or partial fulfillment of one or more 850s. It may include purchase order information, product descriptions, physical characteristics, markings, packaging, carrier, and other logistical information.

861 Receiving Advice: sent from the buyer to the supplier, providing notification and information concerning the receipt of goods or services and their condition.

The above example illustrated one scenario in the use of EDI in a hypothetical industry supply chain. Each X12 document type specifies the segments and data elements contained within the document. An example implementation of the X12 Payment Order (820) is presented below as Table I. This illustration shows the segments used in a typical implementation of the Payment Order (820). The basic structure of all structured business documents is header information, details, and summary information.

Each segment is comprised of a number of data elements, also either mandatory or optional. Each data element is further defined in terms of its base data type, e.g., numeric, character, or categorical. For example, Table II contains the data elements within the BPR segment of the payment order (820).

Table I 820 Payment Order/Remittance Advice

Functional Group ID=RA
Header:

Segment ID	Name	Position	Mandatory/ Optional	Maximum Use
ST	Transaction Set Header	010	M	1
BPR	Beginning Segment for Payment Order/Remittance Advice	020	M	1
TRN	Trace	035	M	1
N1	Name	061	M	1
N1	Name	070	M	1
N2	Additional Name Information	080	O	>1
N3	Address Information	090	O	>1
N4	Geographic	100	O	1

Detail:

	Pos. No.	Seg. ID	Name	Req. Des.	Max. Use	Loop Repeat	Notes and Comments
			LOOP ID–ENT			>1	
	010	ENT	Entity	O	1		
			LOOP ID–RMR			>1	
	150	RMR	Remittance Advice Accounts Receivable Open Item Reference	O	1		
	170	REF	Reference Numbers	O	>1		
			LOOP ID–ADX			>1	
	210	ADX	Adjustment	O	1		

Summary:

	Pos. No.	Seg. ID	Name	Req. Des.	Max. Use	Loop Repeat	Notes and Comments
Must Use	010	SE	Transaction Set Trailer	M	1		

Thus far we have shown just one simple scenario of the transaction flows and the use of X12 (and/or EDIFACT) as the document structure within a business context. There are numerous other standardized business documents used in the day-to-day transactions among trading partners. All documents conform to well-recognized data structures and syntax. The rules for the identification and arrangement of data into specific fields and elements, data segments, and transaction sets are prescribed in the ANSI X12 documents. It is important to note that each industry sector, e.g., finance, manufacturing, transportation, government, etc., has defined its own set of documents to conform to the specific industry practice. In addition, within an industry sector there are further subdivisions, e.g., within the transportation sector there are different documents for use by air, ocean, rail, and truck modalities.

III. EDI IMPLEMENTATION

An EDI implementation consists of at least the following components:

- An on-line network for communication of messages
- Standards for message format

Table II Segment: BPR Beginning Segment for Payment Order/Remittance Advice

Level: Heading

Usage: Mandatory

Max Use: 1

Purpose: (1) To indicate the beginning of a Payment Order/Remittance Advice Transaction Set and total payment amount or (2) to enable related transfer of funds and/or information from payer to payee to occur

Data Element Summary

Ref. Des.	Data Element	Name	Atrributes		
BPR01	305	Transaction Handling Code Code designating the action to be taken by all parties	M	ID	1/1
BPR02	782	Monetary Amount Monetary amount	M	R	1/15
BPR03	478	Credit/Debit Flag Code Code indicating whether amount is a credit or debit	M	ID	1/1
BPR04	591	Payment Method Code Code identifying the method for the movement of payment instructions	M	ID	3/3
BPR05	812	Payment Format Code Code identifying the payment format to be used	X	ID	3/3
BPR06	506	(DFI) ID Number Qualifier Code identifying the type of identification number of Depository Financial Institution (DFI)	M	ID	2/2
BPR07	507	(DFI) Identification Number Depository Financial Institution (DFI) identification number	M	AN	3/12
BPR09	508	Account Number Account number assigned	M	AN	1/7
BPR10	509	Originating Company Identifier A unique identifier designating the company initiating the funds transfer instructions. The first character is one-digit ANSI identification code designation (ICD) followed by the nine-digit identification number which may be an IRS employer identification number (EIN), data universal numbering system (DUNS), or a user assigned number; the ICD for an EIN is 1, DUNS is 3, user assigned number is 9	O	AN	10/10
BPR12	506	(DFI) ID Number Qualifier Code identifying the type of identification number of Depository Financial Institution (DFI)	X	ID	2/2
BPR13	507	(DFI) Identification Number Depository Financial Institution (DFI) identification number	X	AN	3/12
BPR15	508	Account Number Account number assigned	X	AN	1/35
BPR16	513	Effective Entry Date Date the originating company intends for the transaction to be settled	M	DT	6/6

- Software for translating to and from the EDI formats
- An interface with other information systems

EDI does not specify the medium by which the messages will be passed between trading partners. In the past, EDI users either developed private connections

with trading partners or used a VAN. VAN services include translation between different EDI protocols, store and forward mailboxes, security, authentication, notification, as well as the actual transmission of messages. The pricing structure for a VAN typically includes a significant up-front cost, as well as an ongoing per kilobyte or per message transaction fee that may total as much as $100,000 per year for a large organization. Currently, EDI protocols are increasingly being used in combination with the Internet as a transmission medium. The chief benefit of Internet-based EDI is the potential for cost savings, in terms of lower message costs. Efforts are also underway to improve the reliability and security of Internet-based EDI using message tracing and encryption. Another attraction of these Internet-based EDI systems is their ability to support trading partners who have access to the Internet, but do not have their own EDI system. These partners would use the World Wide Web to access and fill out an electronic business form, such as a PO, which is then translated into an EDI document. Thus, web-based EDI is an attractive option for very small firms that wish to be EDI-enabled, perhaps to satisfy the requirements of a larger trading partner, at a minimum of cost.

However, Internet-based EDI is not a panacea, and has several drawbacks. First, the standards underlying Internet-based EDI are still in flux, as discussed below in the section EDI and EC. Second, because the Internet is a public network, message delivery and security cannot be guaranteed to the extent it is with the use of a VAN. Thus, the Internet may not be appropriate for either time-critical messages (e.g., supporting a just-in-time (JIT) process), or sensitive messages. Third, web-based EDI requires that messages be typed in by hand by one of the trading partners, meaning that this partner cannot integrate this system with other information systems in the value chain, such as inventory, production, accounting, and billing. This integration has been cited as a major source of benefits from EDI adoption.

IV. EXAMPLE OF AN EDI IMPLEMENTATION

Printronix is an integrated manufacturer of dot matrix and laser printers headquartered in Orange County, CA. Printronix is the foremost supplier of line matrix printers, and an innovator in continuous form laser printers and thermal bar code printers; many of its products are sold under other brand names. Its annual revenues exceed $150 million.

Printronix adopted EDI in 1997 at the request of one of its major customers. EDI messages in X12 format are received in batch mode three times per day using a VAN. For its initial EDI implementation, Printronix made use of translation services provided by a third party to convert its internal enterprise resource planning (ERP) documents into EDI format. The costs of this translation service comprised both a fixed cost for creating each "map" from a particular ERP document to the associated EDI document, as well as a variable cost for each message translated. Ultimately, the volume of messages reached the point at which it was less costly to bring the translation in-house, and Printronix purchased an EDI package and translation software for its mainframe system.

Printronix conducts EDI with three customers, but does so extensively with only its major EDI-initiating customer. Printronix also exchanges electronic documents with other customers in other, proprietary formats. The EDI system has the advantage of being interfaced or integrated with its internal systems, so that incoming purchase orders are automatically routed to the manufacturing system and appear on the shop floor within minutes or even seconds of having been received. Once orders are completed, shipping notices are automatically generated and forwarded via EDI to this customer and its logistics provider, who then picks up the orders. Invoices are then sent via EDI.

From Printronix's perspective, this EDI system provides a number of benefits including faster and more accurate communications, more rapid receipt of payments, and the ability to respond more rapidly to customers. The costs of the EDI system include the software purchase, ongoing fees paid to the VAN for message transport, and maintenance of the EDI system, which involves two full-time information systems people. The primary benefit, of course, has been the ability to retain its most valuable customer.

V. SECURITY, LEGAL, AND AUDITING ISSUES

There are several ancillary issues arising from the implementation of EDI that the firm needs to consider. Security issues abound because the transactions move electronically across organizational boundaries. Therefore both trading partners need to assure proper authorization of the transaction sets and proper access to the information systems. All transactions must be validated, i.e., both the send and receive segments need verification. Care must be taken to ensure that all parties to the transactions are properly authenticated, i.e., they must be whom they purport to be. Proper accuracy is even more important because of the multiple firms involved in the transactions; therefore edit con-

trols are necessary. The firms should also consider encryption of the data, especially when using the public Internet as their communication medium.

Legal agreements must be established between all trading partners. The contracting process provides all the rules of behavior (terms and conditions of transactions) to which the partners must adhere. For example, in the physical world a PO (offer) is deemed to be received when the PO is put in the mail; however, in the electronic world the PO is deemed to be received when it arrives at the trading partner's system. Further, because business is to take place via electronic means, electronic or digital signatures are required. An EDI transaction should not be able to be repudiated, i.e., once the transaction occurs, neither side can deny having agreed to it.

EDI systems must be capable of audit. Therefore, the firm's audit group and legal staff should both be involved in setting up the contract. Auditing within an EDI environment is complicated by the fact that two or more firms are involved in the implementation of transactions. Nonetheless, the complete transaction cycle from initiation to completion must be auditable. Evidence of the complete cycle needs to be stored on both firms' computer systems, so that either firm may be able to recreate the transactions if necessary. VAN providers should also store transaction histories to support their auditing capabilities.

VI. EDI COST AND BENEFITS

An EDI system can provide benefits of a number of types, including both tangible and intangible, operational, and strategic. The major difference between the tangible and intangible benefits arises from the ease of quantifying the benefit. Tangible, operational benefits include: increased productivity through reduced paper work, lower clerical labor costs, lower postage and courier fees, lower data entry costs, faster document exchange, reduced error rates (and therefore lower rework to correct), streamlined processes, reduced inventory levels (with attendant lower storage and handling costs, reduction in safety stocks, etc.), and rapid confirmation of orders and responses to requests. Cost savings accrue from the decreased administrative costs and improved cash management.

EDI intangible benefits are numerous. Intangible operational benefits arise from: increased data accuracy, higher levels of customer service, more rapid access to information, a higher level of certainty about orders and deliveries because of the computerized tracking, and decreased delivery times. Additional intangible benefits also derive from EDI adoption, such as improved access to more accurate and more timely operational data. Strategic advantage might occur if the EDI system is integrated with the rest of the information systems in the organization and provides the adopting firm with a better foundation for competing in the industry. As examples, EDI enables business process reengineering and supports industry value chain integration, such as JIT inventory, continuous replenishment, and quick response retailing. Firms often tout their enhanced business partnerships from closer working relationships with trading partners. These include improved supplier relationships and customer service.

For many organizations, such as Printronix, the most tangible benefit of EDI is the ability to retain key customers who demand that their trading partners be EDI-enabled. Depending on the importance of these customers or key suppliers, the use of EDI may simply be a necessity. Wal-Mart and General Motors have been cited as examples of key firms in their respective industries that have required adoption of EDI by their trading partners.

The costs of EDI are similar to other information technology innovations. To initiate EDI, startup costs include hardware and software, telecommunications, the development support team, legal and consulting fees, as well as employee and managerial training. For a small organization wishing to merely become EDI compliant, a minimal stand-alone implementation will cost $2,000–20,000. However, such a system will not be integrated with the firm's other systems, and will thus not provide many of the benefits listed above. Following or concurrent with the implementation of EDI are ongoing VAN costs, systems modification and enhancement, legal and other consulting fees, membership in various EDI educational and professional bodies, and education. For a large organization, these recurring fees can run more than $100,000 per year. The firm must also consider the costs of integrating EDI with other management information systems. This is an important consideration, because larger payoffs from an EDI implementation will come from integration throughout the firm. An integrated EDI implementation for a medium or large firm can cost $200,000 or more.

VII. EDI ADOPTION

Firms considering the adoption of EDI need to conduct a feasibility study, similar in many cases to the adoption of other types of information technologies (IT). However, the business case for EDI is more complicated than for IT innovations that fall strictly within

the boundaries of an organization, because two or more organizations are involved in an EDI implementation. Because of the interorganizational nature of the EDI decision, the factors influencing adoption comprise three categories, *perceived benefits, external pressure* due in part to business partner influence, and *organizational readiness.*

Perceived benefits refer to the anticipated advantages that EDI can provide the organization. As explained earlier, benefits are both direct and indirect in nature. Direct benefits include operational cost savings and other internal efficiencies arising from, for example, reduced paperwork, reduced data re-entry, and reduced error rates. Likewise, indirect benefits are opportunities that emerge from the use of EDI, such as improved customer service and the potential for process reengineering.

External pressure is multifaceted, encapsulating the influences arising from a number of sources within the competitive environment surrounding an organization. These sources include: *competitive pressure,* relating to the ability of EDI to maintain or increase competitiveness within the industry; *industry pressure,* relating to the efforts of industry associations or lobby groups to promulgate EDI standards and encourage adoption, and *trading partner influences.*

The latter factor captures the potential power of a trading partner to "encourage" EDI adoption, and the strength of the partner's exercised power. Imagine a major supplier or purchaser of a firm's goods or services. Potential power increases with size of the trading partner and its criticalness to the firm. Exercised power ranges from subtle persuasion to decrees requiring EDI adoption as a necessary condition for conducting any further business with the partner.

Organizational readiness is the set of factors necessary to enable the firm to adopt EDI. These include sufficient *IT sophistication* and *financial resources.* IT sophistication captures not only the level of technological expertise within the organization, but also assesses the level of management understanding of and support for using IT to achieve organizational objectives. Financial resources is a straightforward measure of an organization's capital available for use toward IT investment.

In the context of IOS, however, readiness is not solely an organization-level concept. Adoption of an IOS requires readiness on the part of, at a minimum, two trading partners. Thus readiness considers a firm that may be motivated to adopt EDI, i.e., having high perceived benefits, and be ready to adopt, i.e., having available financial resources and IT know-how, but is unable to do so due to trading partners that are not ready or able to adopt EDI. For example, the firm may find that its trading partners do not have the resources necessary to adopt EDI, or that the firm anticipates difficulty acquiring the "critical mass" of EDI-enabled trading partners to make its adoption worthwhile. Smaller trading partners, even if financially and technologically able to adopt EDI, may not find it worthwhile to do so on a cost/benefit basis due to their limited volume of transactions (i.e., the lower cost of EDI transactions is not able to offset the largely fixed costs of EDI adoption). Thus, even large, solvent, and technologically sophisticated organizations can have difficulty in expanding their networks of EDI-enabled trading partners.

VIII. THE FUTURE: EDI AND EC

The term electronic commerce has emerged to encapsulate all forms of commercial transactions conducted either in whole or in part using computer systems. This includes EDI, business-to-business (B2B), business-to-consumer (B2C), and consumer-to-consumer (C2C). These digital transactions may require trusted third party services, or may be conducted directly between peers (P2P). To place these terms into proper perspective, it is necessary to distinguish: the communications vehicle, from the business data being exchanged. The communication of business information can be conveyed either over the public or private internet(s), or by using a VAN to move data between systems. Previously with EDI, many firms used VANs as their communications providers. Since the advent of the Internet, many firms are now using the Internet to exchange business documents. However, VAN-based EDI remains a viable technology in its own right. As recently as 2000, electronic transactions conducted via VAN-based EDI ($450 billion) exceeded those conducted over the Internet. Very few, if any, firms have yet to abandon their installed VAN-based EDI systems in favor of Internet-based alternatives. However, the Internet has been gaining popularity as a low-cost alternative to VANs for the transmission of EDI messages.

Apart from communications costs, we need to consider the business data being exchanged. The data moved may be structured in a variety of different means. Efforts are underway to develop a new standard of defining the data for EC based on XML. These initiatives are gaining favor due in part to the lack of flexibility and the high implementation costs of X12 standards facing the SMEs. Unlike HTML, XML provides a flexible system of tags that allows for the embedding of metadata, such as date element names,

within documents. Electronic business XML (ebXML) has been advocated by a number of bodies, including the United Nations Centre for Trade Facilitation and Electronic Business. In addition, ANSI has established an XML Task Group within the X12 committee to develop cross-industry XML business standards based on ebXML. The ANSI XML group is working with the EDIFACT work group in an effort to harmonize not only the two existing EDI standards, but also EDI and XML standards. The state of XML is similar to the early days of the development of EDI standards, with the exception that there is now available a significant store of knowledge about business processes, as embodied in existing EDI standards. Thus, although the development of integrated ebXML/EDI standards will take considerable time, this work may proceed more rapidly than the initial development of EDI standards.

XML differs substantially from X12 and other legacy EDI standards in terms of flexibility and efficiency. While X12 may be efficient in terms of bandwidth and processing requirements, it is inflexible to changes such as adding new document types or data elements. EDI standards were developed during a time of relatively high communication costs, hence the standards minimize the amount of data required to communicate business information. Data element names, and even the decimal point, are abstracted out of the documents. This design decision requires substantial effort to map data items from internal systems into the EDI standard. Documents coded in X12 contain position-sensitive data, requiring a predefined map that is trading partner specific to extract data from standard data elements into internal systems.

ebXML, on the other hand, allows data embedded within the documents to be tagged with data element names. This allows document parsers to easily extract the business data based on names rather than position, resulting in lower implementation and maintenance costs. While ebXML introduces more flexibility into the document structure, this flexibility comes at a price of higher bandwidth and processing requirements. ebXML documents may be 5–10 times larger than the equivalent document in X12. As communication costs fall the added flexibility would appear to compensate for the increased data volume. FiXML (financial information exchange markup language) extends this flexibility concept further by adding an additional map between tag identifiers and data element names. Tags are numerical (rather than longer data element names), thereby reducing communication overhead. This approach moves the coupling between data element names to a separately managed mapping table, which links tag numbers to data element names. Businesses may change or introduce new data element names without affecting their trading partners. Most observers predict that current EDI protocols will eventually be replaced by these more flexible XML-based transaction standards. This change is anticipated to accelerate the diffusion of EDI/EC among SMEs.

A perusal of the costs and benefits of EC shows the same set of evaluation criteria and activities that firms face when considering EDI adoption. Thus, one may speculate that B2B EC will face many of the same challenges that EDI has in terms of adoption, as observed by Chwelos et al. and discussed earlier in Section VII.

ACKNOWLEDGMENT

This research was supported by a Social Sciences and Humanities Research Council of Canada Strategic Research Grant.

SEE ALSO THE FOLLOWING ARTICLES

Data Compression • Digital Goods: An Economic Perspective • Electronic Commerce, Infrastructure for • Electronic Payment Systems • Internet, Overview • Standards and Protocols in Data Communications • Value Chain Analysis • XML (Extensible Mark-up Language)

BIBLIOGRAPHY

Chwelos, P., Benbasat, I., and Dexter, A. S. (2001). Empirical test of an EDI adoption model. *Information Systems Research,* Vol. 12, No. 3, 304–321.

Emmelhainz, M. A. (1990). Electronic data interchange, a total management guide. New York: Van Nostrand-Reinhold.

Kaufman, F. (1966). Data systems that cross company boundaries. *Harvard Business Review,* Vol. 44, 141–155.

Electronic Mail

Michael Sampson
Institute for Effectivity, Christchurch, New Zealand

GLOSSARY

domain name system (DNS) A directory service on the Internet that manages the addressing of Internet-connected computer systems.
e-mail client Software that a person uses to work with their e-mail, including sending and receiving, reading and writing, and filing of e-mails.
e-mail server A software program that runs on a server computer to provide e-mail services to users. Connects with other e-mail servers to exchange e-mail for users on remote servers.
internet message access protocol (IMAP) An Internet standard for accessing messages stored on an Internet e-mail server. Supports advanced server-side and advanced client-side operations.
message transfer agent (MTA) The component of an e-mail server that transfers messages to a remote e-mail server.
message store The component of an e-mail server that stores messages for users.
message switch A software program that allows different kinds of e-mail servers to exchange e-mail.
multipurpose internet mail extensions (MIME) A group of Internet standards that extend the capabilities of Internet e-mail to support attachments, graphics, and rich text.
post office protocol (POP) An Internet standard for accessing messages stored on an Internet e-mail server. Supports basic server-side and advanced client-side operations.
public key infrastructure (PKI) A set of technologies that can be used with e-mail to add security and encryption to messages.
simple mail transfer protocol (SMTP) An Internet standard for the exchange of text-based e-mail between Internet e-mail servers.
SMTP gateway A software program that cooperates with a proprietary e-mail server to translate e-mail into the SMTP format for transfer to an Internet e-mail server.

I. WHAT IS ELECTRONIC MAIL?

Electronic mail (e-mail) is a computer-based application for the exchange of messages between users. A worldwide e-mail network allows people to exchange e-mail messages very quickly. E-mail is the electronic equivalent of a letter, but with advantages in timeliness and flexibility. While a letter will take from one day to a couple of weeks to be delivered, an e-mail is delivered to the intended recipient's mailbox almost instantaneously, usually in the multiple-second to subminute range. This is the case whether the e-mail is exchanged between people on the same floor of a business, or between friends at opposite points on the globe. This article provides a comprehensive, intermediate-level overview of e-mail, including its main functions, historical and current architectures,

key standards, supporting infrastructure, and contemporary and future issues.

II. FUNCTIONS OF AN ELECTRONIC MAIL SYSTEM

A computer system that provides e-mail services offers five functions: access, addressing, transport, storage, and cross-system integration (Fig. 1).

A. Message Access

The first function of an e-mail system is to provide access for working with e-mail messages, including creating new e-mails, reading new e-mails, and organizing e-mails into folders for future reference. E-mail client software includes the tools to perform these functions.

1. Dedicated Client

It is most common today for e-mail users to have a specific piece of software on their computer that is the user's interface into the e-mail system. A software vendor has specifically created the e-mail client program, and the end user or their corporate information technology (IT) department has installed the e-mail client onto their computer—whether a PC, Mac, Unix machine, or thin client.

Popular e-mail clients in business today are Microsoft Outlook and Lotus Notes.

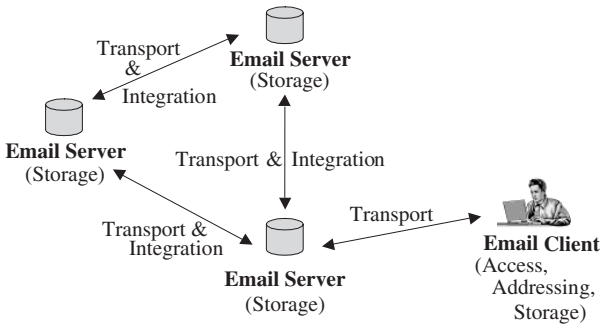


Figure 1 The five functions of e-mail. The five functions of e-mail work together to create an e-mail system. An e-mail client, which is software that a user interacts with, provides the access, addressing, and storage functions. A computer network links the e-mail client to an e-mail server thus providing the transport function, and e-mail servers talk to other e-mail servers over a computer network to provide transport and integration capabilities. E-mail servers provide storage capabilities.

2. Web Browser

The web browser is the new universal client, providing access to many types of computer systems around the world. Since the mid-1990s, e-mail vendors have modified their e-mail servers to support web browser access to e-mail. While these started off as clunky and simple, they have rapidly matured in capability, usability and acceptance.

There are advantages in using a web browser as an e-mail client. For the vendor, they do not have to maintain separate software development efforts to create, maintain, and update dedicated e-mail client software for different operating systems. For the user, all their messages are held on a server that is accessible from any web browser around the world, meaning that they can remain in contact with colleagues and friends from any computer. On the downside, the vendor loses out on creating emotional attachment for users to their specific e-mail client (e.g., a user will typically identify closely with a specific product through an emotionally loaded statement such as *I love Microsoft Outlook and couldn't use anything else*). Web browsers make it easier for an individual consumer or corporate IT department to migrate to a new e-mail system. Finally, web browsers are slower than dedicated e-mail clients at performing standard e-mail functions, which reduces the productivity of an end user. All the major e-mail vendors permit web browser access to their e-mail servers.

3. Wireless

Many corporate employees and members of the management team are mobile during the day. While employees may sit at their desk for some of the day, most people roam between on-site meetings, off-site appointments, and visits with customers or suppliers. Many want access to their e-mail when they are not at their desk and don't have access to a computer. Hence, access from a cellular phone or personal digital assistant (PDA) is a growing area of interest.

The user connects to their e-mail system from a wireless device and reviews their inbox to see if any urgent e-mails have been received. If so, the user reads these on the run, and takes appropriate action as necessary. A short reply can be typed on the keypad of their phone, or a telephone call to discuss the matter with the original sender or another colleague can be initiated.

There are many limitations with using wireless devices for e-mail, e.g., small keypad, small screen, short battery life. However, this is an area of increasing re-

search and focus for e-mail, wireless infrastructure and wireless device manufacturers alike.

B. Message Addressing

When you want to call someone using the telephone, you need to know their telephone number. In the e-mail world, it is their e-mail address you need. An e-mail client should provide a mechanism to correctly address e-mail messages, and this is most often facilitated by recourse to a shared corporate directory or private personal address book. In turn, the e-mail server must know how to deliver a specific e-mail message to the intended recipient. For messages sent locally within the e-mail network the e-mail server will consult its own directory for delivery instructions. For e-mail messages sent outside a given e-mail network, the e-mail server must consult its routing instructions to determine how to connect to the outside world, most frequently the Internet, to deliver its messages.

The standard addressing convention within the Internet world consists of two components in the user@ domain format. The *domain* part is an identifier for an organization or specific server computer at a university (e.g., canterbury.ac.nz), company (ibm.com), Internet service provider (xtra.co.nz), portal (yahoo.com), or other organization (whitehouse.gov). Within a specific domain, the *user* part is a unique identifier for a person. When the two components are concatenated using the "@" symbol, a unique person is identified. For example, if you want to e-mail the author, you would address your e-mail to michael@effectivity.org, which specifies that the user "michael" at the domain "effectivity.org" is the intended recipient. If you incorrectly addressed your e-mail to michael@effectivity.com, mail delivery would fail, or you would get a different person because the domain is wrong.

E-mail systems used within a corporate environment frequently have a secondary addressing schema, or more correctly, a primary addressing schema which is different in format to the secondary Internet addressing schema. A Lotus Notes user will be known uniquely through a Lotus Notes user name, which is in the format of Name/Organizational Unit/Organization (there can be up to four different "Organizational Unit" components, and there is an optional "Country" parameter at the end of the Organization). As a Lotus Notes user, my Notes user name is Michael Sampson/Effectivity (there is no intermediate organizational unit in my Notes name, nor do I use the country identifier), and within a given Lotus Notes network, you can address e-mail to me at Michael Sampson/Effectivity. A secondary Internet name is appended to my directory record, however, to permit users outside of my local Notes network to send e-mail to me through my Internet address of michael@effectivity.org.

C. Message Transport

An e-mail system needs the ability to transmit messages between people. Once a user submits a message to an e-mail server, the server uses a *message transfer protocol,* generally called a *message transfer agent* (MTA), to transfer the message to the server of the intended recipient. A message transfer protocol specifies how one e-mail server establishes a connection to another e-mail server, and defines a common language that the two servers speak.

The *simple mail transfer protocol* (SMTP) is the Internet standard that describes how Internet e-mail servers transfer e-mail between themselves. Before the Internet was widely used within corporate environments, Lotus Notes/Domino used the *Notes remote procedure call* (NRPC) protocol, but now also supports SMTP natively. Microsoft Exchange Server traditionally used *remote procedure call* (RPC) or other protocols, but the default protocol in the latest version of Exchange is SMTP. X.400 was also used for interorganizational e-mail exchange.

D. Message Storage and Retrieval

E-mail messages must be stored in a computer storage system until either the user deletes them, or within a corporate environment, the messages are archived to off-line storage. Most home users store their messages on the local hard drive of their own computer, whereas corporate users have their messages stored on the e-mail server. For users with a *post office protocol* (POP) Internet e-mail service, their messages and folders are stored locally on their own computer. A minority of home users make use of an *Internet message access protocol* (IMAP)—compliant service which provides server-based storage of messages and folders if desired. These two standards are discussed in Section IV.

For the corporate user, messages, folders and preferences are usually stored on a server, although the user can elect to have a client-side copy. The user's mailbox is stored on the server, allowing the MTA to immediately put new messages into the mailbox for later retrieval by the user, e.g., all messages, folders,

and preferences in Lotus Notes are stored in a single NSF file on the e-mail server; the user can elect to replicate a copy of their mailbox to their local computer for off-line access and message editing.

E. Integrating Multiple E-Mail Systems

There are millions of e-mail servers in the world that need the ability to interact with other servers. Within a corporate environment, each branch office, manufacturing location, or other business site will usually have its own e-mail server, and these need to be integrated for message exchange within the corporate network. Further, if one company acquires another it will seek to merge both into a new single enterprise. A shared e-mail system for enterprise-wide message exchanges, calendaring and scheduling, and message addressing through a shared directory is important to facilitate communication within the newly merged business.

There are three ways to integrate disparate e-mail systems: a point-to-point message gateway, a message switch, or the use of Internet protocols.

1. Message Gateway

A *message gateway* is a software application that provides a server-to-server connection between two disparate e-mail servers, e.g., to connect a Microsoft Exchange Server to a Lotus Domino server. Note that the assumption here is that the two e-mail servers do not speak SMTP, or that the business decides not to use SMTP for quality of service or security reasons.

The gateway provides a mechanism for translating the e-mail protocol of the first server into the e-mail protocol of the second server, and vice versa, thereby facilitating cross-server message exchange. The gateway may also permit a shared directory service between the two servers, by periodically merging the directory entries of each individual server and then re-populating the directory on both with the newly updated user listings.

2. Message Switch

A *message switch* is an advanced software application that permits integration between multiple disparate e-mail servers. If a company has historically allowed its departments to install and manage their own e-mail servers, and now wants to enable e-mail communication throughout the entire enterprise, a message switch would facilitate this. Each individual server would be pointed to the message switch, and the switch would deliver messages to the end destination server. For example, a company has 15 different e-mail systems, including Microsoft Mail, Microsoft Exchange, Lotus Notes/Domino, and Novell GroupWise servers. Rather than installing a message gateway between each set of two servers, each individual e-mail server is connected to the message switch which in turn knows how to deliver e-mail to each connected server.

3. Internet Connectivity

The final way to integrate e-mail servers is to use SMTP. Each e-mail server is connected to an Internet-based network, and given an Internet-based name and address. Internet standards are used for transferring messages between the different e-mail servers.

III. E-MAIL ARCHITECTURES

There have been five e-mail system architectures, which deal with how the various components of an e-mail system are organized. These are host-based e-mail, file-sharing e-mail, proprietary client/server, Internet client/server, and web mail. The five architectures are discussed below.

A. Host-Based E-Mail

The original architecture of e-mail consisted of powerful mainframe or minicomputers powering dumb terminals. A single vendor provided the entire solution, which meant that the administration tools were well integrated into the product.

Host-based e-mail systems continued into the 1990s to retain market share against newer client/server e-mail systems. While client/server e-mail was poor at scaling to handle lots of users, and didn't offer robust enterprise-class calendaring and scheduling, host-based e-mail did, and many organizations elected not to migrate to a poor cousin of their existing system. While there were widely recognized user benefits with client/server e-mail, such as the offer of a graphical user interface and the ability to transfer attachments between e-mail users, these user benefits were insufficient to overcome the enterprise- and administrator-benefits that came with host-based e-mail. Only a few organizations continue to use host-based e-mail today. Representative products of this architecture are IBM OfficeVision and DEC All-In-One.

B. File Sharing

The first class of departmental, LAN-based e-mail systems followed a file-sharing architecture. The e-mail server functioned on a file server and maintained a database of files. The e-mail client was given read/write access to a shared drive on the file server, and would download new e-mails as a file, which the e-mail client would interpret and present as a list of new messages. Outgoing e-mails were posted to a different shared directory on the file server, which the e-mail server would poll periodically and pass on through its MTA to the destination post office.

File-sharing systems were designed to be easy to install and easy to operate and use due to their graphical interfaces, and generally be helpful for small workgroups. They were not capable of scaling to serve the enterprise. Those companies that replaced a mainframe e-mail system with a file-sharing architecture ended up with hundreds, if not thousands, of e-mail servers throughout the organization linked through a hodgepodge of e-mail gateways and e-mail message switches. Representative file-sharing products are Lotus cc:Mail, Microsoft Mail, Da Vinci, Banyan, and QuickMail.

C. Proprietary Client/Server

The client/server e-mail systems of the early 1990s were the last vestibules of proprietary e-mail. The functioning of the client and server were separated, but proprietary protocols were used to permit the client to work with a given server. This had a number of benefits. First, it permitted greater choice for organizations in the selection of an e-mail server as a separate issue from the selection of an e-mail client. No longer were the two tightly coupled, as had been the case with file-sharing architectures. Secondly, vendors were able to separate the development of an e-mail solution into two linked groups—one focused on the client, a second on the server.

Client/server e-mail brought substantial benefits to the table. First, it could scale well beyond traditional file-sharing e-mail systems, permitting a consolidation in the number of servers, and hence reducing the administration time and effort involved. Secondly, the servers were more reliable and crashed less frequently than file-sharing systems.

MAPI, VIM, and CMC all fought for the right to be called the "standard" protocol for client/server e-mail, but the messaging application programming interface (MAPI) won. It was a Microsoft protocol to permit client/server e-mail. Vendor independent messaging (VIM) was championed by Lotus and others, but eventually lost out to the market momentum behind MAPI. Common messaging calls (CMC) supported cross-platform e-mail and other advanced functions.

Leading products that followed a client/server architecture were Lotus Notes, Microsoft Exchange, Novell GroupWise, and Hewlett-Packard's OpenMail.

D. Internet-Based Client/Server

The Internet changed the world of e-mail. While it continued with the client/server paradigm, it specified open, noncomplex standards for ensuring interoperability between e-mail servers themselves for message exchange, and between e-mail clients and e-mail servers. In addition, it provided a low cost of entry for any organization that wanted to share e-mail between multiple parties. All that was needed was an e-mail server that spoke Internet protocols, a connection to the Internet, and a registered name in the Internet directory service.

E. Web Mail

Web mail represented a revolution in user access to e-mail. People want the ability to keep their e-mail private to themselves, and have access from anywhere, not just when they are at home at their home computer. Hotmail and Rocketmail were early free web mail services, allowing an individual to have a private e-mail inbox that could be accessed from any web browser around the world. Both were quickly acquired by larger companies, Hotmail by Microsoft, and Rocketmail by Yahoo!

All major e-mail vendors now support web mail access. This essentially means that any client that can speak HTTP, usually a web browser, can connect to their e-mail server, authenticate with a user name and password, and have access to all messages. Their inbox is painted within the browser window, and the user is able to create new messages, review new messages, add contact details for individuals to their address book, and a whole raft of other standard e-mail client services. There is no downloading of messages to a file on the local hard drive, allowing anywhere, anytime access to all messages.

Web mail is much slower than a traditional e-mail client, for three reasons. First, the server inherits additional processing that was traditionally performed by the e-mail client. This will improve over time as servers gain additional processing prowess. Secondly,

the bandwidth connecting a web mail client to a web mail server is usually lower than that connecting an e-mail client to an e-mail server. With web mail, most people dial into the mail server from a home modem, but most users reside on a corporate network with faster links in the second scenario. This second factor will be mitigated as broadband networks are deployed more widely. Finally, web mail clients use HTML for rendering the inbox, which means that every action initiated against the e-mail server requires that the inbox be repainted within the browser. Dynamic HTML eliminates this requirement, and will streamline the process in years to come.

IV. STANDARDS FOR E-MAIL

There are five key Internet standards for e-mail: SMTP, MIME, POP3, IMAP4, and DNS. These standards work together to facilitate the exchange of e-mail between e-mail users through Internet e-mail servers. The X.400 e-mail standard, the most widely used precursor to Internet e-mail, is also brief discussed (Table I).

A. Simple Mail Transfer Protocol (SMTP)

The SMTP describes how an Internet e-mail client transfers e-mail to an Internet e-mail server, and how Internet e-mail servers transfer e-mail between themselves. SMTP is a text-based standard.

SMTP is defined by a number of IETF standards documents. RFC821, RFC1869, and RFC1870 define the transport of messages over a network. RFC822 describes the format of a text-based e-mail message (RFC's are available at www.ietf.org/rfc/rfcXXX.txt, where the number of the RFCs is substituted for the XXX). An SMTP message is composed of two parts:

1. *Message header.* Includes such fields as the message sender, the intended recipient, and the subject of the message.
2. *Message body.* An SMTP message can carry a payload of one text-based message.

The original SMTP standards have been extended greatly to handle additional functions. The negotiation of service extensions (RFC1869), the declaration of message sizes (RFC1870), delivery status notifications (RFC1894), and the return of error codes (RFC2034) are a sampling of such extensions.

Table I E-mail Standards

Standard	Brief description
SMTP	The "Simple Mail Transfer Protocol," used for transferring messages between an email client and an email server, and between email servers.
MIME	A series of "Multipurpose Internet Mail Extensions," designed to enhance the base SMTP standard for advanced tasks.
POP3	The "Post Office Protocol, Version 3," for accessing email messages stored on an email server from an email client.
IMAP4	The "Internet Message Access Protocol, Version 4," for accesing email messages stored on an email server from an email client. More advanced than POP3.
DNS	The "Domain Name System," the directory service for the Internet. Email servers use the DNS to determine where email is sent for delivery to the end recipient.
X.400	A group of standards from the ISO for electronic mail. Not used widely.

B. Multipurpose Internet Mail Extensions (MIME)

The MIME standard describes how to represent and encode electronic objects—such as spreadsheets, images, audio, video, and other binary files—for transmission over an SMTP network. A vendor will frequently concatenate the two standards, saying that they support "SMTP/MIME." MIME is not just one standard document from the IETF, but consists instead of multiple standard documents covering individual aspects for extending Internet mail. See RFC2045 to RFC2049 for the current standards (see http://www.ietf.org/rfc/rfc20XX.txt where "XX" is swapped for 45, 46, 47, 48, or 49).

MIME extends SMTP in five key areas:

1. *Permits multiple body parts.* Whereas SMTP supports only a single text-based body part, MIME extends SMTP so it can carry a payload of multiple body parts, where each can use a different encoding mechanism, but where each encoding mechanism is known and recognized.
2. *Binary encoding.* Provides the ability to encode binary data for transportation over a text-oriented SMTP network.
3. *Alternative character sets in the message header.* Allows the use of character sets other than ASCII for representing the message header. This aids with internationalization of the Internet.

4. *Delivery and nondelivery reports.* Allows users to request confirmation that a message has, or has not, been delivered to the intended recipient.
5. *Security.* Secure MIME adds a digital certificate for encrypting messages, thereby aiding with nonrepudiation, authenticity, access control, privacy, and message integrity.

C. Post Office Protocol (POP3)

The post office protocol (POP) is the most commonly used message request protocol in the Internet world for transferring messages from an e-mail server to an e-mail client. With POP3, the e-mail client requests new messages from the e-mail server, and the server "pops" all new messages out to the client. The server does not retain a copy of the messages, unless specifically requested to do so by the e-mail client. The only copy of the message is now stored locally on the user's PC, using files on the local hard disk. For example, each folder in Microsoft Outlook Express has an associated MBX file on the hard disk that stores messages contained in the folder.

The current base standard for POP3 is described in *RFC1939 Post Office Protocol—Version 3* (http://www.ietf.org/rfc/rfc1939.txt) and *RFC2449 POP3 Extension Mechanism* (http://www.ietf.org/rfc/rfc2449.txt).

D. Internet Message Access Protocol (IMAP4)

The *internet message access protocol* (IMAP) details how e-mail clients interact with e-mail servers for receiving e-mail. IMAP provides a client-to-server message manipulation mechanism, allowing for server-based storage of messages, server-based folder manipulation, and selective downloads of messages or message headers to an e-mail client. While most ISPs have IMAP-compliant e-mail servers, most users remain with POP service. The current base standard for IMAP is described in *RFC2060 Internet Message Access Protocol Version 4 Revision 1,* available from http://www.ietf.org/rfc/rfc2060.txt.

IMAP makes life much easier for users with slow Internet connections, as well as mobile corporate users dialing into the e-mail server from a laptop over a cellular phone. It gives much greater control to the end user over which e-mail messages they download during a given e-mail session, because the user can elect to download only the headers of messages, and then delete certain message headers and identify other e-mails for which they want a full download. When they next synchronize with their e-mail server, those message headers that were deleted in the client are deleted in full off the server, and those e-mail messages that the user wanted to download are downloaded in full.

IMAP provides additional benefits:

1. *Server-based foldering.* E-mail messages can be stored remotely on the e-mail server in folders that the user has established. This frees the user from having access to their e-mail through only one computer, by providing centralized storage with access from anywhere.
2. *Server-based search.* The user can request the server to find messages meeting specific criteria, and the search will be carried out on the server rather than locally on a local message store as would be the case with POP3.
3. *Share-level access to folders.* Users can identify folders that they want to share with others, and those that they want to keep private.

E. Domain Name System (DNS)

The DNS is the Internet directory service that translates human-facing names (such as www.microsoft.com) to the corresponding TCP/IP address on the Internet (one of which for Microsoft is 207.46.230.218 as of March 2001). When a user creates an e-mail for bill@microsoft.com, their e-mail server consults the DNS to determine which e-mail server is responsible for the microsoft.com domain.

The mail exchanger (MX) record in the DNS defines which e-mail servers can receive e-mail messages for a specific domain. There can be multiple MX records for a single domain, as most large corporations will have multiple e-mail servers to receive and send Internet e-mail so as to guarantee reliability of delivery in the case of one e-mail server failing. The e-mail server sending e-mail to a given domain selects the MX record for the destination e-mail server using the preference order that is assigned in the MX record. For example, the MX records for microsoft.com might be:

```
microsoft.com     IN   MX    0   mail.microsoft.com
microsoft.com     IN   MX   10   inbox.microsoft.com
microsoft.com     IN   MX   20   message.microsoft.com
```

These three MX records say that an e-mail server sending e-mail to a microsoft.com address should first use the mail.microsoft.com e-mail server, because it has the lowest preference value of 0. In the situation that a connection cannot be established to mail.

microsoft.com, the e-mail server should try inbox.microsoft.com, the e-mail server with the next highest preference value of 10. Finally, if the first two e-mail servers are unavailable, try the final e-mail server of message.microsoft.com, which has the highest preference value of 20.

F. X.400

Before the widespread adoption of Internet e-mail standards in the 1990s for the exchange of e-mail between organizations, X.400 e-mail held an important role. X.400 is actually a group of technical standards that define the hardware and software requirements for e-mail, and is managed by the ISO (www.iso.ch). It was more advanced in functionality (e.g., with built-in security, read receipts) and reliability (e.g., with guaranteed message delivery times) than Internet mail, but also more complicated and difficult to implement. Very few organizations use X.400 e-mail now.

V. E-MAIL-ENABLED APPLICATIONS

The value of e-mail is compounded when integrated with other applications. While e-mail is an application service in its own right, it can also work closely with other applications within a computing infrastructure to offer notification and information services to users. These applications are referred to as being "e-mail-enabled."

A. Workflow Management

Workflow management systems provide an electronic mechanism for routing work tasks between people within a workgroup, department, business unit, or organization. When one person finishes a task on a given item of work, the workflow system examines the business logic that has been established for the given business process, and notifies the next person in the chain that they have an outstanding next action to complete on the piece of work. For example, when a lawyer completes the first draft of a new contract the workflow management system could automatically notify the appropriate senior partner that a new contract is ready for review.

Workflow management systems are usually e-mail-enabled in three ways:

1. *Notification of new action.* The next person in the business process is notified by e-mail that they have a new action to complete. An e-mail is sent by the workflow application to the specified user, who follows a link within the e-mail to the appropriate work item.
2. *Reminder of outstanding actions.* If a user ignores the first notification of a new action, the workflow system will usually notify them by e-mail again to request completion of the outstanding task. This e-mail may be marked "Urgent" to signal priority and importance.
3. *Automatic escalation of overdue items.* If a user neglects to complete a work item within the time allowed, a manager or other process supervisor can be notified by e-mail. The e-mail would state the work item to be completed, the individual originally assigned to the task, and the duration of time that has lapsed with no action being undertaken. The manager or supervisor can elect what action to take to ensure the work item is completed in a timely fashion.

Workflow management applications from Action Technologies, Lotus Development, FileNET, JetForm, TIBCO Software, Staffware, and TeamWARE Group are all e-mail-enabled. In addition, enterprise resource planning software includes workflow components and e-mail integration (e.g., Oracle, SAP, and J. D. Edwards).

B. Document Management

Document management systems provide life cycle management tools for documents from the point of creation through the eventual archiving or deletion of a document.

Document management systems are e-mail-enabled in the following ways:

1. *Notification of document to review in lifecycle.* When a user checks a new document into the document management system, the life cycle rules of the document may specify that a particular person needs to review the document prior to formal publication. As for the workflow management systems above, this individual can be notified by e-mail of an outstanding work item.
2. *Notification of new documents in repository.* People who work in a corporate environment suffer from too many documents to read and understand, and do not have time to personally search through document repositories to find new documents of relevance to their work. These users can establish an "interest profile" that

specifies authors, keywords, or subjects that are of interest, and according to a nominated time frame (either hourly, daily, weekly, etc.), the document management system will e-mail the user a list of new documents that aligns with their interest profile.

Illustrative document management systems that are e-mail-enabled are Documentum, FileNET, Lotus Domino.Doc, Hummingbird, Open Text, iManage, Interwoven, and Microsoft SharePoint Portal Server.

C. Instant Messaging (IM)

Instant messaging provides the ability to send textual messages, usually short ones in rapid sequence, between business colleagues and personal friends. It permits a real-time conversation medium without the use of a telephone. In addition, IM provides an indicator that shows whether or not the other person is on-line and available to communicate.

Instant messaging is an e-mail-enabled application in two ways:

1. *Presence indicator within inbox.* When you are reviewing the list of new e-mails in your inbox, your IM service can indicate whether the senders of recent e-mails are on-line at that specific time. If so, you can initiate an IM session to ask a specific follow-up question about the e-mail you have just received.
2. *Presence indicator within an e-mail message.* When you open an e-mail message from a colleague, a presence indicator could be embedded into the message that shows the availability status of the individual at that specific time. The recipient can click on the presence indicator to automatically initiate an IM session with the original sender.

Check out ICQ, AOL Instant Messenger, Microsoft's MSN Messenger, or Yahoo Messenger for personal use. Corporate offerings are available from Lotus Development, Microsoft, and PresenceWorks.

VI. SUPPORTING INFRASTRUCTURE

E-mail does not stand alone. It requires other supporting factors within a computing infrastructure. Six important aspects of the supporting infrastructure are help desk support, directory services, content protection, virus protection, public key infrastructure, and message archiving. These six aspects are analyzed below.

A. Help Desk Support

Users struggling to complete a function using their e-mail client require someone in which to turn. This is usually a technically minded co-worker, or if that fails, technically trained help desk staff. The latter group of people should have a good understanding of the e-mail system being used, and be avid users themselves so they can help individual users troubleshoot their way through their own individual problems. In the situation that help desk staff are located remotely and there is no on-site support staff, remote control software allows the help desk staff members to take control of the user's computer over the network and lead the user step-by-step to a resolution.

B. Directory Services

We talked above about e-mail-enabled applications; some take the perspective that e-mail itself is a "directory-enabled" application—that without a directory of users, e-mail doesn't work. Two types of directories are helpful to e-mail.

The first is the user's personal address book, or directory, of colleagues and contacts. The address book allows the user to store the name, address, phone number, and e-mail address of their contacts, and then select these names from within a dialog box when composing a new e-mail message. This saves forcing the user to remember the e-mail address of all the people with whom they communicate.

The second and more important directory is the enterprise directory that stores the routing information for new messages. When a new message arrives at an organization, the e-mail server consults the directory to determine in which mailbox to deliver the message. As we discussed in Section II.B, many organizations have dual addressing schemas: the first is the addressing schema of the e-mail system they use internally, and the second is the addressing schema for Internet users. The enterprise directory translates the addresses between these two addressing schemas; all e-mail destined for an external recipient is given the Internet e-mail address for the user, and all e-mail entering the organization from outside has the delivery address translated from the Internet address to the internal address.

C. Content Protection

Most companies love free publicity that gets their brand name in front of a global population of consumers.

However, when the publicity has to do with employees distributing pornography or crude jokes, such as happened with RoyalSunAlliance, Dow Chemical, and others during 2000, it sends the wrong signal. Hence, organizations are implementing content management systems that review and filter all e-mail traffic to ensure only business purposes are being served by the e-mail being sent. Questionable e-mails, or e-mails with certain types of attachments, are quarantined for review by a systems administrator, and escalation to management for disciplinary action if necessary.

Within organizations, the philosophy behind the implementation of these systems is that the organization owns the e-mail system and provides it for business use. Individuals are not permitted to use the e-mail system for personal purposes, and enforcement mechanisms are put in place to ensure compliance with the policy. Companies such as Content Technologies, Trend Micro, and Tumbleweed are active in this area.

D. Virus Protection

E-mail has become a key way to inflict damage on organizations through the distribution of viruses. A virus is attached to an e-mail message, either within the message itself, or more commonly as an attachment to the message, and when the user reviews the e-mail and its attachments, the virus initiates. Some do relatively harmless, but nonetheless annoying things, such as sending a copy of itself to every person listed in the user's e-mail address book; others are more malicious and delete or corrupt user and system files on the local hard drive before sending itself to others for the same action to be initiated. Needless to say, this is a huge productivity killer for organizations, in terms of lost time for individual users, and time and expense to eradicate the viruses by the IT department.

Many corporations are now implementing virus-scanning software on their firewall and e-mail servers. New e-mail and attachments are automatically scanned for known viruses, and offending e-mails quarantined for deletion or review by a systems administrator. Organizations must be vigilant in keeping their virus signatures up to date, otherwise new versions and strains of the virus will get through undetected. Trend Micro, Symantec, and McAfee offer products in this area.

E. Public Key Infrastructure (PKI)

It is relatively easy to forge messages and make it appear that a message has come from someone else. For organizations conducting business over e-mail, this is unacceptable, and the addition of security to an e-mail system is implemented to overcome this problem. The most reliable way to do this involves the use of public key/private key encryption, using digital signatures managed through a public key infrastructure. When a user creates a message, their private key is used to sign the message. When the recipient receives the message, the e-mail is checked against the public key held in the public key infrastructure. If the public key can unlock the signed message, then the end recipient has a high level of confidence that the message is authentic, reliable, and unaltered. Representative vendors in this market are Entrust Technologies, IBM, and VeriSign.

F. Message Archiving

Most users hoard e-mail and refuse to regularly purge their mailboxes of outdated e-mail. Message archiving software proactively enforces an archiving policy whereby old or outdated e-mail is migrated from online storage on the e-mail server to a secondary server or storage medium. This helps ensure that storage devices in an e-mail server are kept free for new and current e-mail. Representative vendors in this market are K-Vault, OTG, and IXOS.

VII. CONTEMPORARY ISSUES WITH E-MAIL

The e-mail world isn't standing still—there are many changes going on in the industry, and the contemporary challenges are many and varied. We discuss these in this section.

A. Secure E-Mail

E-mail can be intercepted by third parties and read, modified, or deleted without the intended recipient knowing. In situations where business is conducted over e-mail, this is potentially hazardous to the health and well-being of organizations and employees. There are various methods of adding security to e-mail, with digital certificates playing a central role. When an e-mail is sent, the digital certificate, or "signature," of the sender is added to the e-mail. This digital certificate prevents the e-mail from being modified without breaking the original digital certificate. It also encrypts the message so that only those people with a corresponding key can unlock the message and read its contents.

B. E-Mail for Wireless and Mobile Devices

Many business people travel frequently, to meet with a customer, negotiate with a supplier, or network with colleagues and competitors at a conference or tradeshow. Such travel removes the individual from the standard suite of productivity tools available to them at the office. Given the centrality that e-mail currently holds as the conduit of much organizational communication, there is little wonder that an increasing demand for off-site access to e-mail is evident. Access from wireless and mobile devices is the latest of these demands.

A very high proportion of mobile business people carry a mobile phone. The argument goes that it would be great if access to e-mail could be obtained through the phone device, rather than always having to set up a laptop, log into the Web, or call back a secretary and request that the latest bunch of e-mails be faxed to the hotel. Hence phone designs are evolving to offer advanced capabilities to simplify the access to e-mail over a cellular network.

This demand is driving a change in the design of phones. Whereas a small keypad and even smaller screen is sufficient for dialing a phone number, the standard business phone is nearly useless for e-mail. The top three phone manufacturers (Nokia, Ericsson, and Motorola), a host of smaller new entrants (such as NeoPoint and Research In Motion), and the standard bearers of the PC age (Microsoft with Samsung and Ericsson, Lotus with Nokia) are all focused on developing the next greatest thing in phone design for an information-rich world. The emergence of devices with bigger screens, color screens, new methods of text-input, larger keyboards, combination phone/ PDA devices, and speech recognition are the result of this fervent episode of creativity.

C. Personal Use of Corporate E-Mail

Business managers provide resources and other appropriate productivity tools to assist its employees in getting their work done. E-mail is one such resource.

There are two arguments for disallowing personal use of a corporate e-mail system. The first is that it reduces productivity. The employee is getting paid to do their work, and instead is using their time for nonbusiness reasons. There are obviously degrees of misuse, ranging from the employee who sends one or two e-mails a day to friends and uses no more than five minutes to do so, or the individual who spends 90% of their time e-mailing various friends about upcoming parties. Each organization needs to decide where they draw the line on this; some help may be found by looking at the current policy on using business phones for calls to friends.

The second is potentially more damaging to the business. An employee who is given a corporate e-mail address is a representative of that organization to the public. In 2000, a number of high-profile organizations found themselves in the public news over inappropriate personal e-mail sent by members of their staff, most to do with sexual lewdness. This is unhelpful for brand management and establishing the professionalism of the organization in the mind of its various stakeholders.

D. Monitoring Employees' E-Mail

Organizations taking the approach that e-mail is a business resource for use on business matters usually want the right to monitor employee compliance with the corporate guidelines. This allows nominated individuals under management direction to open the mailbox of specified employees and look through their e-mail.

It is highly recommended that an organization has in place a written policy that all employees have signed that outlines what rights of monitoring the organization retains. While the specific rights of the organization depend on the legal jurisdictions in which it operates, having a written policy helps in communicating these rights to employees.

It is important to place limits on the power of system administrators. Most system administrators have unreserved and untraceable access to all e-mail inboxes throughout an enterprise, allowing them to read at will what they want. Given the sensitivity and confidentiality around much business communication, these powers should be limited so that two individuals have to cooperate in order to review the e-mail of another.

E. E-Mail Outsourcing

The economic theory of transaction costs posits that businesses will organize their value creation activities using internal hierarchies or external market players depending on the cost of transacting business. Developments in technology that reduce transaction costs encourage a move to market-based sourcing of value-creating activities, and a consolidation within the internal hierarchy around the essential elements of "Why are we in business?" The rise of the Internet as a communications backbone between organizations

is driving a substantial worldwide reduction in transaction costs. In terms of an internal e-mail system, many companies are deciding to focus management time and attention on other priorities, and let specialist outsourcers take responsibility for the delivery of e-mail across the organization.

There are two general approaches to the outsourcing of e-mail: where e-mail outsourcing is a separate project from the outsourcing of general information systems functions, or secondly, where it is an integrated element of an overall outsourcing plan.

The specialist outsourcer maintains an e-mail infrastructure within the corporate organization on behalf of the organization. Servers, backbone network connectivity, and administration personnel are located on-site by the outsourcer at the major locations of the client organization. All that really has changed is that the responsibility for operating the system has moved outside the management hierarchy.

The alternative for the specialist outsourcer is to provide servers and backbone connectivity through its own network of data centers. This provides much greater scale opportunities for the outsourcer, and hence a reduction in the unit cost of delivering e-mail to the client organization. The outsourcer provides secure, fast, and probably redundant network connections between the client organization and its data center. Servers and messaging data are hence managed through their life cycle off-site to the client organization.

Any client organization moving toward outsourcing, whether provided internally or through an external network of data centers, should put in place a *service level agreement* (SLA) between itself and the outsourcer. This is essentially a contractual understanding of expectations between the two organizations, and should cover such items as delivery time for e-mail, backup procedures, response time for major and minor problems, costs incurred in adding a new user or new site, and the financial terms of the outsourcing agreement.

F. Time-Limited E-Mail

E-Mail can get you in trouble—big trouble—particularly if your e-mail contains sensitive company information, or highly private information, that you don't want to share with others. New add-on products to e-mail clients and servers permit users to specify how long they want an e-mail to remain readable. After the expiration date and time is reached, the e-mail becomes unreadable and cannot be accessed by others. In addition, detailed statistics about who opened the e-mail and what they did with it can be tracked and reported on. This is a relatively new technology area, but more companies will adopt such offerings to protect themselves and their confidential corporate information.

G. E-Mail Overload

Many people receive too much e-mail, e.g., it is not uncommon for business people to receive 50–200 messages per day. That is a lot of e-mail to read, let alone intelligently respond to in a timely fashion.

There are some technology solutions that aid in processing such a huge volume of e-mail, but the battle is won or lost in personal habits. On the technology side, use filters within the e-mail client to sort new e-mails into predefined folders. Perhaps you are working on six high-priority projects and will receive an avalanche of e-mail about each one for the next couple of weeks. Set up a rule that automatically filters all project e-mail into separate project-specific folders, so that you don't have to perform this task manually and can quickly review specific messages about the project at a time you choose.

In addition, blacklist people you don't want to receive e-mail from. If identifiable people within your organization, or external to it, are sending you e-mail which clogs up your inbox and drains your productivity, set up a rule that automatically deletes their e-mail.

On the personal habits front, set aside time each day to process e-mail, and for this most people need 30–60 minutes. Work through each e-mail in sequence, and take whatever action needs to be taken as quickly as possible. If a direct report requests feedback on an idea, provide it there and then. If a new task can be delegated or redirected to someone else in the business, do it quickly. If there is a major new piece of work coming out of an e-mail, note it in your time and task management system so it can be prioritized and completed as appropriate.

Disconnect from your e-mail when you require focus and concentration for a current task. Elect not to be notified of new e-mail as soon as they arrive; instead take control of your inbox and use it as your tool, not your master. If you turn on the ubiquitous "beep" and pop-up notification for every time a new e-mail arrives, you'll be constantly interrupted and distracted from your current task at hand. Leave e-mail to gather, and then when you have a few spare moments during the day, or you come to your "e-mail hour," you can give maximum focus to your new messages.

Sign up for newsletter e-mails judiciously. There are a plethora of e-mail newsletters available that promise to keep you informed about developments in

your industry, technology, and other high-priority items of interest. Subscribe to the best ones, and regularly prune those that add little value.

Encourage people to take responsibility for the decisions they make, rather than feeling like they have to *carbon copy* (CC:) every piece of project communication to their manager. This will reduce a large portion of new e-mail each day; the individuals concerned can always review project progress with their manager at a weekly or otherwise regular catch-up to review project status.

Finally, don't send e-mail at the expense of relating to people in person. E-mail is a great tool for the quick exchange of information, reports, documents, etc., but the nature of e-mail, and the brevity which most people use in their e-mail, can quickly escalate misunderstandings to cause substantial organizational dysfunction. Regular face-to-face meetings, or phone calls if this is financially practical, with the key people in your organization, and external to it, will ensure that any misunderstandings can be quickly resolved without causing massive problems.

H. Junk E-Mail, or Spam

Junk e-mail clogs up corporate servers, fills personal inboxes, and wastes time for each individual user as they sort through new e-mail to remove unwanted mail. Junk e-mail is also called "spam." Junk e-mail falls into three categories:

1. E-mail from someone you don't know that offers to sell something of purported value, e.g., cable TV service
2. Repeated e-mails from someone you do know that are irrelevant to your work tasks and responsibilities
3. E-mail from a known individual with whom you don't want to communicate anymore.

Many of the content management administration applications available offer content scanning for junk e-mail. If certain words or expressions are found within inbound e-mail, these e-mails are moved to special quarantine databases for examination by a systems administrator. E-mail from known domains can be rejected automatically, as well as e-mail from specific individuals. Keeping these "blacklists" up to date requires a high level of discipline from the systems administrator.

E-mail clients offer capabilities to individuals to assist them in their fight against junk e-mail. Microsoft Outlook includes a menu item called "Junk E-mail," and gives the user the ability to add the name of a specific individual to a blacklist. All e-mails from this person in the future will be automatically rejected.

I. E-Mail Retention Policies

An e-mail retention policy formally mandates the length of time that employees must retain copies of e-mail. With most people using e-mail to conduct business, there is a need to manage these communication records, deleting e-mails that are not needed, and protecting e-mails that cover sensitive and confidential discussions about employees, business strategy, and competitors. Draft your e-mail retention policy in close consultation with your lawyers, human resources department, and the IT staff.

J. Unified Messaging and Unified Communications

E-mail is only one of many personal productivity tools used by employees—others include the telephone, voice mail, fax, instant messaging, and discussion groups. These tools all help people communicate and collaborate within organizational boundaries and external to them. However, they are all different in the user interface, access options, administration and support, and infrastructure integration capabilities. Unified messaging and unified communications attempt to bring all messages into a single place where users can interact with those messages, decide how and when to respond using different response channels (e.g., e-mail, instant messaging, a phone call), and integrates the user address book and knowledge of the capabilities and preferences of each recipient in order to optimize the distribution of responses (i.e., if the user sends e-mail, but three recipients prefer fax, the e-mail will be automatically delivered as fax to these recipients).

There are many benefits of this approach, even though few organizations have deployed the capabilities. For the user, they have a unified interface into all their messaging and real-time communications, rather than a series of stovepipe applications that reduces productivity. For the administrator, they have centralized administration and management, a single message store, and a single infrastructure for all messaging and real-time communications. This also adds great productivity, and simplifies future planning, for the administrator.

There is much work to be done in this area, including working out optimal user interface designs, getting message translations to work correctly between all devices, and incorporating intelligence into the delivery preferences of individuals across the world.

VIII. FUTURE ISSUES AND TRENDS

A. Will E-Mail Retain Its Centrality in User Experience?

E-mail currently reigns supreme as the central virtual working space for business people. It is from e-mail that they communicate with others inside and outside their organization, from where they receive new tasks and report on tasks completed, and in which they maintain a quasi-project management system through e-mail folders that group related messages. But will this situation remain?

Both portals and IM threaten the centrality of e-mail to users. Portals have an advantage over e-mail in that they interface with additional organizational and web-based services to provide an alternative central virtual working space. While a well-designed e-mail client gives access to e-mail, calendar and tasks, a portal has much greater flexibility. The main screen of a portal can be configured with the three standard e-mail services, news channels (corporate or web-based), reporting modules (e.g., financial performance), and other indicators of corporate health. Given this increased functionality, will portals reign supreme in the years to come, relegating e-mail as "just one of many" important corporate applications?

Instant Messaging also has an advantage over e-mail. Both share surface similarities, but IM offers additional value in certain areas. With both it is possible to send text messages to other people, but instant messaging adds the ability to know whether the recipient is on-line and available to chat. This presence information, when coupled with the real-time nature of text message exchange over instant messaging, makes it a very powerful communication channel to support virtual, distributed workgroups. No longer is it necessary to send an urgent e-mail and hope for a reply within a couple of hours; with IM you have immediate awareness of whether the person you need to communicate with is currently available. If so, the first party can start talking with them by exchanging text messages, and migrate to a shared meeting room with application sharing if necessary. Alternatively, recourse to a telephone for a voice conversation is an option once you know the other party is available.

Such interactions offer fast turnaround for additional feedback to ensure that each party has fully comprehended the intent of the communication. Additionally, such interactions can quickly escalate from a purely text-based message exchange to a full meeting permitting the sharing of applications and application data to permit deeper understandings to develop between individuals.

B. XML in E-Mail

Extensible markup language (XML) is taking the technology world by storm. It provides a simple way for exchanging structured data between applications within an organization, and external to it. This is a revolutionary concept within the IS world, and addresses key problems that have been experienced for many years. An XML document follows a structured layout as specified by a certain XML document definition, and the receiving application knows how to interpret the XML code and do something intelligent with it. Some e-mail vendors, both established and new entrants, are working toward the use of XML within e-mail.

XML in e-mail will provide two benefits. First, it will allow messages to intelligently handle themselves without user intervention. This will obviously only apply to some messages, as a good proportion of messages require personal review. E-mails that confirm meetings, announce flight delays, or provide time-limited information can automatically update the users schedule, notify them by pager of the flight delay, and delete themselves when the relevant time has passed.

The second area of benefit relates to integration with mobile and wireless devices. The display capabilities on such devices are vastly inferior to a desktop or laptop computer. Hence, an e-mail with XML tags will provide specific instructions to the device on how to appropriately display and format the content.

E-mail messages that can automatically initiate actions within an e-mail client and infrastructure hold the potential for great harm. For example, this would be an ideal way to transfer viruses or other damaging payloads between applications. Hence, vendors offering scanning and protection utilities will need to increase their capabilities to prevent against such negative occurrences.

SEE ALSO THE FOLLOWING ARTICLES

Computer Viruses • Desktop Publishing • End User Computing, Managing • Internet, Overview • Intranets • Security Issues and Measures • Video Conferencing • Voice Communication • XML

BIBLIOGRAPHY

Albitz, P., and Liu, C. (1998). *DNS and BIND,* 3rd ed., Sebastopol, CA: O'Reilly & Associates, Inc.

Hughes, L. (1998). *Internet e-mail: protocols, standards, and implementation.* Boston, MA: Artech House.

Sampson, M., and Ferris, D. (2001). *The global business email market 2000–2005.* San Francisco, CA: Ferris Research Inc.

(March 1999) The 10 Commandments of Email. Harvard Communications Update. Volume 2 Number 3.

Electronic Payment Systems

Jane K. Winn
Southern Methodist University

GLOSSARY

automated clearinghouse An institution used by banks to exchange claims on each other in the form of electronic debits and credits.
CHIPS (Clearing House for Interbank Payments System) An automated clearing house maintained by New York City banks.
electronic funds transfer Movement of a financial value from one party to another by means of an electronic communications network.
e-money or **electronic money** Monetary value stored in electronic form that is widely accepted as a form of payment in a manner similar to cash in the form of a national currency.
FedWire The wholesale electronic funds transfer system maintained by the U.S. Federal Reserve Banks.
giro A system for transferring credits between banks and post offices that is available in places such as Europe, Japan, and Australia that have post office banks.
NACHA (National Automated Clearing House Association) The national organization for financial institutions that participate in the U.S. automated clearing house system.
SWIFT (Society for Worldwide Interbank Financial Telecommunication) An electronic communication system that supports cross-border electronic funds transfers between depository institutions.
TARGET (Trans-European Automated Real-time Gross settlement Express Transfer) An automated clearinghouse established by the European Central Bank for Euro electronic funds transfers.

AN ELECTRONIC PAYMENT SYSTEM is a networked system for sending and receiving messages in electronic form that transfers financial values. Electronic payment systems in widespread use today depend on the services of regulated financial intermediaries in order to function. In the future, more radical alternative electronic payment systems might eliminate the use of national currencies as the unit of account, or disintermediate traditional banking institutions, but have not yet gained widespread acceptance among users. Electronic payment systems have not yet been well integrated with other electronic commerce systems, such as those for negotiating and forming contracts, but it is likely that these systems will be integrated with more success in the future. While some electronic payment systems such as credit cards and automated teller machine debit cards are now available in most countries around the world, most electronic payment systems remain rooted in national currencies and have difficulty supporting cross-border payments.

I. THE ROLE OF ELECTRONIC PAYMENT SYSTEMS WITHIN INFORMATION SYSTEMS

The creation of a wholly electronic form of money has been one of the most compelling visions of the brave new world of cyberspace. Yet decades after the advent of large-scale computer networking, electronic money remains just that, a vision rather than a reality. Although the creation of a stateless "e-money" valorized only by

an allegiance of netizens and liberated from the fetters of government regulation has not been accomplished, every day trillions of dollars of value is transferred using much less glamorous "electronic payment systems." Electronic payment systems, in contrast to e-money, are currently in widespread use, based on well-established technologies, operated by existing financial institutions, and regulated by national governments.

Electronic payment systems are a key element of any electronic commerce system. Given the breakneck pace of innovation that is characteristic of many information technology markets, however, the world of electronic payment systems is characterized by an often painfully slow cycle of new product development and implementation. In addition, while the global information infrastructure effortlessly crosses national boundaries, electronic payment systems seem mired in national economic and legal systems and cross borders only with great difficulty. Electronic payment systems diverge from the general model of information economy business cycles that are characterized by compression of product development cycles combined with rapid adoption of new products and services. This is because of lock-in to mature technologies that are very reliable and relatively inexpensive to maintain, the need for very high levels of security, and the need to integrate electronic payments with other business processes which may themselves be very complex and resistant to rapid change.

Electronic payment systems seem to be an area where the proliferation of new "e-business" models might have the greatest impact. The transfer of value that is the heart of a payment is an abstract notion that can be represented in digital media and automatically processed. Yet the most radical innovations in the world of electronic payment systems seem unable to make the transition from a promising concept or even pilot to enjoying widespread acceptance in the marketplace. As a result, traditional payment systems seem unlikely to be killed off suddenly by a manifestly superior alternative, but rather are condemned to "death by a thousand nibbles" as small, incremental improvements gradually transform existing systems.

Electronic payment systems originated within national economies, and it has proven to be difficult, slow work to extend them into global electronic payment systems. This article focuses on electronic payment systems in one country, the United States, with only limited discussion of electronic payment systems operating in selected other countries. Electronic payment systems in the U.S. are similar to electronic payment systems of other countries in certain respects. This is often true because U.S. providers of electronic payment services have been at the forefront of developing new solutions to electronic payment systems problems and have been successful in exporting those solutions. For example, this is true of the automated teller machine technology for consumer electronic funds transfers (EFTs) and credit card processing systems which have created global networks for consumer electronic funds transfers. However, U.S. electronic payment systems also differ significantly from electronic payment systems of other countries in many important respects as well. For example, checks play a larger role in the U.S. payment system today than they do in the payment systems of many other developed economies, while the U.S. did not develop an equivalent to the giro system used in European countries. Attempts to promote the use of electronic payment systems such as stored value cards or electronic funds transfer systems similar to the giro have not achieved widespread adoption in the U.S., due in part to the continued popularity of checks as a form of payment.

The magnitude of electronic payments relative to all forms of payments may be measured in terms of

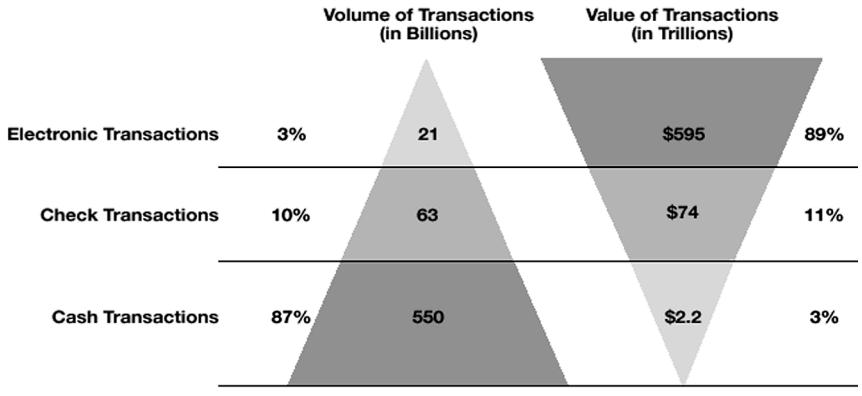


Figure 1 Inverse relationship between volume and value in the U.S. payment system in 1999. Source: National Automated Clearing House Association, copyright 1999, Federal Reserve Bank of Chicago. Used with permission.

volume of transactions, or transaction value. Figure 1 shows the inverse relationship between volume and value in the U.S. payment system in 1999: while the number of cash transactions dwarfed the number of electronic transactions, the value of electronic transactions dwarfed that of cash.

Although each national payment system is unique in many respects, Figs. 2 and 3 show that in recent years, payment systems in developed countries all follow a common trend toward greater reliance on electronic payment systems.

II. WHAT IS A PAYMENT SYSTEM?

Payment is one element of a commercial transaction. A payment system is an institutional process for handling the payment element of many transactions. A payment system can be distinguished based on:

- its legal status as money or a near-money equivalent
- the difficulty of reversing a payment once made, or its "finality"; the risk that a payment once received will prove valueless, or the "systemic" risk that a payment will fail because the payment system itself has failed
- the rules established by law or contract, or system rules, governing the responsibilities of the transacting parties should fraud or error resulting in an unexpected loss to one of the parties
- the complexity of the system within which transfers take place, or content of rules governing the "clearance and settlement system" that effectuates the transfer of value

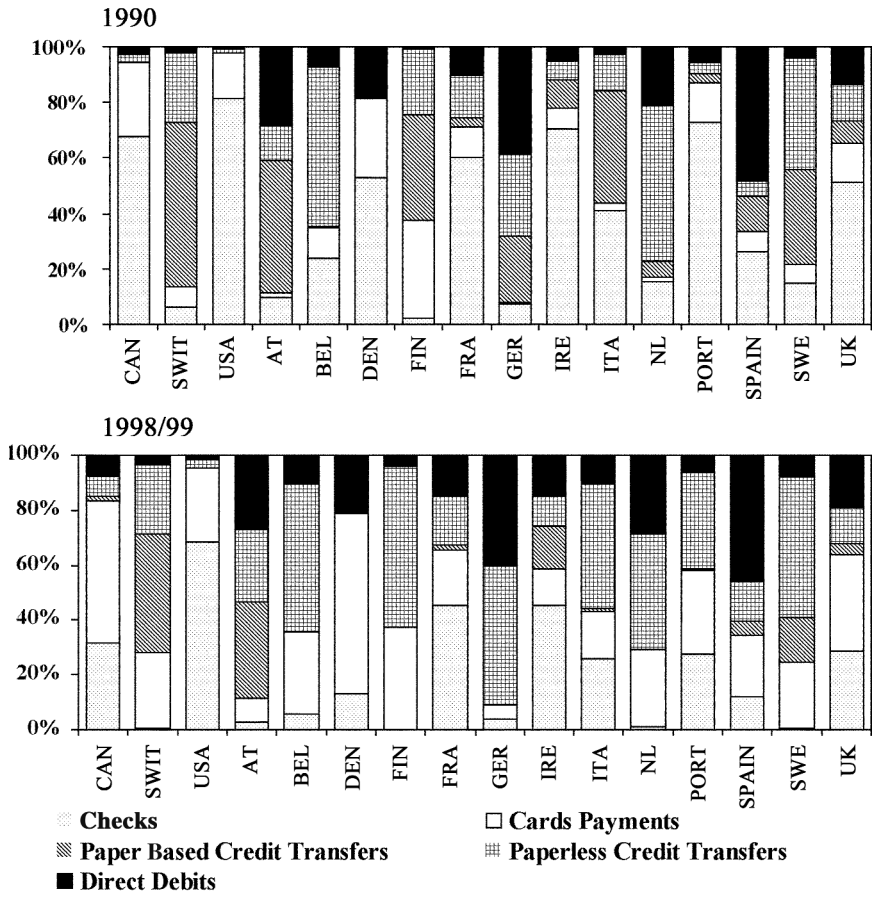


Figure 2 Relative importance of noncash payment instruments: volume of transactions (%). There are negligible shares for electronic money in Austria (0.15%), Belgium (2.26%), Denmark (1.35%), Finland (0.03%), Germany (0.10%), Italy (0.01%), Portugal (0.71%), and Spain (0.12%). This is not shown in the graphs. Data for Canada, Switzerland, USA, Belgium, Germany, Italy, Netherlands, Sweden, and UK refer to 1990 and 1999; data for Austria, Denmark, Finland, France, Ireland, and Spain refer to 1990 and 1998; data for Portugal refer to 1991 and 1998.

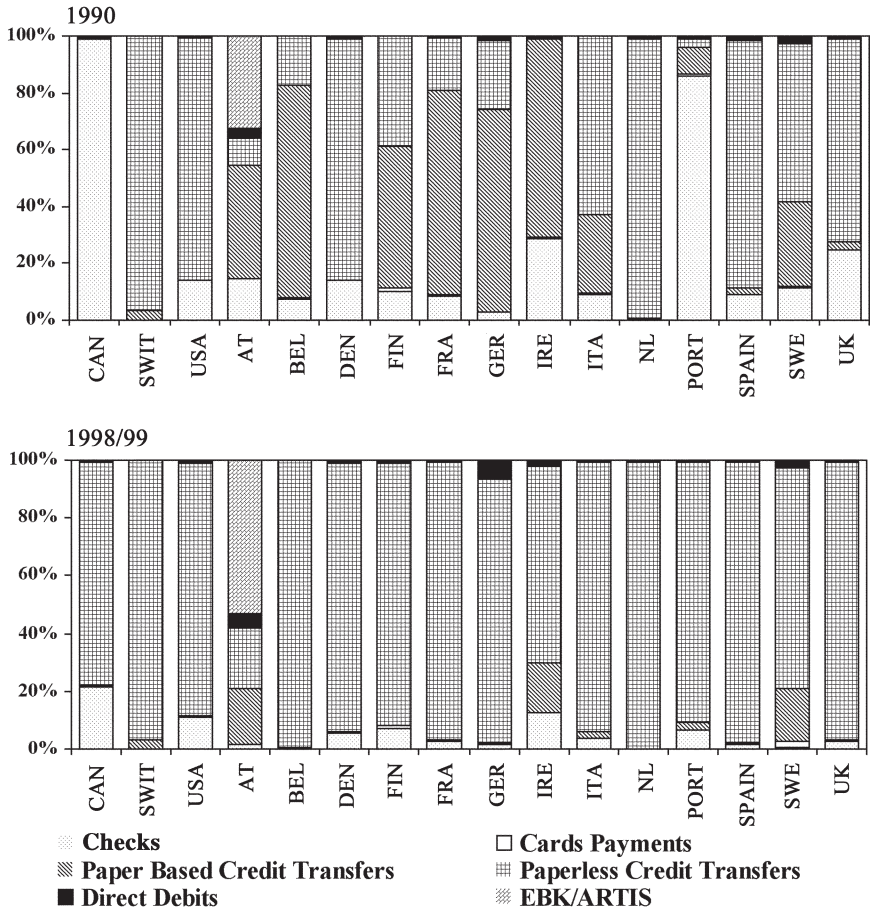


Figure 3 Relative importance of noncash payment instruments: value of transactions (%). There are negligible shares for electronic money in Austria (0.0007%), Belgium (0.001%), Denmark (0.27%), Finland (0.00002%), Germany (0.001%), Italy (0.000003%), Portugal (0.002%), and Spain (0.003%) in 1998, and they are not shown on the graphs. In Austria, 32.4% in 1990 and 52.8% in 1998 of the total value of noncash transactions is attributed to EBK (1990) and ARTIS (1998). Data for Canada, Switzerland, USA, Belgium, Germany, Italy, Netherlands, Sweden, and UK refer to 1990 and 1999; data for Austria, Denmark, Finland, France, Ireland, and Spain refer to 1990 and 1998; data for Portugal refer to 1991 and 1998. Source: Bank of International Settlements (1993, 1996, 1999, 2001), *Payment Systems in G10 Countries,* Basle; European Monetary Institute (1994, 1996, 1999, 2000), *Payment Systems in European Union,* Frankfurt. Copyright 2002 by S. M. Markos and Y. J. Loke; see http://privatewww.essex.ac.uk/~scher/updated trends in payment systems.doc. Used with the authors' permission.

- the character of the payment as a present exchange of values or a grant of credit for which a subsequent settlement will later be required
- the amenability of a payment system to being established within or converted to an electronic format

A. Barter or Gift Exchanges

Some exchanges of value do not count as "payment systems" even though each party to the transaction has been compensated. A barter transaction does not require the use of a payment system. An example of a barter transaction might be an itinerant trader exchanging manufactured goods for furs from the indigenous people in the territory he is crossing. A more modern example of a barter transaction might be the 1986 exchange of the right to market Pepsi-Cola in the USSR for the right to export Stolichnaya Vodka to the U.S. Many discussions of electronic payment systems mistakenly assume that before money was the most common medium of exchange, economic relationships in traditional societies were based on barter, but this is not historically accurate. Economic relationships in traditional societies are grounded in a complex web of noneconomic relationships. Economic ties forged through a lifetime of carefully calibrated

"gift" exchanges not only ensure that the material needs are met, they also ensure that questions of loyalty, hierarchy, and cultural values are addressed at the same time. In modern market economies a vestige of the traditional gift economy remains in the system of "relational" contracts in which the parties make long-term commitments to each other and do not expect a simple, mechanical reciprocity to be enforced for each exchange of value. While individuals with face-to-face relationships may be willing to cope with the inherent ambiguity and lack of finality characteristic of gift exchanges or relational contracts, such payment systems are difficult to adapt to electronic media.

B. Money

Before there were "electronic payment systems," there were payment systems. The simplest form of modern payment systems are based on cash payments, or the use of money that is a physical token designated as legal tender by a national sovereign. Money created by a sovereign serves more than just as a payment device, however. Money is also a store of value and unit of account. In addition, token money permits payments to be made anonymously and makes it more difficult to audit transfers of value.

The most important rights and obligations to parties using money as a payment device are simple and easy to determine once it is clear which party has possession of the physical token. Cash payments are final when the money changes hands; the risk of loss, theft, or destruction of cash normally lies with the party from whose control the loss arises.

Because legal tender in the form of physical tokens serves many important and valuable functions, considerable effort has been expended in the realm of electronic payments in the pursuit of some form of "e-money." Several plausible contenders have emerged in recent years, but market adoption of e-money remains slow and tentative. Part of the reason that e-money has not proven as popular as many expected is that the significance of cash as a payment device is declining in commercial transactions outside of retail markets.

C. Near Money

Transacting parties in many modern economies recognize certain forms of value as the functional equivalent of money. These normally include money held on deposit at a regulated financial institution such as a bank. The reason these deposits are treated as nearly money is that governments with modern financial regulatory systems limit the number of institutions that are permitted to take deposits from the public and engage in fractional reserve banking. In effect, a banking license in a modern economy is a license to issue near money. The quid pro quo normally demanded in exchange for that franchise by government regulators is that banks must submit to highly invasive regulatory oversight of their activities, including payment system activities.

The conversion of bank account money from paper to electronic form permitted the creation of one of the earliest types of electronic payment systems. EFT systems to move money in and out of bank accounts are currently the most important electronic payment systems in the U.S. The challenge for electronic commerce in the future is not how to process electronic payments, but how to adapt mature electronic payment systems to accommodate the demands of newer electronic commerce technologies and business models.

D. Clearance and Settlement Systems

For several centuries, commercial transactions in Western nations have been based not simply on barter, gift, money, or bank credit as a system of exchange, but have also relied on complex clearance and settlement systems. The creation of commercial credit devices, such as the letter of credit, the check, and the promissory note, and the creation of double-entry bookkeeping systems and organized financial and commodity exchanges gave businesses a wide range of alternatives to the physical exchange of cash as a payment system. A clearing system permits transactions to be aggregated and netted out before settlement. A settlement system permits final payment to be made for a transaction or group of transactions. For example, thousands or even millions of transactions may clear within an EFT network before a handful of final settlements are made at the end of the business day to close the books of the members of the network.

E. Nonbank Payment Systems

Many nonbank institutions now provide electronic payment systems that compete to some degree with those offered by regulated financial institutions. For example, telephone service providers or mass transit systems may provide users with stored value cards that can be used in combination with dedicated card readers to permit payment to be made at the time the service is

consumed. These electronic payment systems can be distinguished from money or near money electronic payment systems by the limited number of vendors that will accept payment in this form.

III. EXISTING ELECTRONIC PAYMENT SYSTEMS

In the U.S., there are several very mature, highly sophisticated electronic payment systems that are widely used. These include a wholesale funds transfer systems including the FedWire that is maintained by the Federal Reserve Banks and CHIPS, the Clearing House for Interbank Payments System that is maintained by the New York Clearinghouse; the automated clearinghouse (ACH) system which provides a low cost, relatively slow batch process system for recurring payments; the automated teller machine (ATM) network that provides retail electronic funds transfers for consumers; and the credit card network that provides retail electronic payments for consumers. While these systems provide almost all the electronic payment services used in the U.S. today, it is unclear how these systems will adapt to new business models and electronic commerce technologies.

A. Tested Telex

Tested telexes were probably the first modern electronic payment system. Telex was an electronic communication network that was in widespread use around the world from the 1930s to the 1990s until it was finally superseded by faxes and computerized electronic communication networks. Operators typed messages into telex terminals and then transmitted them over dedicated telephone lines. Banks used encrypted telex messages to send each other funds transfer instructions and other sensitive information in a manner that could not be intercepted or read by anyone but the intended recipient. A telex message would be encrypted using "symmetric key" encryption that required an actual tangible physical key be inserted into the telex machine. Banks shared copies of their encryption keys with other banks that kept them stored within secure locations in their funds transfer departments.

B. Check Collection

In the 19th century, U.S. banks were permitted to issue banknotes that circulated as currency until 1863, when Congress reserved the issuance of banknotes to the federal government. Banks that had depended on the issuance of banknotes as a source of capital were left scrambling to find alternative sources of capital, and discovered checking accounts. Banks in the U.S. have heavily promoted the use of checking accounts by their customers for over a century. In the 1970s, when electronic alternatives to payment by check became available to U.S. banks and their customers, U.S. banks first began trying to convince their customers to stop writing checks and take advantage of electronic alternatives. These efforts have been spectacular failures. By 2000, banks had barely managed to slow the rate of increase in the volume of checks written in the U.S., which by that time had reached nearly 70 billion a year.

The check collection process in the U.S. is a staggeringly large and complex system that relies heavily on computers to process and move hundreds of millions of pieces of paper a day. Checks are processed by having computers read the magnetic ink character recognition (MICR) numbers printed on the bottom of the check. The MICR line is preprinted with a number identifying the bank, a customer account number, and the check number, and after the check has been issued and deposited in a bank, it is encoded with the dollar amount of the check. MICR line technology was developed in the late 1950s by banks that then invested in computerized check reading technologies throughout the 1960s and 1970s, speeding up the check collection process and lowering its costs. Today, efforts are being made to improve further the efficiency of the system by sending the information contained in the MICR line in electronic form to the payee bank, either in advance of the paper check ("electronic presentment") or in lieu of the paper check ("truncation"). These improvements can marginally speed up the process and lower the cost of check collection, but limitations of MICR line technology limit the number of advances that can be made in the check collection system itself. For example, there is no way to encode the name of the payee on a check using MICR line technology, so if the paper check is converted to electronic form and destroyed, it may later be difficult to determine who was the payee of the check.

Businesses and consumers in the U.S. prefer payment by paper check to electronic alternatives because it is an open system that permits payment to be made to anyone in any amount, no matter how large or small; the check writer retains control over the amount and timing of the payment; bank charges for handling checks are relatively low; customers can re-

ceive credit for having made a payment while still retaining funds in their checking accounts because of the delays in the process of clearing checks (a benefit known as the "float"); and they traditionally receive their cancelled check back from the bank which could be used as a receipt or proof of payment in the event of a dispute with a payee (although many banks now charge extra for this service and it is not as common as it once was).

C. ACH

In the 1970s, the automated clearinghouse (ACH) was created, building on the check collection system and MICR line technology. Checks are cleared through "clearinghouses" and the automation of the check collection process made it only a small step to create a wholly electronic clearinghouse which could process instructions to credit or debit money from checking accounts using electronic messages formatted in a manner similar to the information encoded on the bottom of paper checks. Management of the ACH system is handled through the National Automated Clearing House Association (NACHA) and regional clearinghouse associations whose members are depository institutions. Although all depository institutions in the U.S. participate in the ACH system, over 90% of the processing of ACH payments is still performed by the Federal Reserve Banks (the "Fed"). The Fed got into the business of providing ACH data processing services to ensure the new ACH payment system was a success. In the following 25 years, a few competitors have emerged so the market for processing ACH payments may become more competitive in the future.

The ACH is a "closed" network that accepts inputs only from regulated financial institutions. Individuals or businesses wishing to make or receive payments by ACH must give instructions to their respective financial institutions. The ACH is a batch processed system, not a real time system, so instructions to debit and credit accounts are accumulated into batches for processing. As a result, a payment made by ACH requires a minimum of one to two days to settle. Because of the cumbersome interface for submitting ACH payment instructions and the delays caused by batch processing, the ACH is most popular in the U.S. for recurring payments in relatively stable amounts, such as salary and wage payments made by employers, and bill payments such as monthly mortgage or insurance premium payments by consumers that are for predictable amounts. Businesses have had some success in setting up payments to other businesses by ACH in lieu of checks, but it is difficult to make spontaneous or one time payments through the ACH.

A system for bundling business-to-business ACH payments with the remittance information required for a trading partner's accounts receivable department to process the payment was established in the 1980s. This system is known as Financial Electronic Data Interchange (FEDI). The system failed to catch on for two important reasons: the lack of standardization in the formatting of remittance information made it difficult for the recipient to interpret it and apply payments correctly in reliance on it; and the failure of many of the nearly 10,000 regulated depository institutions in the U.S. to upgrade their ACH systems to support FEDI. A business could not covert its accounts payable system to use FEDI exclusively because it could predict that many of its trading partners would bank with depository institutions that did not support FEDI and so would not receive the remittance information in electronic form. When the U.S. federal government mandated greater use of EFTs for payments to and from the government by December 31, 1999, depository institutions across the country finally upgraded their ACH systems to support FEDI. Progress in standardizing remittance information is proceeding more slowly, however.

D. FedWire

FedWire is a real time electronic funds transfer system set up by the Fed at the same time the ACH system was created. Like the ACH, FedWire is a service offered to banks; banks in turn offer the service to their customers and must transmit their customers' instructions to the Fed for processing. FedWire moves money out of one account at the Fed into another, and it is final for most practical purposes when the EFT instruction has been processed within the Fed's computer system. The service charges for using FedWire are greater than for using the ACH, but those higher service charges may be justified if there is a business need for faster, more certain movement of the funds, and they normally remain small relative to the amount transferred. Because a claim on a Federal Reserve Bank constitutes money in the U.S., the FedWire transfers are considered money, not merely a "claim" against one private party such as a bank to another private party such as a second bank. While for most purposes, the difference between a claim on a private commercial bank and a claim on a Federal Reserve Bank is not significant, if a funds transfer is for

a large amount of money or the recipient wants to be certain the payment is final and will not be reversed, FedWire is the best EFT system to use.

E. CHIPS

The Clearing House for Interbank Payments System (CHIPS) is a private clearinghouse that provides a real time funds transfer service similar to FedWire. At one time payments processed through CHIPS were made on a "net settlement" basis—at the end of the day, members of CHIPS paid only the net balance due to other members; they did not settle for each payment transaction individually. In recent years, CHIPS has moved to a continuous settlement system, thereby reducing the risk of a payment transaction being later reversed. Because CHIPS is a private organization rather than a government agency such as a central bank, there is a slightly greater risk that payments made via CHIPS might fail than payments made through a government agency. This is because, while some governments have defaulted on their payment obligations, such occurrences are less likely than the failure of regulated financial institutions that make up the membership of CHIPS. However, the member banks that make up CHIPS have all agreed to guarantee the security of payments made through that system, so short of a collapse of the entire commercial banking system in the U.S., it is believed that CHIPS as a system is unlikely to fail. In the event of a banking crisis of that severity, it is unclear what value even a FedWire would have given that it would still have to be credited to a customer's account at a commercial bank after the transfer was completed by the Fed.

F. Credit Card

For over 100 years, some retail establishments in the U.S. have granted credit to their customers to finance their purchases. By the 1950s, many merchants had created charge plates for their customers and inscribed the customer's and account number that could be used to facilitate charges to the customer's account. These merchant charge cards as well as checking accounts met the needs of most individuals shopping near home, but were often not useful away from home. In response to the need of business travelers wishing to charge their travel expenses the first modern third party credit cards were created: Diners Club, Carte Blanche, and American Express. In the 1960s, Bank of America created one of the first third party bankcards, which later became the Visa card; other banks joined together to form what became known as MasterCard.

These systems were originally based on paper accounting processes, but throughout the 1980s and 1990s, they increasingly switched to electronic processes. When a credit card is used today to make a purchase from a retail merchant, it is conventional for the merchant to use a point of sale (POS) terminal to transmit information about the card and the amount of the transaction to a credit card authorization service. That service makes a real-time determination whether there is sufficient credit available on the account to cover the transaction and whether the card has been reported lost or stolen. The merchant then receives back an authorization number indicating the transaction has been accepted by the issuer before accepting the charge in payment. The actual processing of the credit card charge itself, as opposed to the prior authorization, takes place later. It may be processed electronically immediately after the transaction has been completed, or several days later after the merchant has submitted paper credit card charge slips to the merchant's bank for processing. Once a month, the credit card issuer sends a statement to its customer and the customer has the option of paying all or part of the charges on the card.

G. ATM and Debit Cards

In the 1970s, banks began to offer their customers the option of making after-hours deposits and withdrawals from automated teller machines (ATMs) built into the walls of bank branches. In order to access a checking account using an ATM, a consumer normally needs an ATM debit card and a personal identification number (PIN). Banks began building ATMs in locations separate from banks and joining into alliances with other banks to create networks of ATMs. Some of these networks today are known as Pulse, Cirrus, NYCE (New York Cash Exchange), Plus, and Interlink. These ATM networks themselves established technological standards that permitted interoperability and entered into interchange agreements with each other and then with foreign bank ATM networks. As a result, there is now a global network for ATM-originated consumer EFTs.

In the U.S., there have been many attempts to promote the use of ATM cards to purchase goods and services at the point of sale. Consumers in the U.S. have been reluctant to stop writing checks, which give consumers the benefit of the float for several days, in

favor of ATM debit cards, which normally instantly debit their checking accounts at the time of the transaction. Because of the reluctance of many consumers to make purchases with ATM debit cards, many merchants have been unwilling to invest in POS electronic terminals capable of processing checking account debits using ATM cards together with a PIN. If consumers were more willing to make purchases with an ATM debit card, merchants would benefit several ways: a merchant is normally charged by its own bank just a few cents to process an ATM debit card transaction; payment is final when made and normally cannot be reversed by the consumer; and fraud problems are much lower with ATM debit cards than with credit cards because of the extra security provided by the PIN number. By contrast, the true cost of handling checks is quite high (once personnel costs and bounced check costs are taken into account); banks normally charge a percentage (from 1–5%) of the amount of the transaction to process credit card charges; consumers can stop payment on a check or credit card transaction more easily than an ATM debit card transaction; and fraud costs associated with both checks and credit cards are relatively high.

Most retail merchants in the U.S. have already leased or purchased POS terminals to process credit card transactions. Because credit card POS terminals are nearly ubiquitous in the U.S., Visa and MasterCard introduced a new payment service in the late 1990s that used credit card POS terminals but which debited money from the consumer's checking account instead of charging the purchase to the consumer's credit card account. These are known as "offline" debit cards because these transactions clear through the credit card communications network before they are finally settled and are contrasted with "online" debit cards, which are generally bank ATM cards that clear immediately against a consumer's bank account. Offline debit cards are marketed under the names CheckCard for the Visa product and MasterMoney for the MasterCard product and can be used without entering a PIN number. From the card holder's point of view, the only differences between using an offline and online debit card are the requirement to enter a PIN number and faster clearing with the online debit card. From the merchant's point of view, however, the charges for processing offline debit card payments are like those for credit cards (a percentage of the amount of the transaction), not like those for processing online debit transactions (just a few cents per transaction). While merchants are happy that consumers are more willing to use debit cards, they hate the higher charges associated with the offline debit cards. In 1999, major U.S. retailers including WalMart and Sears sued the credit card associations to challenge this pricing structure based on claims it violated U.S. antitrust laws.

H. SWIFT

The Society for Worldwide International Financial Telecommunications (SWIFT) is a wholesale funds transfer system widely used for funds transfers that take place outside the U.S. Unlike FedWire, for which funds transfers are executed by the U.S. central bank on its own books, and CHIPS, which is a centralized clearinghouse operated on a cooperative basis by the major New York commercial banks, SWIFT establishes a system of bilateral message exchanges between banks. Whether SWIFT funds transfers are electronic will depend on the arrangements between the two banks executing the funds transfer.

I. Giro

In many European countries, national post office systems provide basic banking services to the public. One of the banking services originally offered by post office savings banks and now also offered by commercial banks to their retail customers is a system of credit transfers known as giro. Unlike the U.S. ACH system, which can process both debit and credit funds transfer instructions, the giro system processes credit transfers. In other words, the giro only accepts instructions from account holders regarding payments to be made out of the customer's account; they do not process instructions to debit customer's accounts originated by merchants. In recent years, giro systems have been automated so their actual operations are similar to those of the U.S. ACH. With the creation of a single European currency, the Euro, efforts are underway by the European Central Bank to integrate separate national giro systems into a single clearinghouse for electronic payments known as TARGET (Trans-European Automated Real-time Gross settlement Express Transfer).

J. Stored Value Cards

It is possible to store a record of funds available for payment on a card and then use card reader devices to record withdrawals from and further deposits to that fund. A "stored value" card can be created using

a simple plastic card with a magnetic stripe on one side like those used for credit cards and debit cards. The security and functionality of the stored value card can be increased if the magnetic stripe is replaced with an integrated circuit (IC) computer chip. An IC card is often referred to as a "smart card" because some computing functions can actually be performed on the card itself, such as retrieve and apply a digital signature to an electronic record, not just the storage of data, which is all a magnetic stripe card can do.

Smart cards as stored value cards have achieved more widespread use outside the U.S. than inside. Smart cards first came into widespread use in Europe and Japan as telephone cards, and now the same technology is used for some retail payment services. In the U.S., stored value cards have achieved much more limited success, for example, as public transportation fare cards, photocopy cards, and student accounts for campus catering and retail services. Repeated attempts by major U.S. commercial banks to promote the use of smart cards as a retail payment device equivalent to cash, checks, credit cards, or debit cards among U.S. consumers and merchants have not yet succeeded.

IV. REGULATION OF ELECTRONIC PAYMENT SYSTEMS

Most electronic payment systems in the U.S. are operated by banks or other regulated financial institutions. The safe and sound operation of U.S. banks and their payment systems is something that bank regulators, such as the Federal Reserve Board, the Office of the Comptroller of the Currency, and the Office of Thrift Supervision at the federal level, or equivalent agencies at the state level, seek to guarantee. The rights and obligations of individual parties to payment transactions may be governed by special payment laws, or in the absence of a special law, by the agreement of the parties.

A. Contract

In the absence of a more specific statute or regulation, the rights and obligations of parties to a payment transaction are governed by private agreement. For example, many aspects of payments by smart card, e-money, or other emerging payment systems (described later) are governed by nothing more than the contract between the payment service provider and the users of the service. Under these circumstances, users may find that under their contract with the payment system provider, if a payment transaction is not completed successfully they have fewer rights and are offered fewer remedies than expected based on their experience with more mature, more highly regulated payment systems. This may be one of the reasons U.S. consumers have not to date shown much interest in emerging payment systems.

B. Negotiable Instruments Law

The rights and obligations of parties using checks are generally governed by negotiable instruments law. Negotiable instruments law developed several hundred years ago and although this body of law was organized and modernized in some areas when it was codified as Uniform Commercial Code (UCC) Article 3, it remains in many respects very complex and anachronistic. Under U.S. law, the fact that a check must be written on a piece of paper is one of the major obstacles preventing the development of electronic checks. In other areas of commercial law, archaic requirements that contracts be written on paper have recently been abolished, but negotiable instrument law was expressly exempted from the scope of such law reforms. Negotiable instruments law has retained the requirement that checks be on paper because bank regulators are worried that they are not yet prepared to regulate the use of electronic checks effectively and that the premature recognition of electronic checks might jeopardize the overall safety and soundness of the U.S. payment system.

C. Federal Reserve Board Regulations Z and E

Under the authority granted by Congress, the Federal Reserve Board has issued Regulation Z governing, among other things, the use of credit cards, and Regulation E governing, among other things, the use of debit cards by consumers. These regulations provide consumers strong protections against liability for unauthorized use of either a credit card or a debit card. If a consumer's credit or debit card is used without authorization and the consumer promptly notifies the card issuing institution of that fact, the consumer's liability in most instances will be capped at $50.

While the limit on liability in both cases is similar, the practical consequences of experiencing unauthorized transactions with a credit card or debit card remain quite different. If an unauthorized charge appears on a consumer's credit card, the consumer can

refuse to pay while the card issuer researches the consumer's complaint. If an unauthorized debit is made from a consumer's checking account, the bank issuing the debit card has the right to research the matter for 10 days before it is required either to deny the consumer's claim the charge is unauthorized or to recredit the consumer account. While some banks may recredit a consumer's card before the 10-day period is up, a consumer may nevertheless have serious problems from the time the funds disappeared out of his or her checking account until a recredit is issued.

Another significant difference between credit and debit card consumer protection concerns the consumer's right to dispute the terms of the transaction with the merchant. If payment is made by debit card, it is treated like a cash payment and the consumer has no right under Regulation E to ask the bank to recredit the consumer's checking account because of a dispute with the merchant. If payment is made by credit card, the card issuer is required to resolve the consumer's dispute with the merchant. Furthermore, as a practical matter, because the bank is required to resolve the dispute and the bank has a long-term relationship with the consumer, which will not normally have any relationship with the merchant, many merchants feel this dispute resolution process favors consumers.

D. Wholesale Funds Transfer Law

For the first 15 years that FedWire, CHIPS, and the ACH systems operated, there were only government agency regulations governing selected aspects of wholesale electronic funds transfers. The fact that there was no comprehensive legislative system regulating the rights and responsibilities of parties sending and receiving such wholesale funds transfers is quite surprising in light of the huge volume of funds being transferred through these systems. Finally in 1989, a new model law, UCC Article 4A, was issued, and because that model has now been enacted in all 50 states, it provides a statutory basis for determining the rights and obligations of parties to electronic funds transfers who are not consumers.

In most instances, UCC Article 4A applies very strict rules that force parties of funds transfers to pay for their own mistakes. For example, if a bank and the bank's customer have agreed on a reasonable online authentication system that permits the bank to receive electronic messages from the customer containing electronic funds transfer instructions, and an employee of the customer manages to circumvent the controls on the system and embezzle millions of dollars in the form of unauthorized electronic funds transfers, then it will be the customer's loss, not the bank's. The only exception to this rule is if the customer can prove the losses were caused by a hacker operating outside the customer's own computer system, which as a practical matter, is rarely the cause of unauthorized funds transfers. Similar rules allocate losses caused by the use of incorrect account numbers in funds transfers and other errors bank customers are likely to make. On the other hand, banks as a group are held responsible for the integrity of the system as a whole.

The loss allocation regime for wholesale funds transfers is in some sense the opposite of the loss allocation regime for consumer payments. In the consumer payment context, lawmakers more or less assume mistakes will happen and require payment system providers to fix consumers' mistakes for them. As a result, the prices charged for consumer electronic payment systems must be set high enough to cover the costs of fixing those mistakes. In the wholesale funds transfer context, where a single funds transfer may be for a billion dollars or more, it is not feasible to try to protect participants from the consequences of their mistakes. Banks can agree to service charges of only a few dollars per transaction to process electronic funds transfers for millions of dollars or more because in most instances, the bank will not be expected to bail out its customer if a mistake occurs.

E. Clearinghouse Rules and System Rules

Organizations such as check clearinghouses and the ACH have clearinghouse rules that define the rights and obligations of anyone using the clearinghouse. For example, a clearinghouse rule might specify a deadline for requesting that a payment that cleared through the clearinghouse be reversed. Financial institutions normally join the clearinghouse as members and take steps to ensure their employees learn the clearinghouse rules. Financial institution customers may be surprised to learn that they are also bound by clearinghouse rules, even though they have never heard of them and certainly never read them. This is normally accomplished by including a provision in the contract between the bank and customer that informs the customer that he or she is bound by the rules of any clearinghouse to which the bank belongs.

A similar system operates in the world of ATM debit cards and credit cards. System rules define the rights and obligations of all the banks participating in the

ATM network or credit card association. These rules are made binding on individual customers by contract. A major difference between the rules of the ACH, on the one hand, and debit or credit card system rules on the other is that the ACH rules are published as a book and can be made available to any member of the public interested in seeing them. Debit and credit card system rules are trade secrets that cannot be disclosed to members of the public. In effect, debit and credit card system rules are a sort of secret law that serves many of the same functions as payments laws such as negotiable instruments law, Regulations Z and E, or UCC Article 4A.

F. Money Services Businesses Laws

Many providers of electronic payment services such as Western Union or Paypal.com are not banks, and so the safety and soundness of the payment services they offer cannot be guaranteed by bank regulators. Many states have moved to fill this void by passing laws regulating money services businesses. For example, such a law might require a non-bank payment service to set up a trust fund to hold customer credit balances until they are paid out on behalf of customers. Individuals using non-bank payment services in states without such laws have no guarantee that the payment service provider has set aside adequate reserves to meet all its payment obligations. The European Union recently enacted a directive governing e-money, which after it has been transposed into the laws of member states, will regulate providers of e-money services in Europe to guarantee their safety and soundness.

G. Money Laundering Laws

One of the most important tools modern law enforcement authorities have for fighting organized crime is money laundering laws. The criminal laws in most developed countries now have laws prohibiting the practice of depositing funds earned in a criminal enterprise into a legitimate enterprise such as a bank or business in order to obscure their character as the profits of a criminal enterprise. As a result, many prosecutions of criminal activities proceed as money laundering cases because it is easier for law enforcement agencies to prove that money that had no legitimate source was deposited into a bank account or legitimate business than it is to prove just how the money was earned. Money laundering prosecutions are possible in part because most modern financial transactions leave a clear audit trail that can later be analyzed by law enforcement personnel. If electronic payment systems that do not leave an audit trail came into widespread use, then law enforcement agencies would see their ability to prosecute organized crime under money laundering laws erode. As a result of the fears of law enforcement agencies, there has been considerable debate over whether innovative electronic payment systems should be regulated or even prohibited if they are likely to be exploited by organized crime to escape detection by law enforcement. In recent years, this controversy has subsided in large part because none of the innovative electronic payment systems causing the most consternation have achieved any significant success in the marketplace, making the issue moot for the time being.

V. EMERGING ELECTRONIC PAYMENT SYSTEMS

The marketplace is now littered with failed recent attempts to start new electronic payment systems. While enthusiasm for innovative electronic payment services remains unabated in some quarters, in other quarters there is now considerable skepticism about the ability of emerging payments systems to achieve lasting success. For example, in 1994, First Virtual was established to offer a secure payments technology to support Internet commerce. In 1996, First Virtual enjoyed a very successful initial public offering, positioning itself as a major new Internet payments company. Individual subscribers authorized First Virtual to charge their credit cards for an initial allocation of funds to spend. Then subscribers visited the Web sites of merchants who had signed up with First Virtual and authorized the merchants to debit their balances by using a PIN number; First Virtual obtained confirmation of this authorization by a separate e-mail to the subscriber. The success of such simple alternatives as the secure sockets layer communication protocol (discussed later) eroded demand for the First Virtual service, however, which never achieved a critical mass of individual or merchant subscribers. In 1998, First Virtual announced the cessation of its payment services and the refocusing of the company on electronic messaging services only.

A. e-Money

e-Money electronic payment services use electronic tokens to permit online payments in more or less the same manner that cash payments are made offline. Several e-money products have been developed in re-

cent years, but none are still on the market in the U.S. today. DigiCash was a promising e-money technology that relied on patented cryptographic protocols developed by David Chaum which permitted the purchaser to spend e-money while remaining anonymous. Part of the value of Chaum's patented technology was that it would permit the identity of the purchaser to be uncovered in the event of fraud while keeping it hidden if the purchaser complied with the rules of the system. Individuals wishing to use DigiCash to make Internet purchases could send money to the DigiCash issuer and then download e-coins for safekeeping in a "wallet" on the hard drive of their personal computer. When the individual wished to make a purchase, the software would deduct an appropriate amount of e-coins from the wallet, and transfer it to the merchant. A similar product that did not rely on Chaum's blinded signature cryptographic protocols was CyberCoin developed by CyberCash. Although Chaum's basic system did not require its adoption by traditional banks to work, some banks did offer DigiCash. A bank could permit a merchant wishing to accept DigiCash to transfer the e-money to the bank to confirm that it had not already been spent before finalizing the transaction. Both DigiCash and CyberCash eventually filed for bankruptcy.

B. Micropayments

Micropayment technologies permit the payment of amounts as small as a fraction of a cent, which are too small to be processed economically using electronic payment systems available today such as debit or credit cards. In the mid 1990s, micropayment technologies seemed destined to be an essential building block for Internet commerce because many Internet content providers wanted to charge for their content but their potential customers were unwilling to pay subscription fees for access. Micropayment technologies such as CyberCoin and Millicent would have permitted consumers to download electronic money to a personal computer that could then be spent in small increments with participating merchants. These micropayment services never gained any substantial market share and are no longer offered, although it remains possible that future versions of this concept will enjoy greater success.

C. Stored Value Cards

Many observers believe stored value cards have promise to serve as more than subway fare cards or telephone cards. Mondex is an example of a more sophisticated form of stored value card that can take the place of cash by permitting transfers of value onto and off of the card. One of the primary advantages of the Mondex product from a consumer's point of view (and one of its primary disadvantages from a law enforcement point of view) is that it will permit person-to-person transfers of value from one card to another without clearing the transfer through a bank or otherwise creating an audit trail unless the transaction information is offloaded from the card. While this feature reduces the transaction costs of the system, it also increases the risks of forgery or money laundering. Because a smart card does not merely store data, but can perform processing functions, the electronic cash function of Mondex can be combined with other functions to enhance its appeal to consumers and merchants, such as tracking loyalty program credits. Although Mondex has been heavily promoted by its developers, it has not yet gained any substantial market share either in the U.S. or in Europe.

D. Network Security Protocols

One of the reasons credit cards are widely used in Internet commerce is that most browsers are capable of creating a secure communication channel between the Internet merchant's Web server and the consumer's browser using the Secure Sockets Layer (SSL) protocol developed by Netscape. SSL uses digital signature technology to provide assurance to individuals visiting Web sites on the Internet that the sites are genuine and are not a mere hacker masquerading as someone else. The SSL service also provides assurance that transfers of information between the local computer (or "client") and the server are confidential and are received intact. Web server applications that support electronic commerce come with software that manages the keys and the encryption processes in a way that is "transparent" to the visitor to the Web site. There is now an Internet Engineering Task Force (IETF) standard called Transport Layer Security (TLS) based on SSL.

One standard that was developed in the mid-1990s but failed to achieve any significant market share was the Secure Electronic Transaction (SET) protocol. This standard was established by Visa, Mastercard, IBM, and other industry associations and vendors to improve the security of credit card transactions over the Internet. SET was widely touted as superior to SSL in a variety of ways: it would prevent consumers from fraudulently disputing credit card

charges by establishing a stronger connection between a consumer's online identity and a particular charge and it would permit a merchant to be paid even though the merchant would never learn a consumer's credit card number. It is unclear whether SET could ever have lived up to these promises. It was never given a chance because implementation of SET would have required a huge investment by banks and merchants in an untested, and ultimately unsuccessful technology. As with micropayments, an updated version of SET or a similar concept may enjoy more success in the future, especially in light of the great need for stronger online authentication systems for Internet commerce.

E. Person to Person Internet Payment Services

Most electronic payment systems operate among banks and merchants and provide consumers with few access points to the system outside of those managed by banks and merchants. PayPal is one example of a new person-to-person electronic payment service that tries to solve this problem. PayPal and its competitors are not really so much new electronic payment systems as a new user-friendly interface for processing transactions executed using old payment systems such as ACH payments, credit card payments, and even payments by check. PayPal assists individuals such as sellers and bidders on Internet auction sites such as eBay and handles online payments quickly and easily. PayPal relied on "viral" marketing to spread the word about its service: anyone referring a new customer to PayPal was credited $10, later reduced to $5, to his or her PayPal account. In its first 6 months of operations, it had signed up over one million customers. PayPal's popularity began to wane in 2000 after it was forced to tighten up its administration of accounts following attacks on its system by organized crime, and traditional banks began to offer very similar competing products, such as Citibank's c2it service.

F. Internet Scrip and Loyalty Programs

Some services offered over the Internet are nonbank payment systems. For example, Beenz provided a kind of loyalty program similar to airline frequent-flyer miles programs. Subscribers who earned enough Beenz by making online purchases or visiting sites could spend their Beenz with participating merchants. Flooz was an "Internet scrip" service. It offered gift certificate currency that could be sent by e-mail and spent with participating merchants. Products such as Beenz and Flooz did not circulate generally enough to be considered equivalent to money or near-money systems; however, they might have been subject to regulation under state money services businesses laws especially if they accepted deposits from the public. In 2001, both Beenz and Flooz ceased operations and filed for bankruptcy; it was unclear if either program would be revived.

G. Electronic Letters of Credit

Letters of credit are commonly used to make payments associated with cross-border trade in goods. An importer will open a letter of credit with a local bank that will then confirm that the importer's credit is good with the exporter's bank in a foreign country. When the exporter places the goods destined for the importer on a ship, it will normally receive a bill of lading in return. The exporter turns the bill of lading over to its local bank in return for payment, which then sends the bill of lading to the importer's bank in return for payment, which then turns the bill of lading over to the importer in return for payment. The importer then uses the bill of lading to obtain possession of the goods from the shipper. While all the payments made under a bill of lading have generally been in electronic form since the days of the tested telex, bills of lading are still required by warehousemen and some national laws to be on paper. Various attempts are now underway to replace the paper bill of lading with an electronic record and integrate the electronic payment process with the remittance of an electronic bill of lading. One such attempt is the Bill of Lading Electronic Registry Organization (BOLERO or Bolero.net), which was organized by SWIFT and TT Club (Through Transport Club, a mutual insurance association specializing in transportation risks) in 1999. It remains unclear whether Bolero or any of its competitors such as LC Connect or Tradecard will gain any significant market share.

H. e-Checks

Given that checks are the such a popular payment device in the U.S. today, it might seem reasonable to expect that an electronic version of checks might be popular, too. In fact, legal obstacles to the recognition of e-checks have made it difficult to conduct experiments in this area, and the current technology for

processing checks has proven remarkably resistant to adaptation to either partially or entirely electronic alternatives to paper checks. For example, one "electronic check" product converts a bounced check into an ACH debit. This product is popular with retail merchants because it lowers the cost of processing bounced checks and improves the odds that the check will be paid when represented. Check "truncation" systems stop paper checks at some point in the collection process and replace them with electronic messages sent though the ACH system. One payment product called an electronic check for marketing purposes uses a paper check to originate an ACH debit from a consumers account. Therefore it is not really a check transaction at all, but an electronic funds transfer transaction, even though the distinction may not be obvious to the consumer. A wholly electronic check developed using advanced encryption technologies was successful in pilot, but never achieved any subsequent adoptions after that pilot.

I. Internet ACH Transactions

The ACH has the potential to be the foundation for a whole new generation of electronic payment applications for Internet commerce, but only if some very fundamental issues can be resolved. For business-to-business payments, applications will have to be developed that take advantage of existing standards such as FEDI, or newer standards aimed at resolving similar problems based on newer electronic commerce technologies developed specifically for the Internet such as XML (*e*xtensible *m*arkup *l*anguage). Business processes generally will need to be more standardized than they are today, particularly among small and medium sized businesses, before electronic payment systems can make much progress in replacing checks as the preferred medium of payment among U.S. businesses.

The primary obstacles to greater use of ACH payments in business-to-consumer Internet commerce are the difficulty in authorizing and initiating spontaneous, one-time debits from consumer accounts, and the natural preference of consumers for payment systems that give them more generous payment terms. Before banks can accept an ACH debit from a merchant seeking payment from a consumer account, the merchant must obtain the consumer's prior authorization in writing for the debit. Various systems are now in development that will make it easier for the merchant to obtain the electronic equivalent of a signed written authorization from a consumer to electronically debit the consumer's account. Prices consumers pay today in online transactions reflect the high costs of accepting payment by credit card, but consumers are generally not aware of the magnitude of the costs passed onto merchants as a group as a result of the high fraud and dispute resolution problems associated with credit card use. Merchants may be able to offer price incentives to consumers to persuade them to switch to online electronic payment systems with lower overheads than the credit card system.

VI. PROSPECTS FOR FUTURE DEVELOPMENTS

It is too soon to say whether the lumbering behemoths of today's electronic payment systems will someday be supplanted by more agile, technologically advanced e-money systems. However, it seems clear that migration away from the very mature systems operating today towards the more advanced systems likely to be deployed in the future will continue to be a cautious, incremental process of evolution for some time. This is because current users of electronic payment services are locked-in to those systems and have insufficient incentives to pay what it would cost to move to more advanced systems.

A network effect exists if individual members participating in a system value that system more whenever there are more participants in the system. Because the value of a payment system increases for individual payers when there are more people willing to accept it, the market for payment systems clearly shows strong network effects. If the costs of switching from one network to another is very expensive, then participants in that network will suffer from lock-in to the network they are currently using. The experience of recent years indicates that electronic payment systems are networks characterized by very high switching costs for financial institutions, merchants, and even the consumers using them, so users of electronic payment systems today suffer from a high degree of lock-in to existing technologies and products. In markets with strong network effects and large problems with lock-in, changes from one technology to another may be characterized by a sudden "tipping" of the market from one network standard to another, more technologically advanced. Proponents of e-money products clearly expected this to happen in the market for electronic payment services, but it did not. While it remains possible that at some point in the future there will be a sudden switch from current electronic payment systems to one more technologically advanced, the path to technological innovation in electronic payments has been characterized by slow,

incremental progress, not sudden, massive changes. Because of the degree to which electronic payment systems are already woven into a complex fabric of consumer habits and preferences and existing business management and information systems, the pattern of slow evolution seems likely to continue with regard to most forms of electronic payment systems for some time.

SEE ALSO THE FOLLOWING ARTICLES

Business-to-Business Electronic Commerce • Crime, Use of Computers in • Electronic Commerce • Electronic Commerce, Infrastructure for • Electronic Data Interchange • Sales

BIBLIOGRAPHY

Bank for International Settlements. Available at www.bis.org.

Banks, E. (2001). *e-Finance: The electronic revolution*. New York: Wiley

Clearing House Interbank Payments System. Available at www.chips.org.

European Central Bank. Available at www.ecb.int.

Fieler, K. (1999). *Electronic money*. Chicago: Federal Reserve Bank of Chicago. [Revised by Tim Schilling]

Johnson, E. G. O. (1998). *Payment systems, monetary policy, and the role of the central bank*. Washington, DC: International Monetary Fund.

Markose, S. M., and Loke, Y. J. (2001). Changing trends in payment systems for selected G10 and EU countries 1990–1999, in *International Correspondent Banking Review Yearbook 2000/2001*. London: Euromoney.

Mauss, M. (1990). *The gift: The form and reason for exchange in archaic societies* (W. D. Hall, trans.). New York: Norton. [original French edition, 1925]

National Automated Clearing House Association. Available at www.nacha.org.

O'Mahony, D., Peirce, M., and Tewari, H. (2001). *Electronic payment systems for E-commerce,* 2nd ed. Boston, MA: Artech House.

Society for Worldwide Interbank Financial Telecommunication. Available at www.swift.com.

Turner, P. S. (1999). *Law of payment systems and EFT*. Gaithersburg, MD: Aspen Law & Business.

U.S. Federal Reserve Board. Available at www.federalreserve.gov.

Vartanian, T. P., Ledig, R. H., and Bruneau, L. (1998). *21st century money, banking & commerce*. Washington, DC: Fried, Frank, Harris, Shriver & Jacobson.

Winn, J. K. (1999). Clash of the Titans: Regulating the competition between established and emerging electronic payment systems. *14 Berkeley Technology Law Journal*, Vol. 675.

Encryption

Jeff Gilchrist
Elytra Enterprises, Inc., Yanier, Ontario, Canada

GLOSSARY

cipher An encryption algorithm.
ciphertext A message after it has been encrypted.
cryptography The science of keeping information secure.
factor Any number that divides a given integer.
factoring Reducing an integer into its prime factors.
hash The output of a hash function.
hash function Takes an arbitrary message of arbitrary length and creates an output of a fixed length.
MAC Message authentication code—the output of a keyed hash function.
NIST National Institute of Standards and Technology in the United States.
plaintext A message before it has been encrypted.
polynomial An expression of two or more terms.
prime factor A prime number that is a factor of another number.
prime number An integer >1 that can only be divided by 1 and itself.
relatively prime Two integers that do not have any common factors.
s-box Substitution box.

ENCRYPTION is the process of scrambling a message in such as way that its original content cannot be seen. The original message is called plaintext and the encrypted message is ciphertext. Decryption is the process of converting ciphertext back into the original plaintext. Cryptography is the science of keeping information secure using techniques to provide confidentiality, authentication, data integrity, and nonrepudiation. Confidentiality is the ability to keep the message secret from everyone but those authorized to see it. Authentication is the ability to determine the origin of the message. Data integrity is the ability to detect any changes made to the data, and nonrepudiation is the ability to prevent someone from denying a previous action.

I. SYMMETRIC ENCRYPTION

Symmetric encryption uses a single or very similar key for both encryption and decryption. The security of symmetric cryptography rests in the key so anyone who is able to obtain the key can decrypt and encrypt messages. These keys are measured in bits, the recommended key size today being 128 bits. The key size indicates how many possible keys are available and thus how difficult it would be to perform a brute-force attack (trying all possible keys). Having a 20-bit key means there are 2^{20} or 1,048,576 possible keys. The larger the key size, the larger the number of possible keys and the harder it will be to conduct a brute-force attack. For two parties to communicate securely, they need to agree on a symmetric key beforehand. The two types of symmetric encryption are stream ciphers and block ciphers.

A. Stream Ciphers

1. Basic Principles

Stream ciphers are a type of encryption algorithm that process an individual bit, byte, or character of

plaintext at a time. Stream ciphers are often faster than block ciphers in hardware and require circuitry that is less complex. They are also useful when transmission errors are likely to occur because they have little or no error propagation.

Stream ciphers can be classified into synchronous, self-synchronizing, and the one-time pad. Synchronous ciphers have an independently generated keystream from the plaintext and ciphertext. They need to be in the same state using the same key in order to decrypt the data properly. If a ciphertext character is modified, it does not affect the decryption of the rest of the ciphertext, but if a character is deleted or inserted, synchronization will be lost and the rest of the decryption will fail. Self-synchronizing (asynchronous) ciphers have a keystream that is generated from the key and a specified number of previous ciphertext characters. This type of stream cipher can better handle characters being deleted or inserted as the state only depends on the specified number of previous ciphertext characters. After that number has been processed, the cipher will be synchronized again. A synchronous stream cipher has no error propagation, but a self-synchronizing cipher has limited error propagation. If a character is modified, the maximum number of incorrect decryptions would be limited to the specified number of previous ciphertext characters after which correct decryption would resume.

2. One-Time Pad

The one-time pad is the most secure encryption algorithm available if used properly. It is a Vernam cipher, the plaintext characters are XORed (bitwise add, modulo 2) with the keystream characters to form the ciphertext where the keystream characters are generated randomly and independently. The keystream characters must be truly random and not generated from a pseudo-random number generator. The same number of keystream characters must be generated as there are plaintext characters. This set of keystream characters becomes the one-time pad. A duplicate pad must be given to the recipient in order to decrypt the message. The pad may never be used again to encrypt another message and must be destroyed. Because the keystream was generated randomly, every key sequence is equally likely and thus unconditionally secure if the attacker has only the ciphertext.

For example, the ASCII plaintext message "Canada to invade USA" would require 20 keystream characters to be generated. Using a truly random source such as measuring the thermal noise of a device, the following 20 ASCII characters were generated: FOquLwXrmnHDoFmVCKtg. The encryption process can be seen in Table I. To decrypt the message, the receiver takes the ciphertext and XORs each character with the corresponding keystream character in the one-time pad to reveal the plaintext.

Although the one-time pad is the most secure encryption algorithm, it does have problems. The one-time pad key must be delivered to the recipient in a secure manner and the pads can never be reused. The keystream characters in the pad must be truly random (another source is measuring radioactive decay).

3. RC4

RC4 is a stream cipher that supports variable key sizes and was developed in 1987 by Dr. Ron Rivest for RSA Data Security Inc. One of RC4's most popular uses is in the secure socket layer (SSL) protocol used in Web browsers to encrypt the web session between a browser and a Web server.

The keystream of RC4 is independent of the plaintext. The design has an 8×8 substitution box (S-box), so 256 entries. These 256 entries are permutations and the permutation is a function of the key. The S-box entries are labeled $S_0, S_1, \ldots, S_{255}$ and the algorithm contains two counters, m and n. The S-box must first be initialized before encryption can start. This is done by filling it linearly so $S_0 = 0$, $S_1 = 1$, $\ldots$, $S_{255} = 255$. A second 256-byte array must be created and filled with the encryption key, repeated if necessary to fill the entire array so $K_0, K_1, \ldots, K_{255}$. Finally the key material is used to randomize the S-boxes. The following pseudo-code describes the initialization:

For $m = 0$ to 255
    $S[m] = m$
$n = 0$
For $m = 0$ to 255
    $n = (n + S[m] + K[m]) \bmod 256$
    exchange $S[m]$ and $S[n]$

Now that the S-box has been initialized, the encryption process can begin. Using the RC4 algorithm, a random byte is generated from the S-boxes. This byte, R, is then XORed with a byte of plaintext to form ciphertext. The process is repeated until all of the plaintext has been encrypted. The decryption process is the same except R is XORed with a byte of ciphertext to reveal the plaintext. The following pseudo-code describes the encryption process:

Table I One-Time Pad Encryption Example[2]

Plaintext			Keystream			Ciphertext	
ASCII	Hex	XOR	ASCII	Hex	=	ASCII	Hex
C	0×43	⊕	F	0×46	=		0×05
a	0×61	⊕	O	0×4F	=	.	0×2E
n	0×6E	⊕	q	0×71	=		0×1F
a	0×61	⊕	u	0×75	=		0×14
d	0×64	⊕	L	0×4C	=	(	0×28
a	0×61	⊕	w	0×77	=		0×16
	0×20	⊕	X	0×58	=	x	0×78
t	0×74	⊕	r	0×72	=		0×06
o	0×6F	⊕	m	0×6D	=		0×02
	0×20	⊕	n	0×6E	=	N	0×4E
i	0×69	⊕	H	0×48	=	!	0×21
n	0×6E	⊕	D	0×44	=	*	0×2A
v	0×76	⊕	o	0×6F	=		0×19
a	0×61	⊕	F	0×46	=	‘	0×27
d	0×64	⊕	m	0×6D	=		0×09
e	0×65	⊕	V	0×56	=	3	0×33
	0×20	⊕	C	0×43	=	c	0×63
U	0×55	⊕	K	0×4B	=		0×1E
S	0×53	⊕	t	0×74	=	‘	0×27
A	0×41	⊕	g	0×67	=	&	0×26

$m = (m + 1) \bmod 256$

$n = (n + S[m]) \bmod 256$

exchange $S[m]$ and $S[n]$

$x = (S[m] + S[n]) \bmod 256$

$R = S[x]$

Ciphertext = R XOR Plaintext

The RC4 algorithm is simple enough to memorize and easy to implement.

B. Block Ciphers

1. Basic Principles

Block ciphers are a type of encryption algorithm that process one block of plaintext at a time. Plaintext blocks of length m are generally mapped to ciphertext blocks of length m. The value m is referred to as the block size and is usually measured in bits. While stream ciphers usually process a bit or a byte of data at a time, block ciphers generally process at least 64 bits at a time. If the plaintext is larger than the block size of the encryption algorithm, multiple blocks of plaintext are encrypted into multiple blocks of ciphertext. This can be done using different modes of operation, two common ones being ECB and CBC.

In electronic codebook (ECB) mode, the plaintext is divided into blocks of size specified by the algorithm. Each block is then encrypted into a ciphertext block. Using the same encryption key, identical plaintext blocks always encrypt into the same ciphertext block so data pattern analysis can be performed. For this reason, messages longer than one block are not recommended to be sent in ECB mode. Errors in a ciphertext block only affect the decryption of that block.

In cipher block chaining (CBC) mode, the plaintext is divided into blocks of size specified by the algorithm. An initialization vector (IV) the size of the block is also generated and this need not be secret. The first plaintext block is XORed with the IV before it is encrypted. The second and subsequent plaintext blocks are XORed with the ciphertext block that was created from the previous plaintext block and then

encrypted. This removes the problem in ECB mode where every identical plaintext block always encrypts to the same ciphertext block. If an error occurs in one ciphertext block, it will affect the decryption of that block and the following one. The CBC process is shown in Fig. 1.

Two important principles of block ciphers are confusion and diffusion. Confusion tries to conceal any link between the key, plaintext, and ciphertext. A simple way to accomplish this is by using substitution. Diffusion hides statistical relationships by spreading out any redundancy in the plaintext over the ciphertext. This can be done by using permutations.

2. DES (Data Encryption Standard)

The DES algorithm is the most widely known block cipher in the world. It was created by IBM and defined in 1977 as U.S. standard FIPS 46. It is a 64-bit block cipher with 56 bit keys and 16 rounds. A round in DES is a substitution (confusion), followed by a permutation (diffusion).

For each 64-bit block of plaintext that DES processes, an initial permutation is performed and the block is broken into two 32-bit halves, a left half (L_i) and a right half (R_i). The 16 rounds of substitutions and permutations, called function *f*, are then performed. For each round, a DES round key (K_i) of 48 bits and the current R_i are input into function *f*. The output of *f* is then XORed with the current L_i to give R_{i+1}. The current R_i becomes L_{i+1}. After the 16 rounds, the two halves are rejoined and a final permutation is the last step. This process is shown in Fig. 2.

The DES algorithm, even today, is resistant to most practical attacks. The use of DES, however, is no longer recommended because of the small key size. With 56 bit keys, DES is susceptible to brute-force attacks, where every possible DES key is tried to find the cor-

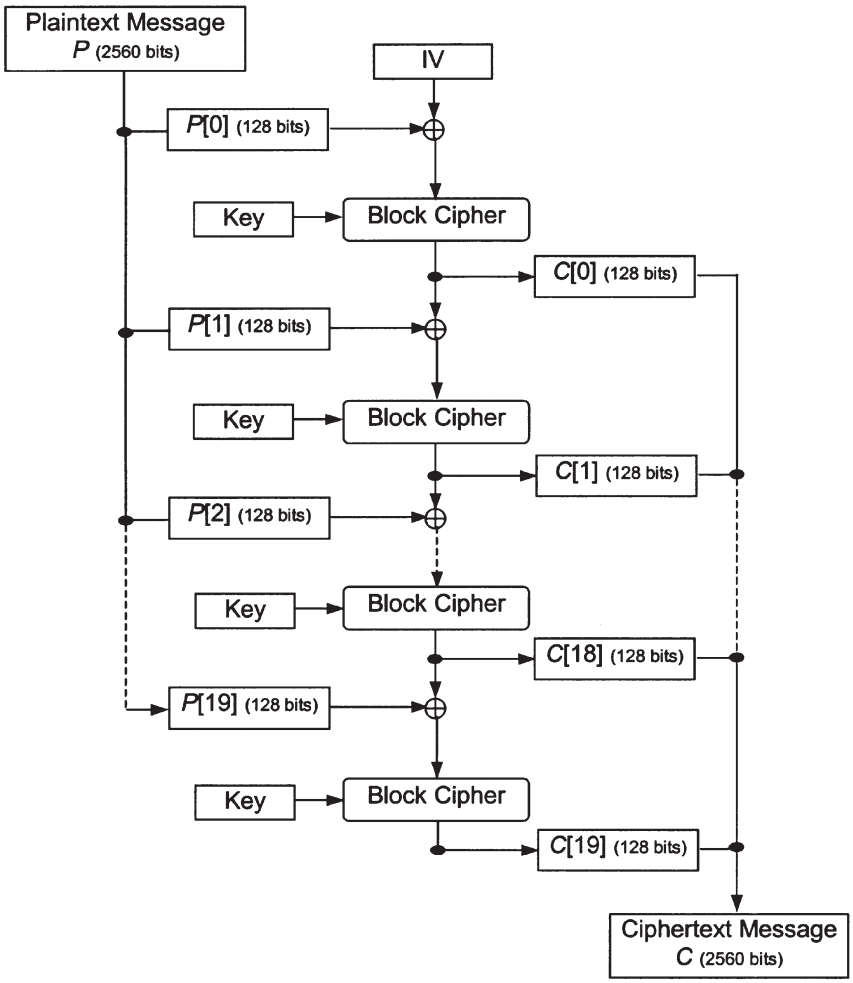


Figure 1 CBC mode of a block cipher with 128-bit block size.

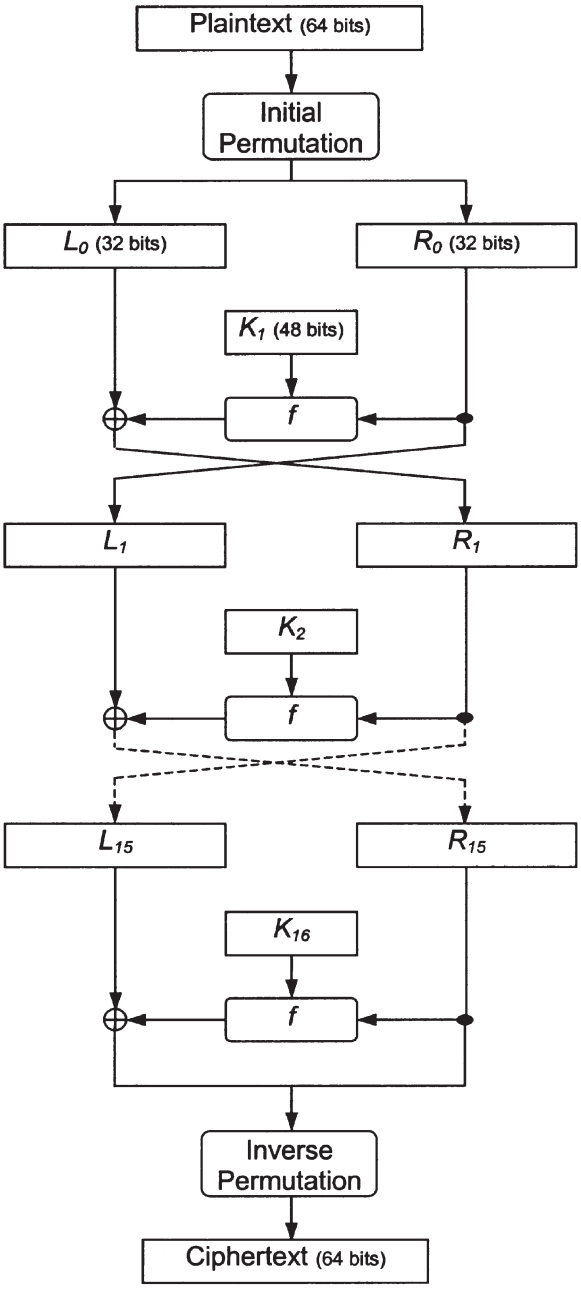


Figure 2 The DES algorithm.

rect one to decrypt the ciphertext. With a brute-force attack, only 50% of the keys (on average) need to be tried before the correct one is found. In January 1999, a DES cracking contest was held by RSA Data Security Inc. After only 22 hours and 15 minutes, the DES encrypted message was cracked by an international team consisting of a custom hardware DES cracking device built by the Electronic Frontier Foundation (EFF) and a group of distributed computing enthusiasts known as distributed.net. The EFF hardware cost $250,000 U.S. to design and build. Creating a second device would be much cheaper. Distributed.net used the idle CPU time of the members' computers at no extra cost. With more money, a DES cracking hardware device could be built to decrypt messages within a small number of hours or even minutes.

3. AES (Advanced Encryption Standard)

In January 1997, the U.S. National Institute of Standards and Technology (NIST) announced the AES initiative to find a replacement for the aging DES algorithm. The AES effort would specify a publicly disclosed encryption algorithm that was available worldwide on a royalty-free basis. The minimum requirements for AES were a symmetric key block cipher with a 128-bit block size and 128-, 192-, and 256-bit key sizes. A series of conferences was held to evaluate the 15 submitted candidates and by August 1999, five AES finalists were chosen: MARS, RC6, Rijndael, Serpent, and Twofish. NIST announced in October 2000 that Rijndael was chosen to be the AES. Rijndael (pronounced "Rhine-doll") was designed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen. The official AES specification was expected to be complete by summer of 2001.

The AES algorithm (Rijndael) has a variable block length and variable key length. The current specification has the AES block size set to 128 bits, with three possible key sizes of 128, 192, and 256 bits. With a 128-bit block size, the AES algorithm consists of an initial round key addition, $R - 1$ rounds, and a final round. The number of rounds is a function of block and key size. With a 128-bit block size, a 128-bit key has 10 rounds, a 192-bit key has 12 rounds, and a 256-bit key has 14 rounds. The round transformation for the first $R - 1$ rounds has four steps: ByteSub, ShiftRow, MixColumn, and AddRoundKey. The final round is slightly different with the MixColumn step removed. The ByteSub step is a nonlinear (behaving in an unpredictable fashion) byte substitution that acts independently on the State (the input into the function) bytes. The ShiftRow step has the rows of the State shifted over various offsets. The MixColumn step has the columns of the State multiplied modulo $x^4 + 1$ with a fixed polynomial. Finally, the AddRoundKey step has the round key XORed with the State. The round key (K) is derived from a cipher key expansion. The total round key bits required is the number of rounds plus one multiplied by the block size. In the case of AES with a 128-bit block, $10 + 1$ rounds multiplied by 128 bits equals 1408 bits

required. The first round uses the first 128 bits of the round key bits, and each round that follows uses subsequent 128-bit blocks of the round key bits. The AES process for a 128-bit block and 128-bit key is shown in Fig. 3.

Because the minimum key size specified by AES is 128 bits, the cipher should not succumb to brute-force attacks any time soon. With 128 bits, 3.4×10^{38} different keys are possible, which is 4.72×10^{21} times more keys than with DES. Using enough computing power to try all possible 56-bit keys in 1 sec, it would take 150 trillion (1.5×10^{14}) years to try all possible 128-bit keys, 2.7×10^{33} years to try all possible 192-bit keys, and 5.1×10^{52} years to try all possible 256-bit keys.

4. CAST

CAST is a design procedure for block ciphers developed in 1993 by Carlisle Adams and Stafford Tavares. It was later patented by Entrust Technologies (U.S. Patent No. 5,511,123) but made available worldwide on a royalty-free basis. A specific implementation of CAST known as CAST-128 was developed by Carlisle Adams in 1997 and is described in RFC 2144.

CAST-128 is a 64-bit block cipher with key sizes ranging from 40 to 128 bits in 8-bit increments. For key sizes up to 80 bits, 12 rounds are performed; for keys greater than 80 bits, 16 rounds are performed. Keys less than 128 bits in size are padded with zeros to form a 128-bit key. It has 8 S-boxes, 4 of which are

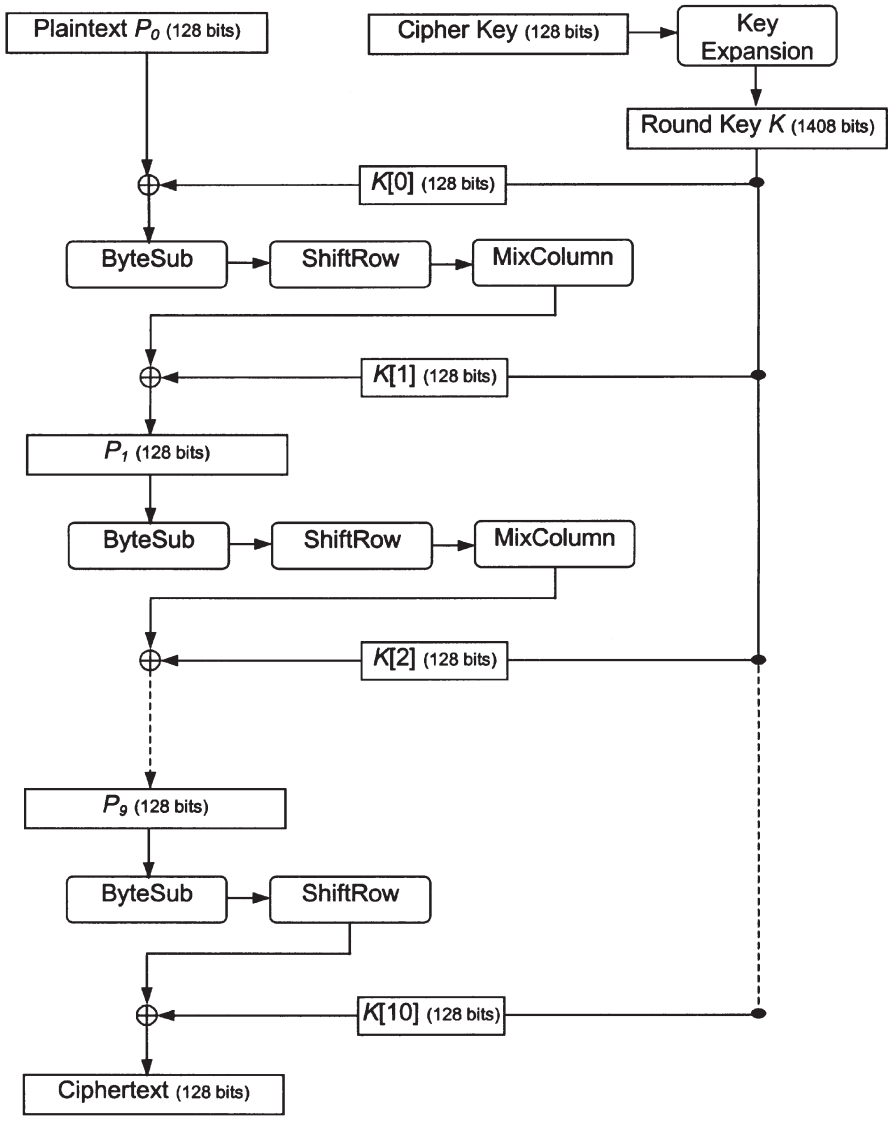


Figure 3 Rijndael (proposed AES) with 128-bit block and 128-bit key.

round function S-boxes and 4 of which are key schedule S-boxes. Using the key schedule, 16 pairs of subkeys are computed: Km_i and Kr_i. The 64-bit plaintext block is broken into two 32-bit halves: L_i and R_i. To protect against cryptographic attacks, CAST-128 uses rotations, XOR, addition, and subtractions in the round function f. The subkeys and 32-bit halves are input into the f function. There are also three variations of the round function: f_1, f_2, f_3. Round function f_1 is used for rounds 1, 4, 7, 10, 13, and 16. Round function f_2 is used for rounds 2, 5, 8, 11, and 14. Round function f_3 is used for rounds 3, 6, 9, 12, and 15. After the 16th round, the two 32-bit halves are concatenated to form the 64-bit ciphertext block. This process is shown in Fig. 4.

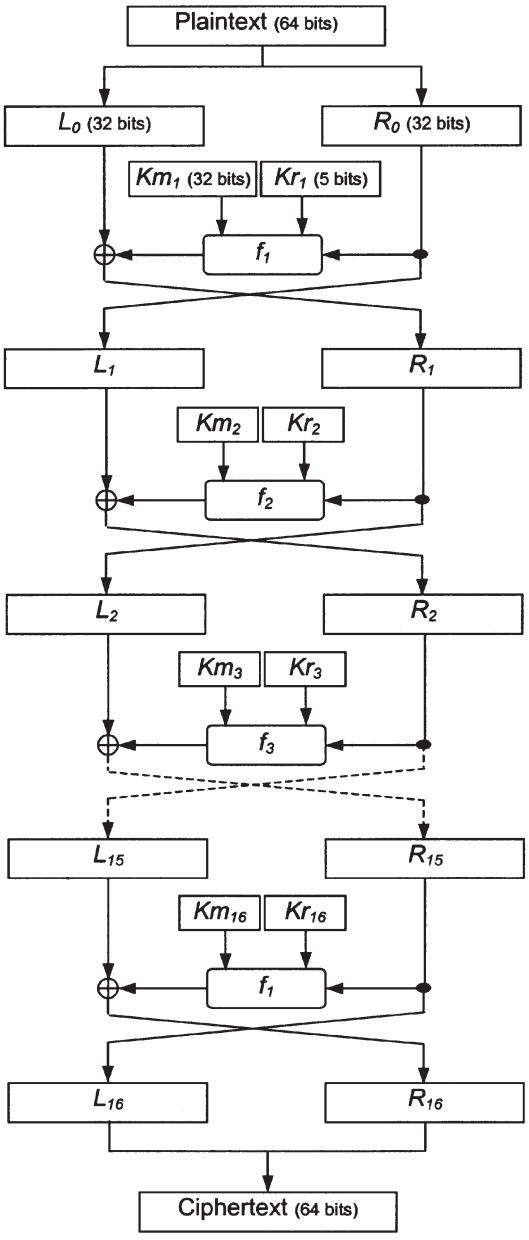


Figure 4 CAST-128 with 64-bit block and 128-bit key.

Another implementation of CAST known as CAST-256 was developed by Carlisle Adams in 1998 (RFC 2612). CAST-256 is a 128-bit block cipher with variable key sizes of 128, 160, 192, 224, and 256 bits. It is built on the CAST-128 algorithm and was a candidate for the AES competition. CAST-256 has 12 quadrounds (48 partial rounds) for all key sizes and improves on CAST-128 by using the larger block size and key sizes with little sacrifice in speed. CAST-256 is also available worldwide on a royalty-free basis.

CAST-128 and CAST-256 have been implemented in many applications. There are no attacks currently known for these two ciphers.

II. PUBLIC KEY ENCRYPTION

A. Basic Principles

Public key cryptography has made some aspects of the protection of communications much easier. With symmetric key encryption, both the sender and the receiver needed the same key to decrypt the message so there is the problem of distributing the keys in a secure manner. With public key encryption, a key pair is created: a public key and a private key. The public key can be given out to the people you need to communicate with or it can even be placed in a public database. The private key is kept secret and known only to yourself. To communicate securely, the public key of the receiver is retrieved. The message is then encrypted using the receiver's public key. The encrypted message is then sent to the receiver, and they use their private key to decrypt the message. Keys no longer need to be distributed in a confidential way.

Because an attacker can obtain the public keys of people communicating, public key cryptography can always be analyzed with respect to chosen plaintext attacks. That is, an attacker can pick whatever plaintext they want, and encrypt it using the intercepted public key to perform analysis on the resulting ciphertext. This information is then used to try to uncover plaintext from real intercepted ciphertext. Another problem with public key cryptography is that the sender must be sure he is encrypting the message with an authentic public key from the receiver. If an attacker can replace the receiver's public key in a public database (or other means) with her own and the sender does not verify that he is in fact using the correct public key, the sender will mistakenly encrypt the message

with the attacker's public key and the attacker will be able to decrypt the message.

B. RSA

RSA is a public key algorithm that was created by Ron Rivest, Adi Shamir, and Leonard Adleman. The U.S. patent for RSA (No. 4,405,829), granted in September 1983, expired in September 2000, so the algorithm is now freely available to use.

To create an RSA key pair, two large random prime numbers p and q of similar size must be generated. Multiply p and q to form n ($n = pq$). An encryption key e must be chosen (e is often chosen to be 3 or 65537) such that it is relatively prime to $(p - 1)(q - 1)$. A procedure known as the extended Euclidean algorithm is then used to compute the private key d, such that $d = e^{-1} \bmod ((p - 1)(q - 1))$. The values n and e are the RSA public key, and the value d is the RSA private key.

To encrypt data using the RSA algorithm, the following formula is applied, where m is the plaintext message and c is the resulting ciphertext: $c = m^e \bmod n$. To decrypt data, the following formula is applied: $m = c^d \bmod n$.

The security of RSA relies on the fact that currently there is no efficient algorithm known to factor large numbers. An attacker trying to determine an RSA private key would try to factor out the two primes p and q from the value n in the public key. With this information, the private key could be found. If an efficient algorithm for factoring is discovered, RSA private keys would then be easily found. In August 1999, a group of 14 researchers successfully factored a 512-bit RSA key after 4 months of computing using a factoring algorithm known as the Number Field Sieve (NFS). At the time, 512-bit RSA was widely in use on the Internet. It is now recommended that a minimum of 1024-bit RSA keys be used. It is also interesting to note that the RSA patent number (No. 4,405,829) itself is prime.

C. Diffie–Hellman

Whitfield Diffie and Martin Hellman created the first public key algorithm in 1976. The Diffie–Hellman algorithm is used for key exchange and it is still in use today. The security of the algorithm comes from the complexity of computing discrete logarithms.

If two people wish to communicate securely over an insecure line, they need to agree on a symmetric key with which to encrypt their communications. First a large prime number p and generator g must be produced. The value g must be primitive mod p, that is, it must be able to generate all elements in the field. Person A and B both require these values; they do not need to be secret so p and g may be sent in the clear. Person A then generates a random integer x and sends person B the result $M_a = g^x \bmod p$. Person B generates a random integer y and sends person A the result $M_b = g^y \bmod p$. Person B receives M_a and computes the symmetric key $K = M_a^y \bmod p = g^{xy} \bmod p$ and person A receives M_b and computes the same symmetric key $K = M_b^x \bmod p = g^{xy} \bmod p$. K can then be used as the key in a symmetric encryption algorithm to encrypt communications between person A and person B. The Diffie–Hellman algorithm is illustrated in Fig. 5.

With the Diffie–Hellman algorithm, a passive attacker (someone who can only read all communications between persons A and B) cannot determine the secret key K. The algorithm does not provide any authentication of either party so an active attacker who can intercept, inject, or modify messages could perform a man-in-the-middle attack and read all encrypted communications. In this case, the attacker would see p and g. When person A sends her $M_a = g^x \bmod p$ to person B, the attacker would intercept the message, calculate his own u value and send back $M_c = g^u \bmod p$ to person A. The attacker would also create his own v value and send $M_d = g^v \bmod p$ to person B, then intercept the message $M_b = g^y \bmod p$ from person B. The attacker has now performed two Diffie–Hellman key agreements, one with person A and one with person B. The attacker and person A

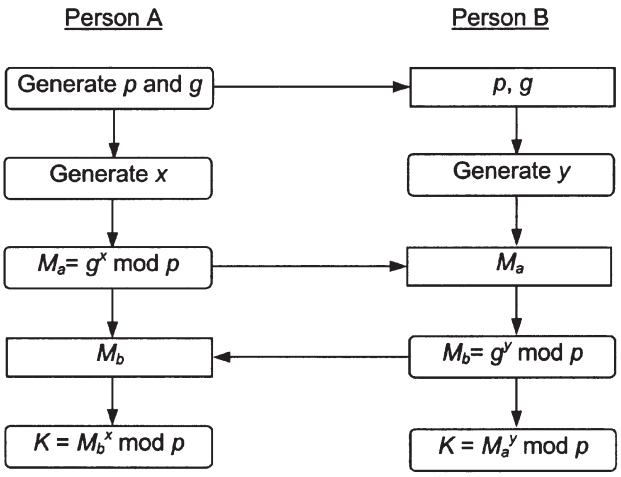


Figure 5 Diffie–Hellman key agreement.

would calculate the shared symmetric key and start communicating in encrypted form. The same would happen with the attacker and person B. The attacker would decrypt the message from person A, read the information, re-encrypt it for person B, and send the new message to person B and vice versa. Neither person A nor B has any idea that the attacker is reading the communication. However, methods exist such as the station-to-station (STS) protocol to augment the Diffie–Hellman algorithm to provide authentication of the parties involved.

D. Elliptic Curves

Elliptic curve cryptography (ECC) is a fairly new class of public key algorithms that was proposed by Neil Koblitz and Victor Miller in 1985. It can be implemented using smaller keys than RSA and Diffie–Hellman while keeping similar strengths. In some cases, performance can also be superior.

The security of RSA comes from the difficulty of factoring large integers, and the security of Diffie–Hellman comes from the difficulty of computing discrete logarithms. ECC is based on a mathematical structure called an elliptic curve. Similar to Diffie-Hellman, the strength of ECC comes from discrete logarithms but this time computed over an elliptic curve group. There are two types of elliptic curves that are used in ECC, and they are incompatible with each other. One is of odd characteristic F_p (or modulo p) and is more suitable for software implementation. The other is of even characteristic $F_2{}^m$ (or over a field having 2^m elements) and is more suitable for hardware implementation.

ECC algorithms are based on already known public key algorithms but the underlying math is performed over the points on an elliptic curve instead of over a finite field of integers. For digital signatures, the DSA (Digital Signature Algorithm) is modified and becomes ECDSA (Elliptic Curve Digital Signature Algorithm). For key exchange, the Diffie–Hellman algorithm is modified and becomes ECDH (Elliptic Curve Diffie–Hellman). An ECC public key of 160 bits is generated by selecting a point on the elliptic curve that consists of two elements (x, y), each one being 160 bits. This means that to store an ECC public key, 320 bits is actually required (it is possible to compress this key to 161 bits using a patented method). An RSA public key of 1024 bits consists of the product of two primes (which is 1024 bits) and the public exponent (which is often set to 3, so 2 bits) for a total of 1026 bits. Signature sizes are similar, the ECC signature requires two 160-bit values giving 320 bits total, and the RSA signature consists of one 1024-bit value.

While factoring integers and computing discrete logarithms seem to be the same level of difficulty, computing elliptic curve discrete logarithms appears to be much more difficult. The best known method for solving the first two problems is the NFS and runs in subexponential time. The best known method for solving the ECDL problem is the parallel collision search algorithm, which runs in fully exponential time. This is why a smaller ECC key can be used for public key cryptography than with RSA or Diffie–Hellman. In April 2000, a group of more than 1300 people working together in a distributed computing effort used the parallel collision search method to crack a 109-bit ECC public key after 4 months of computing. This effort used about 2.5 times more computing power than the 56-bit DES crack, and about 50 times more computing power than the 512-bit RSA crack. It is recommended that a minimum of 160-bit ECC keys be used.

III. HASH FUNCTIONS

A. Basic Principles

A hash function, otherwise known as a one-way hash function, takes an arbitrary message of arbitrary length and creates an output (a hash) of a fixed length. The main characteristics of a cryptographic hash function are that given a message, it is easy to compute the hash; given the hash, it is difficult to compute the message; and that given a message, it is difficult to find a different message that would produce the same hash (this is known as a collision).

Hash functions are used for data integrity and often in combination with digital signatures. With a good hash function, even a 1-bit change in a message will produce a different hash (on average, half of the bits change). With digital signatures, a message is hashed and then the hash itself is signed. The signature will show if the hash value has been tampered with and the hash will show if the message has been modified. The two types of hash functions are unkeyed (MD5, SHA-1) and keyed (MAC).

B. MD5

The MD5 hashing algorithm (RFC 1321) was designed in 1992 by Ron Rivest as an improved version of MD4. It is an unkeyed hash with an output of 128 bits. The

message to be hashed is processed by MD5 in 512-bit blocks. The message is first padded so that its length is a multiple of 512 bits. Four 32-bit chaining variables are initialized to (hex): cv_1 = 0x01234567, cv_2 = 0x89abcdef, cv_3 = 0xfedcba98, and cv_4 = 0x76543210. For each message block, four rounds of the main loop are performed for a total of 64 operations (16 operations per round). The message block of 512 bits is further divided into 16 sub-blocks of 32 bits each. The chaining variables (cv_1, cv_2, cv_3, cv_4) are copied into round variables (rv_1, rv_2, rv_3, rv_4), respectively. For every operation, there is a nonlinear function on three of the four round variables. The result (R_1) is added to a constant, the remaining round variable, and a 32-bit sub-block of the message to give R_2. This new result (R_2) is rotated to the right a variable number of bits and added to one of the round variables. R_2 also replaces one of the round variables. The round variables (rv_1, rv_2, rv_3, rv_4) are then added to the chaining variables (cv_1, cv_2, cv_3, cv_4), respectively. The main loop is repeated until all message blocks have been processed after which the chaining variables are concatenated to give the 128-bit MD5 hash.

The hash of the ASCII text "MD5" using the MD5 algorithm is:

0x7f138a09169b250e9dcb378140907378

Changing the last bit in "MD5" from a 1 to a 0 results in the ASCII text "MD4." Even a 1-bit change creates a totally different hash. Using MD5 on the text "MD4" results in the hash:

0x59b6d1f8ea235402832256aa62415fe0

Although no collisions have been found in MD5 itself, collisions have been found by den Boer and Bosselaers in the MD5 compression function. Therefore, it is generally recommended that a different hashing algorithm, such as SHA-1, be used.

C. SHA-1

The Secure Hash Algorithm (SHA) was developed in 1992 by NIST and is based on the MD4 algorithm. A flaw was found in SHA, and 2 years later a revision (SHA-1) was published as U.S. standard FIPS 180-1. Unlike MD4 and MD5, which have an output of 128 bits, SHA-1 has an output of 160 bits.

The message to be hashed is processed by SHA-1 in 512-bit blocks. Like MD5, the message is first padded so that its length is a multiple of 512 bits. Five 32-bit chaining variables are initialized to (hex): cv_1 = 0x67452301, cv_2 = 0xefcdabfe, cv_3 = 0x98badcfe, cv_4 = 0x10325476, and cv_5 = 0xc3d2e1f0. For each message block, four rounds of 20 operations each are performed. The message block of 512 bits is further divided into 16 sub-blocks of 32 bits each and then expanded to 80 sub-blocks of 32 bits each. The chaining variables (cv_1, cv_2, cv_3, cv_4, cv_5) are copied into round variables (rv_1, rv_2, rv_3, rv_4, rv_5), respectively.

Four constants are used in the main loop: C_1 = 0x5a827999 for the first round, C_2 = 0x6ed9eba1 for the second round, C_3 = 0x8f1bbcdc for the third round, and C_4 = 0xca62c1d6 for the final round. For every operation, there is a nonlinear function on round variables rv_2, rv_3, and rv_4. The result (R_1) is added to the constant for that round, rv_5, and a 32-bit sub-block of the message to give R_2. Round variable rv_1 is rotated to the left 5 bits and then added to R_2 to give R_3. Round variable rv_5 is then replaced by rv_4, rv_4 is replaced by rv_3, rv_3 is replaced by rv_2 after it has been rotated to the left 30 bits, rv_2 is replaced by rv_1, and rv_1 is replaced by R_3. The round variables (rv_1, rv_2, rv_3, rv_4, rv_5) are then added to the chaining variables (cv_1, cv_2, cv_3, cv_4, cv_5), respectively. The main loop is repeated until all message blocks have been processed, after which the chaining variables are concatenated to give the 160-bit SHA-1 hash.

The 160-bit output from SHA-1 provides greater protection from brute-force attacks than the 128-bit hash algorithms (collisions are more difficult to find). Currently, there are no published cryptographic attacks against SHA-1.

D. Keyed Hash Functions (MACs)

A keyed hash function, also known as a MAC (message authentication code), takes a message and a secret key as input and outputs a fixed-length hash. A MAC has characteristic that unkeyed hash functions do not have. It has the property that given a message, it is difficult to produce the same MAC output without knowledge of the secret key. Thus, a MAC can provide data integrity like an unkeyed hash but it can also provide authentication without the need for any other additional means of authentication such as digital signatures. MACs are good for determining if a message has been modified and authenticating who created the MAC. Only someone with the secret key can produce or verify the hash.

A common way to implement a MAC algorithm is by using a block cipher in CBC mode. Using AES for example with a 128-bit block size and 128-bit key K, a MAC can be generated for a message by encrypting the message using AES in CBC mode with IV = 0.

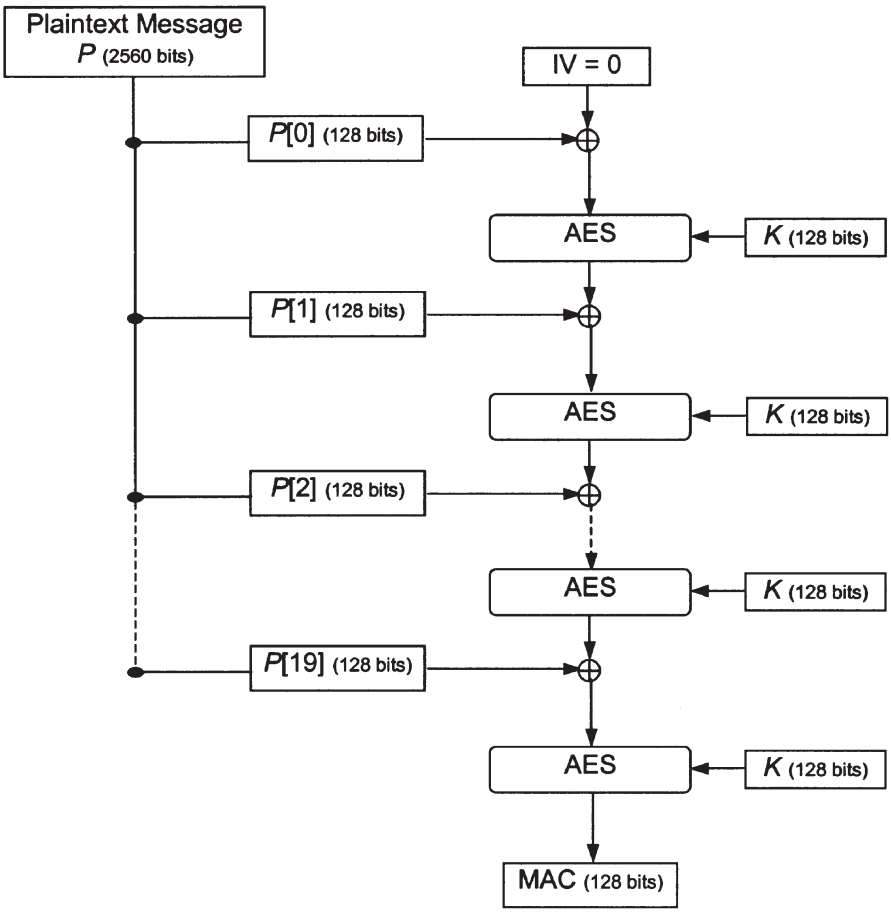


Figure 6 MAC algorithm using 128-bit block AES-CBC and 128-bit key.

The last encrypted block becomes the MAC. For this case, because AES has a 128-bit block size, the MAC size is also 128 bits. The MAC secret key is the 128-bit AES key *K*. This process is shown in Fig. 6. To verify the MAC, the message is encrypted using the same method as above with the MAC secret key. The final output is then compared to the MAC and, if they match, the message has not been modified and only someone with knowledge of the secret key was able to create that MAC.

IV. DIGITAL SIGNATURES

A. Basic Principles

Digital signatures are very different from the handwritten signatures used on paper. Because data on a computer can be easily copied, an image of a person's written signature could be cut and pasted into a new document, making signature forgery a simple task. A different type of signature had to be designed for the digital realm.

Digital signatures can provide data integrity, authentication, and support for nonrepudiation. After a message has been signed, it cannot be modified without being detected. A valid digital signature can only be created by the original signer (i.e., cannot be forged) and thus can prove who signed the message. While signature creation relies on private information, signature verification must be possible with public information. The signer cannot later deny signing the message.

Public key cryptography can also be used for digital signatures. The RSA algorithm and DSA (Digital Signature Algorithm) are such examples.

B. RSA

The RSA public key algorithm makes use of a public/private key pair. The public key is used to encrypt

messages and the private key is used to decrypt messages. The reverse is done to create a digital signature. Only the owner of the key pair knows the private key, but everyone can know the public key. The owner uses his private key this time, instead of someone's public key, to encrypt a message ($c = m^d \bmod n$). This is the signature since only the owner of the private key could have performed this task. Anyone can take the owner's public key and decrypt the message, thus verifying the signature ($m = c^e \bmod n$).

The RSA algorithm is slow for large messages so most implementations of RSA signatures use one-way hash functions in conjunction with signing. A message is hashed using an algorithm such as SHA-1. The hash is then signed (by encrypting the hash with the RSA private key). The message and signed hash are then sent together to the recipient. The recipient can verify the signature by decrypting the encrypted hash using the RSA public key of the sender. The message is then hashed using the same hash algorithm (SHA-1 in this case) and, if it matches the decrypted hash, the signature is valid.

C. DSA

The Digital Signature Algorithm (DSA) was developed in 1991 by NIST and published as U.S. standard FIPS 186. The DSA uses the SHA-1 hash function and is a variant of the ElGamal signature algorithm. Unlike the RSA algorithm, which can be used for both encryption and digital signatures, the DSA algorithm can only be used for signing. DSA has a key size between 512 and 1024 bits.

As with RSA, a public and private key pair is generated. A 160-bit prime number q is generated. A prime number p is then generated such that q divides $(p - 1)$ and p is between 512 bits and 1024 bits (in 64-bit increments). An element e is selected where $e < (p - 1)$ and $(e^{(p-1)/q} \bmod p) > 1$. A generator is calculated using $g = e^{(p-1)/q} \bmod p$. The DSA private key x is generated by randomly selecting an integer $x < q$. The public key y is generated using $y = g^x \bmod p$ and is published along with the values p, q, and g.

A DSA signature can be created for the message m using the following steps [*Note:* SHA(m) represents the message m being hashed by the SHA-1 algorithm]:

1. A random integer k is selected such that $k < q$.
2. $r = (g^k \bmod p) \bmod q$.
3. $s = (k^{-1} (\text{SHA}(m) + xr)) \bmod q$.

The digital signature (the values r and s) is sent with the message m.

The DSA signature (r and s) for message m can be verified using the following steps:

1. Obtain the signer's public key (y, p, q, and g).
2. Verify that r and s are less than q and not = 0. If not, signature is invalid.
3. $w = s^{-1} \bmod q$.
4. $u_1 = (w * \text{SHA}(m)) \bmod q$.
5. $u_2 = rw \bmod q$.
6. $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$.

The signature is valid if $v = r$.

The DSA relies on the difficulty of computing discrete logarithms for its security. A 512-bit DSA key is no longer strong enough for long-term security so using DSA with 1024-bit keys is recommended.

V. POLITICAL ISSUES

Encryption and the use of encryption technology are very political topics. Originally, encryption was used by the military to protect its communications and only the government had access to it. With the advent of home computers and the Internet, citizens now have access to strong encryption. Governments want to control the use of encryption and encryption technology so that it cannot be used against them. They want to control the export of strong encryption so that foreign governments, drug cartels, and terrorists cannot use it to protect their communications from intelligence agencies. They want to control the use of encryption by their citizens so criminals cannot use it to hide their plans from law enforcement. Citizens are lobbying to reduce the control on encryption so they can protect their privacy. Corporations are lobbying to reduce the control on encryption to help promote electronic commerce, so they can sell their products outside of the country, and to protect their foreign offices.

A. Export Control

Most countries have laws on the export of encryption technology. Canada and most European countries have fairly relaxed export restrictions on encryption, allowing source code and programs in the public domain (freely available) of any strength to be exported without a permit. The U.S. government had a more restrictive export control law. Prior to 1997, only products

with 40-bit keys or less could be easily exported from the United States. At that time, a 40-bit key could be cracked with a dozen computers in less than a day, so very few people were willing to buy encryption products from U.S. companies. In 1997 the U.S. government increased the maximum key size from 40 to 56 bits (the same key size as DES) and stipulated that any company exporting 56-bit products had to include a key recovery mechanism into the product within 2 years.

In January 2000 the U.S. government relaxed export control to a level where an encryption product of any key size could be exported to nongovernment users after it had undergone a technical review. Exception is made for several countries (Cuba, Iran, Iraq, Libya, North Korea, Sudan, and Syria) as classified by the U.S. government. In July 2000, the export control was further relaxed to allow special export status to the 15 European Union countries, and Australia, Norway, the Czech Republic, Hungary, Poland, Japan, New Zealand, and Switzerland. Export to those countries could now include government users, and products could be shipped immediately without waiting for a technical review to be complete after submitting a commodity classification request to the Department of Commerce. Export of strong encryption between the United States and Canada has been allowed for some time.

B. Domestic Use

Many countries such as Australia, Canada, Germany, Japan, Mexico, and the United States do not restrict the use of encryption by its citizens. In the United States, a citizen can use any encryption product for protecting her data and communications. Law enforcement agencies such as the Federal Bureau of Investigation (FBI) have been trying to get legislation introduced for years to enable it to be able to decrypt encrypted data of suspected criminals. The idea is that the FBI would only be able to do this with a court order, similar to the way wiretaps work. Several techniques have been proposed for implementing such a system, the most well known being the Clipper Initiative. In 1993, the Clipper chip was proposed. It is an encryption chip that would be installed in all new telephones, fax machines, and modems. A copy of the private key would be held by the government in case law enforcement needed to decrypt the communications for a wiretap. There was a large outcry from the people and the law was never enacted.

Other countries have strict rules on the use of encryption by citizens. Until recently in France, citizens had to submit their private keys to the government if they wanted to use encryption technology. Countries such as Belarus, China, and Russia severely restrict the use of cryptography; a license is required to use encryption technology. In the United Kingdom, there are no restrictions on what encryption technology can be used, but a court can force someone to either decrypt their data or turn over their private keys. If the order is not complied with, the offense is punishable by 2 years in prison. Before using encryption technology, it is best to first check the laws of your country.

SEE ALSO THE FOLLOWING ARTICLES

Copyright Law • Crime, Use of Computers in • Ethical Issues in Artifical Intelligence • Firewalls • Game Theory • Security Issues and Measures • Software Piracy

BIBLIOGRAPHY

Adams, C. (1997). The CAST-128 encryption algorithm, RFC 2144. Available at http://www.ietf.org/rfc/rfc2144.txt.

Adams, C., and Gilchrist, J. (1999). The CAST-256 encryption algorithm, RFC 2612. Available at http://www.ietf.org/rfc/rfc2612.txt.

Daemen, J., and Rijmen, V. (1998). AES proposal: Rijndael. Available at http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf.

den Boer, B., and Bosselaers, A. (1994). Collisions for the compression function of MD5, *Advances in Cryptology—EUROCRYPT '93 Proc.* New York: Springer-Verlag, 293–304.

Diffie W., and Hellman, M. E. (1976). New directions in cryptography. *IEEE Trans. Information Theory,* Vol. IT-22, No. 6, 109–112.

Diffie, W., van Oorschot, P., and Wiener, J. (1992). Authentication and authenticated key exchanges. *Designs, Codes and Cryptography.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Electronic Frontier Foundation (1998). Cracking DES, O'Reilley. Available at http://www.eff.org/pub/Privacy/Crypto/Crypto misc/DESCracker/.

Harley, R. (2000). Elliptic curve discrete logarithms: ECC2K-108. Available at http://pauillac.inria.fr/~harley/ecdl7/.

Johnston, M. (2000). U.S. updates encryption export policy. IDG News Service, Washington. Available at http://www.idg.net/idgns/2000/07/17/UPDATE1USUpdatesEncryptionExportPolicy.shtml.

Koops, B. (2000). Crypto law survey. Version 18.3. Available at http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm.

Menezes, A. (1993). *Elliptic curve public key cryptosystems.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Menezes, A., van Oorschot, P., and Vanstone, S. (1996). *Handbook of applied cryptography.* Boca Raton, FL: CRC Press, 191–195, 352–354.

Mills, E. (1997). Consensus needed for encryption export policy to succeed, IDG News Service, San Francisco. Available at http://sunsite.nstu.ru/sunworldonline/swol-02-1997/swol-02-encryption.html.

National Bureau of Standards (1976). Data encryption standard, NBS FIPS PUB 46. Gaithersburg, MD: National Bureau of Standards, U.S. Department of Commerce.

National Institute of Standards and Technology (1994). Digital signature standard, FIPS PUB 186. Gaithersburg, MD: National Institute of Standards and Technology, U.S. Department of Commerce.

National Institute of Standards and Technology (1995). Secure hash standard, FIPS PUB 180-1. Gaithersburg, MD: National Institute of Standards and Technology, U.S. Department of Commerce.

National Institute of Standards and Technology (1997). Advanced Encryption Standard (AES) development effort. Available at http://csrc.nist.gov/encryption/aes/.

National Institute of Standards and Technology (2000). Rijndael: NIST's selection for the AES. Available at http://csrc.nist.gov/encryption/aes/rijndael/.

Rivest, R. L. (1992). The MD5 message digest algorithm, RFC 1321. Available at http://wwwietf.org/rfc/rfc1321.txt.

Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM,* Vol. 21, No. 2, 120–126.

RSA Security (1999). DES challenge III. Available at http://www.rsaesecurity.com/rsalabs/des3/.

RSA Security (1999). Factorization of RSA-155. Available at http://www.rsasecurity.com/rsalabs/challenges/factoring/rsa155.html.

Schneier, B. (1996). *Applied cryptography,* 2nd ed. Toronto: John Wiley & Sons, 397–398.

Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell System Technical Journal,* Vol. 29, No. 4, 656–715.

End-User Computing Concepts

Joseph B. O'Donnell *Canisius College*
G. Lawrence Sanders *State University of New York, Buffalo*

GLOSSARY

agency theory An economic theory that views a firm as a set of contracts among self-interested individuals. Users avoiding agency costs with IS professionals is considered to be a major motivation for end-user computing.
command-level end user End user/developer who creates software applications through the use of fourth-generation language commands.
data mining The process of finding relationships that are unknown to the user. Data mining in the context of data warehouses is the automated analysis of large data sets to find patterns and trends that might go undiscovered.
data warehouse A centralized data resource that allows users to manipulate and use data for knowledge discovery. Includes end-user tools for data mining and online application processing.
end-user computing (EUC) The development by users of all or part of their computer-based systems.
end-user programmer The most sophisticated level of end user who develops software applications by creating customized computer code through programming languages.
fourth-generation language Category of programming languages that uses nonprocedural instructions and involves specifying what needs to be accomplished rather than providing the details of the task to be performed.
menu-level end user Least sophisticated level of end-user development in which the user creates software applications through the use of menu commands.
online application processing (OLAP) Consists of data warehouse tools that allow users to analyze data through multidimensional views in order to increase the meaningfulness of the information.
prototyping The process of building an experimental system for demonstration and evaluation so that the users can determine requirements.

I. INTRODUCTION

End-user computing (EUC) is the development by users of all or part of their computer-based systems. EUC differs from other forms of computing in that the system is built by the end user rather than by an information systems (IS) professional. The beginnings of end-user computing are tied to the emergence of personal computers, which provide a lower cost alternative to mainframe computer processing. EUC began with individual user, decision support systems such as spreadsheets for analysis and have expanded to include group-based systems in personal computer, data warehouse, and Internet environments. Fourth-generation languages and increasing familiarity with personal computers have fueled the growth of EUC. User-developed software has the advantages of flexibility and rapid development but involves control,

security, and coordination issues for the organization. This article discusses EUC fundamentals, motivations for EUC, EUC issues, and EUC in data warehouse, Internet, and future mobile computing environments.

II. FUNDAMENTALS OF END-USER COMPUTING

A. End-User Computing and Applications

The emergence and growth of users developing software has paralleled that of personal computers. Personal computers have provided a lower cost alternative to mainframe computer processing. In addition, personal computers have provided increasingly user-friendly software that has enabled users to develop their own software. End users are not considered to have specialized IS skills but rather use the information system for business or personal purposes. Prior to EUC, systems were developed and maintained by IS professionals. IS professionals have job titles such as systems analysts, database administrators, programmers and operators.

As shown in Fig. 1, EUC began with the advent of personal computers in the early 1980s. Early EUC involved spreadsheet software packages that enabled users to perform calculations to solve business problems. In performing these calculations, users developed formulas that would perform repetitive calculations. In the 1980s these spreadsheets were typically used individually, but the trend toward group computing systems was growing. By the early 1990s, EUC had expanded beyond decision support spreadsheet systems to include operational systems throughout the organization. Advances in graphical user interfaces and fourth-generation languages further spread EUC to include large database systems, known as data warehouses, and Internet Web sites by the mid-1990s. Finally by the late 1990s, handheld digital devices emerged as another viable technology for end-user computing.

The growth of the Internet has expanded EUC from primarily individual user applications to include group applications. From an organizational perspective, EUC started as intraorganizational systems in the personal computer environment and has grown to include interorganization and organization-to-customer systems on the Internet. Intraorganization systems facilitate communication of business information within organizations, simplify information retrieval, and provide decision support capabilities. Interorganizational EUC systems enable business partners to communicate and possibly process business-to-business (B2B) transactions in an efficient manner. Organization-to-customer EUC systems facilitate communication of product, service, and company information to the consumer, and possibly process business-to-consumer (B2C) electronic commerce.

B. End-User System Development Skills

End user/developers range from those who use menus to create their own applications to those who use programming languages to develop software. Menu-level end user/developers use menus and templates to generate software applications. The next level of end-user development is the command level where users are able to generate applications through commands that are beyond the menus. For examples, this level of user is able to create commands such as formulas in fourth-generation language spreadsheets. Finally, the end-user programmer, the most sophisticated level of user, develops applications by using customized computer

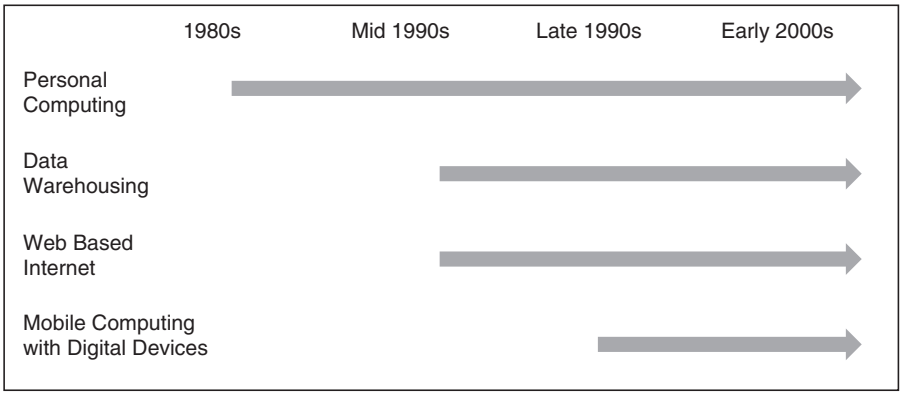


Figure 1 Timeline of end-user computing expansion.

code through programming languages such as Java. The end-user programmer level consists of individuals who understand the underlying syntax and logic of a programming language. For efficiency reasons, this level of developer may decide to generate program code through fourth-generation languages using menus and commands and then modify the generated code to meet the requirements of the application.

C. Prototype Development

The EUC computing environment is geared toward a flexible and iterative systems development process. EUC is well suited for use of the prototyping systems development methodology, which is also a repetitive process. Prototyping is the process of building an experimental system for demonstration and evaluation so that the users can determine requirements. The steps in the EUC prototyping process are as follows:

1. Users identify their basic requirements through their firsthand knowledge of the process.
2. Users develop an EUC initial prototype (experimental model) through use of fourth-generation language software. This prototype may only contain some of the functionality of the proposed system, but provides an overall feel and look of the proposed system.
3. Users interact with the prototype and suggest changes. The end user evaluates the functionality of the system functions by comparing the operation of the prototype to the system requirements.
4. Users revise and improve the EUC prototype. The user makes the changes to the prototype to meet the system requirements.
5. Repeat steps 3 and 4 until users are satisfied.

The prototyping process is very different than the traditional systems development life-cycle approach that stresses completion of planning and analysis before beginning the design process. Benefits of prototyping include facilitating the learning process, speed of development, flexibility, and a better ability to meet end-user needs. EUC prototyping allows the user to learn the requirements throughout the process of creating and testing the system rather than the more abstract process involved in the analysis phase of the systems development life cycle. Some researchers believe that prototyping represents a trial-and-error approach that is fundamental to the human learning process. Another benefit of the prototyping approach is rapid development of the system by providing a working model with limited functionality early in the process. EUC prototyping is flexible as the design of the application continually changes. Finally, researchers have found that systems built through prototyping better meet the needs of the users.

A major disadvantage of prototyping is that it may not be appropriate for large or complex systems, which may require significant amounts of coordination. However, some use of prototyping may be very useful for large projects when integrated with systems development life-cycle methodologies. Furthermore, use of EUC for large and complex projects may not be appropriate due to the limited systems analysis skills of most users. EUC and prototyping may both be best suited for small, less complex systems.

III. MOTIVATIONS FOR END-USER COMPUTING

A. Flexibility and Efficiency

The flexibility of EUC results from the end user developing a software application without assistance from IS professionals. This avoids communication issues, reduces problems of managing limited IS resources, and diminishes the restrictions of IS capabilities. Because the end user understands the purpose and functional requirements of an application, he or she can design the application to meet these requirements. Conversely, in the non-EUC environment, the end user must communicate the requirements to the IS professional who may be more well versed in the technical aspects of computing than with the organization's business processing needs. In addition, the availability of IS professionals is often limited, potentially delaying the development of the software application. Finally, system development options may be limited by the technical capabilities of the organization. Implementation of systems that requires limited available technical expertise may cause a development project to be bypassed. Conversely, EUC uses fourth-generation language, development software that reduces the technical expertise required to develop the application. Thus, the organization can be less concerned about the technical skills and focus on the project itself.

Another aspect of flexibility is the capability for end users to design ad hoc software applications for one-time or infrequent use. This allows end users to quickly and efficiently obtain information without concern for the availability of IS professionals to assist in the project. In summary, EUC provides opportunities for

developing software more efficiently by reducing the need to communicate detailed requirements and by avoiding the need for extensive IS professional assistance. It is also more flexible in meeting the needs of the users.

B. Agency Theory

Agency theory is an economic theory that views the firm as a set of contracts among self-interested individuals. An agency relationship is created when a person (the principal) authorizes another person (the agent) to act on his or her behalf. Issues arise in this relationship based on the risk that the agent will act opportunistically and not act in the best interests of the principal. Therefore, a contract or agreement should be designed to eliminate or reduce the potential of the agent acting opportunistically. Applying agency theory to traditional software development, the end user acts as the principal authorizing the IS professional, the agent, to design and build a software application. To avoid the costs of this agency relationship, the business user of this system may adopt EUC to develop this system. To understand the users' motivations for avoiding an agency relationship, it is useful to describe the nature of an agency relationship and the costs of maintaining it.

The nature of an agency relationship is that the principal delegates authority to the agent and cannot observe the agent's actions. However, the principal controls the agent's actions through contract arrangements. Problems arise in the arrangements due to different goals and information asymmetry of the principal and the agent. In terms of corporate computing, the end user is interested in obtaining system functionality at the least cost, while IS professionals may have different objectives. For instance, a user may receive a less than optimal system because it better matches the skill set of the IS professional. Information asymmetry results from IS professionals having a much greater understanding of the technical requirements of systems development. An IS professional's information advantage may lead to two problems known as moral hazard and adverse selection. Moral hazards occur when the IS professional exhibits a lack of effort, while adverse selection occurs when the IS professional misrepresents his or her ability to the user.

The agency relationship between an end user and IS professional involves the costs of monitoring, coordinating, and experiencing residual loss. Monitoring costs are incurred by the user to ensure that the IS professional is not taking advantage of the agreement with the user. For instance, this includes such costs as reviewing and evaluating the progress of project plans and budgets. Coordination costs involve the expense to the user of organizing the activities of the IS professional. This would include any meetings or communications that are initiated by the user to coordinate the systems development project. Finally, the user's residual loss is the amount of loss that the user experiences because the IS professional has different interests and the optimal system is not implemented.

A chargeback system is a commonly used method to manage the user–IS professional relationship in order to achieve goal congruence, i.e., similar goals, among the parties. Under this approach, IS costs are allocated back to the user departments based on criteria such as use of IS resources. Chargeback systems motivate IS professionals to provide services and systems that are in the greatest demand. Chargeback systems also reduce the information asymmetry as the user gains an improved understanding of the IS resources expended to complete a task.

In 1990, Gurbuxani and Kemerer suggested that adoption of EUC is a result of the goal incongruence and information asymmetry of the user and IS professional relationship. EUC increases users' ability to have direct control over their computing needs. Also, the users have a better understanding of the systems development process. In summary, EUC represents an opportunity to reduce agency costs by reducing the IS professional's role in developing systems and increasing the users understanding of the processes and requirements of developing and maintaining information systems.

IV. ISSUES IN END-USER COMPUTING

A. Growth

The growth of EUC and EUC-related tools, such as personal computers and the Internet, is advancing at a much faster rate than traditional computing and mainframe systems. Factors affecting EUC growth include growth accelerators that relate to EUC benefits and potential growth inhibitors that involve organizational issues in adopting EUC (Fig. 2). Reasons for their growth can be viewed through characteristics identified in adoption of innovation research. Rogers has identified five of these characteristics: relative advantage, compatibility, complexity, trialability, and observability. The relative advantage of EUC over the traditional computing environment includes in-

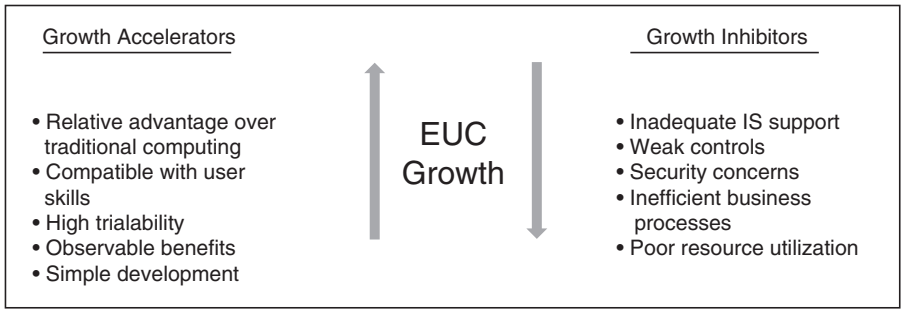


Figure 2 Factors affecting end-user computer growth.

creased flexibility and efficiency, and reduced agency-related costs. Theoretically, the greater the relative advantage of EUC over traditional computing, the faster the growth of EUC.

EUC is also compatible with the information management habits of many individuals in their personal lives. Younger employees entering the marketplace have developed EUC applications, such as analytical tools, for educational reasons and as means of communication, such as Web sites. Many older employees have also learned to adapt to EUC through their personal and professional experiences. Accordingly, the use of EUC in an organizational setting parallels and is compatible with its use in a personal setting.

The advent of graphical user interfaces (GUIs) and fourth-generation languages has decreased the complexity of designing and using EUC. GUIs make extensive use of bars, buttons, and boxes to perform commands that were previously performed by text-based, menu-driven systems. Fourth-generation languages are nonprocedural languages that involve specifying what needs to be accomplished rather than providing the details of the task to be performed. These technological advancements make it easier to build and use EUC and, thus, increases its growth rate.

EUC has a high level of trialability in that the software applications can be tried on an individual basis without impacting other users. For instance, individuals using personal computers can potentially build and use an EUC on a personal computer without disrupting the activities of other individuals in the organization. In addition, by developing prototypes, the users are able to test portions of the software application during the development process. EUC's high level of trialability increases its rate of EUC growth.

Finally, the growth of EUC is related to its benefits being readily observable to users. For example, benefits of EUC applications can easily be viewed through the quality of information that is obtained or generated. With regard to the Internet, individuals can easily search the Web to view EUC Web sites and determine the resulting benefits of the site. The ease of observing EUC benefits increases the growth rate of EUC. In summary, EUC's high levels of relative advantage, compatibility, trialability, and observability and low levels of complexity are major reasons for the growth of EUC.

B. IS Support

Users indicate that benefits of EUC include faster response to user needs, reduced application development backlog, and more successful system implementations. However, disadvantages include lack of experience with developing software and incompatibility of hardware and software. Support of end-user activities is important to the success of the software application in limiting the disadvantages of EUC.

IS support of EUC may include end-user training, hardware and software selection, coordination, and operational support. The process of learning how to use systems development tools has been perceived as too time consuming and infringing on job responsibilities. Researchers have found that decision-making satisfaction is positively related to the overall satisfaction with the level of training. Fourth-generation languages address the learning issue by including user-friendly interfaces, tutorials, and extensive help information. Due to the ease of using fourth-generation language, the training has become less focused on technical issues, such as programming, and more concentrated on how to most efficiently and effectively use the software to develop EUC. Organizations can also address this concern by providing training and learning aids such as communicating software tips and capabilities through newsletters or Web sites.

In terms of software and hardware compatibility, the IS professional's role in EUC is to define a computing platform that facilitates the users' development

and use of software. Thus, the IS professional's role is to assist in selecting company-wide development software and hardware that is used for user development. In addition, the IS professional supports the operation of the user-developed software. Due to the growth of networking and the Internet, support may need to be provided to remote, as well as centralized, computing locations. Thus, under EUC, the IS department role is transformed from being primarily responsible for systems development to a technical support role.

IS support may also involve coordination of EUC development and standardization of EUC design. The coordination issue involves ensuring that efforts are not duplicated in other parts of the organization. Further, the IS department is responsible for providing database information that is meaningful to users throughout the organization. Next, with regard to standardization, the EUC software should contain standard design features, such as screen formats, so that the users of the system will be able to more easily learn the features of the system. In addition, maintaining standard design features reinforces corporate communication strategies such as company image and quality standards. IS professionals may be required to assist in supporting the operation of the software. This requires that the IS professionals be trained to properly establish software parameters and troubleshoot difficulties when the software is down.

C. Organizational Control Issues

1. Risks of EUC

One of the greatest challenges of EUC relates to the control issues of users developing software. This loss of control can be a large impediment as organizations grow. End-user-developed applications are not designed for the long haul. They are usually not scalable and readily maintainable. Mcleod and Schell have identified the following risks related to end-user computing:

- *Poorly aimed systems.* Systems built by users could be built in a different more effective way and the EUC system actually might not solve the right problem.
- *Poorly designed and documented systems.* Users do not have the required experience to build systems to be maintained over the long run. They also do not have the breadth of knowledge to apply the proper design techniques to solve a problem.
- *Inefficient use of organizational resources.* Incompatible hardware and software proliferates. Systems and their accompanying data tend to be redundant. This leads to decreased levels of data integrity.
- *Loss of security.* End users do not safeguard their data and software. Data abound on hard drives, diskettes, and CD-ROM. Printouts are not disposed and maintained in a secure fashion. Steps are not taken to back up and secure all type of media containing data.

2. EUC System Reliability Issues

Control issues to address these risk concerns include reliability and maintainability of the EUC system information and processes. EUC systems are typically not built under recommended quality control standards for systems development. For instance, user-developed software often is not subjected to extensive system and user testing procedures before being used by the organization. Also, updates to the software are often performed based on limited testing. Consequently, the information provided from the EUC may be inaccurate and unreliable. Poorly managed EUC systems development may result in duplicate, unreliable, and inaccurate EUC systems.

Maintainability of systems is highly dependent on proper system documentation to enable use of the EUC after the user/developer is no longer involved with the system. Unfortunately, EUC systems are typically poorly documented due to lack of training, user interest, and availability of resources. First, end users are not trained in the methodologies to properly document system flows and program features through flowcharts and descriptive narrative. Second, end user/developers are often drawn to EUC as a method of designing an application to solve a business problem. Their interest typically doesn't include the tedious task of documenting the system. Lastly, the end user/developer generally has many other business responsibilities and is typically not provided with sufficient time to document the system.

3. Security Issues

End users are typically not knowledgeable about the requirements and processes for designing secure systems. Furthermore, early EUC software development tools did not include easy-to-use, menu-driven features to develop secure applications. Consequently, some EUC systems are built with inadequate security controls and, thus, are exposed to the risk of inadvertent changes to applications and data, and hackers gaining unauthorized access to the organization's computing resources.

Traditional IS applications are usually maintained in secure computer libraries that severely restrict unauthorized access. Conversely, many EUC systems are maintained on the user's personal computer that may have poor security. For poorly secure systems, users may inadvertently change the application programs or incorrectly change the application data. In addition, poor security increases the risk of hackers corrupting EUC applications programs and data, and stealing confidential company information.

D. Innovation

Though control is an important issue in the unbridled proliferation of EUC applications, there looms the larger issue of facilitating technological innovation and its diffusion. EUC is a critical mechanism for fostering technological innovation because end-user computing enclaves often implement and pilot emerging technology applications. As Fig. 3 illustrates, the principle of network externalities (often referred to as Metcalf's law) states that utility of a network equals the square of the number of users. Simply put, the more people you have involved in end-user computing, the more valuable it becomes as the dynamic interaction of the participants creates an engine for change and innovation. End users attracting other end users to the technology power this engine of innovation; this in turn increases the utility and adoption of the technology throughout the entire organization.

E. Process Implications

The decision to use EUC often fails to consider the impact of the software on the business process. EUC may provide a quick and effective method for developing a business tool for processing, however, this may not be the most efficient method for processing information. An organization has numerous processes in areas such as product/service delivery, new product development, customer relations, and supply chain management. These processes include integrated activities that are dependent on delivering the end product of the process. Accordingly, changing one component of the process could potentially impact many of the subsequent activities. Thus, implications of EUC changes should be considered in the overall process design.

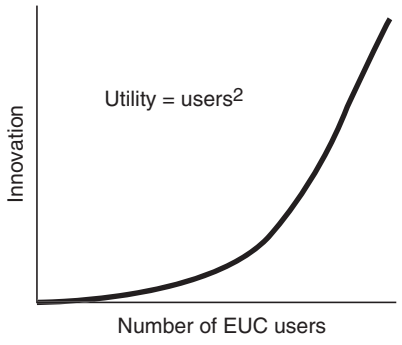


Figure 3 Effect of network externalities on innovation.

The traditional IT development life cycle considers the process impacts during the systems analysis and design phase of the development project. Trained IS professionals would consider existing processes and the requirements of new or changed processing when designing the new system. Under EUC, the user is performing the analysis and design of the new system and processes. The impact of the user performing analysis and design has both positive and negative aspects. On the positive side, the user is very familiar with the requirements of his or her portion of the process. Thus, the systems analysis and design are done efficiently and effectively to meet the requirements of the user/developer's portion of the process. On the negative side, the user may not be aware of other portions of the process and, thus, be unable to identify other opportunities for improving the process. Many IS professionals are trained and have experience in analyzing process flows and identifying opportunities for improvement, while users commonly are not trained in this area. Consequently, EUC may not maximize the benefit of the technology or may actually reduce the efficiency and effectiveness of the overall process. One example of underutilizing technology is to simply automate a process that was previously performed manually. This approach omits opportunities to gain improvement through process reengineering. Process reengineering involves radical redesign of a process to reduce cycle times and improve quality.

Another potential underutilization results from the absence of potential improvement of other components of the process. This results from the user who develops the EUC not being aware of the other components of the process and the recognition that a change in process would be beneficial. The user/developer may not have the skills or desire to investigate the opportunities for process improvement. Also, increasing the scope of the EUC to include other component processes may increase the risk of the process failing. This increased uncertainty may provide incentive for the EUC computing to be used for

incremental changes rather than broad comprehensive changes.

A more negative scenario of EUC involves a degradation of the overall process. Under this scenario, the implementation of an EUC results in the other component processes being less effective or efficient. This may occur as a result of transaction volumes rising above a level where the EUC operates in an efficient manner. Thus, designers of EUC must consider the impact of increasing volume on the process flow.

In analyzing EUC decisions, users should consider the impact on each of the process performance measurements. Common methods to measure the performance of a process include, but are not limited to, cycle time, on-time delivery rates, error rates, costs, and satisfaction levels. Cycle time relates to the time elapsed from the initiation to the completion of the process. On-time delivery rates reflect how often the process delivers its end product on schedule. Error rates are the percentage of the process's errors in comparison to its total output. Costs include capital and human resource expenditures to create and maintain the process. Satisfaction levels capture perceived success of the process and are typically measured through survey instruments. Theoretically, EUC applications that have the greatest impact on performance measures are of the greatest benefit to the organization.

F. Resource Efficiency Issues

EUC applications are often generated through use of fourth-generation languages. These languages tend to create resource-intensive EUC applications that include functionality that is not required for the application. Conversely, when an application is built specifically for an application, the resulting code is typically more streamlined and possibly operates more efficiently. Resource utilization is of greater importance for larger transaction volumes when response time is a concern. Therefore, consideration of whether to use the fourth-generation language or manually customized code may be related to an application's expected size and transaction volume.

V. DATA WAREHOUSE END-USER COMPUTING

A. Data Warehouse Structure

The main goal of the data warehouse (DW) is providing end-user analysts and managers with access to data and data analysis methods that improve strategic and operational decision making. The focus of the DW is to provide information at a level of aggregation that is most meaningful to the decision maker. This optimal level of aggregation may vary by user or the problem being solved. In fact, the DW should be capable of providing a level of detail ranging from detailed transactions through to summary, organizational-level information. In addition, this information may originate from various sources within, and external to, the organization. The IS professional must aggregate and massage these data into a meaningful form for the use in EUC. Data warehouse end-user tools are comprised of two types of decision support tools: (1) data mining and (2) data access and manipulation.

B. Data Mining

Data mining is the process of finding relationships, which are unknown to the user. Data mining in the context of data warehouses is the automated analysis of large data sets to find patterns and trends that might go undiscovered. For example, end users in a large international bank used data mining tools to analyze complex customer deposit trends to identify opportunities to improve customer satisfaction and bank profitability.

End-user data mining tools include easy-to-use statistical analysis tools for knowledge discovery. The tools include use of menus and graphics to simplify the creation of the statistical inquiry and interpretation of the results. Data mining tools assist the users by providing menus of statistical parameters that ensure that the appropriate information is provided to build statistical models for analyzing information. After analyzing the results, the user can easily change statistical parameters through menus and rerun the results. Data mining tools empower end users to perform analysis that traditionally required statistical specialists to develop customized statistical programs.

Data mining tools provide automated end-user tools to assist in identifying patterns and trends through capabilities such as standard statistical functionality and artificial intelligence. Data mining tools include fundamental statistical features, such as regression analysis, in a format that facilitates use without knowledge of the underlying programming requirements. In addition, artificial intelligence tools perform problem solving in a manner similar to human cognition. Although the underpinnings of artificial intelligence methods are based on sophisticated searching, interpretation, categorization, and opti-

mization algorithms, their use is relatively simple due to automated end-user data warehouse tools.

C. Data Access and Manipulation

End-user data warehouse access and manipulation tools include ad hoc query, drill down, and online application processing (OLAP). Ad hoc query capabilities enable users to make informal information searches without concern for the technical components of the search. End-user queries involve the accumulation and filtering of information across various tables of a database.

Drill-down capabilities provide the ability for the end-user to access different levels of information without creating ad hoc queries for each level of information. Instead the user is able to drill down to a detail level, or elevate to the summary level, through pull-down menus or mouse clicks. This data warehouse feature limits the risk of information overload in the knowledge discovery process by allowing the user to view the appropriate level of information for his or her purposes.

Finally, OLAP encompasses a set of data warehouse tools that allows users to view data in multidimensional views in order to increase the meaningfulness of the information. These tools allow individuals to easily look at information from a different perspective. For instance, OLAP would allow an end-user manager to first look at sales figures summarized by product line and then by geographic region without having to query the data warehouse a second time. This provides a fast and efficient method for users to view the information in multiple ways.

Although these data warehousing tools are extremely powerful, they do not replace the need for the end user to understand the meaning of the information. This includes the user comprehending the meaning of statistical results and the content of data fields. For instance, the user should be aware of the underlying assumptions used for pattern recognition in the data mining tools. This would enable the user to make more accurate inferences about patterns recognized by the data mining tools. The user must also understand the meaning of information provided in the data warehouse. Often, data warehouse data are obtained from various nonintegrated transaction processing systems. Thus, it is possible that similar field names can be used in different transaction systems and have different meanings. Interestingly, some users spend the vast majority of their data warehouses analysis time investigating the meaning and source of the information provided in a data field. Thus, the underlying meaning communicated by EUC data warehouse data and tools must be understood by the user to facilitate effective decision making.

D. IS Support

Data warehouse design is a challenging, yet critical task for IS professionals who must anticipate data and ad hoc query requirements to address future decision making. The real difficulty lies with the fact that most users are not aware of what data will be needed for future, currently unforeseen opportunities or problems. At best, these designers can develop a flexible system to adapt to ever-changing information needs.

Operational support issues relate to maintaining a complex infrastructure that often includes telecommunication networks, different hardware platforms, disparate systems, and a large database system. Data warehouse data must be continually updated from various systems that may operate under different system architectures. This can be particularly difficult when the company has legacy systems that are under a different operating system environment than the data warehouse. In addition, the IS professionals must support acceptable system response times to facilitate timely and efficient end-user decision making. Also, the data warehouse must be a secure environment where end users are only provided system access to those resources that they need for their job responsibilities.

VI. INTERNET AND E-COMMERCE END-USER COMPUTING

A. Web Design Consistency

A major design consideration is consistency of format for Web page design. Because many of a company's Web site users may be from outside of the organization, the Web site communicates the company's image to the public. EUC Web design involves the risk that the users will build Web sites independently without regard to corporate standards. Sites should be built using authorized software to facilitate the IS department support of the Web site.

The EUC Web sites should provide links to official company Web sites to assist users in reviewing relevant sites. In addition, controls are required to ensure that

EUC sites include updated company information. Also, uniform resource locators (URLs) must be kept up to date so that users do not encounter broken links.

B. Legal Issues

Internet and e-commerce Web sites represent a medium through which the organization and members of the organization can communicate with the public. Accordingly in litigious societies like the United States and Western Europe, Web site content is a potential source of lawsuits. Litigation may involve areas such as discrimination suits, intellectual property rights, consumer fraud, and contract law. The potential financial loss due to litigation costs and negative press coverage must be considered when designing a Web site. Consequently, legal experts should be consulted when designing appropriate standards for Web site content.

C. Communicating Trust through Design

Communicating trust is a major issue of EUC Web development, especially in the area of e-commerce sites. Individuals accessing e-commerce Web sites view information and have opportunities to perform transactions such as purchasing goods. A critical component of a Web site's success is that individuals trust the information on the site and that the site transactions will be performed as promised. This is especially true of EUC Web sites that may not have been developed in accordance with organizational and/or industry standards. A Web site can communicate trustworthiness by expressing traits such as dependability, reliability, and honesty. Trustworthiness can be communicated through the design and content of the Web site. Researchers have identified six components of e-commerce trustworthiness: image, navigation, presentation, fulfillment, technological sophistication, and seals of approval. This section analyzes the six components and discusses their impact on EUC e-commerce Web sites.

Image relates to a corporation's promise to deliver goods and services and its credibility based on reputation and the Web site visitor's previous experience with the company. The image is built through various communication channels such as television, newspaper, billboards, radio, and the Internet. The image of the company is further defined through personal experiences with company products, employees, and Web sites. The EUC Web site should encompass the company image and complement other communication media. This can be accomplished through use of corporate logos, slogans, and design styles. Thus, corporate communication officials should approve the Web site design. Further the use of company-based graphics should not adversely impact response time. Both of these requirements reduce the EUC flexibility in design and may increase the length of the development process. However, integrating image components will increase the trustworthiness of the site.

Navigation relates to the ease of finding what the user seeks. Effective navigation has been found to be associated with a site visitor's trust in the Web site. A critical component is to maintain a site structure and this structure should reflect the visitor's view of the information or service. Without a site structure, the site may evolve into a disorganized collection of random directories. The most effective site structures reflect the visitor's view rather than a company view such as organizational structure. The organizational structure may be easier in terms of dividing Web ownership and responsibility, but this structure often does not meet the needs of the visitor. Instead a Web visitor-focused design often results in a Web page containing information from several departments. Thus, the end user/developer must coordinate the Web site development process among several departments.

Additionally, navigation can be improved through reducing navigational clutter, providing search capabilities, and using meaningful page descriptions and keywords. Navigational clutter can be reduced through aggregation, summarization, filtering, and truncation. Aggregation involves showing a single unit that represents a grouping of smaller ones. Summarization involves presenting a large amount of information in a smaller fashion. Filtering involves limiting the amount of information based on some criteria such as what information has been found to be most useful by previous viewers of the site. Finally, truncation involves providing a portion of the message that allows the users to mouse click to another Web page to obtain the rest of the information. These approaches are geared toward limiting the risk of information overload for the visitor of the Web site. To limit information overload, the user/developer must not only consider the content of the Web site but how it is organized.

Providing web site search engines improves the ability of visitors to quickly find the sought-after information. Research by Nielson indicates that more

than 50% of site users go directly to a search engine, when available, rather than navigating through links. The search engine provides greater control of the navigation for the site visitor. Accordingly, user/developers should include a search engine in their site and consider the logic of the search engine and its performance capabilities.

Use of meaningful Web page descriptions and keywords increases the ability of site visitors to distinguish the purpose of the Web pages. Some of the Internet search engines will show the author's page description or abstract rather than trying to generate their own text. The length of the description involves a trade-off between brevity, to avoid information overload, and providing enough information to adequately explain the purpose of the page. The keyword list includes the terms that will be referenced in finding the Web pages. The end user/developer should provide descriptors and keywords that increase the chance that the site will be visited.

Presentation relates to design attributes that communicate quality. It involves use of graphics and layout to communicate the purpose of the site. Further, the presentation can reflect a high skill level of site design or may resemble other trusted sites. These attributes communicate a high quality level that prompts trust in the site. End user/developers should be aware of methods they can use to provide their sites with a professional look.

The fulfillment attribute indicates how orders will be processed and how recourse can be taken if there are problems with the order. A major issue with fulfillment is the privacy of customer information. Explicit statements that the information will be kept private helps to address site visitor concerns. In addition, tracking mechanisms allow customers to monitor the progress of the order and potentially reduce concerns about delivery of goods or services. Statements related to recourse involve return polices and descriptions of processes involved in returning the good. For the end user/developer, addressing system capabilities and return policies will build trustworthiness into the site.

The state of technology attribute connotes professionalism and sophistication of a company's technologies. This attribute includes the use of modern technology tools, the functionality provided at the Web site, and the speed at which page text and images appear. End user/developers should also consider the process of updating the Web site to gain maximum use of new technology such as Web browsers. In addition, inclusion of functionality that appropriately delivers relevant content and demonstrates technical capabilities should be considered. Finally, the processing speed of the site may be a concern when developing the site with a fourth-generation language.

Web site seals of approval are provided by outside organizations to assure the security of the Web site or the reliability of the company. Security seals may occur at the network, technology, or company level. At the network level, displaying icons related to network security, such as Verisign, communicates the safety of the site from hacker attacks. Icons from e-commerce-enabling functions, such as IBM e.business mark, connote the use of established technology. Finally, company-level icons from business partners or credit card companies symbolize the credibility of the organization hosting the site. Business partner icons provide an opportunity to build confidence through the reputation of the partner. Credit card icons symbolize reimbursement protection against fraudulent use of credit card numbers. Further, this icon demonstrates the importance of credit cards as a form of electronic payment.

Another form of certification comes through relationships with intermediaries. These middlemen provide conduits to the end retailer through comparison shopping sites. Comparison shopping Web sites facilitate the buyer's search for vendors and discovery of product features and prices. The intermediary is typically compensated for retailer sales of goods and services when the customer accesses the retailer's site via links from the intermediary's site. The intermediary has an incentive to maintain relationships with reliable retailers because those good relationships build customer satisfaction with the intermediary's service. Thus, inclusion of a retailer's link on a comparison shopper site is a form of certification for the retailer.

Another significant issue is the use of streamlined design to promote quick user response times for access to the site and its available functions. As of the year 2000, research indicates that users want Web site response times of less than 1 sec, whereas organizations have goals of approximately 10 sec. Thus, users want request times that cannot be met through current technology. However, the faster the response times of the site, the closer the organization is to meeting user needs. Research also indicates that variability in response time reduces user satisfaction. Both of the time response issues are of concern when EUC computing produces the Web site through fourth-generation languages, which generate applications that are generally not as efficient as manually coded

applications. Therefore, the possibility of variability in download speed exists when different methods are used by IS professionals versus end user/developers for designing Web sites. Also, response times of site functions such as downloading of files impact user satisfaction. In summary, EUC Web sites communicate an image—whether intended or not—to the general public. EUC sites that are poorly designed or not designed according to company standards communicate a negative corporate image.

D. Organizational Support

The overall impact of Internet and e-commerce Web sites on EUC development is that support from IS, corporate communications, business development, and legal departments is needed to create more effective sites. Similar to the PC environment, the users can use fourth-generation languages to build the application, while IS supports the operation of the application. Unlike the PC environment, the Web sites are focused on an audience that is outside the organization. Thus, experts in areas such as corporate communications and marketing need to be involved in Web site development. Also, business development specialists may be needed to develop relationships with business partners to facilitate the use of seal of approval icons. Finally, use of organizational icons, business logos, and wording on the Web site may require legal department approval to limit any possible litigation.

Although fourth-generation languages simplify the Web development process, the complexity of the business relationships that promote trustworthiness may require Web site development to be performed by IS professionals. These professionals may be better trained in identifying requirements and coordinating multidisciplinary projects. Furthermore, these individuals may be better able to identify the impacts of high transaction volume sites on back-office operations.

VII. FUTURE DIRECTIONS FOR END-USER COMPUTING: MOBILE COMPUTING

A. Trends in Mobile Computing

The population of mobile computing users is growing. Mobile computing started as the use of laptop computers from remote locations and has expanded to the use of handheld computing devices accessing wired or wireless networks. Users of EUC are able to create decision support software, communicate with others, and access information from handheld devices. This trend is expected to spread to domestic living quarters where the home network will coordinate activities of digital-based appliances. For instance, a user may be able to remotely control the operation of his or her digital heating unit to ensure that the home is warm when he or she arrives home from work. EUC is also expected to become more prominent in automobiles through the use of global positioning systems that provide the location of the automobile and direct drivers to their destination. These technologies are expected to provide new platforms for user-developed applications.

EUC potentially could move from a text- and graphic-based medium to a verbal environment where voice recognition technology translates voice commands into a digitized form for computer processing. Once perfected, this technology will simplify the process of end-user computing even further. This simplicity and the growing proliferation of handheld computing devices promises to increase the number of people using EUC.

B. Lean Applications

The nature of developing EUC is expected to change from developing resource-inefficient applications on personal computers to generating efficient programs that operate in a thin client computing environment of handheld digital devices. In addition, for the near future, these smaller devices are generally not able to store the same volume of information as personal computers. Thus smaller and simpler EUC programs may be preferable for processing. This is in sharp contrast to most fourth-generation languages that produce applications that include excess functionality.

C. Importance of Compatibility

Mobile computing development may be more concerned with compatibility and the efficiency of the computing code than previous EUC. Compatibility of processing is critical because the application needs to be able to perform on any of a myriad of digital devices. Also, some industry experts believe that a virtual network will emerge in which mobile users will seamlessly share resources. For instance, in this future network, a user with a digital device would be able to locate the geographically nearest printer, even if it is owned by another organization, and then print a document on that printer. Compatibility and some pay-

ment mechanism would need to span organizations to support this resource sharing scenario.

VIII. CONCLUSION

EUC provides opportunities for users to quickly develop applications with limited IS knowledge. The concepts of EUC apply to varying computing platforms such as personal computers, data warehouses, the Internet, and mobile computing. EUC concepts are consistent for these different computing platforms but the specific benefits of, and risks associated with, EUC differ with each of the technologies. Also, the trade-off between the ease of application development and the required organizational support varies under the different platforms. Still, EUC provides a computing environment to effectively meet user needs when sufficiently supported by IS professionals and other necessary organizational departments.

SEE ALSO THE FOLLOWING ARTICLES

Data Mining • Data Warehousing and Data Marts • Electronic Commerce, Infrastructure for • End-User Computing, Managing • End-User Computing Tools • Internet, Overview • On-Line Analytical Processing • Prototyping

BIBLIOGRAPHY

Edberg, D. T., and Bowman, B. J. (1996). User developed applications. *Journal of Management Information Systems,* Vol. 13, No. 1, 167–185.

Ferraro, A. (1999). Electronic commerce: The issues and challenges to creating trust and a positive image in consumer sales in the World Wide Web. Available at http://www.firstmonday.dk.

Gardner, S. R. (1998). Building the data warehouse. *Communications of the ACM,* Vol. 41, No. 9, 52–60.

Gurbaxani, V., and Kemerer, C. F. (1990). An agency theory view of the management of information systems. *Proc. International Conference on Information Systems,* 279–288.

McLean, E. R., Kappelman, L. A., and Thompson, J. P. (1993). Converging end-user and corporate computing. *Communications of the ACM,* Vol. 36, No. 12, 79–92.

McLeod, R., and Schell, G. (2001). *Management information systems.* Upper Saddle River, NJ: Prentice Hall.

Nielson, J. (2000). *Designing web usability: The practice of simplicity.* Indianapolis, IN: New Riders Publishing.

Regan, E. A., and O'Connor, B. N. (1996). *End user information systems: Perspectives for managers and information system professionals.* New York: Macmillan.

Rogers, E. M. (1995). *Diffusion of innovations,* 4th ed. New York: The Free Press.

End-User Computing, Managing

Eduardo Gelbstein
International Computing Centre, United Nations

GLOSSARY

end user computing Using a computer at the application level. The term end-user is used to distinguish the person for whom the application was designed from the person who programs, services, or installs the product. End-user computing may involve the design of macros, spreadsheet formulae, and simple databases by individuals who are not professional computer programmers.

maintainability The degree to which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. Maintainability is increased when formal development methodologies are followed and when detailed documentation is produced.

scalability The ability of an end-user computer application to continue to function well as it is changed in size or volume in order to meet an organization's need. Typically, such rescaling is to a larger size or volume.

service level agreements (SLA) A contract between a service provider and a customer or group of customers. This contract specifies what services the service provider will furnish, how their performance will be measured, and performance targets, as well as the responsibilities of all parties involved in this contract.

MANAGING A MULTIUSER NETWORK of workstations and personal computers in a corporate environment presents several challenges. End-user computing was born out of the PC revolution. For the first time end users were empowered to develop their own computing solutions. However, in the beginning they were limited by the power of the early PCs, which could do little more than act as glorified typewriters and calculators. Consequently, business-critical activities remained on centrally managed hosts. The power of the PC has grown to the extent that sophisticated and powerful systems can be implemented on them. It is this ability that raises the greatest concerns among corporate management. The response has in general been balanced—exerting control to ensure that key databases are centrally managed and secured while permitting end users to analyze and manipulate data using their newly developed skills.

This balance must also reflect the cost of providing facilities and a robust enterprise-wide computing environment. This results in the need to maintain the continuous availability of networks, servers, and desktop computers and their peripherals; to deliver quality of service and value for money accepted as enough by both users and executives; to ensure the security of all data and information; to support end users as and when required, etc.

A proven way to achieve these objectives is that of: establishing a framework for end-user computing defined in clearly formulated and communicated *policies,* adopting well-documented and tested systematic *processes* for all operational activities, and treating major changes to the end-user computing environment as *projects.* This article focuses on these three topics.

I. BACKGROUND AND SCOPE

The number of computer users in an organization has grown from a few finance and personnel people

in the 1950s to a very large population. Many organizations now have inventories of personal computers larger than the number of employees, arising from policies where employees are also provided with home and portable computing equipment.

The stand-alone personal computer of the 1980s gave way to those connected to local-area networks designed to share files and printers. These were the early days of end-user computing and at that time it was believed that this was going to end the tyranny of the traditional data center or "glasshouse" IS organization that operated corporate and departmental systems.

As computing power is integrated into business processes, it will become increasingly critical to safeguard the integrity of data security and continuity of operations. Since the mid-1990s:

- The ubiquity of end-user computing, the rapid adoption of Internet technologies (Internet access, corporate intranets), the deployment of client–server applications, the adoption of wide-area networks, the use of e-mail, etc. all combined to create a well-connected enterprise.
- This in turn, presents a problem of scale: It is much simpler to manage a local-area network with 15 personal computers than an enterprise network of several thousand computers.
- There is an added challenge of growing power: UNIX and NT servers now have the level of complexity of the mainframe (Windows NT consists of more lines of code than the IBM mainframe operating system MVS/ESA), running multiple applications on several partitions, but without the management tools available for mainframes.
- End-user computing is now expected to have availability targets no different from those of the traditional data center, i.e. higher than 99.9% (this implies a downtime of less than 9 hours a year), have a very high degree of security, and operate 7 days a week, 24 hours a day.
- The accepted average expenditure needed to provide a fully functional, networked, and supported personal computer in 1999 was $10,000 U.S. per year, that is, a visible and substantial cost that continues to rise.

It is recognized that the lessons learned in the operations of large networks and large computer systems typical of the glasshouse can and should be applied to the networked environment of end-user computing.

This article focuses on three aspects of end-user computing in a multiuser environment: *policies* that define the end-user computing environment, *processes* that enable it to be operated and supported to an appropriate level of service on a continuous basis, and *projects* through which major changes to infrastructure, hardware, and software are implemented and rolled out to the end-user community.

II. END-USER COMPUTING

There appears to be no clear definition of the scope of end-user computing. Its origins are closely associated with the early models of personal computers, targeted at enthusiasts with simple operating systems (such as CP/M) and limited software, which obliged users to write "code" to perform useful functions using languages such as Basic, C, Pascal, and others.

The introduction of spreadsheets in the late 1970s allowed a new type of programming, namely, the user-defined mathematical and statistical formulas and the recording of keystrokes to create macros. These permitted end users with relatively little knowledge of information technology and, in particular, of applications development, to become self-sufficient in a wide range of tasks.

Subsequent products, such as databases, some of which also had compilers, which allowed an application to be created, compiled, and then distributed without having to have a copy of the original software, also became very popular and end users started to create increasingly more sophisticated applications. At this time, networked PCs were still fairly uncommon and where local-area networks existed, their functionality was limited to file and print sharing. Nevertheless, many utilities and applications designed by end users started to be copied on floppy disks and distributed (as did unlicensed copies of software).

It was then that management, initially information technology organizations that were mere spectators of these developments, became aware that there was a problem. The software distributed in this way was undocumented, of doubtful quality, and it occupied a growing amount of the working hours of the employees. Unlicensed software created legal exposures at the corporate level and it became clear that a multiuser networked environment needed to be managed in a different way.

The tools for end-user computing continued to become more sophisticated and by the 1990s various classes of "users" had emerged:

- The person capable of creating a simple macro through keystroke recording using the tools provided with integrated office suites of programs, introducing some animation in a slide

presentation or creating a family of linked spreadsheets
- The person capable of developing complex database queries using Structured Query Language and similar tools, writing programs to customize the way office suites are used with programming languages or vendor-supported development environments
- The person capable of creating linked HTML documents, Internet and intranet Web sites, and other complex applications that include multiple documents and file formats.

Most organizations are now connected through networks, not just local ones but also global. Their performance is critical and there are strong demands for security, quality, and reliability to support environments such as Web sites, data warehouses, e-commerce, and intranets.

Organizations that have lived with end-user computing for some time have come to appreciate the need to return to many of the disciplines of the "traditional" development environment, such as systems development life-cycle and change management, quality assurance, version control, and well-maintained documentations. Without these the freedom and flexibility of end-user computing can create vulnerabilities in terms of, for example:

- GIGO applications (garbage in, garbage out)
- The maintainability of undocumented or poorly documented applications
- The scalability of end-user applications for which there is demand that they become used on a corporate-wide basis
- The ease of design of virus, worms, and other forms of malicious software and the ease with which these can spread through networks.

This article examines how these can be balanced with the continuing need for end users to be able to use products and skills to maximize the benefits of the vast amounts of information available within a corporate environment.

III. POLICIES

Attempting to manage any, but particularly a large-scale, end-user computing environment without governance is a recipe for anarchy. Policies are the mechanism for defining governance. Policies must be compatible with the business needs and culture of an organization. Implementing and disseminating policies, monitoring compliance, and taking corrective action are management issues outside the scope of this article.

Although the range of policies needed for effective governance and the importance bestowed on them will vary from one organization to another, the policies discussed in the following subsections are in common use.

A. Policies on the Degree of Decentralization

The three main activities that define how an organization uses and manages information systems and technology are (1) visioning, (2) innovation, and (3) development and operations. The focus of this article is on operations.

From a corporate perspective, the functions of technology assessment, innovation, development, and visioning—defining how the organization proposes to use information systems and technology or operate in the Information Age—deliver higher value when there is a substantial degree of central management. The degree to which operations are decentralized (to a department, business unit, geographical region, etc.) will be defined by the management culture of the organization and its chief information officer. Decisions on how this work will be sourced, that is, by employees of the organization or by an external service provider (ESP), are discussed later in this article.

The greater the degree of decentralization, the smaller the potential economies of scale and the higher the risk of operating practices and standards diverging across the organization, as well as potential for duplication of development efforts and incompatibilities. On the other hand, decentralization enables flexibility and responsiveness to meet the needs of the individual departments or business units.

B. Policies on Technical Standards

In his book *The Politics of Information Management,* Paul Strassman states "Standards are a compromise between chaos and stagnation." End-user computing has become an enterprise-wide activity. A minimum set of standards is needed to ensure the following:

- Data are captured and entered only once.
- Documents can be exchanged in electronic form.
- Data (e.g., accounting) can be aggregated without intermediate processing.

- Technical diversity (the number of different technologies, software versions, etc.) is minimized to reduce complexity.
- Suppliers, clients, and other interested parties can exchange information in the form of e-mail, invoices, orders, etc. in the simplest possible manner.
- Economies of scale in the form of improved terms and conditions of contract are gained.
- Operations remain a manageable task—diversity increases complexity.

Given the short product cycles associated with information technology products, defining and deploying standards becomes a frequent activity and those responsible must exercise sound judgment in the selection process and in its timing. They also need to have the political skills required to ensure that these standards are adopted across the organization and the project management skills to ensure that the rollout of new products and facilities is successful.

C. Policies on the Use of Enterprise Resources

Since employees have access to telephones, fax machines, photocopiers, e-mail systems, the Internet, enterprise data and information, documents, and equipment on loan such as cell phones, notebook computers, and home computing facilities, it is necessary to define what represents acceptable personal use of these facilities.

These policies need to describe how the organization deals with situations where the policies were not followed. Implementing these policies requires specialized tools for monitoring and recording activity, both in terms of usage, for example, who is accessing what Web sites and the duration of each session, in addition to financial, such as the personal use of a cell phone. Additional policies may be needed to cater for situations such as the loss or theft of equipment, for example, a notebook computer containing sensitive or confidential information.

D. Policies on Privacy and Confidentiality

In certain situations an organization may need to access an end user's workstation, for instance, with tools that support remote diagnostics, monitor phone calls and e-mail, and audit the software in a personal computer. In these circumstances, those responsible for end-user computing management need to ensure that adequate provision exists to advise employees of how compliance with policies will be operated and how this relates to their legal and statutory rights with regard to privacy and confidentiality, defined through national legislation.

Similarly, employees need to respect the confidentiality of the organization's data, information, and documents and appropriate policies need to be in place.

E. Security Policies

In the networked environment, security has become a major issue. Enterprises seek to protect their internal networks by the use of firewalls and related technology. However this substantial investment can be totally negated by the deliberate or inadvertent use of, for example, a dial-up connection to another network.

An organization should have a data protection policy. This is necessary to comply with national and International legislation (e.g., European Union, EU). It is easy for an end user to create an extract from a corporate data file and then break the law by using it for a purpose other than that for which the data subject's permission was obtained. A further problem occurs if data are taken across a national boundary to a country that does not meet EU data protection criteria, as is currently the case with the United Sates. These issues must be addressed through appropriate awareness programs and staff training.

Information security covers three main domains:

1. Availability of information and information systems
2. Integrity of information and data
3. Confidentiality of information and data.

Individual policies are, therefore, needed to cover such topics as:

- Backup and restore, disaster recovery, and contingency planning
- Hacking and malicious software (denial of service, virus, worm, trap door, Trojan horse, etc.)
- Access rights to classified information and data (user identification, authentication, etc.)
- Software configuration and installation
- Providing access to vendors and maintenance organizations to support remote diagnostics of, for example, servers and enterprise storage systems
- Remote access to information.

These security policies would normally include a description of the procedures for testing security, methodologies for conducting regular security reviews, and the actions to be taken in the event of a security breach (investigation, strengthening security arrangements, disciplinary action, etc.).

F. Policies on Document and Data Retention

The growing number of documents available in electronic form has changed the nature of archiving. Policies for document and data retention need to take into account legal and statutory frameworks and requirements. A major concern is the use of the local hard disks on a laptop or desktop computer to hold key enterprise data. Many surveys (including those by Gartner Group) have shown that no less than half of all business documents are held in this way. At a time when effective knowledge management is seen as a key differentiator, the lack of a coherent and systematic mechanism to store and share such documents becomes a major liability.

From a technical perspective, it would be relatively simple to keep an electronic archive of all documents. However, these would need to be indexed so as to enable reliable retrieval. Without this capability, archives are as good as useless.

Such policies also require taking into account that changes in technology need to be reflected in the way data are archived. For example, by the year 2000, it has become impossible to read an 8-in. floppy disk formatted under the operating system CP/M. In the case of large systems, there are terabytes of unreadable data because the applications that generated this data no longer exist or because new versions of software cannot read data generated by older versions.

G. Policies on Sourcing Options

Many organizations rely on their own employees to carry out the operational and support tasks of end-user computing, while others have chosen to outsource. The outsourcing of operational work has created the highly competitive industry of external service providers (ESPs). In an outsourcing contract, the ESP commits to deliver defined levels of service against payments. Typical contract lengths are in excess of 3 years. ESPs have legitimized the concept of payment for services and created reliable benchmarks for the total cost of providing a fully functional and supported workstation to an end user.

The organization outsourcing these services divests the responsibilities of funding capital investments, recruiting and training employees, defining technology choices and configurations, operating procedures, and so on, to the ESP. An important consequence of outsourcing is the loss to the outsourcing organization of the knowledge on how operating processes are carried out.

Other issues to consider in relation to outsourcing include:

- Loss of control over employee appointments and a potential adverse impact on the organization's security policies. On the other hand, a strong reason for outsourcing in the 21st century is the difficulty of recruiting and retaining suitably skilled staff.
- The cost of changes to requirements as well as the initial terms and conditions of an ESP contract are usually very favourable. Changes to the original specification, however, are subject to separate negotiations and the ESP may seek premium pricing. Changing ESPs is a complex and costly undertaking.
- A small organization dealing with a large ESP may find itself disadvantaged at a time when resources are scarce, because a large client may have greater negotiating leverage and thus be able to obtain preferential treatment.
- The "cost of sale", i.e., the costs incurred by ESPs in order to respond to a request for proposals and the subsequent contractual negotiations, which can be substantial, are invariably included in ESP fees. Small organizations are more likely to be adversely affected by this than large ones.

H. Funding

As stated in the introduction to this article, the total cost of providing a networked personal computer system is significant—several hundred million dollars a year for a large multinational company. Policies on the financial responsibilities of end-user departments relate to those on the degree of decentralization and are independent of whether operations and support are outsourced or carried out by in-house resources.

Funding end-user computing can be included as part of the corporate overhead since it is simple to administer, costs are transparent to top-level management, and end users are not burdened with establishing such costs. This approach is most successful when the governance arrangements allow for any special requirements of end users or business units to be identified and addressed.

Alternatively, a charge for each user connected to a network or using a help desk can be raised. Here, individual user groups or business units make a contribution that represents the cost of providing these services. These groups can, where appropriate, define different service levels representing specific needs, such as the requirement for technical support 7 days a week instead of 5.

The charging methodology used must be simple to understand by both management and end users.

I. Service-Level Agreements

When the operations and support of end-user computing are outsourced, contracts invariably include the following:

- A detailed description of the services and facilities provided
- The level of performance and security
- The procedures to follow if performance targets are not met.

These and often other definitions constitute a service-level agreement (SLA). SLAs have been found to be valuable insofar as they:

- Document the quality of the services provided in quantitative terms
- Define the responsibilities of all the parties involved in end-user computing
- Provide mechanisms for performance and service providers and service users.
- Greatly improve communications between service providers and service users.

For these reasons, many organizations that have not outsourced these operations have also implemented SLAs with their in-house service teams. Here, the policies need to define how failure to consistently meet the SLA will be dealt with.

J. Audits and Benchmarking Policies

Independent technical reviews and measurements to comparative benchmarks are often used to safeguard the interests of an organization in terms of best practices, policy compliance, and value for money. Policies in this area should define the frequency with which technical reviews are performed, including whether this will be done within the organization or by an independent third party. The same applies to benchmarks, which must be carried out on a like-for-like basis if valid conclusions are to be drawn.

IV. PROCESSES

Policies create a framework within which day-to-day operations and support activities are carried out. As stated earlier, the end-user computing setting is becoming increasingly complex as both the number of end users and devices connected to networks increases. Moreover, the complexity of both hardware and software in servers, routers, switches, and storage subsystems is increasing at the same time.

Increasing complexity may lead to a loss of manageability. Processes are mechanisms that enable the delivery of consistent, robust services. These processes need to be documented, explained, tested, and carried out in a systematic way. The principles of total quality management (TQM) are relevant to process design, improvement, and documentation. Clearly, training on the application of these processes and their monitoring and periodic review are essential for a successful and systematic approach to operations.

Processes are divided in five distinct categories:

1. Day-to-day operations
2. Introduction of changes
3. Technical and end-user support
4. Business planning
5. Communications.

The documentation of each process becomes, in practice, a detailed guide to *how* and, where appropriate, *when* each particular activity should be performed. This documentation needs to be kept up to date as changes in technology and the processes themselves are introduced.

A. Processes for Day-to-Day Operations

In an ideal world, the operation of information systems would be invisible:

- All systems and facilities would operate to an availability of 99.999% or better (a downtime of 5 minutes a year) and do so with absolute security.
- Changes would take place without disturbing the end user or disrupting the continuity of operations.
- Information systems would continue to operate and the work of the organization would go on without pause regardless of any disaster that may occur.

- In the event of failures, all systems and applications could be restored to operational status immediately.

These targets are essentially attainable, but not easily or cheaply: Technologies need to be configured for fault tolerance, diagnostic tools have to be in place, all employees must be knowledgeable, processes must be robust, and effective management structures must be in place. The main processes relating to day-to-day operations are discussed next.

1. Availability and Performance Management

The focus of *availability management* is the monitoring of all servers, storage devices, routers, switches, network links, power distribution, and other equipment for hardware faults and software errors. It is assumed that the design and configuration of the hardware, network, and other infrastructure have given due consideration to techniques such as clustering redundancy and other fault-tolerant measures as may be appropriate to meet availability targets. Software errors may arise from inappropriate configuration parameters required for the interoperation of various devices, from buggy software (applications designed in-house or from a third party vendor), from operating systems, from software installed or downloaded by end users, etc.

Performance management looks for the causes of degraded response time, latency or indicators of insufficient processor power or poorly designed applications, insufficient network bandwidth, excessive input/output transactions, inappropriate database design, incorrect query design, software loops, etc. All instances of degraded availability or performance should be treated as an operational incident (see later discussion of support processes). Whenever practical, diagnostic systems capable of identifying a potential problem before it manifests itself to the end user should be used to reduce operational disruptions.

All actions taken to restore normal operations should be recorded and be subsequently reviewed. For large-scale operations, best practices have shown that a daily review of operating incidents by the operations manager and his or her team is effective.

2. Operations Staffing

A robust operation requires motivated and capable employees to perform routine work, monitor events, and take appropriate actions. Continuous operations 7 days a week, 24 hours a day, require several people to be involved to cover shifts, vacation periods, absences due to training, sickness, etc.

The roles and responsibilities of every individual associated with these operations need to be defined in detail and be supported by appropriate documentation on the processes to be carried out. An appropriate level of training and supervision is also required. The above must be complemented by suitable escalation rules should a problem require on-call staff to intervene and, when appropriate, senior management to be advised.

3. Security

Information and data are valuable organizational assets and their integrity and confidentiality must be adequately protected against loss, unauthorized access, or modification. Important aspects of security are discussed next.

a. Physical Security Measures and Practices

Physical security addresses the environmental needs to house and operate information systems, networks, and related facilities. Good security starts with the design of the facility itself, which should take into account fire resistance and fire fighting facilities (sprinklers, inert gas, etc.) and the avoidance of physical hazards (obstructions, changes in floor level, cables on the floor, etc.).

Best practices also include providing a resilient power supply, which would typically involve installing battery backup and standby power generators and preventing electrical surges. Good security processes and practices limit physical access to the data center through the use of access control systems, CCTV, etc. In the data center itself, security measures should include safekeeping of manuals and operating procedures as well as tightly managed change control practices.

In addition, a culture in which security awareness among the employees is complemented by training on actions to be taken in an emergency and the use of appliances such as extinguishers should be encouraged and systematically applied.

b. Logical Security

The scope of logical security has grown to cover all of the following topics:

- Resource access controls (operating system, configuration tables, application software, password files, operations employees, end-user registration, definition of rights)
- Remote access for problem solving (employees and vendors)

- Management and configuration of firewalls and other security devices
- Password rules and password changes
- Malicious software (applications)
- Malicious software virus, worm, attachments, scanning, actions.

Logical security involves those responsible for the operation of information systems, those designing them, those using them, and also those people outside the organization. As a result, managing logical security has acquired great importance. The successful performance of this function requires, as a minimum:

- A judicious choice of (1) policies and procedures to assign access to resources and (2) tools to analyze and monitor events that may impact security
- Appropriate rules and technologies to ensure the proper identification and authentication of those seeking access to resources
- Best practices (e.g., so that the password to access a root directory is not "admin")
- Tools to provide resource access control, authentication, and so on
- Tools to protect confidential information (e.g., encryption)
- Tools to detect malicious software such as viruses and worms
- Periodic reviews to identify if shortcuts or other mechanisms to bypass resource access controls (trap doors) or other malicious software have been embedded in an application (time bomb)
- Periodic reviews to ensure that user access rights to data are consistent with job responsibilities and security policies.

In addition to servers, networks, and operating systems, it is necessary to take special security precautions with stored data and information, in particular, to ensure that complete copies of important data are kept in the form of backups and archives and that these can be used to recover the information in the case of loss or corruption of the primary source. From a security point of view, it is also essential to ensure that the physical security of the site where copies of the primary data are kept is adequate. For sensitive data it may be necessary to encrypt it before it is transported to a second site and during its temporary storage at this site.

c. Business Continuity

For most, if not all, organizations the ability to continue their activities—at least to some degree—in the event of a disaster is essential. The concept of business continuity is an umbrella that includes disaster recovery, contingency plans, and crisis management.

Disaster recovery consists of being able to recreate key information systems and their related communications, (usually) at another physical location, in a period of time defined by the criticality of the activities to the organization. Such time frames could, for example, range from less than an hour to several days. The shorter the recovery time frame, the higher the cost and complexity of the arrangements because a very rapid recovery can only be achieved if a replica of all systems and facilities is continuously operated as a hot standby.

Contingency plans describe the recovery priorities, the processes through which people will be advised of their changed responsibilities, location, etc., the release of information to the public and to the press, and the process for returning to the normal status. Recovery priorities are determined by the outcome of a business impact analysis that needs to be kept up to date. Contingency plans also need to be tested and reviewed periodically to validate their effectiveness. Crisis management is not related to the operation of information systems and deals with communications when the disruption to an organization has become highly visible.

B. Introduction of Changes

Although this section does not specifically mention end-user computing per se, this is one area where end-user computing is found to be weak due to the lack of awareness or skill in these methodologies and best practices.

1. Configuration Management, Software, and Change Control

The essence of *configuration management* is to ensure that items of equipment from different vendors and for different purposes interoperate in a seamless manner. It also includes the means to provide identification, control, status accounting, and verification of the components of the IT infrastructure and related assets. Processes in this category include inventory management, the coordination and follow-up of maintenance contracts, and the definition and control of access rights to vendors and maintenance companies, etc.

The process of capacity management is designed to monitor and tune existing service capacity to ensure that optimum use is made of the available re-

sources and that sufficient capacity is available to deliver an agreed service level. It also examines the best ways and timing for providing additional capacity.

Fault tolerance and resilience are also associated with configuration management and are designed to provide levels of hardware, software, and networking redundancy compatible with the service levels to be delivered.

This is a topic where potential conflict could arise. The way to simplify the task of support staff—and hence to reduce costs—is to set tight standards and enforce them rigorously. This is in contrast to the objectives of the end user who seeks maximum flexibility to exploit the technology as he or she sees fit. There is no easy solution to this—a compromise will always be required and will take time and effort to evolve.

2. Software Control

Software control is the process of the physical storage in a definitive software library of all software to ensure that only correctly released, licensed, and authorized versions of software are in use. Processes in this category include the tracking of releases, patches, service packs, and bug fixes. Software distribution is an additional process through which all software from the definitive software library is disseminated to authorized users to ensure that only correctly released and licensed versions of software are in use. This is often the single most expensive activity to a support organization.

3. Change Control

Change control is the process through which the initiation, implementation, and review of all proposed changes to the operational IT infrastructure and facilities are controlled to avoid disruption to operations.

The processes described in Sections IV.B.1 through IV.B.3 should be treated as critical to operations management. Many best practices have been documented and published.

4. Periodic Review

Reviews fall into various categories. Of particular relevance to end-user computing are technical reviews covering security policies and compliance, best practices in operations and the use of the computing facilities by end users, and compliance with the terms of software licences.

Security reviews can also focus on identifying vulnerabilities to both internal and external threats and testing the security arrangements currently in place. Internal threats arise from dishonest employees whose objective is to defraud the organization. This can be done by abusing legitimate access rights or by obtaining other means of system access (stolen or copied passwords, trap doors, Trojan horse software, etc.). Disgruntled employees, on the other hand, have as an objective sabotage or damage to the organization. This can involve systems access, malicious software, or physical damage. It is widely acknowledged that internal threats represent the most serious type of security exposure.

External threats have grown as a result of global connectivity. These threats include access to confidential information, the corruption or deletion of data or information, the injection of malicious software into an organization, and denial of service.

C. Support

1. Incident Management

The incident management process is designed to ensure quick recovery of any degradation to services. The help desk is the single point of contact for all users and records all incidents reported to it in an appropriate system.

2. Problem Management

The objective of problem management processes is to reduce the number of incidents by structurally removing causes of (systematic) error in the IS infrastructure. Typically, this category includes separate processes for problem control including escalation and management information systems as well as compliance with applications development standards and best practices.

For substantial operations, the work of the help desk needs to be supplemented with a robust problem management system that could be integrated with change management. These systems are used to record all incidents, to whom they were assigned, and with what level of priority. Problems are subsequently tracked until their closure. These kinds of systems usually include databases of all users, inventories of all equipment, including their physical location and a problem history for each of them. The solutions to all incidents and problems are also recorded and analyzed in order to identify nonrandom occurrences. Such a system can also provide a “knowledge bank” with proven solutions to recurrent problems of a nonsystematic nature (e.g., known difficulties with software packages). High priority problems that remain

open longer than a predefined threshold time are automatically escalated to technical support managers.

Management reports from these systems can be used in the daily review of operating incidents, to identify recurrent problems or callers, and to obtain measurements of overall service level. Management information obtained from a problem management system can be used to support the analysis of training needs for both technical personnel and for end users. Frequent inquiries as to how to perform certain tasks are a good indicator that training would be beneficial or that existing training programs are ineffective.

Another responsibility of the help desk or call center is that of end-user administration and management of all moves, additions, and changes (MACs). The tasks covered by end-user administration include the creation of:

- Log-on identities and often an initial password
- Definition of software profiles
- Definition of resource access rights
- Creation of e-mail accounts
- Assignment of a network address, telephone number, etc
- Creating a record of all of the above in the problem management system
- Updating the appropriate directories (phone, e-mail, etc.)

and the subsequent maintenance of the above as employees change offices, user profiles, departments, etc. Similarly, all access rights should be removed in a timely manner when an employee leaves the organization. The question as to whether the help desk should also be responsible for installation work and repairs to simple problems (e.g., replacing the toner in a printer) is a matter for assessment under each set of circumstances.

D. Business Planning

Because of space limitations, only some business planning processes are covered in this article.

In the rapidly changing world of information systems, there is a constant need to track innovations, spot opportunities to develop new facilities, improve quality of service, and reduce cost. *Technology assessment,* product selection, preimplementation testing, the use of advisory services and other sources of expertise are all part of this process. Investment in technologies, including those required for future infrastructure and capacity requirements, needs to be supported by a robust business case, usually as part of a *business plan* for information systems and of a *portfolio of information system assets.*

The development of detailed business continuity, disaster recovery, and contingency plans needs to be shared between the IS function and individual business units or departments. The involvement of senior executives in this activity is of great importance.

Relationship management is more than a process, it is a culture: Although traditionally IS organizations have been regarded as inward looking and user hostile, the profound dependence of organizations on information systems has increasingly demanded that IS organizations, regardless of whether they are in-house or outsourced, operate as a service provider. This role implies that many relationships must be maintained, some contractual and some of a more social nature: with vendors and other service providers, with maintenance organizations, with senior management, and with end users (see next section).

E. Communications

What should those responsible for end-user computing management tell the end users? "Not much" is always the wrong answer. End users, particularly those active in end-user computing, are rapidly becoming literate in information systems (but not necessarily as knowledgeable as they may think). Good communications are essential to create a "caring environment" and avoid misunderstandings. The following list should be taken as a starting point:

- Planned changes and upgrades: purpose, timing, potential impact
- Completed changes and upgrades: successes, problems remaining
- Training programs and other information events: content, date, time, location
- Explanations of the reasons for current and recent performance problems
- Notifications and alerts concerning malicious software
- Who should be contacted when there are problems
- Who should be contacted about special needs (e.g., weekend work).

Multinational and international organizations dealing with multiple locations and cultures face a consid-

erably greater challenge to ensure clarity of communications, compliance with policies and coordination. This becomes critical when dealing with large IT projects. There is a large collection of humor dealing with computer problems and IT support organizations that clearly confirms that "caring environments" and effective communications are *not* the norm.

V. PROJECTS

The change control processes described above apply to single changes made in an operational environment. Larger changes, such as the rollout of an application (which could have started life as a modest end-user computing project), major infrastructure upgrades, large-scale hardware and/or software changes, or the introduction of new facilities and systems, have a much greater probability of success when treated as formal projects. In a large organization, several hundred or thousands of employees will be affected by the rollout of such projects.

A formal project contains, as a minimum, the following components:

- A sponsor who has approved the business case for the project
- A full-time project manager
- A detailed project plan describing in sufficient detail activities, dependencies, resources, dates, and a critical path analysis
- A communications plan and format to inform the end-user community
- A mechanism to provide for representation of the end-user community at various stages of the project
- Training plans for end users.

Any large-scale end-user computing project also requires that attention be given to the topics discussed in the following subsections.

A. Design for Simplicity and Standards

For example, complex desktop operating systems such as Windows NT need to be configured in a systematic manner in all the personal computers to ensure that subsequent technical support is dealing with a homogeneous environment. Software distribution tools can be used to advantage to control end-user environments and ensure compliance with the terms and conditions of software licences.

B. Distribution Logistics

Any project involving the more or less simultaneous deliveries of various new equipment needs to give attention to the facilities for storing boxes from vendors, unpacking, the removal of packaging material, installation of equipment, ensuring inventory labels correspond to the (updated) inventory database, removal of equipment no longer required, disposals, etc. Work involving lifting floor panels and/or electrical supplies constitutes a hazard to employees and adequate safety measures need to be in place.

C. Off-Hours Working

Major rollouts involving hardware and software are highly disruptive activities. Most of the installation work is best undertaken outside normal working hours and this, in turn, is often subcontracted to implementation technicians. The schedule and procedures for validating and accepting work done need to be specified in the contract with the implementers.

D. Heartbeat

Because any major project of this kind will require from several days to several months to be completed, the project plan needs to define the rhythm or heartbeat with which new facilities will be rolled out to groups of end users and the priorities assigned to these groups. The staged rollout presents several challenges:

- The organizational politics of groups wishing to move up the priority table to be the "first" to have the new facilities or systems
- The potential lack of compatibility between documents or data created by the two families of facilities or systems that have to coexist for the duration of the project
- The need to convert old documents and data to new formats
- The need for the support organization to deal with a heterogeneous environment of the old and the new and to supply additional temporary resources to deal with the complexity of the transitional situation and the added workload of users as yet unfamiliar with the new facilities or systems
- The need to synchronize training programs with installation work.

E. Postimplementation Review

It is good practice to carry out a rigorous review of any major project as soon as it has been completed. The review should include, in particular:

- The identification of those activities that were implemented as planned and, where appropriate, the identification of people who performed particularly well
- The identification of those activities that "could have been done better" and the lessons to be learned as a result
- The study of all known problems encountered during the project and the reasons they occured.

This review should also identify the corrective actions taken and their impact on the overall project.

The postimplementation review team should include representatives of the end-user computing community.

SEE ALSO THE FOLLOWING ARTICLES

Data, Information, and Knowledge • Desktop Publishing • Electronic Mail • End-User Computing Concepts • End-User Computing Tools • Firewalls • Outsourcing • Security Issues and Measures • Spreadsheets • Strategic Planning, Use of Information Systems for

BIBLIOGRAPHY

Information Technology Infrastructure Library (ITIL). London: U.K. Stationery Office. To order: http://www.thestationeryoffice.com. For more information on ITIL: http://www.ccta.gov.uk/services.itil.htm.

Lewis. L. (1999). *Service level management of enterprise networks.* Norwood, MA: Artech House.

Page, S. (1998). *Establishing a system of policies and procedures.* Published by Steven B. Page, ISBN 1929065000.

Strassman, P. (1995). *The policy of information management.* New Canaan, CT. Information Economics Press.

Strassman, P. (1998). *The squandered computer.* New Canaan, CT. Information Economics Press.

Sturm, R., Morris, W., and Jander, M. (2000). *Foundations of service level management.* New Canaan, CT. SAMS.

End-User Computing Tools

Jose Stigliano and Marco Bruni
International Fund for Agricultural Development, United Nations

GLOSSARY

application development environment An integrated set of tools that supports the development of applications typically includes a language editor, an interpreter, a compiler, a debugger, program libraries, and reusable components.
application generator Software that generates application programs without formal programming.
fourth-generation language (4GL) A specification language, often nonprocedural, that specifies the results that must be obtained rather than the precise steps that must be followed.
office productivity tools Software typically used in an office environment such as word processors, spreadsheets, personal databases, and presentation graphics.
third-generation language (3GL) A high-level (conventional) programming language such as BASIC, C, and Java.
workgroup computing tools Software that supports the work of multiple users on common tasks.

I. SCOPE

End-user computing tools are a means to an end, as are any tools; the means are computer programs and the end is to solve end-user problems using available computing power. End-user computing tools encompass primarily office productivity and workgroup computing tools as well as application development tools used by individuals who are not information technology (IT) professionals.

In comparison with corporate or enterprise computing tools, including those used for professional software development, end-user computing tools tend to be more accessible to end-user organizations in terms of cost, installation, and training requirements and, for application development tools, in terms of level of programming knowledge required.

Programming is an issue of fundamental concern in end-user computing tools: Ideally, end users should be able to put computers to work on any task without having to do any programming. Office productivity and workgroup computing tools meet this objective; users can create applications for individual use and collaborate with others through a network without programming, although some advanced functions may require an understanding of basic programming notions.

Application development tools require knowledge of programming that varies from none for macro recorders and some application generators, through moderate for some fourth-generation languages (4GLs) and Web-based tools, to advanced for conventional programming languages. The need for programming usually arises when the end-user application is intended to support a multiuser task.

Given the learning curves associated with programming, most application development tools, especially those designed with the end user in mind, tend to focus on making programming easier. This is achieved through a number of techniques such as

visual interfaces, preprogrammed reusable components, program code generators, and development environments that take care of many technical implementation details.

A trade-off usually exists between ease of use and the power and flexibility of development tools: Easy tools suitable for self-training tend to offer a fairly limited range of solutions; powerful and flexible tools suitable for a wide range of solutions tend to have a significant learning curve.

This article focuses on three main categories of end-user computing tools: office productivity, workgroup computing, and application development tools. The main tools for office productivity and workgroup computing are discussed briefly because they are treated in detail in related articles. The limitations of macros are illustrated in the context of office productivity tools.

Application development tools are classified into four types for the purpose of this discussion: application generators, 4GLs, application development environments, and scripted Web pages. The most important characteristics, relative ease of use, power, and flexibility of each type of tool are discussed from an end-user perspective. The article concludes with a brief look at some trends and new developments in the area of end-user computing tools.

II. BACKGROUND

End-user computing tools were virtually unknown in organizations before the 1980s. With the diffusion of the personal computer (PC), the number of commercial applications available to end users began to grow at a rapid pace. Among these applications, office productivity packages such as word processing, spreadsheets, database management systems, and presentation graphics became ubiquitous end-user computing tools. These tools, designed to create applications for individual use, began to offer facilities for users to distribute their applications to other users inside and outside the organization.

In the 1990s, the growing use of local- and wide-area networks, and eventually the Internet, offered end users the possibility of working with computers in progressively larger groups. This led to the emergence of workgroup computing tools or *groupware,* some of which became virtually indistinguishable from office productivity tools. Early groupware products provided different models of collaborative computing based on proprietary solutions. Internet standards fostered the development of more "open" solutions, and the advent of the World Wide Web brought a significant simplification of the computer-based collaboration paradigm.

In spite of the growing number of commercial applications available for office productivity and workgroup computing, end users at some point invariably found the need to solve specific problems for which off-the-shelf solutions did not exist or were not suitable. The lack of suitable existing solutions, coupled with the application development backlog typical of IT organizations, motivated end users to use application development tools to build the missing tools (i.e., the computer programs) with which they could create the needed custom applications.

Taking advantage of conventional programming languages available in their computing environment such as BASIC on PCs and C on UNIX machines, some end users ventured into conventional programming. This practice, however, often added a significant burden to the users' work, forcing them to address a variety of unanticipated technical issues beyond the scope of the intended solution.

As a result, users tried solutions demanding less technical knowledge such as application generators and 4GLs. Application generators helped end users develop a limited range of database applications without conventional programming. Fourth-generation languages enabled users to develop more complex solutions without having to worry about many of the technical implementation details.

More recently, application development environments offering visual interfaces and reusable components made the power and versatility of third-generation languages (3GLs) more easily available to those end users willing to invest sufficient time and effort in understanding the necessary programming concepts and techniques.

As happened with groupware, the advent of the World Wide Web brought a conceptually simple but powerful environment in which end users could develop applications for use by an unlimited number of users inside and outside the organization. In this new environment, end-user applications became conceptually simpler, taking the form of "executable" documents capable of linking a variety of new and existing information resources.

III. OFFICE PRODUCTIVITY TOOLS

Office productivity tools aim at increasing the productivity of individuals by providing computer support for common office tasks. Originally designed for single-user applications, these tools increasingly provide support for more than one user. They include some of the most common end-user computing tools such as word processors, spreadsheets, personal data-

base management systems, and presentation graphics. These tools are available both as individual products as well as "suites" of products offering an integrated environment with common functionality and features across products. A common feature of office productivity tools is a macro recording function that allows users to automate the execution of repetitive tasks and extend the standard functionality of the tool in simple ways without programming.

Office productivity tools provide an automation tool known as a *macro* that allows users to store sequences of standard commands and execute them as a single command. Essentially a mechanism to save keystrokes, a macro can be seen as a tool for creating "custom" commands from standard ones. Using the macro facility requires no knowledge of programming: The user invokes a macro recording function to record a sequence of manual keyboard actions and mouse clicks and saves the recording with a name. The user can then play back the recording as needed by invoking the macro name.

A. Word Processors

Word processors are tools specifically designed to process textual information, that is, information consisting primarily of words in arbitrary arrangements called documents. Word processors typically read input entered by the user through the keyboard, process the text according to the commands given by the user, and create a file containing the user's application such as a letter, office memo, or report. Word processors support the task of writing, letting end users create, edit, store, search, and retrieve documents containing formatted text and graphics. This text, which has been produced with a word processor, provides an example of the formatting capabilities of the tool.

A variety of tasks can be automated using standard built-in functions: replacing a string of text throughout the document, generating a table of contents, or merging the text of a letter with a list of addressees for mailing purposes. These functions perform the relevant task in response to commands issued by the user. However, for tasks that need to be repeated often, issuing a command each time can be too time consuming; macros help automate the execution of repetitive tasks.

B. Spreadsheets

Electronic spreadsheets are tools specifically designed to process numeric information, allowing also some text manipulation and the construction of charts and graphs. Spreadsheets provide a user-programmable rectangular workspace consisting of rows and columns forming *cells* at their intersection that can contain user input and *formulas*. Formulas are a combination of column–row address identifiers, such as A1 and D9, and user-defined mathematical expressions consisting of arithmetical operators and predefined functions.

Formulas can be thought of as the "visual programming language" in which spreadsheets are programmed; they are used to relate cells to other cells and define the processing requirements of the user. Figure 1 provides a simple example of spreadsheet

	A	B	C	D
1	Item Description	Unit Price	Quantity	Total Amount ($)
2				
3				= B3*C3
4				= B4*C4
5				= B5*C5
6				= B6*C6
7				= B7*C7
8				= SUM (ABOVE)
9				

Figure 1 Example of spreadsheet formulas.

formulas. Figure 2 provides an example of the result of applying the formulas in Fig. 1 to some arbitrary user input.

Spreadsheets support the tasks of making calculations, automatically processing the formulas, and displaying the computed results in the corresponding cells; each spreadsheet is a reusable application capable of producing results as a function of user input. A key feature of spreadsheets is the ability to instantly recalculate all formulas so that users can immediately see the effect of changes in input values; this makes this tool particularly suitable for "what-if" analyses and goal seeking.

Most spreadsheets organize the workspace into individual *worksheets* and collections of related worksheets called *workbooks*. Users can create links that relate data across worksheets and workbooks and can include links to spreadsheet applications in applications created with other office productivity tools.

For example, a workbook can contain a consolidated worksheet with total sales in each region, which is linked to individual worksheets with details of sales, one for each region. A link to the consolidated worksheet, which may include a chart, could then be included in a word processing document that discusses market trends. Also, advanced spreadsheets can support the simultaneous use of workbooks by more than one user, merging changes made by different users and handling conflicting requests.

In general, spreadsheets provide end users with an intuitive computing environment, suitable for calculation-intensive applications, which can be programmed without conventional programming skills. In addition, macros can be used to automate the execution of a series of steps through a single command.

C. Personal Database Management Systems

Personal database management systems (DBMSs) are tools specifically designed to store, retrieve, and manage large amounts of data in both numeric and textual formats. DBMSs can be used by end users to manipulate lists of data as well as by other applications that need to store data for further processing. Virtually all DBMSs found in office productivity suites implement the so-called "relational" model of DBMS (RDBMS) in which data are stored as tables composed of rows and columns.

Personal RDBMSs, which are distinguished from the more robust "enterprise" RDBMSs, are typically implemented as a file containing the database that must be read entirely before users can access the tables. Enterprise RDBMSs are implemented as a set of specialized processes that handle multiple user connections, optimizing the response time and overall performance.

RDBMSs use Structured Query Language (SQL) to create and manage databases through queries. Although SQL is a database manipulation language in principle intended for end users, it is not a particularly friendly tool for nonprogrammers, especially when complex queries are involved.

	A	B	C	D
1	Item Description	Unit Price	Quantity	Total Amount ($)
2				
3	7" Frame – Metal (regular)	11.99	100	1199.00
4	7" Frame – Metal (special)	14.99	50	749.50
5	10" Frame – Wood (regular)	17.99	100	1799.00
6	10" Frame – Wood (special)	21.99	50	1099.50
7	15" Frame – Wood (special)	28.99	30	869.70
8				**5716.70**
9				

Figure 2 Example of input and calculated values.

```
SELECT      LAST_NAME, FIRST_NAME, AREA_CODE, PHONE
FROM        CUSTOMERS
WHERE       TOT_PURCHASE_AMT > #500#
AND           (AREA_CODE = #617#
            OR AREA_CODE = #508#)
ORDER BY    LAST_NAME, FIRST_NAME;
```

Figure 3 Example of SQL query.

SQL uses a *query-by-command* approach, in which the user manipulates data in the database using English-like statements such as `SELECT` and `ORDER BY` that are interpreted by the RDBMS. Figure 3 illustrates a simple SQL query that interrogates a customer database and lists in alphabetical order all the customers in the Boston area with total purchases of more than $500.

Most personal RDBMSs also offer a more user-friendly approach to database manipulation through visual query interfaces. Visual queries use an approach called *query-by-example* (QBE), where the user interrogates the database by illustrating the results to be obtained. The user supplies the example by filling in blanks in one or more screen forms.

Forms correspond to tables in the database that the user has selected for the query, and columns or cells in the form correspond to columns in a specific table. Specific values and relational operators such as "less than" and "greater than" indicate the data selection criteria and simple "yes/no" checkboxes indicate whether the corresponding value in a given column is to be displayed.

For example, a form field containing `>1000` tells the system that only rows with values greater than 1000 in the corresponding column are to be retrieved. Figure 4 illustrates how the query above would look using a tool that supports QBE; Fig. 5 illustrates a hypothetical result. In addition to query manipulation, visual interfaces provide forms to allow users to input and maintain data in the database. These forms, which are associated with the database tables, also provide basic data validation and formatting.

Advanced personal RDBMSs support both visual and SQL-based queries. Query generators combine the example and command approaches by generating the appropriate SQL statements behind the scenes from forms that the user visually fills in. Users can save the queries and then edit the SQL code for minor modifications. These RDBMSs also offer macros to automate repetitive work and a development environment in which the user can associate queries, forms, and online and printed reports to create database applications that can be distributed to other users.

FIELD	LAST_NAME	FIRST_NAME	TOT_PURCHASE_AMT	AREA_CODE	PHONE
TABLE	CUSTOMERS	CUSTOMERS	CUSTOMERS	CUSTOMERS	CUSTOMERS
SORT	ASCENDENT	ASCENDENT			
SHOW	YES	YES	NO	YES	YES
CRITERIA			>500.00	617	
OR				508	

Figure 4 QBE version of query in Fig. 3.

LAST_NAME	FIRST_NAME	AREA_CODE	PHONE
Brown	Patrick	617	23405050
Simon	Paul	508	33590384
Simon	Paul	508	33590385
Thompson	Marie	617	47638287
Tonen	Nigel	508	34636878

Figure 5 Hypothetical result of query in Fig. 4.

D. Presentation Graphics

Presentation graphics are tools specifically designed to create and manage business presentations containing text and graphics. These tools allow users to lay out a graphic workspace consisting of a series of "slides" or pages that make up a business presentation. A characteristic feature of presentation graphics tools is a variety of ready-to-use layouts that users customize through text-formatting and graphic-manipulation functions similar to those offered by word processors. To enhance the impact of presentations, these tools usually offer audio and visual effects such as sound playback and animation. Macros can be used to record animated scenes.

E. Limits of Recorded Macros

The advantage of macro recording is that user tasks can be automated without programming. On the one hand, end users need not be aware of the program code generated by the macro facility to be able to use it. On the other hand, by inspecting the code generated by the recording, users may be able to make simple changes to the recording and use it for other problems. The chief limitation of this approach is that these recordings can be used only with the specific configuration of the problem at hand at the time they were recorded. They cannot be automatically generalized for use with similar problems with different configurations.

For example, recording the action of replacing all occurrences of the symbol "$" with "¢" in a document can produce the Visual Basic code illustrated in Fig. 6. By inspecting the code and with no knowledge of programming, the user can easily adapt this recorded macro

```
Sub replace_all_$_with_¢()
'
' Macro recorded
'
    Selection.Find.ClearFormatting
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = "$"
        .Replacement.Text = "¢"
        .Forward = True
        .Wrap = wdFindAsk
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll
End Sub
```

Figure 6 Example of code generated by a macro recording.

to solve a similar problem such as replacing £ with $.

However, if the user needs to automate a slightly more involved task such as capitalizing the first word in every cell of every table present in a document, where the number of tables in any given document is arbitrary, the macro recorder will not help. If the user records the keystrokes resulting from manually performing this task on a given document, such a recording will only work properly in documents with an identical format (configuration), i.e. same number of tables, each with the same number of rows, and will fail with any other document configuration. More importantly, the code generated by the recording cannot be used as a basis for solving the problem at hand by simply modifying appropriate parameters.

Solving the problem of capitalization in the above example involves knowledge of programming concepts such as variable definition and assignment, iteration control and iteration block, as well as knowledge of the type of instructions available in a specific programming language. The Visual Basic code in Fig. 7 illustrates this. In general, as this example shows, many relatively simple automation tasks that involve an arbitrary number of iterations cannot be solved by using a macro recording facility.

IV. WORKGROUP COMPUTING TOOLS

The tools presented in the context of office productivity are used to develop applications that enhance the efficiency of individual users. Workgroup computing tools, also referred to as groupware, support the efforts of groups of users working on related tasks. Messaging systems and shared databases are at the heart of these tools.

Workgroup computing includes familiar end-user computing tools for communication, coordination, and group collaboration such as e-mail, group calendars, workflow systems, document management, and hypertext. Other important but less frequently used groupware applications include collaborative writing systems, shared whiteboards, and decision support systems.

Two widely used groupware products are Lotus Notes/Domino and Novell GroupWise. Notes/Domino is basically organized as a collection of shared and local databases using a common structure to store documents, applications and administrative information (e.g., access control list) and as a set of programs that manipulate the information in the databases. GroupWise is basically organized as a repository for all messages (e.g., documents, attachments, address lists) and a "post office" module or "agent" that routes messages, coordinates calendars, and allows information sharing between users.

While the basic messaging and collaboration functionality of Notes/Domino and GroupWise is similar, the former is a more complex product that offers a family of tools for application development and real-time collaboration tasks such as online meetings and shared applications and projects. For example, both products provide advanced e-mail including electronic discussions and newsgroups, calendaring/scheduling, automated workflow, and task and document management. However, Notes/Domino offers tools for the development of sophisticated workflow and document management systems that are beyond the scope of GroupWise.

A. Group Communication

Electronic mail (e-mail), a system that allows users to send and receive messages through a computer network, is the most common groupware application and a central component of several other groupware tools based on a messaging system. Most e-mail systems include features for forwarding and attaching files to messages, providing delivery receipts, and creating mailing groups. Advanced e-mail systems include "rule" processors capable of automatically sorting and processing messages.

Mailing lists and *newsgroups* are similar to e-mail systems but are intended for the distribution of messages among large groups of users. While mailing lists deliver messages as they become available, newsgroups only deliver messages after receiving an explicit user request. *Computer conferencing* and *electronic discussions* are also similar to (and in most cases based on) e-mail, but messages are posted to a conference or discussion group, rather than to individual addressees.

B. Group Coordination

Group calendars allow task and meeting scheduling and coordination among groups of users by sharing information on individual calendars. Typical features detect conflicting schedules and help establish schedules that are acceptable to all participants. As well as sharing information on scheduled appointments and locations of events, group calendars are also helpful for locating people in organizations.

Workflow systems allow electronic documents to be automatically routed and acted on throughout organizations on the basis of user-defined processes. These systems combine the rules governing the tasks to be performed with the roles of the various participants

```
Sub in_tables_first_word_in_cell_cap()

Dim intnumcolms, intnumrows, k, l, m As Long                          Variable definition
Dim t As Table                                                        Variable definition
Dim c As Cells                                                        Variable definition

For Each t In ActiveDocument.Tables                                   Iteration control

t.Select
   intnumcolms = t.Columns.Count                                      Variable assignment
   intnumrows = t.Rows.Count                                          Variable assignment

      For k = 1 To intnumcolms                                        Iteration control
      For l = 1 To intnumrows                                         Iteration control
            For Each c In t.Range.Cells                               Iteration control
            t.Cell(l, k).h

            Selection.MoveLeft Unit:=wdCharacter, Count:=1            Iteration block
            Selection.MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdExtend  Iteration block
            Selection.Range.Case = wdTitleWord                        Iteration block
            Selection.TypeText Text:="hi"                             Iteration block

            For m = 1 To ((intnumrows * intnumcolms) - 1)             Iteration control

               Selection.MoveRight Unit:=wdCell                       Iteration block
               Selection.MoveLeft Unit:=wdCharacter, Count:=1         Iteration block
               Selection.MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdExtend  Iteration block
               Selection.Range.Case = wdTitleWord                     Iteration block
               Selection.TypeText Text:="hi"                          Iteration block
            Next m
      Next l
      Next k
Next t
End Sub
```

Figure 7 Example of code that cannot be generated by a macro recording.

and coordinate the information required to support these tasks. Workflow systems may not physically "move" documents across a network; often the "flow" is simulated by providing coordinated access to documents stored in a common location and by keeping track and displaying status information such as approved, rejected, or in progress.

C. Group Collaboration

Collaborative writing systems are designed to support group writing, allowing authors to track changes and see each other's changes as they are made. These systems also provide methods for locking parts of a common document and linking separately authored docu-

ments. *Shared whiteboards* allow two or more users to view and draw on a shared drawing surface from different locations. Whiteboard systems can identify which user is currently drawing by means of color-coded or labeled "telepointers" associated with each user. *Decision support* systems are designed to facilitate decision making in group meetings, providing support for tasks such as brainstorming, critiquing of ideas, weighting and specifying probabilities of events, and voting.

Document management systems allow users to categorize and share their documents with other users. Users can create electronic documents with ordinary office productivity tools, store them in a shared repository, and "profile" these documents to facilitate subsequent retrieval by other users. Profiles are used to classify documents into system-defined categories and provide additional "metadata" such as keywords. A "search engine" is a key component of document management systems that allows users to retrieve documents from the shared repository in various ways, e.g., by category, keyword, or full-text search. Authors can selectively define the granting of access rights to their documents for other users. Document management systems can also maintain profiles with details and physical location of paper-based documents.

D. Hypertext

Hypertext is a system for linking related text documents that allows the participation of multiple users. In a hypertext document, any word or phrase can be "hyperlinked" to information related to that word or phrase residing in the same document or in another document. When a hyperlink is activated, the hypertext system retrieves the related information. For example, by selecting a word in a sentence, the definition of that word is retrieved.

Some hypertext systems allow users to see who has visited a certain document or link, or how often a link has been followed, thus giving users a basic awareness of what other users are doing in the system. Another common multiuser feature in hypertext is the ability for any user to create links in documents authored by others so that the original author can become aware of new related material.

The World Wide Web is a distributed hypertext information system based on the HyperText Markup Language (HTML), a standard language that describes the basic structure and layout of hypertext documents called *pages* (HTML pages or Web pages). HTML pages typically contain links with addresses, in the form of uniform resource locators (URLs), of other Web pages that can be stored on the user's computer or on any other Web *server* in the world. Users of the Web "read" HTML pages with an application called a *browser*.

HTML documents are "plaintext" files with *tags* describing structure and layout elements embedded in the text. Tags are codes surrounded by angle brackets and usually paired (e.g., `<h1> </h1>`, `<p> </p>`, and `<table> </table>`) that describe elements such as headers, paragraphs, and tables. To be readable by a browser, an HTML document must contain at least three pairs of tags, namely, `<HTML> </HTML>`, `<Head> </Head>`, and `<Body> </Body>`, to indicate that it is a well-formed HTML document with a header and a body.

The Web implements a simple and extremely powerful model of collaboration in which *clients* (Web browsers) request pages and web servers send the requested pages in response, using the HyperText Transfer Protocol (HTTP). Using simple text editors or word processors, users can create applications in the form of HTML documents containing structured information; they can then post these documents on an intranet or the Internet, where they become instantly available to an unlimited number of users.

V. APPLICATION DEVELOPMENT TOOLS

Office productivity and workgroup computing tools are primarily used to create user applications, but some of these tools also provide some facilities for users to develop reusable programs. This section presents four types of application development tools that are used to develop reusable programs with which end users create custom applications.

End-user *application generators* provide application development capabilities suitable for standard applications such as data entry and reporting without requiring conventional programming. These tools offer simplicity of use and quick results, but give developers little freedom of decision in terms of application design and tend to have performance, integration, and deployment limitations.

Fourth-generation languages are more powerful tools that can be used to develop full-fledged applications. Some 4GLs provide a relatively friendly programming environment that shields the developer from much of the complexity of the software development process. Introduced mainly as declarative languages for database manipulation, many 4GLs have evolved into sophisticated visual development environments.

Third-generation languages, initially offering little more than a text editor and the language compiler,

have also evolved into comprehensive *application development environments* offering visual interfaces, reusable components, and extensive online help. This has encouraged some end users to acquire sufficient programming skills to venture into ambitious application development projects.

Scripted Web pages use *scripting* languages to create a newer form of application, sometimes referred to as *executable content,* in which the program code is merged with the application content (text and graphics) and packaged into a Web page. Many scripting languages are less complex than traditional 3GLs and easier for end users to learn. More importantly, applications developed as scripted HTML pages can be instantly accessed by an unlimited number of users.

While office productivity and workgroup computing tools have a clearly defined target—the end user—many application development tools are not as clear-cut; both professional programmers as well as end users use 4GLs, application development environments, and scripting languages. The difference usually lies not so much in the tools but in the way the tools are used, the extent to which advanced features are taken advantage of, and the application development process followed in terms of requirements analysis, application design, and program documentation.

A. Application Generators

Application generators are software tools that automate the process of developing computer applications, generating the necessary application programs, or program components, from high-level descriptions of the problem. A range of products has been classified as application generators, from early software engineering tools intended to increase the productivity of professional programmers, to specialized components of 4GLs for less technical users, to end-user tools associated with DBMSs that automatically generate relatively simple database applications.

End-user application generators are usually a component of a DBMS or are part of a development environment that includes a database. These tools are used mainly to create menu-driven database applications and are based on a *data dictionary* or repository stored in the database.

In the simplest tools, the user only needs to specify the data to be used and the type and number of items to be included in the menu and then select options to specify the actions assigned to each item. The application generator typically interprets the user selections, merges those selections with prebuilt code, and draws on the data dictionary of the associated DBMS to understand the relevant data structures. The tool then generates all the necessary program code for the application, which is usually built as a single component with a predefined organization.

In more advanced application generators, the developer can define business rules that are stored in the data dictionary, store application design preferences in style sheets, and use templates to generate the appropriate program code. As with the simpler tools, the developer does not need to write program code if a standard design for the application is acceptable.

In general, end-user application generation tools offer limited functionality as well as relatively little control over the design and performance of the application obtained, thereby offering a limited range of information processing possibilities. Applications created through this kind of automatic generation often suffer a "performance penalty" at runtime due to excess (nonoptimized) code. This penalty becomes increasingly severe as the volume of processing grows, although the increasing processing speed of computers sometimes makes performance a noncritical issue.

Besides functionality and performance, applications generated with some application generators may be difficult to integrate with other existing user applications, in particular, with those created with office productivity and workgroup computing tools. Some application generators require a runtime license to execute the applications, limiting the deployment possibilities.

B. Fourth-Generation Languages

Fourth generation languages are proprietary "specification" languages aimed at providing greater productivity with respect to conventional 3GLs. Greater productivity usually means that the developer needs to write less program code, simpler code, or both.

The 4GL category comprises a wide range of application development tools including "declarative" programming languages, very high-level programming languages, and visual development environments. Query languages and report writers are sometimes also considered 4GLs due to the use these tools make of declarative languages.

While most of these tools aim at increasing the productivity of professional application developers, some 4GL products aim at making computing power available to nonprogrammers by hiding many of the technical details involved in the software development process.

4GLs address the objectives of attaining greater productivity and freeing developers from implemen-

tation details by means of high-level abstractions. Two types of high-level abstractions are typically used in 4GLs: declarative statements and *high-level constructs,* sometimes referred to as very-high-level languages, and prebuilt components that the developer can readily customize and include in new applications.

Declarative statements and high-level constructs provide a specification of what the application is required to do, rather than defining exactly how it is supposed to do it. Similarly, prebuilt components, which include reports, windows, and a variety of graphical user interface (GUI) controls, provide functional building blocks that can be used without knowledge of how they have been implemented.

4GLs using declarative statements and high-level constructs provide a language editor in which the developer writes program statements in the 4GL specification language; these products are generally associated with a DBMS. The developer uses commands such as `SCREEN` and `REPORT` to declare such application components. Commands such as `BUILD` instruct the system to generate the necessary code for these components from data structures and data definitions maintained in the system's data dictionary.

The developer can edit the resulting code to refine or further customize the component. The language of these 4GLs usually includes several procedural statements similar to those found in conventional programming languages in order to specify processing steps that must be executed in a precise sequence.

4GLs offering prebuilt components usually provide a visual environment in which the developer manipulates the components as graphical objects with drag-and-drop techniques and specifies the various "properties" of these objects. Applications are developed by customizing predefined components such as windows, menus, and reports that are supplied by the 4GL environment.

In addition to components, a scripting language is provided for the developer to specify detailed processing logic. As the developer builds the application, the corresponding program code for each component is assembled from the 4GL libraries and automatically merged with the user-specified customizations by the 4GL environment.

Command-based 4GLs are mostly used to create character-based applications, while 4GLs with visual components are generally used to create event-driven applications. Regardless of the specification language used, the developer usually has a choice between generating *pseudo-code* and generating either machine code directly, that is, an executable program, or 3GL code such as C or C++, which is then compiled into machine code.

Pseudo-code can be generated very quickly but does not execute fast and cannot be freely deployed since it requires a proprietary interpreter. Machine code takes longer to generate but is readily executable and is optimized by the 4GL compiler to achieve performance comparable to that of conventional programming languages.

4GLs introduce a significant degree of automation in the application development process yet are flexible enough to address a fairly wide range of information-processing requirements. This is because the developer is free to organize the internal components of the application and define detailed business logic for each component while the tool automatically takes care of integrating the various components. 4GL environments permit developers to integrate the resulting applications with other external applications through predefined interfaces.

From the point of view of making computing power available to nonprogrammers, this approach to application development requires at least a basic understanding of programming principles on the part of the developer. The developer needs to define correct procedural processing logic, expressed in a 3GL fashion and, when visual components are involved, organize the various event-driven actions associated with each component. However, 4GLs offer "wizards" or "experts" that can guide the developer step by step through necessary principles and good practice, making these tools less demanding in terms of programming knowledge requirements.

The choice of 4GL product and the associated specification language usually determines a set of trade-offs. Easier 4GLs may not be sufficiently expressive, limiting the range of applications that can be developed; highly expressive 4GLs, on the other hand, may be as difficult to master as conventional programming languages.

C. Application Development Environments

Application development environments, also known as application development systems and integrated development environments (IDEs), provide support for third-generation programming languages such as Visual Basic, Pascal, C, C++, Java, and other application development tools typically used by professional developers. Some of these tools also make programming languages more accessible to end users willing to invest in acquiring the necessary programming knowledge.

Applications written in 3GLs can be developed using a text editor and a compiler or interpreter, as

many programmers have done for years, but this approach can only be used by highly skilled professionals and tends to hinder the programmer's productivity. Application development environments facilitate the task of developing applications by offering a variety of tools such as prebuilt reusable components, wizards that walk the developer through the various steps of the development process, code examples that can be copied and modified, debuggers that help find and correct programming errors, and templates that automatically generate program code. In addition, developers' "toolkits" offer additional software routines, utilities, and program libraries.

Some 3GL environments have grown in sophistication to the extent of blurring the dividing line with 4GLs. However, programming with 3GLs usually calls for a deeper understanding of programming concepts and techniques and a greater involvement in implementation details. In a 3GL development environment, developers are not limited to the use of predefined components; they develop applications by writing new program code in addition to customizing reusable components and modifying code from existing program libraries. More importantly, it is the developer's task to define and implement the organization of the application and its various components and to integrate these components into appropriate executable modules.

The advantage of 3GL application development environments over 4GLs and other more "automated" tools, such as application generators, is that 3GLs generally give the developer the necessary freedom of design and power of expression to address a virtually unlimited range of information processing requirements. 3GLs not only allow the developer to implement more sophisticated program logic, but also provide greater flexibility for integrating the resulting application with other existing applications. This integration among existing, often heterogeneous, applications is an increasingly important issue for legacy applications that need to be quickly adapted for use with Web-based e-business applications.

D. Scripted Web Pages

Scripted Web pages are HTML documents that contain small programs or *scripts* that can be executed to generate content that is understood by Web browsers. The program code or script can be either a separate component or embedded in the Web pages. These applications are sometimes referred to as *executable documents* or *executable content* since the content of the Web page causes the execution of programs by the Web server. In contrast, JavaScript and similar scripts are executed directly on the user's computer and are typically used in HTML documents to perform common tasks such as checking user input, performing calculations, and displaying messages.

Scripting languages are programming languages that tend to be "lighter" (i.e., having simpler syntax) than more traditional languages, and tend to be interpreted rather than compiled. However, many scripting languages are becoming increasingly powerful (some can even be compiled), making them difficult to distinguish from more conventional languages. Some of the Web technologies available to end users involve the use of full-fledged programming languages such as Visual Basic and Java for scripting purposes, typically to launch or control the execution of programs by the Web server.

One of the main strengths of using scripted Web pages is the portability of the resulting applications across the Web, where Web servers execute the scripts and return Web pages, leaving to browsers the task of translating those pages into a familiar user interface. Thanks to HTML, which is the *lingua franca* of the Web, applications developed as scripted HTML pages, once deployed to a Web server, can be instantly accessed by an unlimited number of users.

VI. TRENDS

Mergers and acquisitions in the software industry and the prominence of standards continue to foster the blending and convergence of software technologies and the use of common infrastructure services, components, and "metaphors" across different product categories for the benefit of end users.

This trend is leading to fewer categories of functionally richer tools, where several formerly distinct categories become nearly indistinguishable from other, more traditional ones. Object-based and object-oriented software and the growing use of standards account for much of the success in integrating dissimilar technologies.

The blend of software technologies across product categories is rapidly changing the landscape of end-user computing tools, from operating systems to applications and application development tools. Several product categories that were popular and in some cases essential in the 1990s have quietly merged with traditional ones, as the following examples illustrate.

Utilities, once essential end-user tools to manage the resources of desktop computers, have become an integral part of modern operating systems. Network operating systems, key components that enabled users

to share network resources, are becoming increasingly hard to distinguish from the computer's operating system. The functionality of desktop publishing (DTP) tools has been reached and surpassed by that of word processors; as a result, DTP has merged with the word processing category.

Office productivity tools are increasingly supporting group collaboration through online meetings, meeting scheduling, and Web discussions, providing functionality that is hard to distinguish from that of workgroup computing tools. Both office productivity and groupware tools are providing increasing support for Web-based applications based on Internet standards.

Many conventional 3GL programming languages have evolved into integrated development environments with facilities for visual programming similar to those found in advanced 4GLs. Also, many 3GLs and 4GLs have focused on the development of Web-based applications and thus have evolved toward object-based and object-oriented techniques. In addition, different application modeling languages associated with proprietary development methods and techniques have converged into the Unified Modeling Language, which is rapidly gaining acceptance among developer and user communities alike.

A promising new technology in the area of database application development, called *business rules,* is of particular interest for nonprogrammers. Business rules technology implements declarative development, that is, a specification of what needs to be done rather than instructions on how to do it, but goes beyond 4GLs to unify the business and application specifications. Instead of using a separate specification for the application, business rules technology uses an inference "engine" to derive the necessary computational steps from the data relationships defined by the business specification. The goal is to achieve a fully compilable and executable business specification that enables business analysts and other non-IT experts to develop and maintain complex business applications.

The Web is changing the way computing resources are used across networks. Web-deployed office productivity tools, an interesting new development, are an example of this change. Using Web-deployment technology, office tools such as word processors, spreadsheets, and presentation graphics can be used as browser-based, online services over the Internet. A key aspect of this technology is that it makes office productivity tools independent of the desktop platform of the end user; used in conjunction with secure connections and Web-based storage, users can create and access office applications from anywhere.

E-mail messages increasingly include live links to databases and intranet/Internet applications, and the design of many information systems increasingly revolves around documents. The document object model, a standard interface for accessing and manipulating the content, structure, and style of documents, in particular HTML and Extensible Markup Language (XML) documents, is fostering the development of "document-centric" applications.

The capabilities of Web applications are being significantly enhanced with the use of XML, an open (nonproprietary) standard for describing how data are structured. Unlike HTML, which uses a fixed set of tags, XML is extensible, that is, it allows developers to create new tags to describe the content of documents. Increasingly, XML is being used to define data elements stored in documents such as Web pages, which thus become functionally similar to databases. The rapid adoption of XML is simplifying the development of electronic data interchange and business-to-business application integration.

Taking advantage of the extensibility of XML, the World Wide Web Consortium (http://www.w3c.org) has reformulated HTML as an XML application, resulting in a new format called extensible HTML (XHTML), which is a candidate new standard for Web pages. An important advantage of an extensible format for Web pages over a fixed one is that Web developers no longer need to wait for new versions of the markup language to implement new features in Web applications. XML, together with the family of new Internet standards, is beginning to bring in a flexible and powerful new breed of end-user computing tools.

SEE ALSO THE FOLLOWING ARTICLES

Electronic Mail • End-User Computing Concepts • End-User Computing, Managing • Groupware • Network Database Systems • Object-Oriented Programming • Productivity • Spreadsheets • Word Processing

BIBLIOGRAPHY

Date, C. J. (2000). *What not how: The business rule approach to application development.* Reading, MA: Addison-Wesley.

IEEE standard glossary of software engineering terminology, IEEE Std 610.12-1990. New York: IEEE.

Nardi, Bonnie A. (1993). *A small matter of programming: Perspectives on end user computing.* Cambridge, MA: The MIT Press.

Simon, A. R., Marion, W. (1996). *Workgroup computing: Workflow, groupware, and messaging.* New York: McGraw-Hill.

Trimbley, J., and Chappell, D. (1989). *A visual introduction to SQL.* New York: John Wiley & Sons.

Engineering, Artificial Intelligence in

Peter C. Y. Chen and Aun-Neow Poo
National University of Singapore

GLOSSARY

artificial intelligence Computational models and techniques that attempt to solve problems as competently as (and possibly faster than) humans can.
fuzzy logic A logic that deals with the concept of partial truth by allowing true values to be between "completely true" and "completely false."
genetic algorithms Exploratory search and optimization procedures based on the principle of natural (biological) evolution and population genetics.
knowledge-based systems Domain-specific computerized systems that emulate human reasoning.
neural networks A group of interconnected processing units whose connection strength can be adjusted in order to produce certain output for a given input.

This article reviews the application of four artificial intelligence (AI) techniques in the field of engineering. These techniques are knowledge-based systems, neural networks, fuzzy logic, and genetic algorithms. A brief historical perspective on and the fundamentals of each of these techniques are presented, followed by a discussion on the general application of these techniques in four main areas of engineering, namely, automatic control, scheduling, fault diagnosis, and concurrent engineering. Limitations, current trends, and potential future development of these techniques are discussed.

I. INTRODUCTION

Engineering concerns the creation of physical systems (e.g., an industrial robot). The behavior of such systems is intrinsically governed by physical laws, such as Newton's laws. Creating an engineering system requires not only knowledge of the physical laws, but also the skills to apply such knowledge in building a system. In general, intelligence refers to the ability to accumulate such knowledge and skills, and to apply them (possibly under uncertain or poorly structured environmental conditions), in order to achieve certain goals.

Engineering has long been a human-centered activity. To build an industrial robot, for instance, requires engineers with various types of knowledge (e.g., of kinematics and dynamics) and skills (e.g., of fabricating mechanical parts). Thus, it is reasonable to say that an engineer obviously possesses (human) intelligence. A man-made system exhibiting to some degree this ability may be characterized as possessing *artificial intelligence* (AI).

An artificially intelligent system (or simply, intelligent system) usually performs some specific tasks by design. A robotic workstation that can recognize and remove faulty integrated-circuit boards from a production line is one example. What makes such a system appear to be intelligent is the internal computational algorithms. These algorithms are the results of the application of certain computational techniques in designing and building the system. Such techniques can be considered AI techniques.

AI techniques are conceptual analytical tools as opposed to application-specific methods. For instance, an AI technique (such as the so-called neural networks) can be used in the above-mentioned robotic workstation, but it may also be used, for example, in a pattern-recognition system for reading handwritten postal codes on envelopes at a mail-sorting plant. In the field of engineering, AI is generally treated as a catch-all term for characterizing computational techniques that can be used to perform, or to assist a human being in performing, certain tasks "intelligently." A computational technique is usually considered artificially intelligent if it can solve (or learn to solve) problems without detailed preprogrammed instructions, and if it can do so at least as competently as a human being could have done. The key aspect of this notion of artificial intelligence is that, in solving a particular problem, an AI technique does not require explicit instructions on what steps to take; it is capable of generating a solution through its internal "reasoning" mechanism.

By this rough measure of artificial intelligence, an explicit sorting method (by itself) for arranging the integers from 1 to 100 in descending order, for instance, is normally not considered an AI technique. However, a neural network capable of "learning" to recognize integers from 1 to 100 that are handwritten by 100 different people is generally considered to be artificially intelligent.

This article discusses AI techniques that are useful in engineering. A number of commonly used AI techniques and their applications in various areas of engineering are presented. This article is organized as follows. Section II presents a historical perspective on the development of artificial intelligence in engineering. Section III presents the fundamentals of a number of commonly used AI techniques. Section IV discusses the applications of these AI techniques in various areas of engineering. Section V discusses the limitations of current AI techniques, and some emerging trends in the application of these techniques in engineering. Section VI discusses possible directions of future development in this field.

II. HISTORICAL PERSPECTIVE

Four types of AI techniques appear to be dominant in engineering: knowledge-based systems, neural networks, fuzzy logic, and genetic algorithms. Early development of the individual techniques took on apparently independent courses. It was not until their widespread use in engineering (especially in the 1990s) that integration of two or more techniques in a particular application became more common. This section briefly reviews the historical background of each technique.

A. Knowledge-Based Systems

Knowledge-based systems are computerized systems that emulate human reasoning. Such systems are built with specific knowledge in certain domains of application, and operate in a way similar to that of a human expert. They were created in an attempt to capture and emulate (generally "high-level") human intelligence in symbolic form, usually in a set of if–then rules. Given an input, the system triggers a corresponding rule to produce a response. For example, a knowledge-based system for managing an automated manufacturing workcell may have a rule such as "If robot A fails during operation, then activate robot B to execute tasks that were originally assigned to robot A." It has been established that the performance of such systems is highly dependent on the amount of domain-specific knowledge contained in the system.

These types of knowledge-based systems first appeared in the 1970s, and were generally known as *expert systems*. They often contained "hard-coded" knowledge in a narrow domain. A pioneering commercial application of such systems was reported in 1980. This application involved an expert system called R1 that can be used to configure VAX computers made by the Digital Equipment Corporation.

Development of knowledge-based systems was facilitated by the advent of so-called expert system shells. These are software programs that are equipped with a basic inference mechanism (also called an inference engine) but without the domain-specific knowledge. The first example of this type of software appears to be EMYCIN, originally developed at Stanford University for medical diagnostic applications. Such shells allow a user with domain-specific knowledge but minimal programming skills to build an expert system for a particular application quickly, by simply entering the necessary data and rules for manipulating such data.

It appears that the current use of knowledge-based systems is mainly in assisting human experts on carrying out certain tasks. The expert system usually assumes a consultative role for analysis, monitoring, and diagnosis, while the ultimate decision is still made by the human expert.

B. Neural Networks

The "modern" concept of artificial neural networks emerged in the 1940s. A neural network was originally conceived to be a group of interconnected logical elements whose connection strength can be adjusted by the so-called Hebbian learning rule. These ideas, novel at the time, lacked analytical formalism. It was not until the late 1950s and early 1960s that significant analytical rigor was established for a class of neural network called *perceptrons*. The original model of perceptrons consists of a collection of neurons arranged in two layers (i.e., an input layer and an output layer), where there is no connection between any two neurons in the same layer and signals travel only from the input layer to the output layer. The key result was the so-called perceptron convergence theorem, which analytically established the learning capability of the two-layer perceptrons.

Perceptrons were considered to be a class of elegant yet simple devices that can learn to solve various problems. However, it was soon discovered that the capability of perceptrons is limited. Specifically, the problem lies in the fact that, although a perceptron can learn anything that it can represent, its representation capability is inherently limited. The often-cited case that exemplifies such a limitation is that perceptrons are incapable of solving the exclusive OR problem, because this problem is not linearly separable, whereas the two-layer perceptrons can represent only linearly separable problems.

The limited representational capability of perceptrons became a major obstacle to the development of the field of neural networks throughout the 1970s and early 1980s. This limitation was overcome in late 1980s, with the publication of two key results. One is the proof that a multilayer perceptron (i.e., a perceptron with an additional layer—called the hidden layer—between the input layer and the output layer, and a nonlinear sigmoidal activation function) is capable of representing any smooth function to any desired degree of accuracy, if the hidden layer contains a sufficient number of neurons. The other result is the rediscovery of the so-called back-propagation algorithm that apparently enables a multilayer perceptron to learn to represent any given function to some degree of accuracy. Due to these results, neural networks (i.e., the multilayer perceptrons and other variants) have finally become a general tool for a wide range of applications, including handwriting recognition, robotics, and chemical process control.

C. Fuzzy Logic

Fuzzy logic is a multivalued logic that allows intermediate values to be defined between binary evaluations (such as either "warm" or "cold"). In fuzzy logic, notions such as "rather warm" or "pretty cold" can be formulated mathematically, and thus can be processed by computers. By enabling computers to process such imprecise notions, it may be possible to develop machines that mimic human-like thinking.

Fuzzy logic was originally developed as a mathematical theory for modeling the uncertainty inherent in natural languages. Its first engineering application was reported in 1975, where fuzzy logic was used in the controller of a steam engine boiler in a laboratory. Subsequent research and development of fuzzy logic applications remained largely in academia until the early 1990s, when fuzzy logic became an industrial tool. Although fuzzy logic has been used in a wide range of applications (from handwriting recognition to cancer diagnosis), its use appears to be the most pervasive in consumer electronics. Fuzzy logic-based controllers can now be found in various consumer products, such as camcorders, vacuum cleaners, and washing machines.

Among the various AI techniques, fuzzy logic appears to be the most successful to date in commercial application. Such success to some extent reflects the fact that fuzzy logic appeals to human intuition and is relatively simple to implement.

D. Genetic Algorithms

Genetic algorithms are computational procedures suitable for dealing with optimization problems (such as the traveling salesman problem, for instance). These procedures are conceptual abstractions of biological evolutionary processes. The basic idea is that, by encoding a candidate solution to an optimization problem as a string of numbers (whose fitness value can be evaluated), and by altering this string iteratively through certain (usually stochastic) operations based on biological processes and the principle of survival of the fittest, an "optimal" solution can be obtained.

Genetic algorithms emerged during the 1960s and 1970s. Their development was originally motivated by the desire to find methods for the design and implementation of robust and adaptive systems that are capable of operating in an uncertain and changing environment. The original framework focused on developing systems that self-adapt over time based on

feedback from the environment. This led to the so-called reproductive plans, which are now generally referred to as simple genetic algorithms. Such algorithms were not designed *a priori* for solving specific problems, but rather as tools for modeling complex systems with emergent behavior.

Throughout the 1970s, research efforts focused mainly on understanding (by empirical exploration) the behavior of simple genetic algorithms, and on developing guidelines for choosing various parameters that influence the behavior of such algorithms. In the 1980s, genetic algorithms were widely used for function optimization. Issues such as representation, efficiency, and convergence became important, because they determine the performance of genetic algorithms in a particular application. By the 1990s, genetic algorithms had become an established optimization method, and have been applied to various engineering problems such as telephone call-routing, manufacturing job-shop scheduling, and database query optimization.

III. TECHNIQUES OF AI

Among the various computational techniques that may be considered artificially intelligent, four appear to be dominant in the field of engineering. They are knowledge-based systems, neural networks, fuzzy logic, and genetic algorithms.

A. Knowledge-Based Systems

A knowledge-based system in general consists of four main components: a database, a knowledge base, an inference engine, and a user interface, as illustrated in Fig. 1. A user interacts with the system through the user interface. The interaction usually takes the form of *query-and-response*. The knowledge base usually contains a set of if–then rules. The database contains the current data (also called context) of a query–response process. The inference engine is the reasoning mechanism; it manipulates the if–then rules and the context to produce a response. For a rule in the form of "If *A* then *B*," *A* is referred to as the antecedent and *B* the consequent. Intuitively, *A* and *B* represent the condition and action of a rule, respectively.

The operation of a typical knowledge-based system can be described as follows. Data associated with a query are entered through the user interface, and stored in the database. These data represent a context, whose arrival causes the inference engine to search the knowledge base with the aim of matching this context with an antecedent of a rule. If a match is found, the rule will "fire," generating a consequent. This consequent may be taken as the response of the system to the query, or it may itself become a new context to be processed by the system. A final response is obtained when no context is left to be processed.

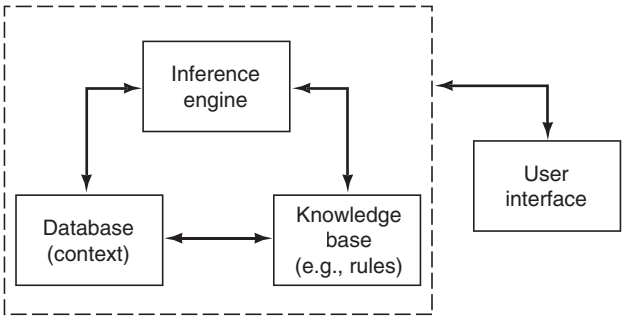


Figure 1 Structure of a knowledge-based system.

To search the knowledge base, an inference engine may use two strategies, namely, forward chaining and backward chaining. In forward chaining, the inference engine searches the knowledge base for rule(s) with an antecedent that matches a given context. If a match is found, the corresponding rule fires to produce a consequent. The action dictated by the consequent may include creation, deletion, and updating of items in the database, and thus may result in the firing of more rules. This strategy is useful in situations where the knowledge-based system is required to specify certain action under the context provided. In backward chaining, a consequent is selected as a hypothesized response. The inference engine then searches the knowledge base to find the context that leads to this hypothesis. This strategy is useful in situations where a logical explanation is required for each proposed action (such as in diagnosis).

It is possible that, during a search, the inference engine finds more than one rule whose antecedent matches the given context. This is referred to as a *conflict,* and the multiple rules represent a conflict set. There are four common methods for resolving a conflict: (1) first match, (2) most recent match, (3) toughest match, and (4) privileged match. In the first method, the first rule found by the search is fired. In the second method, the rule that satisfies the most recent context in the database is fired. In the third method, the eligible rule that contains the highest number of elements in the antecedent is fired. In the last method, each rule in the knowledge base is assigned *a priori* a specific priority, and the rule (in the conflict set) with the highest priority is fired.

Software programs that contain the basic structure of a knowledge-based system are available for facilitating system development. Such software programs are called *shells* because they do not contain any rules. To make a shell into a functional system, the domain-specific rules or expert knowledge need to be added. The development of such rules is usually referred to as knowledge engineering. It involves acquiring knowledge from some human experts, and representing the knowledge in a certain formalism, such as rules, frames, and so on.

B. Neural Networks

There are various types of neural networks. The one most commonly used in engineering applications is the so-called multilayer feedforward neural network. Other variants include radial basis function networks, self-organizing networks, and Hopfield networks.

A multilayer feedforward neural network consists of a collection of processing elements (or units) arranged in a layered structure as shown in Fig. 2. The layer that receives signals from some source external to the network is called the input layer; the layer that sends out signals to some entity external to the network is called the output layer; a layer located between the input layer and the output layer is call a hidden layer. The term *feedforward* indicates the manner by which signals propagate through the network from the input layer to the hidden layer(s) to the output layer. Any given unit, except those in the input layer, receives signals from every unit in the preceding layer, then (based on these signals) generates a response and transmits it to every unit in the next layer, or transmits it to some entity external to the network if the given unit is in the output layer. The function that maps the input signal to a given unit into a response signal of the unit is called the *activation function*. One type of commonly used activation function

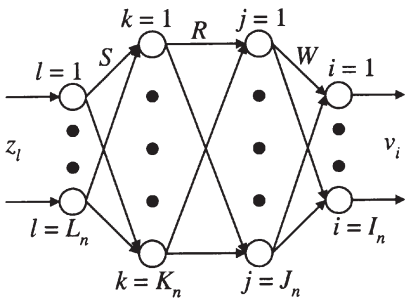


Figure 2 Neural network structure.

is the hyperbolic tangent function $g(x) = c\tanh(x)$, where the constant c is referred to as the scaling factor. The units in the input layer do not have an activation function; each unit in the input layer simply "relays" the network input to every unit in the next layer.

For the neural network with two hidden layers, as depicted in Fig. 2, the network output v_i (of the unit i in the output layer) is generated according to the following sets of nonlinear mappings. Let (1) the number of units in the input layer, the first hidden layer, the second hidden layer, and the output layer be L_n, K_n, J_n, and I_n, respectively; (2) the activation function of the units in the hidden layers and the output layer be $g(x) = c\tanh(x)$; (3) $\bar{\bar{r}}_k$, $\bar{r}_j$, and r_i denote the input to the kth unit in the first hidden layer, jth unit of the second hidden layer, and the ith unit of the output layer, respectively; and (4) $\bar{\bar{v}}_k$, $\bar{v}_j$, and v_i denote the output of the kth unit in the first hidden layer, the jth unit of the second hidden layer, and the ith unit of the output layer, respectively. Then $\bar{\bar{r}}_k = \sum_{l=1}^{L_n} S_{kl}z_l$, $\bar{r}_j = \sum_{k=1}^{K_n} R_{jk}\bar{\bar{v}}_k$, $r_i = \sum_{j=1}^{J_n} W_{ij}\bar{v}_j$, $\bar{\bar{v}}_k = g(\bar{\bar{r}}_k)$, $\bar{v}_j = g(\bar{r}_j)$, and $v_i = g(r_i)$, where W, R, and S are the weight matrices. Alternatively, v_i can be expressed as

$$v_i = g\left(\sum_{j=1}^{J_n} W_{ij}g\left(\sum_{k=1}^{K_n} R_{jk}g\left(\sum_{l=1}^{L_n} S_{kl}z_l\right)\right)\right).$$

For convenience a generalized weight vector Θ is defined as $\Theta = [W_1, \ldots, W_i, \ldots, W_{I_n}, R_1, \ldots, R_j, \ldots, R_{J_n}, S_1, \ldots, S_k, \ldots, S_{K_n}] \in \Re^{c_\theta}$, where W_i, R_j, and S_k represent the ith row of W, the jth row of R, and the kth row of S, respectively, and c_θ is the total number of weights in the network, i.e., $c_\theta = I_n \times J_n + J_n \times K_n + K_n \times L_n$. The mapping realized by the network can then be compactly expressed as $v = N(Z,\Theta)$, where Z is the input vector, i.e., $Z = (z_1, z_2, \ldots, z_l, \ldots, z_{L_n})$, and N is used as a convenient notation to represent the mapping achieved by the network.

It is known that a multilayer feedforward network with one hidden layer (containing a sufficient number of units) is capable of approximating any continuous function to any degree of accuracy. It is in this sense that multilayer feedforward networks have been established as a class of universal approximators. Thus, for a given function $y = f(Z)$, there exists a set of weights Θ^* for a multilayer feedforward neural network (containing a sufficient number of hidden units) with the output $v^d = N(Z, \Theta^*)$, such that, for some ϵ, $\|y - v^d\| \equiv \|f(Z) - N(Z,\Theta^*)\| \le \epsilon$, $\forall\epsilon \ge 0$, where $\|(\cdot)\|$ denotes the supremum of $(\cdot)$. Note that the above statement only assures that the weights Θ^* exist, it does not indicate what their values are, or how to find them. To determine these weights is the objective of neural network learning.

Neural network learning involves the adjustment of the weights. In general, the transformation of the network input into the network output can be referred to as a global mapping. A global mapping achieved by the network is the aggregation of all the local mappings achieved by the individual units in the network. The operation performed by each unit is fixed (i.e., the local mappings, such as the *tanh* function in the example above, are not modifiable), but the connection weights can be modified so as to alter the global mapping. Thus, a multilayer feedforward neural network can be used to represent various input/output relationships by simply adjusting its connection weights according to some specific rule (called a learning rule or a learning algorithm). This process of weight adjustment is called learning (or training). The learning rule is usually derived so as to minimize the network output error, which is defined as the difference between the desired output and the actual output of the network.

A typical learning process is as follows. First, the values of the weights of the network are randomly set. An input is selected with the desired network output (corresponding to this input) specified. The input is fed into the network to generate an output. This output is then compared with the desired output corresponding to the given input. If the difference between the actual output and the desired output (i.e., the output error) is not within a certain tolerance, then the connection weights are adjusted according to the learning rule. This process is repeated until the output error is within the specified tolerance.

The so-called error-backpropagation algorithm is an effective learning rule. Let Δv denote the network output error, i.e., $\Delta v = y - v$ (where y is the desired output of the network), and let the cost function to be minimized be $J = \frac{1}{2}\Delta v^T \Delta v$. The error-backpropagation algorithm specifies that the weights be adjusted in proportion to (but in the opposite direction of) the gradient of $J_{\Delta v}$ with respect to the weights Θ, i.e., $\dot{\Theta} = -\lambda_n \frac{\partial J_{\Delta v}}{\partial \Theta} = -\lambda_n \Delta v^T \frac{\partial \Delta v}{\partial \Theta}$, where λ_n is the learning rate. The entity λ_n determines how fast the connection weights are updated. It is usually set to be small, i.e., $0 < \lambda_n < 1$, to prevent the weights from oscillating around the point of convergence. Since $\Delta v = y - v$, so $\frac{\partial y}{\partial \Theta} = 0$, and $\frac{\partial \Delta v}{\partial \Theta} = -\frac{\partial v}{\partial \Theta}$. The learning rule then becomes $\dot{\Theta} = \lambda_n \Delta v^T \frac{\partial v}{\partial \Theta}$. Specifically, the dynamics of the weights W_{ij}, R_{jk}, and S_{kl} can be expressed as $\dot{W}_{ij} = \lambda_n \Gamma_i \bar{v}_j$, $R_{jk} = \lambda_n \bar{\Gamma}_j \bar{\bar{v}}_k$, $\dot{S}_{kl} = \lambda_n \bar{\bar{\Gamma}}_k z_l$, where $\Gamma_i = \Delta v_i g'(v_i)$, $\bar{\Gamma}_j = g'(\bar{v}_j) \sum_{i=1}^{I_n} \Gamma_i W_{ij}$, $\bar{\bar{\Gamma}}_k = g'(\bar{\bar{v}}_k) \sum_{j=1}^{J_n} \bar{\Gamma}_j R_{jk}$, and $g'(\cdot) = \frac{\partial g(\cdot)}{\partial (\cdot)}$.

C. Fuzzy Logic

In classical (crisp) set theory, a set (if nonempty) contains a group of elements. The boundary of a crisp set A is defined by the following characteristic function $x_A(x)$: $x_A(x) = 1$ if $x \in A$, and $x_A(x) = 0$ if $x \notin A$, where x is an element of the universe of discourse $\mathfrak{X}$. In contrast to a crisp set, the characteristic function of a fuzzy set (also called the *membership function* and denoted by μ_A) expresses the degree to which an element in $\mathfrak{X}$ belongs to the set; it assigns to each element of $\mathfrak{X}$ a number in the closed interval [0,1], i.e., μ_A: $\mathfrak{X} \rightarrow [0,1]$.

Fuzzy logic refers to logical operations on fuzzy sets. The concept of linguistic variables plays a central role in the applications of fuzzy logic. The possible values of a linguistic variable are linguistic terms (or simply, terms). These terms are linguistic interpretations of technical variables. For example, the technical variable *distance* (measured in meters) can have linguistic interpretations such as very far, far, medium, close, or very close. Each linguistic value is elucidated as a label of a fuzzy set in its universe of discourse and each set is defined by a membership function that maps one or more variables to a degree of membership, usually specified in the range between 0 and 1, in a fuzzy set.

A membership function of a fuzzy set is a possibility function. A membership function with a value of zero implies that the corresponding element definitely does not belong to the fuzzy set. When the element is absolutely a member of the fuzzy set, it will have a membership function value of 1. An intermediate value between 0 and 1 implies that the corresponding element falls partially inside the fuzzy boundary of the set, as illustrated in Fig. 3. In practice, the membership function usually has a trapezoidal or triangular shape.

The basic logical operations are NOT (negation, denoted by the symbol $-$), AND (conjunction, $\wedge$), and OR (disjunction, $\vee$). In Boolean logic, NOT X is true if and only if X is not true; X AND Y is true if and only if both X and Y are true; X OR Y is true if and only if at least one of X or Y is true. In fuzzy logic, a different set of rules applies. One version of this set of rules is as follows. Negation of a membership is found by subtracting the membership value from 1, i.e., $\mu_{\bar{A}} = 1 - \mu_A$; conjunction is found by taking the minimum value, i.e., $\mu_{A \wedge B} = \min[\mu_A, \mu_B]$; and disjunction is found by taking the maximum, i.e., $\mu_{A \vee B} = \max[\mu_A, \mu_B]$.

The logical operations of negation, conjunction, and disjunction are the foundations for fuzzy rule-based processing, which is primarily based on infer-

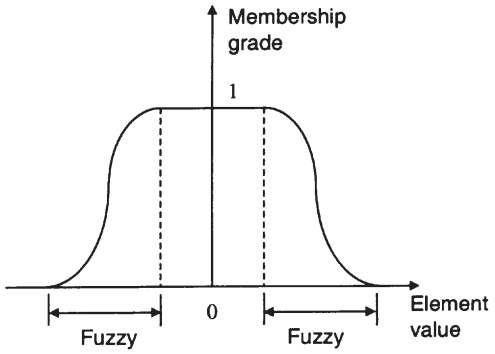


Figure 3 A membership function of a fuzzy set.

ence. Conditional statements (i.e., if–then rules) are the key to making fuzzy logic useful. These statements, when linked through the use of fuzzy operators, can produce quite complicated conditions. A single if–then rule is generally of this form: *If x is A then y is B*, where A and B are linguistic values of variables x and y, respectively. It has become a standard interpretation that this if–then rule is interpreted as a fuzzy constraint on the variables x and y; that is, the rule is equivalent to $A \times B$, where $A \times B$ is the Cartesian product of A and B, which is defined by $\mu_{A\times B}(u,v) = \mu_A(u) \wedge \mu_B(v)$, with $u \in U$ and $v \in V$, where μ_A and μ_B are the membership functions of A and B, respectively, and U and V are the universes of discourse of X and Y, respectively. When the dependence of x on y is characterized by a collection of n if–then rules, the rule becomes: *If x is* A_i *then y is* B_i, $i = 1, \ldots, n$, which is interpreted as: $A_1 \times B_1 + A_2 \times B_2 + \ldots + A_n \times B_n$. Here + denotes the OR operation.

D. Genetic Algorithms

Inspired by Darwin's principle of genetics and natural selection, genetic algorithms are exploratory search and optimization procedures based on the principle of natural (biological) evolution and population genetics. The search and optimization processes imitate natural evolution, and hence include genetic operations such as reproduction, crossover, and mutation. Simple genetic algorithms are procedures that operate in cycles called generations, and are generally composed of coded genotype strings, statistically defined control parameters, a fitness function, genetic operations (reproduction, crossover and mutation), and mechanisms for selection and encoding of the solutions as genotype strings. The basic flowchart of a genetic algorithm is shown in Fig. 4.

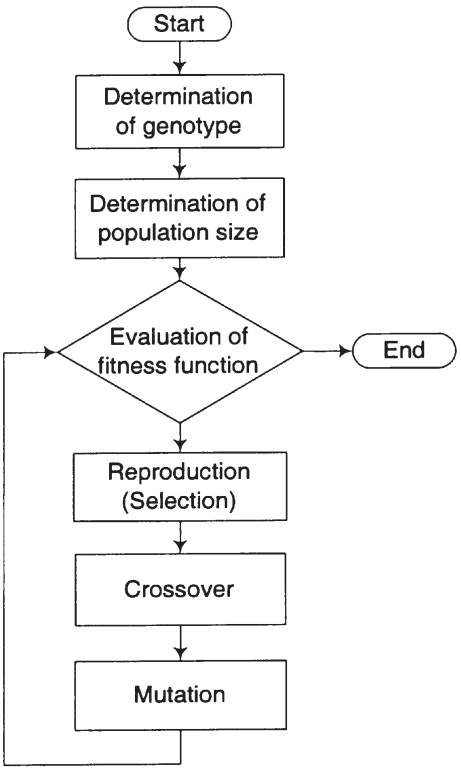


Figure 4 Simple genetic algorithms.

A population of genotype strings called chromosomes is initially generated randomly at the start of the genetic optimization process. At each generation, a new population of strings is generated based on the fitness of the previous generation. Each string in a population encodes a candidate solution. The existing population (maintained by a selection mechanism) undergoes three genetic operations: reproduction, crossover, and mutation.

A few choices of genotype encoding methods are available. Selection of a method depends on the specific problem to be solved. Binary encoding is commonly used. Each chromosome, which represents a probable solution, is a string of zeros and ones—the so-called bits (e.g., {10100101011100101101}). Binary encoding gives many possible chromosomes, even with a small number of alleles. For cases where binary encoding is not natural, some corrections are required after crossover and/or mutation. Permutation encoding is preferred for solving task-ordering problems. Every chromosome is represented by a string of numbers in a sequence (e.g., {293741856} or {239681547}). When dealing with special problems where some complicated values (such as real numbers) are used, value encoding is suitable. In such cases, every chromosome is a string of some meaningful value (e.g., {2.7632,

2.2068, 7.4329, 5.123}, or {(*back*), (*front*), (*right*), (*left*)}, or {*AFEBCEFDBCFEBAFDCEBA*}). An example of such a special case is the problem of finding the weights for a neural network, in which case the real values in the chromosomes represent the weights.

Reproduction is governed by the general principle of survival of the fittest. In this process, a new generation of population is formed by randomly selecting strings from an existing population based on their fitness. As a result, individuals (parents) with higher fitness values are copied to produce a new population of individuals (offspring) in the subsequent generation, while less fit individuals are assigned lower probabilities of contributing to subsequent generations.

After reproduction, a crossover operation is executed to create new individuals (offspring) with a pair of parent chromosomes. This process, analogous to nature's sexual reproduction, takes two chromosomes and swaps part of its genetic information to produce new chromosomes. In the simplest case, some crossover point is randomly chosen; bits before this point is copied from the first parent, and then everything after that crossover point is copied from the second parent. For example, the crossover between Parent 1 with binary strings {001101|100111010} and Parent 2 with strings {110010|010101011} will produce two offspring having chromosomes {001101|0101 01011} and {110010|100111010}.

Mutation is the last genetic operation in a generation before the fitness of each individual of the population is evaluated. It is a localized or bitwise operation, which is applied with a very low probability, to alter the value of a random position in a gene string. Thus mutation acts as an insurance against total loss of any gene in the population by its ability to introduce a gene that may not initially have existed or that was lost through application of reproduction and/or crossover. It also acts to prevent the falling of all solutions in a population into local minima. For binary encoding, this operator simply alters a bit (from 0 to 1, or 1 to 0) at an arbitrarily chosen site. For example, if a chromosome is represented as {1111111111} and mutation occurs at position 8, the new mutated chromosome becomes {1111111011}.

IV. APPLICATION OF AI IN ENGINEERING

AI techniques are increasingly being accepted in various areas of engineering. Four areas in which applications of AI techniques commonly appear are automatic control, scheduling, fault diagnosis, and concurrent engineering.

A. Automatic Control

Figure 5 shows a schematic diagram of a typical feedback control system which comprises three main components: (1) the plant or controlled process, (2) the measurement/observer system (measuring or estimating the states of the plant), and (3) the controller, which, based on the inputs/commands and the state feedback on the plant, controls the plant so as to achieve the desired response at the output.

In a regulator, the control objective is to maintain the plant's output at a desired value (or set point). An example of this is the control of an air-conditioning system to maintain a certain desired temperature. In a servomechanism, the control objective is to make the output follow a desired trajectory (with respect to time) as specified by the inputs/commands. An example of this is the control of the joints of a robot such that the end effector follows a desired path.

Over the years, various conventional control techniques have been developed, including proportional-integral-derivative (PID) control, nonlinear feedback control, adaptive control, sliding mode or variable structure control, linear quadratic Gaussian (LQG) control, and H_∞ control. Most of these techniques are model based, in the sense that when the dynamic model of the plant and its parameters are exactly known, the desired control signals can be generated exactly. These techniques provide good control performance and have many advantages, such as good stability, robustness to model uncertainties and disturbances, and good accuracy and speed of response when the underlying assumptions are met.

PID control is one of the simplest and earliest control techniques. Here, the design of the controller involves mainly the proper selection of the three controller gains, namely, the proportional gain, the integral gain, and the derivative gain. For a linear plant with a known dynamic model, selection of these gains is relatively straightforward and many well-developed techniques exist.

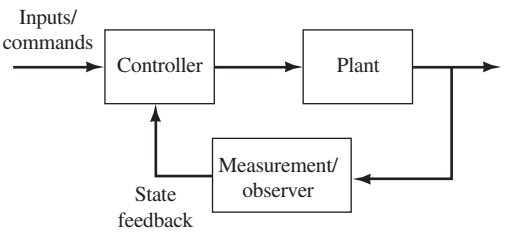


Figure 5 Components of a feedback control system.

For highly nonlinear and highly coupled systems (e.g., robots), linear control techniques often prove to be insufficient, particularly where high performance is required in high-speed trajectory-following operations. One approach to improving performance is to include within the control loop a linearizing and decoupling controller, a technique sometimes called feedback linearization. This nonlinear control technique is model based and requires an accurate dynamic model of the plant to generate a control signal that effectively linearizes and decouples the system.

In an adaptive control system, the values of some or all of the parameters of the feedback controller are modified, or adapted, during operation in responses to changes in the system (e.g., changes in the dynamic model or the presence of external disturbances) so as to maintain good system performance. Various approaches have been developed for adaptation, that is, changing the values of the controller parameters. Many of these are model based and require on-line estimation or system identification to determine the plant's dynamic model in order to tune the controller. Adaptive control systems have been successfully applied in many complex, nonlinear and coupled systems.

Sliding mode control makes use of a sliding surface and the switching controller is designed so as to force the state of the plant onto this surface and to remain on it. The sliding (or switching) surface is defined such that when the state of the system is on the surface, the error of the output is zero. The advantages of sliding mode control are that an exact model of the plant is not required in generating the control signal and it has good robustness against factors such as nonlinearity, model uncertainties, external disturbances, and parameter variations. Its disadvantage is that, in the presence of large modeling errors and disturbances, the high switching frequencies and significant control effort required can lead to undesirable excitation of the high-frequency modes in the plant.

Both LQG control and H_∞ control are optimal control techniques meant for essentially linear time-invariant systems. The control laws are formulated so as to optimize some performance objective. While they have many advantages including guaranteed stability with good stability margins, they are nevertheless model-based techniques and modeling errors can significantly affect their performance.

Conventional control techniques are well developed and lead to satisfactory performance, when the underlying assumptions are satisfied. However, there are several disadvantages. In general, they are model based and any significant modeling errors can significantly affect the performance. The control algorithms are also "hard" and generally cannot handle situations that may involve reasoning or inference making, or where information is incomplete, vague, or qualitative.

In recent years, the development of AI techniques has seen their fusion with conventional feedback control strategies to form what is loosely called intelligent control. These include fuzzy logic control, neural control, knowledge-based (expert) control, genetic algorithm control and combinations of these techniques in hybrid intelligent control (e.g., fuzzy-neural control, genetic-fuzzy control, genetic-neural control, and genetic-neural-fuzzy control).

In general, intelligent control techniques can be classified either as the direct approach or the indirect (or supervisory) approach. In the direct approach, the intelligent controller replaces the conventional controller in the feedback control loop in which, for example, the controller in Fig. 5 would be the intelligent controller instead of a conventional controller. In the indirect approach, the intelligent controller is incorporated into the control system as an auxiliary (supervisory) controller complementing the main conventional controller. This is shown schematically in Fig. 6.

Knowledge-based (expert) control is generally implemented in a supervisory (indirect) control mode, as illustrated in Fig. 6. The intelligent controller comprises two main components, namely, a knowledge base and an inference engine. The knowledge base contains encoded information gained from a human expert with experience on the operation of the plant. Sensors measure information on the desired inputs/commands, and the states of the system. Using this information, the inference engine navigates through the knowledge base, makes the necessary deductions, arrives at a reasonable course of action, and then instructs the controller, or tunes the controller parameters, accordingly. This type of intelligent controller is capable of perception, reasoning, learning,

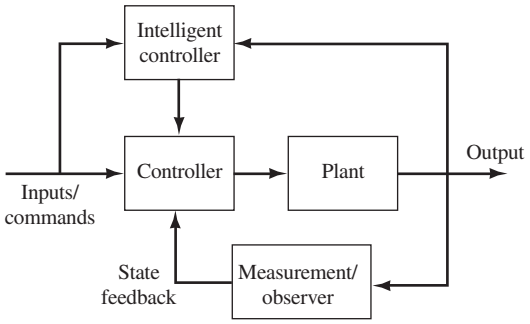


Figure 6 Indirect (supervisory) intelligent control.

and making inferences from incomplete information or knowledge of the process. In conventional controllers, invariably only a single mathematical model of the system under control is used. For expert systems, however, a multiplicity of model descriptions can be used and, depending on the sensor information received, the controller can switch between models to obtain the best performance. Fuzzy logic may also be employed in a supervisory knowledge-based control system. In this case, the measured information from the input commands and the plant are first fuzzified into fuzzy quantities using a fuzzifier. This is then processed by a fuzzy inference engine which works through a fuzzy rule or knowledge base to produce the desired actions. The outputs are finally defuzzified and applied to the controller.

Fuzzy logic has also been applied successfully in many applications using the direct control mode as shown in Fig. 5. In this case, the intelligent controller is a fuzzy logic controller and contains the fuzzy rule base together with an inference engine. The expertise of an experienced control engineer is captured qualitatively using linguistic rules, which are then imbedded in the fuzzy rule base. The error, representing the difference in the output of the plant from its desired value as specified by the input commands, is first fuzzified before processing by the inference engine based on the fuzzy rule base. The control output, in fuzzy quantity, is defuzzified into "crisp" values before input to the plant. Significant improvements in system performance can be achieved, particularly in complex systems which are not amenable to accurate modeling and where knowledge of the dynamics is incomplete.

One main weakness of conventional control techniques, as mentioned earlier, is that they are generally model based and work well when an accurate dynamic model of the plant under control is available. Otherwise, system performance can be significantly degraded. The ability of neural networks to map complex functions, using only input/output measured data, can be used to great advantage in such model-based controllers, and various schemes have been reported successfully with improved performance. With the use of neural networks in a model-based control scheme, not having an accurate model of the plant is no longer a constraint. Generally, in a neural controller, a neural network is trained, using actual plant input/output data, to accurately represent the model of the plant (or its inverse) as needed. When properly trained, the neural network will accurately map the plant's model and achieve overall good control performance.

An example of a neural controller in a nonlinear feedback control scheme is shown schematically in Fig. 7. In this model-based control scheme, if the estimates of the plant's parameters $\hat{M}(q)$ and $\hat{H}(q,\dot{q})$ are accurate, then perfect trajectory-following control can be achieved. In a neural controller based on this control scheme, the portion of the system shown within the dotted lines is replaced by a neural network. If trained properly, the neural network maps its inputs (comprising the states q and $\dot{q}$, and the signal u) into the required control effort T according to the schematic diagram shown in Fig. 7.

The neural network can either be trained off-line until the required mapping accuracy is achieved or on-line. With on-line retraining (using actual plant input/output data during operation), the neural network's mapping characteristic will be changed dynamically according to changes in the plant's dynamic characteristic. In this way, the neural controller can adapt to plant parameter changes and to external disturbances during operation and maintain good control performance continuously over time.

B. Scheduling

Scheduling concerns the proper management of available resources so as to complete a given set of tasks on time. It is an important issue in the management of manufacturing systems. A manufacturing system normally consists of a set of resources (e.g., CNC machines) capable of performing certain tasks (e.g., polishing a part). Each task requires a certain amount of time to execute (i.e., the processing time), and has a deadline for completion, while the available resources are limited. The basic objective of scheduling is to find a sequence by which a set of tasks is to be performed by certain resources.

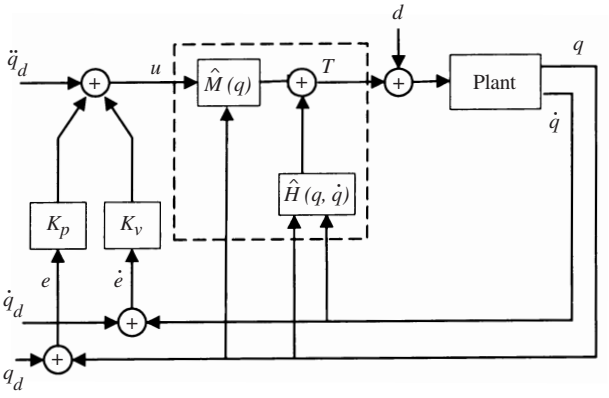


Figure 7 Computed torque control system.

The problem of scheduling has been extensively investigated during the last few decades, with its origin found mainly in *job-shop scheduling*. Such investigations mostly focus on developing analytical methods for managing the activities of various resources to achieve "optimal" operation of a manufacturing system, under the premise that certain tolerance (such as missing a deadline by a certain amount of time) is permitted. The criteria of optimality range from minimum processing time and maximum utilization of resources, to least penalty paid for completing a task before the deadline or for missing the deadline.

Two issues hinder the practical success of analytical approaches to scheduling. One concerns the computational complexity of realistic scheduling problems. Even for a small set of tasks and resources, the computational cost to search for an acceptable schedule could be too high for the search to be beneficial. The other issue concerns the dynamic nature of manufacturing operation. During production, a task may be withdrawn, a machine may break down, and a deadline may be changed. Such disruptions may render a pre-planned schedule obsolete. To deal with these two issues, practical scheduling approaches often rely on heuristics. AI techniques can be used to capture and apply such heuristics.

A human scheduler often formulates schedules based on experience. Knowledge-based systems appear suitable for emulating such formulation. Usually the scheduling heuristics used by a human scheduler are expressed in some form of knowledge representation. The inference engine then uses this knowledge to generate (if possible) a schedule for a given situation. It is also possible to incorporate some "learning" capability into a knowledge-based system for scheduling. One approach is to store known solutions of various types of scheduling problems in the system, and then attempt to match a given scheduling problem to one already stored. If no match is found, the system may try to find a stored problem that "resembles" the given problem, and then use the corresponding stored heuristics to generate a schedule.

Neural networks can also be used to solve certain scheduling problems. One approach is to map the parameters of a scheduling problem into a suitable neural network architecture (e.g., a Hopfield-type network), and then use the corresponding learning algorithm to minimize a certain energy function (e.g., the total completion time of a set of tasks), which represents a solution of the scheduling problem. Another approach is to use a neural network in conjunction with other analytical methods to solve a given scheduling problem. Examples of this approach include using a neural network to "recognize" the type of a given scheduling problem (so that its solution can simply be retrieved from a solution database), and using a neural network to approximate a certain function in the so-called Lagrangian relaxation method for scheduling.

Since a scheduling problem can be formulated as a search of all possible task-execution sequences to find the ones that meet the deadlines of the tasks, genetic algorithms are a suitable tool for such a search. One approach is to choose an appropriate gene structure to represent the candidate solution of a scheduling problem, and then process such a structure using the genetic operations. Unconventional operations based on field-tested manual scheduling procedures can also be added to improve the performance of the solution process. Another approach is to use genetic algorithms to learn heuristic rules for scheduling, so that resulting rules (such as shortest run-time first, scheduling by increasing processing time, etc.) can be used for solving subsequent problems.

Most analytical approaches to scheduling assume that the parameter values of a scheduling problem (e.g., deadline, processing time, etc.) are known *a priori*. In reality, it is most likely that only estimates or ranges of variation for such parameter values are available. For such cases, fuzzy logic can be used to model the uncertainty in the parameters. For instance, imprecise deadlines and processing times can be modeled using fuzzy sets, and then processed by some "fuzzified" conventional scheduling rules to generate a schedule.

C. Fault Diagnosis

Considerable attention has been devoted in recent years to the problem of fault diagnosis, both in plants, systems, and processes. Modern-day systems are extremely complex, highly automated, and are susceptible to equipment malfunction and operator errors. The high level of automation reduces the labor required, particularly the need for highly skilled operators, and gives consistent quality, thereby improving overall efficiency and cost effectiveness. This, together with the complexity of modern-day systems, hampers the operator's ability to quickly diagnose and eliminate potential system faults or equipment failures before they occur.

In fault diagnosis, the primary aim is to detect abnormal system behavior, from measurements made on the process, and to determine the underlying cause of such behavior based on the pattern of these measured

signals. Various AI techniques (e.g., knowledge-based expert systems, fuzzy logic systems, neural networks, and combinations of these) have been employed for system and process fault diagnosis. Such diagnosis could be done off-line (where information on a system or process, which could be a batch process, is processed off-line to determine its health) or on-line (in which the system is continuously monitored and diagnosed).

In a knowledge-based system, a rule base of possible faults is constructed based on the *a priori* knowledge and experience of a domain expert, who could be the designer of the system or process, or an experience operator or maintenance engineer who has experience on such faults. An inference engine in the diagnosis system processes the information gathered, using its fault rule base. How well the diagnosis system works depends on the expertise of the domain expert and how thorough and how well his knowledge has been encoded into the knowledge base. Other than in engineering, knowledge-based diagnosis has also been applied successfully in medical applications.

Among AI techniques in engineering diagnosis, artificial neural networks have perhaps been the most widely applied and many applications have been reported in the literature. One approach is to use actual process parameters during the occurrence of the faults to train the neural network to recognize fault patterns. The usefulness of the neural network for fault diagnosis is due mainly to the ability of neural networks to learn to represent general input/output relationships, to classify patterns, and to generalize.

A neural network is able to learn, using actual measured process data, to map any input/output relationship of a process that is of interest. This is particularly useful for complex processes where a reasonably complete analytical model of the working of the process, including its behavior in the presence of faults, is not available. The more complex a system, the more difficult it will be to diagnose faults. Using a neural network, a "black box" approach can be employed by training a network using actual measured process parameters.

The mapping ability of the neural network also means that it can be trained to recognize or classify patterns. These could represent certain combinations of process parameters, including their historical values, which may indicate the occurrence of certain faults. In such applications, the output from the neural networks would be the presence (or absence) of certain faults or combinations of them. Training involves the use of actual measured process parameters in the presence of the faults.

A neural network is able to not only learn to represent input/output relationships, but also to generalize what it has learned. That is to say that, when properly designed and trained, the input/output relationship computed by the neural network will still be correct (or nearly so) even for combinations of the inputs that it has never seen before (that is, not used in training the network). As an example of the usefulness of this characteristic, a neural network could be trained for multiple single-fault patterns. Data for these could be obtained by inducing in the process or system single faults. Once trained, the neural network will be able to give the correct outputs when a multiple-fault situation occurs even though it has never been trained with multiple-fault data.

Fuzzy logic has also seen many applications in fault diagnosis. It works well in knowledge-based fault diagnosis because human experts generally use linguistic terms that are suitable for use with fuzzy variables. It has also been widely used in combination with artificial neural networks, such as the fuzzy ART (adaptive resonance theory) neural network and the backpropagation neural network.

AI applications for fault diagnosis have been applied to a wide variety of applications, ranging from electrical multicircuit transmission lines and chemical processes to manufacturing processes. Examples include the use of fuzzy ART neural networks to classify faults in complex multicircuit transmission systems, and the application of neural networks for the diagnosis of faults in a discrete-type product assembly line. In the latter case, actual process parameters that exist in the assembly of a subsystem in an optical disk drive were recorded. The actual cause of the defects in subassemblies that failed the final quality checks was also determined by experienced technicians and recorded. The fault data, together with the process parameter values corresponding to these faulty parts, were used to train a neural network. Once the neural network had been trained, it was then used to predict the specific faulty component which causes any subsequent subassembly to fail the final quality check. The neural network diagnosis system helps to reduce the requirement of having highly skilled technicians for the rework of faulty subassemblies. It also helps to reduce the amount of trial and error in the rework process by identifying more accurately, as compared to an unskilled technician, the exact component that causes a subassembly to fail so that it can be the first candidate for replacement.

D. Concurrent Engineering

The process of product development involves various activities. These activities include design, manufactur-

ing, assembly, and service of a product. In traditional product development, these activities are treated as being independent of each other and are performed *sequentially*. Such a sequential approach may result in iterations of the development cycle due to modifications caused by neglecting activities downstream from a particular phase in the sequence.

Concurrent engineering is an approach for product development, whereby various relevant activities are considered *simultaneously*. Multiple design objectives and constraints are investigated in an integrated environment in order to find an optimal solution in the early stage of product development. As a result, design errors can be foreseen and avoided early on, thus reducing (and possibly eliminating) iterative modifications.

Various frameworks have been proposed for implementation of a concurrent engineering approach. Some consider broadly the wide spectrum from business plans and economic feasibility down to the details of manufacturing of specific parts, while others focus narrowly on conceptual design and manufacturability. Regardless of the scope of a particular framework, AI techniques are commonly used in the general areas of design, optimization, and manufacturing process selection.

In developing an acceptable solution for integrated product development, various candidate solutions (related to design, manufacture, assembly, and service) are formulated and evaluated. Such formulation and evaluation may involve qualitative and quantitative measures. For qualitative measures (e.g., of designing parts to be symmetrical), knowledge-based systems are more appropriate, since they may emulate human experts in qualitative design. A typical example is the application of a knowledge-based system for casting design evaluation. The casting process is extremely experience oriented. Exact quantitative characterization of this type of process is very difficult to obtain; the experience of casting designers and foundry experts are critical to a good design. For such a process, knowledge-based systems appear to be well suited. The experience of casting designers and foundry experts can be codified and stored in a knowledge base, and applied to assist less experienced personnel in various situations.

For concurrent engineering activities that involve quantitative measures (such as selecting a particular manufacturing workcell for processing of a specific product), optimization approaches based on genetic algorithms can be employed. Genetic algorithms are suitable for solving problems that involve many discrete and continuous parameters. The combination of these parameters in a design problem (for instance) could result in a large number of possible solutions, with varying degrees of optimality. A genetic algorithm can search the combinations for a set of values for these parameters that leads to an acceptable solution. This is more useful when industry standards are taken into consideration in the selection of the parameter values; that is, a design whose relevant parameters comply with industry standard would be preferred. An example of such an application is the selection of nuts and bolts for a mechanical design, where certain sizes of these components allowed by mechanical design standards are preferred. By using a genetic algorithm, such industry standard requirements can be easily incorporated into the parameter selection process.

Neural networks can also be used for quantitative analysis in product development, particularly for integrating several functions in the concurrent product development process. These functions include product feature recognition, feature family formation, and manufacturing resource selection. The key objective is to select a set of manufacturing resources (machines, tools, and fixtures, etc.) for producing a particular product efficiently. One approach is to use a neural network to recognize the features of parts based on their computer-aided-design (CAD) models, process these features so as to group the parts into families, and then select the appropriate resources to produce each family. The result of this integrated selection process also serves as feedback to the part designer, because potential manufacturing problems (e.g., the available machines are not capable of producing a specific feature) may thus be revealed. By conducting such an integrated selection process at an early stage of the design process, part designs can be evaluated and improved before the actual manufacturing activities start.

In concurrent product development, many concepts and information often cannot be clearly defined and quantified. In such situations, fuzzy logic can be used to model the underlying uncertainty and consequently, to facilitate decision making. For example, there are certain relationships between the function of a part (e.g., torque transmission) and its feature (e.g., dimension, form, surface texture). Such function–feature mappings are difficult to quantify in precise numerical terms, because they exhibit varying degrees of correspondence; that is, a certain function may imply (to some degree) a certain feature, but such implication is not definite, and there is no clear-cut boundaries between such mappings. In such cases, fuzzy logic techniques can be used to model these imprecise mappings to facilitate the design process. Another approach where fuzzy logic could be useful in concurrent

engineering is in incorporating fuzzy knowledge and data representation, and fuzzy inference, in knowledge-based systems. This approach retains the structure of a knowledge-based system, but uses fuzzy facts and fuzzy rules to represent, and fuzzy inference to process, data and knowledge.

V. LIMITATIONS OF CURRENT AI TECHNIQUES AND EMERGING TRENDS

A. Limitations of Current AI Techniques

The four AI techniques discussed in this article have been shown to be useful in various engineering applications. They, however, have certain limitations. The limitations of knowledge-based systems concern the questions on the existence of human expertise and the proper representation of expert knowledge. Application of knowledge-based systems presumes the existence of human expertise in a particular domain. It can be argued that such expertise may not exist at all in certain domains, and that even if it does exist, it may merely lead to *ad hoc* rather than optimal or near-optimal solutions.

These arguments can be illustrated in the context of scheduling. Studies have found that a human scheduler may spend over 80% of his time on understanding the constraints of a scheduling problem rather than on generating a schedule. This may indicate that the so-called scheduling expertise is not about scheduling but merely about constructing a clear definition of a scheduling problem. In real-life situations (e.g., in a production line), events beyond the control of the scheduler may instantly alter the definition of the problem, thus leaving the scheduler with no choice but to produce an *ad hoc* schedule without any detailed analysis.

Even if human expertise does exist in a given domain, the representation of such expertise in proper formats may be elusive, because, due to the "vagueness" of human experience, the knowledge acquired may not be exactly what the expert uses, or may not be complete.

Both genetic algorithms and neural networks are commonly used for optimization. In that context, they share the same two weaknesses. One concerns the fact that there is no guarantee that they will produce the globally optimal solution. The other concerns their slow solution process. In many engineering applications (such as systems control), the degree of optimality of a solution strongly influences the performance of a given system. Ideally, a solution should be globally optimal. However, current algorithms for neural network learning and genetic operations cannot guarantee globally optimal solutions; often the solutions obtained are suboptimal.

The second weakness is that both neural networks and genetic algorithms are slow to produce a solution. This appears to be an inherent characteristic of both techniques. In neural networks, in order for learning to converge, the learning rate is usually set to be small. A small learning rate means a longer time to convergence. In genetic algorithms, the stochastic nature of the genetic operations means that an acceptable solution may not emerge before many generations of candidate solution are produced.

Fuzzy logic may be effective when the problem is simple and intuitive. The simplicity of a problem may mask two main issues concerning the fuzzy-logic technique. One is the apparent lack of systematic methods for applying fuzzy-logic technique in solving engineering problems. Although being intuitive is an advantage of fuzzy logic, when dealing with more complicated problems, systematic methods are more useful. Currently, there seems to be a lack of systematic methods that deal with difficulties concerning matters such as selection of membership functions and inference methods. The other issue concerns the complexity of fuzzy logic-based methods. When a problem is small, the number of fuzzy rules required is also small. However, when a problem is complex and involves many variables, the number of fuzzy rules required to process these variables may become so large that the advantage of fuzzy logic being an intuitive method simply vanishes.

B. Emerging Trends

Although the four main AI techniques took on apparently independent courses through their respective initial development, two trends seem to be emerging concerning the integrated use of AI in engineering. One is the integration of these techniques for a given application. For instance, fuzzy logic and genetic algorithms can be integrated to solve scheduling problems involving tasks with fuzzy deadlines and processing times. A fuzzy set is used to model such tasks, while a genetic algorithm is used to generate the scheduling rules for the tasks. Another example is that of integration of knowledge-based systems with fuzzy logic, where fuzzy sets and rules are used for knowledge representation within the architecture of a knowledge-based system. Various approaches are used to integrate these techniques for

many applications, including neural-fuzzy control and hybrid neuro-expert system for manufacturing process planning.

The other trend is the integration of AI techniques with conventional formal techniques so as to make the solution process more analytically tractable. AI techniques are often regarded as "black boxes" because their solution processes are often not explained analytically. By combining AI techniques with formal methods such as Baysian statistics, Markov chains, and Petri nets, it may be possible to explain to some degree the inner workings of various AI approaches in the context of the formal method involved. As an example, the combination of neural networks and hidden Markov models has become quite useful in speech recognition.

VI. CONCLUSIONS

The field of engineering has increasingly accepted artificial intelligence as a set of useful tools for solving practical problems. This is evident from the amount of research and development reported in academic literature and in trade publications. Although "conventional" analytical approaches still dominate the ways by which engineering problems are defined and solved, AI techniques are beginning to play a more prominent role in complementing the conventional approaches to improve the effectiveness and efficiency of the solution process. However, practical application of AI techniques in engineering, especially in a commercial context, is still very limited. For AI-based approaches to be more widely used in engineering, two major issues need be resolved.

The first issue concerns the general impression that AI-based approaches result in a "black box" solution process. While the final solution may appear to be valid, the process by which an AI-based approach arrives at such a solution is often not amicable to intuitive or analytical understanding. This lack of understanding of the inner workings of AI-based approaches can easily lead to a lack of confidence in the solution itself. Consequently, both the approach and the solution may not be considered acceptable.

The second issue concerns the lack of concrete and impartial evidence demonstrating that, for solving certain engineering problems, AI-based approaches are at least as effective and efficient as conventional approaches. Such evidence must show not only that AI-based approaches can generate working solutions, but also that such solutions are just as reliable and economical. Without such evidence, it is difficult to persuade engineers to consider AI-based techniques in conjunction with conventional approaches in practice.

To resolve the first issue requires analytical tools and techniques for studying the behavior of AI techniques. For that purpose, empirical results based on a few selective computer simulations are not sufficient; only with tractable analysis will such results be convincing. Resolving the second issue requires thorough benchmarking of AI-based approaches against conventional approaches in specific problem domains. Such benchmarks can serve as a reference on deciding if and when to use AI-based approaches instead of conventional ones.

SEE ALSO THE FOLLOWING ARTICLES

Artificial Intelligence Programming • Evolutionary Algorithms • Expert Systems Construction • Hybrid Systems • Industry, Artificial Intelligence in • Knowledge Representation • Medicine, Artificial Intelligence in • Neural Networks • Robotics

BIBLIOGRAPHY

Dym, C. L., and Levitt, R. E. (1991). *Knowledge-based systems in engineering.* New York: McGraw-Hill.

Haykin, S. S. (1994). *Neural networks: a comprehensive foundation.* New York: Macmillan.

Mitsuo, G., and Cheng, R. (2000). *Genetic algorithms and engineering optimization.* New York: John Wiley.

Ross, T. J. (1995). *Fuzzy logic with engineering applications.* New York: McGraw-Hill.

Enterprise Computing

Mark P. Sena
Xavier University

GLOSSARY

application service provider (ASP) An organization that hosts software applications and provides information services for other organizations.
business intelligence software A category of software that enhances the ability of an organization's decision makers to access and analyze enterprise information.
customer relationship management (CRM) systems An integrated information system that supports an organization's planning, scheduling, and control of marketing, sales, promotion, customer service, and related activities.
electronic commerce The use of computer networks for buying, selling, or exchanging information. Electronic commerce can be conducted between two businesses, between businesses and consumers, within a business, or using an electronic intermediary.
enterprise resource planning (ERP) An integrated, packaged software system that supports all major business units of an enterprise.
manufacturing resource planning (MRP II) An information system that integrates manufacturing and related applications such as material requirements planning, accounting, and distribution.
material requirements planning (MRP) An information system that determines the assemblies that must be built and the materials that must be procured in order to meet the scheduled production of a manufactured item by a given date.
supply chain management (SCM) systems A computer system that supports an organization's acquisition, production, and delivery of products and services.
systems integrator (SI) An individual or organization that assists an organization with the selection, implementation, and operation of an enterprise system.

I. INTRODUCTION

In the 1990s, most large organizations changed their approach to building computing systems. Rather than developing software to meet the requirements of a particular business function or process, organizations have increasingly implemented packaged integrated business software known as enterprise systems. The business world's embrace of enterprise systems was among the most important development in the corporate use of information technology in the 1990s. The process of implementing, configuring, maintaining, and operating enterprise systems is known as *enterprise computing*.

Enterprise systems have experienced remarkable growth in the past decade. In 1994, the market for enterprise systems was little more than $200 million annually. The market for enterprise systems is now several billion dollars per year. Major companies now believe that these systems are necessary to compete. The software is now so pervasive that firms that once referred to their information technology according to

their hardware architecture (e.g., "IBM shop") now call themselves "SAP shops," "PeopleSoft shops," etc.

Despite the rapid growth and reported benefits of enterprise systems, many unanswered questions about the field remain. Adopting an enterprise system entails dealing with major organizational issues. Choosing and implementing a package is a difficult process for most firms. Major implementations can cost millions of dollars. Some organizations have blamed their enterprise systems for performance problems or even bankruptcy. Once implemented, the software can dictate important changes to business processes.

Recently, organizations have looked for ways to gain additional value from their enterprise systems. These systems are now viewed as a necessary foundation for major corporations to conduct electronic commerce, to manage supply chains, to build enterprise data warehouses, to conduct business intelligence, and to manage customer relationships. The remainder of this chapter is organized as follows. Section II explores the conditions that led to the popularity of enterprise computing. Section III examines the functions that ERP systems perform, focusing on SAP R/3. Section IV provides an overview of other enterprise systems that complement the core ERP functions. Section V investigates the enterprise system project activities of vendor selection and system implementation. Section VI examines the benefits that organizations can achieve via enterprise computing. Finally, Section VII provides a summary and concluding comments.

II. CONDITIONS LEADING TO THE PROLIFERATION OF ENTERPRISE SYSTEMS

Enterprise computing was among the most important developments in information technology in the 1990s. The rapid growth of the field has been remarkable. Nearly every Fortune 500 firm has implemented some form of enterprise system. Given the formidable costs and risks associated with implementing and operating enterprise systems, the reasons why enterprise systems became attractive to organizations merit further attention.

As a starting point, it may be useful to examine the state of business computing in the days before enterprise systems became so prevalent. Prior to the proliferation of enterprise systems, organizations typically developed customized systems to meet the needs of each functional area. In most companies, systems evolved in a fragmented manner resulting in data that was not kept in a single repository and processes that were not fully integrated. This lack of integration caused considerable inefficiency in the form of storing redundant data, entering and formatting data from one system to another, and so on. With this approach, organizations experienced increasing difficulty and expense in information systems development and maintenance. Chief information officers recognized that many of their large development projects failed to deliver required user functionality within the projected time frame and within the budgeted cost.

As organizations developed these business support systems, events were occurring in the field of information technology that caused executives to reexamine their approach to implementing computing systems. Advances such as client–server computing and graphical user interfaces increased the demand for systems with more user-friendly features. Local-area and global networks emerged that enabled diverse users to share data and resources. Many firms that had developed mainframe-based systems realized that they lacked the expertise to develop these new systems and that the expense of developing systems on this platform would be formidable.

Because multiple firms have similar functional systems (e.g., payroll, accounting), commercial software firms emerged that offered standardized functional software packages. Such software solved the problem of each firm creating its own system but compounded the problem of incompatible data formats and processes. For example, a firm that purchased a standardized payroll system was faced with the task of integrating the data with its existing systems and its users were forced to deal with disparate user interfaces.

In the middle to late 1990s, organizations became increasingly concerned about the year 2000 (Y2K) problem. Managers were faced with decisions about whether to begin the process of fixing the date problems within their functional systems or to purchase and replace the systems. Enterprise systems offered an alternative solution for organizations that chose not to fix their software. In the latter part of the decade, however, the Y2K problem was cited as a reason for declining growth in enterprise systems because many organizations committed resources to fixing the Y2K bug.

Another way to examine the evolution of business computing toward enterprise software is from an operations management perspective. In the 1960s the focus of manufacturing systems was on inventory control. As material requirements planning (MRP) became prominent in the 1970s, firms were able to trans-

late the master schedule of end items into time phased requirements for subassemblies, components, and raw materials procurement. In the 1980s, the concept of manufacturing resources planning (MRP-II) extended MRP to shop floor and distribution management activities. In the 1990s, MRP-II was further extended to include engineering, finance, human resources, and other activities in the business enterprise. Consequently, the term, enterprise resource planning (ERP) was coined.

As a result of the preceding circumstances, enterprise systems became extremely popular. Leading the way in the movement is SAP, founded in 1972 by five former IBM employees. During their work as consultants for IBM, SAP's founders recognized that each client was developing the same or similar computer systems. As a result, the group set out to develop and market standardized enterprise software that would integrate all business processes. In 1979, the company released the R/2 software system for mainframe computers. This was followed in 1992 by the R/3 client–server system that is now deployed for millions of users around the world.

III. ERP FUNCTIONS

ERP is software that integrates and automates traditional back-office functions such as finance, human resources, and operations. To more closely examine the functions provided by these systems, we detail the component modules of SAP R/3, the leading ERP software. SAP R/3 consists of three major functions: financials, logistics, and human resources. Software offered by such vendors as Oracle Applications, PeopleSoft, J. D. Edwards, and Baan offer similar functionality. Table I summarizes the major ERP functions.

A. Financials

The SAP R/3 financials function includes five major modules. The Financial Accounting module allows organizations to process and track financial accounting transactions. It includes such functions as general ledger, accounts receivable and payable, fixed asset accounting, and legal consolidation. The Controlling component enables functions for internal cost

Table I ERP Functionality (SAP R/3)

Financials
Financial Accounting (general ledger, accounts receivable and payable, fixed asset accounting, legal consolidation)
Controlling (overhead cost accounting, cost center accounting, overhead orders, activity-based costing, product cost controlling, cost object controlling, profitability analysis)
Investment Management (corporate-wide budgeting, appropriation requests, investment measures, fixed assets settlement, depreciation forecasts)
Treasury (cash management, treasury management, market risk management, funds management)
Enterprise Controlling (executive information system, profit center accounting, consolidation functions, Business Cockpit)
Logistics
Sales and Distribution (sales support, order entry, pricing, credit checking, availability checking, contract and scheduling agreements, shipping management, billing, sales information system)
Production Planning and Control (sales and operations planning, demand management, material requirements planning, production control)
Materials Management (purchasing—quotations, outline agreements, vendor evaluation—and inventory management, warehouse management, invoicing, inventory control, purchasing information system)
Quality Management (statistical process control, control charting, quality notifications, task assignment, quality information system)
Plant Maintenance (document planning, processing and history of maintenance tasks, maintenance information system)
Logistics Information System (monitoring, reporting logistics information)
Project System (work breakdown structures, cost and schedule planning, monitoring of resources, business workflow, project information system)
Human resources
Payroll
Benefits Administration
Time Management
Employee Self-Service
Manager's Desktop

accounting. It includes overhead cost accounting, cost center accounting, overhead orders, activity-based costing, product cost controlling, cost object controlling, and profitability analysis. The Investment Management component supports strategic decisions for capital spending. It includes such functions as corporate-wide budgeting, appropriation requests, investment measures, fixed assets settlement, and depreciation forecasts. The Treasury component enables users to structure financial assets to ensure liquidity and minimize risk. It includes applications for cash management, treasury management (management of financial deals), market risk management, and funds management. The Enterprise Controlling module monitors company success factors and performance indicators. It consists of an executive information system, profit center accounting, and consolidation functions. These functions enable diverse financial data to be consolidated, combined with external data, and monitored using a "Business Cockpit" that measures performance and enables analysis (drilling) of selected data.

B. Logistics

The SAP R/3 logistics function includes seven major modules. The Sales and Distribution module includes functions for sales support that allow users to manage information on sales leads, sales calls, inquiries, quotations, marketing campaigns, and competitor products. In literature for prior versions of R/3, SAP positioned Sales and Distribution as a fourth functional category. The module enables order entry with pricing that includes surcharges, discounts, price lists, customer agreements, and credit limit functionality. It conducts availability checking and supports contract and scheduling agreements. A shipping management function enables transportation planning, processing, monitoring, and controlling. A billing function enables automatic invoicing via mail, fax, or EDI. Finally, a sales information system supports reporting and analysis of relevant information.

The Production Planning and Control application enables the planning, executing, and control of production. It includes modules for sales and operations planning that enable plans to be formed based on expected sales or other key figures. A demand management module breaks down the figures into product level and creates a demand program. An MRP module then calculates quantities and procurement dates for necessary materials. The application also includes production control modules based on production method (production order processing, repetitive manufacturing, or KANBAN production control).

The Materials Management module is intended to optimize the procurement process and logistics pipeline in an organization. The purchasing function develops purchase orders from requisitions based on reorder levels, forecast data, requirements from other modules, or departmental requests. It also includes functions for quotations, outline agreements, and vendor evaluations. Its functions enable electronic authorizations and purchase order monitoring. The inventory management function supports common types of receipts, issues, and stock transfers in addition to special stocks such as batches or consignment stocks. It supports various inventory evaluation methods. The warehouse management function enables firms to process goods movements and maintain current records of all materials stored. It supports interfaces with handheld terminals, bar-code scanners, and other technologies. The evaluated receipt settlement function creates automatic invoices based on posted goods receipts. Finally, a purchasing information system and inventory control functions enable users to choose data for reports, conduct analysis, identify trends, support purchasing decisions.

The Quality Management application helps firms monitor quality and assists in compliance with international standards. It includes functions to predefine control data and quality requirements and functions to conduct statistical process control and control charting. The module includes quality notifications based on complaints against vendors, internal problem reports, or customer complaints. It also makes use of SAP's business workflow to assign task improvement activities to responsible parties. Additionally, a Quality Management information system enables managers at different levels to plan, monitor, evaluate and control quality.

The Plant Maintenance application supports the planning and execution of maintenance activities with regard to system availability, costs, material, and personnel deployment. The system enables integration with such external systems as geographical information systems, CAD systems, or plant data collection systems. The module allows users to document the planning, processing, and history of maintenance tasks such as inspection, servicing, and repair activities. A catalog system allows users to define causes, activities, and tasks. Like other modules, a maintenance information system supports business reporting, presentation development, and analysis of maintenance data.

In addition to the functional information systems within each logistics module, SAP R/3 includes a Lo-

gistics Information System that brings together information from each of the other modules enabling reporting and analysis of integrated logistical information.

In support of organizational projects ranging from investment management, marketing, research and development, and others, SAP R/3 includes a Project System module. The system allows users to define work breakdown structures to organize project tasks. It includes cost and schedule planning functions that integrate with other R/3 modules. The system checks and monitors the availability of funds, capacities, materials, and other resources. The system is supported by SAP's business workflow functionality for communication and messaging. It also includes a project information system for planning, reporting, and analysis of projects.

C. Human Resources

SAP R/3's Human Resources function includes modules for payroll, time management, benefits administration, and an employee self-service center. In addition to comprehensive payroll functionality, SAP's Payroll module supports government regulation compliance, various calculation methods, data transfer templates, and other services. The Benefits Administration module handles various benefit programs and plans. The Time Management module includes automatic time collection, calculation, reporting, and integration with payroll, financials, and other functions. The Employee Self-Service module empowers employees to view and maintain personal information via Web-based technology or voice-response systems. Human Resources is supported by a Manager's Desktop function that brings together human resources information and enables managers to perform administrative functions such as expense reports, salary changes, and employee transfers, etc. It also enables reporting and ad hoc analysis of human resources information.

IV. ERP EXTENSIONS

Despite continued growth of the overall ERP market in the late 1990s, ERP vendors (PeopleSoft, Baan, and others) experienced declines in revenue growth leading to sharp drops in stock market prices for firms in the sector. In turn, vendors have sought new directions by building on the core ERP systems that so many firms have implemented. Meanwhile, many organizations that implemented ERP in recent years are beginning to look at ways to capitalize on their investment. New applications such as sales force automation, customer relation management, data mining, and supply chain management systems are being built onto ERP platforms to engage customers and drive profits.

To gauge the direction in which ERP vendors are heading, we examine the recent initiatives by ERP market leader SAP. This inspection serves to illustrate the ways in which vendors are expanding their offerings from the transaction-oriented back-office functionality of ERP to areas that serve other needs of organizations. Although our illustration focuses on SAP's offerings, its competitors (e.g., Baan, PeopleSoft, third-party vendors) offer similar types of products. In fact, other vendors (e.g., i2 for supply chain management or Seibel for customer relationship management) are the market leaders in these industries. SAP's product line and functionality, however, are similar to those of industry leaders and can be used to illustrate how these systems complement core ERP systems.

To expand on its core R/3 system and industry solutions (e.g., automotive industry, aerospace), SAP has developed offerings in the areas of electronic commerce (EC), supply chain management (SCM), customer relationship management (front office), and business intelligence. Table II summarizes these initiatives.

A. Electronic Commerce

Electronic commerce applications can be organized into four broad areas: interorganizational (business-to-business), intraorganizational (within business), customer-to-business, and electronic intermediaries. SAP has developed products for each of these classes. For customer-to-business EC, SAP offers the SAP Online Store, which provides functions for product catalogs, shopping basket management, customer registration, quotation and availability checks, payment transactions, order status, and international capabilities.

For interorganizational EC, SAP has developed the Business-to-Business Procurement solution. This system capitalizes on many of the advantages associated with using the World Wide Web. It facilitates online purchasing by enabling suppliers to publish catalogs on the Web using an open-systems architecture via business application programming interfaces (BAPIs) that are XML-enabled (Extensible Markup Language). The system enables real-time integration

Table II Other Enterprise Systems (SAP Business Initiatives)

Electronic Commerce
Customer-to-business EC (Online Store)
Interorganizational EC (Business-to-Business Procurement)
Intraorganizational EC (Employee Self-Service)
Electronic intermediaries (MySAP.com)
Supply Chain Management
Advanced Planner and Optimizer (Supply Chain Cockpit, Demand Planning, Production Planning and Detailed Scheduling, Supply Network Planning, Global Available-to-Promise)
Business-to-Business Procurement
Logistics Execution System (Warehouse Management System, Transportation Management System)
Customer Relationship Management (Front Office)
Marketing (segmentation analysis, database marketing analysis, market research)
Sales (order processing, customer contact management, sales call management, quotations, postsale support, customer cockpit)
Service (installation management, return material authorization, depot repair, scheduling and dispatching, service agreements, mobile service, call management, communications support)
Business Intelligence (Decision Support)
Business Information Warehouse (presentation, analysis, data storage and management, extraction, transformation and loading, data administration, system administration)
Knowledge Management (tools for authoring, translating, and transferring documents, knowledge warehouse, knowledge and documentation from SAP R/3)
Strategic Enterprise Management (business consolidation, planning and simulation, corporate performance monitor, business information collection, stakeholder communication tools)

between buyers and sellers and contains reporting capabilities to conduct vendor performance tracking and cost center analysis.

An example of an intraorganizational EC initiative is SAP's Employee Self-Service application. This product enables employees to use a Web browser to interface with SAP R/3's human resources functions to perform such tasks as reviewing and updating personal information and benefits, conducting time reporting, filing expense reports, and using the employee directory, etc. Each of the initiatives in the three EC areas is enhanced by the ability to link the functions to the core SAP R/3 system.

Intermediaries (or electronic brokers) are economic agents that stand between the parties of a contract (or transactions), namely, buyers and sellers, and perform functions that enable the fulfillment of that contract. Recently, SAP has developed a new EC strategy to extend ERP via an Internet portal site. Through the site, called MySAP.com, SAP plans to host a digital marketplace that packages third-party content (e.g., news, financial information). The portal is intended to become a full-scale application that enables users to purchase goods from parties via the site. Eventually, SAP plans to make many of its core applications available via the Internet, presenting new opportunities to link ERP with EC.

B. Supply Chain Management

A supply chain is a system through which organizations acquire raw material, produce products, and deliver products and services to their customers. The market for supply chain software has rapidly grown into a multibillion industry. Until recently, SAP endorsed third-party vendors, such as i2, that develop SCM software as plug-in applications for firms that use ERP systems. However, as the market became lucrative, SAP developed its own supply chain initiative. Its product has three major components: Advanced Planner and Optimizer (APO), Business-to-Business Procurement, and Logistics Execution System (LES).

SAP's APO system is intended to allow users to model and monitor supply chains "globally, accurately, and dynamically." The system contains five major applications. Supply Chain Cockpit provides a configurable graphical user interface for modeling supply networks, retrieving information, and event triggers that alert users about pending situations. Demand Planning provides advanced forecasting and planning tools that enable planners to combine collaborative forecasts from sales, customers, or partners. The module is integrated with the SAP data warehouse (SAP Business Information Warehouse). Supply Network Planning uses advanced optimization techniques and

what-if simulation to synchronize activities and plan material flow along the supply chain with the intent of supporting purchasing, production, and distribution decisions. Production Planning and Detailed Scheduling combines graphical planning tables with constraint-based optimization tools and planning functionality that includes multiplant planning, materials and capacity checking, simulation capabilities, and other functions. Finally, the Global Available-to-Promise module enables users to simultaneously check multilevel component and capacity availability to match supply with demand.

The second component of SAP's supply chain initiative is the Business-to-Business Procurement solution. As detailed previously, this application is marketed jointly by SAP as a part of its supply chain and electronic commerce initiatives. The solution aims to provide Web-enabled, real-time integration between buying and sellers for procurement of maintenance, repair, and operating (MRO) supplies and services. MRO is a popular term for nonproduction goods and services, such as office supplies, computer equipment, repair parts, and maintenance services.

The third supply chain management component is the SAP Logistics Execution System. This system extends the warehouse management and transportation capabilities present in the core R/3 system. Warehousing functions include the monitoring, planning, and analysis of warehouse performance, inbound and outbound processing, modeling and optimizing of storage space, and interfaces with warehouse automation technologies. Transportation functions include shipment scheduling, routing, and processing, freight cost management, and monitoring and reporting of transportation information networks.

C. Customer Relationship Management (Front Office)

SAP's Customer Relationship Management function is intended to automate business processes associated with sales, marketing, and customer service and to integrate knowledge from these sources with core R/3 business functions. SAP's initiative includes three major functions. SAP Marketing provides tools and functionality to plan, execute, evaluate, and integrate marketing programs. It includes techniques such as segmentation analysis, database marketing analysis, and market research. The system enables analysis of market share from point-of-sale data and tracking of the effectiveness of marketing campaigns. SAP Sales provides functions for order processing, customer contact management, sales call management, inquiries and quotations. It provides postsale support such as order tracking and complaint management. It also includes a customer cockpit for analysis of customer interactions. SAP Service coordinates self-service functions, including a call-center-based customer support center, parts and service delivery, and invoicing. It includes functions for installation management, return material authorization, depot repair, scheduling and dispatching, service agreements, mobile service, call management, and communications support.

D. Business Intelligence

Like its supply chain solution, SAP initially allowed third-party vendors to supply decision support tools that allow organizations to analyze data from the core ERP system, but SAP recently developed a series of systems to perform these functions. Their offerings include the SAP Business Information Warehouse (BW), SAP Knowledge Management (KM), and SAP Strategic Enterprise Management (SEM).

SAP BW is data warehouse software that performs presentation, analysis, data storage and management, transformation and loading, data extraction, data administration, and system administration. The presentation function includes interfaces for standardized report generation, ad hoc queries, a catalog of available reports, Microsoft Excel extraction, Web distribution, and graphical data visualization. The analysis function contains an OLAP (online analytical processing) engine that enables slicing, drill down, statistical reporting, and other OLAP functions. The system allows users to drill into the operational transaction data in addition to accessing data warehouse contents. Additionally, the software provides functions for data storage and management (storing multidimensional views of data), extraction, transformation and loading (procedures for extracting, cleaning, and validating data), data administration (creating schema, cubes, mapping, etc.), and systems administration (scheduling, monitoring systems, security, capacity planning).

SAP divides its KM initiative into three categories, Knowledge Development, Knowledge Transfer, and SAP Content. Knowledge Development includes tools and consulting services to assist organizations in developing knowledge management programs. Consultants assist organizations in defining needs and planning content requirements. Authoring tools help users create (or convert) company information, training

materials, documentation, system simulations, and performance tests into a knowledge repository. The Knowledge Transfer process enables web-based replication of information objects (e.g., documents, presentations) and indexing and retrieval of knowledge content. SAP Content extracts and synthesizes knowledge from the core ERP system in the form of business knowledge, product knowledge, training materials, and documentation. Supporting all functions in the KM initiative is the Knowledge Warehouse (Info DB V.4), which provides the repository and suite of tools to facilitate authoring, translation, distribution, delivery, and retrieval.

SAP SEM is a set of software that enables executives and senior managers to consolidate financial data, conduct corporate planning and simulate business impacts, monitor corporate performance, automate collection of business information, and maintain stakeholder relationships. The software contains five major components. Business Consolidation enables financial consolidation and value-based accounting. Business Planning and Simulation supports the creation of dynamic and linear business models, simulation of scenarios, analysis of scenario results, and rolling forecasts. Corporate Performance Monitor uses industry-specific and customer-developed performance indicators that are linked to a Balanced Scorecard or Management Cockpit to continually monitor performance all levels relative to strategic targets. Business Information Collection supports automated and semiautomated collection of business information from internal and external sources, including an automatic search of Internet information. Stakeholder Relationship Management facilitates the communications with stakeholders regarding business strategy. The module also collects structured feedback from stakeholders and integrates them with the Corporate Performance Monitor and Business Planning and Simulations modules.

V. ENTERPRISE SYSTEM PROJECTS: VENDOR SELECTION AND SYSTEM IMPLEMENTATION

When an organization decides to implement an enterprise system, it is a major undertaking with great potential impact. Many important decisions must be made along the way. First, an organization must consider whether it will benefit from purchasing such a system. Lonzinsky offers the following questions that organizations should ask themselves when conducting such analysis. These guidelines focus on the information quality that organizations derive from current systems and the ability of the current systems to meet operational goals.

- Are the company's current systems incorporated into the company's business and linked to operational and management activities or do they simply record what has happened for later analysis?
- Are data that are presently available reliable as generated by current systems or does the company rely on reconciliation, revisions, and manual adjustments to make numbers useful?
- Is the company's evaluation of its financial position based directly on results obtained from the current system or from resources derived from its reports (requiring extensive additional effort)?
- Are business processes (finance, accounting, receiving, inventory, etc.) naturally integrated or do these functions operate independently, requiring effort to ensure consistent information flow?
- Is the number of persons involved in support activities comparable to other companies in the industry with similar business volumes?
- Are response time and necessary information for client requests satisfactory?
- Does the company's experience with suppliers, recorded by current systems, add value when it comes time to negotiate new contracts?
- Are current systems really used in the company's planning process?
- How much does it cost the company to maintain existing systems at the present level of contribution to the business? Is this cost-benefit relationship satisfactory?
- How much would it cost to migrate from the present situation to an environment of ideal systems based on software packages?
- How much would it cost to maintain such an ideal environment?

After an organization arrives at a decision to acquire an enterprise system, it must begin the process of choosing the most appropriate software to meet its needs. Generally, organizations will form a committee to conduct this analysis. Lonzinsky suggests several factors to be considered in comparing prospective software. These factors suggest that an organization should gather information about vendor support, product strengths and weaknesses, implementation considerations, and the vendor's commitment to improving the product:

- Type of support provided by vendor
- Qualifications of vendor's support personnel
- Extent to which alterations can be made to customize package to user needs
- Reliability of the product
- Vendor response times when called to resolve problems
- Product performance
- Strengths of product functionality
- Product's functional and technical limitations
- Time required to implement package
- Improvements made to package since acquired (or developed) by vendor.

Hecht offers additional suggestions for choosing enterprise software. First, a firm must determine whether to choose a single vendor (integrated solution) or a best-of-breed solution. The integrated approach offers many benefits (e.g., common user interface, integrated knowledge and processes). However, integrated implementations require more consensus among business functions. Hecht also notes that "many products are functionally a mile wide but an inch deep." This implies that some enterprise software contains modules that are not as elaborate as those offered specifically for a particular function.

Given the impact of enterprise software on an organization, selecting the most appropriate vendor is vital. Many organizations struggle with this process. Gartner Group has identified four major stumbling blocks to successful vendor selection: (1) Time—project teams (for vendor selection) can consume up to 20 employees for 14 months. (2) Cost—in addition to employee time, acquisition cost can account for up to 30% of overall cost. Other costs include personnel and travel expense in defining criteria, developing the RFP, gathering and validating data, and interviewing vendors. (3) Finding objective data—companies report that they lack objective, validated data on vendor products and services and are forced to rely on vendor RFP responses, presentations, and marketing to make decisions. (4) Lack of a structured process—companies that lack a rigorous selection method may focus on only a limited set of criteria or may be subject to political agendas or "gut feelings."

While selected publications and Web sites offer comparisons of various vendors, the market changes rapidly as new products and versions of software are introduced. Thus, rather than focus on the results of such investigation, perhaps more importance should be placed on the criteria that have been used to evaluate vendors.

Gartner Group offers some broad comparisons of the major ERP vendors on two comprehensive dimensions: functionality and technical architecture. Among the major vendors, J. D. Edwards is given the highest rating for functionality followed by Oracle, SAP, PeopleSoft, and Baan. In terms of technical architecture, SAP is rated highest followed closely by Baan and J. D. Edwards. Slightly below these three are Oracle and PeopleSoft. Each of the major vendors is rated in the top half of the dimension for technical architecture, and only Baan falls slightly outside of the upper portion in terms of functionality.

Gartner Group also rates vendor performance on two functional areas, manufacturing and general accounting. J. D. Edwards is rated highest in accounting followed by Oracle, PeopleSoft, SAP, and Baan. In manufacturing, J. D. Edwards, SAP, and Baan are rated nearly even followed by Oracle then PeopleSoft.

Once an organization chooses a vendor, it must embark on the arduous process of implementing the system. The impact of failing to succeed in this implementation can be great. ERP systems have a chance of substantially hurting a business due to implementation or performance problems. There are several reports of such failures. FoxMeyer claims that its failed system led the firm to bankruptcy. Dell Computer, Mobile Europe, Dow Chemical, and Applied Materials are among the firms that spent millions before ultimately abandoning the implementation. Even successful implementations frequently require tens of millions of dollars. As a result, it is not surprising that much of the ERP practitioner literature has focused on implementation issues.

Because of the complexity of implementing an enterprise system, nearly every major project is outsourced, in part, by consulting firms that specialize in such implementations. Most consulting firms specialize in implementing software from one or more particular vendors. Vendors such as SAP and PeopleSoft have formed partnerships with selected consultants to manage installations. Vendors have also developed methodologies, such as SAP's "ASAP" program, designed to streamline the difficult process of installing enterprise systems. These programs may be particularly useful for midsized firms that cannot commit the resources available to larger firms.

An alternative that is gaining popularity is the use of application service providers (ASPs) that not only manage enterprise implementations but also the software, hardware and networking technologies. An ASP owns all or part of the infrastructure on which the applications reside and coordinates the various hardware pur-

chases, software licensing or development, and network connections. Customers rent the services from the ASP on a per-user, per-month basis. ASPs share costs among many customers, allowing for the possibility of a lower cost structure than traditional solutions.

Many organizations attempt to manage an ERP implementation like any other large-scale information systems project. Traditional approaches to system development require project teams to gather all requirements up front. The process creates incentives to avoid changes later in the project. However, as an integrated system, ERP systems require considerable flexibility and support from business units. The software may require changes to business process rather than software changes.

Cooke and Peterson conducted a survey of SAP adopters that provides several insights for organizations implementing ERP systems. The authors' findings suggest that successful implementation of SAP depends on executive commitment, clearly defined business objectives, strong project management, and utilization of the best people full time. The study also provides barriers to implementation and factors that impact time and budget. The greatest barriers include skills availability, training, and technical complexity. Similarly, the system complexity is listed as the factor that most contributes to project overrun. Other factors include resistance to change, internal delays, skills availability, and changes in project scope.

VI. POTENTIAL BENEFITS OF ENTERPRISE SYSTEMS

Given the formidable expense and risk of changing an organization's approach to business by implementing an enterprise system, there must be significant benefits that organizations hope to attain. Cooke and Peterson list several reasons why companies implement SAP: standardizing company processes, integrating operations or data, reengineering business processes, optimizing supply chain or inventory, increasing business flexibility, increasing productivity/reduce number of employees, and supporting globalization strategy. They also identified the Y2K problem as a reason for many previous ERP implementations.

Deloitte Consulting reports that ERP has the potential to deliver "significant tactical and bottom-line strategic benefits." They also note that unexpected benefits such as streamlined processes, improved visibility, improved decision making, and enhanced cooperation can be attained.

In terms of motivations for implementing ERP, Deloitte identifies two categories: technical and operational. Technology motivations include systems that are disparate, poor quality of information, difficulty in integrating acquisitions, systems that have become obsolete, and systems that cannot support organizational growth. Operational motivations for implementing ERP include poor business performance, high cost structure, lack of customer responsiveness, complexity or inconsistency of business processes, and inability to support business strategies or to support globalization.

Stemming from these motivations, Deloitte reports several tangible and intangible benefits that are frequently realized from ERP projects. Tangible benefits include inventory reduction, personnel reductions, productivity improvements, order management improvements, financial close cycle reduction, information technology cost reduction, procurement cost reduction, cash management improvement, and transportation and logistics cost reductions. Intangible benefits include improved information and processes, customer responsiveness, integration, standardization, flexibility, and globalization.

However, Deloitte also warns that such benefits may take time to accrue. The study identifies three distinct stages that organizations encounter. After going live, organizations typically experience a dip in performance as they "stabilize." Following this period, organizations tend to "synthesize," realizing additional effectiveness from the better decision-making capabilities afforded by ERP. Finally, firms that are able to enter stage 3 "synergize" around their ERP systems and are able to transform their system's usage into business strategies.

VII. CONCLUSIONS

The proliferation of information technology in the past decade has had a dramatic impact on business and society. While electronic commerce has garnered considerable notoriety, perhaps no other topic has been more important to business computing than ERP. For organizations that adopt them, these systems not only become the foundation of their computing activities but can also fundamentally change the way they conduct their businesses processes. In many industries, ERP systems, along with other types of enterprise systems, are now considered critical for organizations to compete. Effective implementation and operation of enterprise systems can lead to significant

benefits. Conversely, failed implementations have been linked with poor business performance and even bankruptcy.

Enterprise computing continues to evolve each year. The past decade brought changes to organizational computing that few could predict. How will organizations in the next decade utilize computer applications? Will application service providers become the preferred method? What about Web-based intermediaries such as MySAP.com? Will ERP applications become standard worldwide? What future applications, such as customer relationship or supply chain management, will emerge? Although the future of enterprise computing is difficult to predict, it is very likely to remain among the most important topics in business, one that can greatly impact the success of an organization.

SEE ALSO THE FOLLOWING ARTICLES

Computer-Integrated Manufacturing • Electronic Commerce • Executive Information Systems • Procurement • Project Management Techniques • Supply Chain Management • Value Chain Analysis

BIBLIOGRAPHY

Cooke, D. P., and Peterson, W. J. (1998). *SAP implementation: Strategies and results*. New York: The Conference Board.

Davenport, T. H. (July–August 1998). Putting the enterprise into the enterprise system. *Harvard Business Review* 121–131.

Deloitte Consulting (1998). ERP's second wave: Maximizing the value of ERP-enabled processes. Available at http://www.dc.com/what/secondwave/wave2.pdf.

Fan, M., Stallaert, J., and Whinston, A. (2000). The adoption and design methodologies of component-based enterprise systems. *European Journal of Information Systems,* Vol. 9, 25–35.

Hecht, B. (March 1997). Choose the right ERP software. *Datamation*. Available at http://www.datamation.com.

Kumar, K., and van Hillegersberg, J. (April 2000). ERP experiences and evolution. *Communciations of the ACM,* Vol. 43, No. 4, 22–26.

Lonzinsky, S. (1998). *Enterprise-wide software solutions: Integration strategies and practices*. Reading, MA: Addison Wesley

MetaGroup (1999). Enterprise resource management (ERM) solutions and their value. Available at http://www.metagroup.com/products/inforum/ERM.htm.

Technology evaluation: Business applications. (20xx). Available at http://www.technologyevaluation.com/Research/ResearchHighlights/BusinessApplications/BusApps.

3COM (2001). Enterprise resource planning solutions—White paper. Available at http://www.3com.com/technology/tech_net/white_papers/.

Enterprise Resource Planning

Sowmyanarayanan Sadagopan
Indian Institute of Information Technology, Bangalore

GLOSSARY

business process reengineering (BPR) Refers to the critical analysis of the existing ways of carrying out key activities within an enterprise, often referred to as business processes, and optimizes them. In today's IT-intensive business environment, BPR generally accompanies most ERP implementations.

change management Refers to the management of changes in internal policies, revised roles of key positions, and the accompanying resistance to changes that results because of the introduction of ERP software that impacts most employees.

customization The process of making the generic ERP software taking into account firm-specific details and processes.

enterprise computing Refers to the key issues involved in the applications of IT for large-scale, organization-wide, high-performance and mission-critical applications.

"going live" Refers to the process of commissioning the ERP software—with "live" data getting entered into the system.

materials resource planning (MRP) Refers to the detailed calculations that accompany the planning of time-phased availability and ordering of materials, taking into account the complex relationships between assemblies and subassemblies that constitute the final product.

manufacturing resource planning (MRP II) Refers to the planning of manufacturing schedules taking into account the time-phased material availabilities, ordering, and the capacities of production facilities.

"packaged" software Refers to the general-purpose software, such as ERP, that packages business processes that are common to most firms across industries; this permits a firm-level implementation by customizing a generic solution than building it *ab initio* for every firm-level deployment.

process modeling Refers to the abstraction of a set of activities like order processing, as a process that can be analyzed and optimized for efficiency and effectiveness; this is often done through a set of tools built into standard process-modeling software.

ERP is an acronym that stands for enterprise resource planning. ERP software saw phenomenal interest from the corporate sector during the period 1995–2000. It used to be the fastest growing segment among business software. ERP programs offer the capability of integrating all functions of an enterprise—finance and accounts, human resources, sales, logistics, production, materials management, and project management. Naturally, it was appealing enough that most Fortune 500 corporations embraced ERP; this was followed by many small and medium enterprises as well. Substantial investments were made in hardware, software, consulting, and training to support ERP implementations, that a recent McKinsey article estimates ERP market to be in excess of $80 billion in the year 2000. Significant benefits are associated with successful

implementation of ERP—in the form of faster inventory turnover, higher capacity utilization, faster time to market, and overall profitability. Many analysts feel that today's global business environment—products and services customized to suit the individual needs of millions of customers, delivered over multiple timelines in a 24/7 basis—would have been impossible without such enterprise software. Undoubtedly, ERP represents one of the most complex and demanding application software in the corporate environment. Of course, there are instances where corporations plunged into ERP implementation without intense preparation and focus; naturally, there have been a number of disillusions as well. With the maturity of related software in the areas of supply chain management (SCM) and customer relationship management (CRM), ERP has grown into the core of today's enterprise computing.

I. INTRODUCTION

A. Definition of ERP

ERP is a *package software solution* that addresses the *enterprise needs* of an organization by *tightly integrating* the various functions of an organization using a *process view* of the organization.

The core of ERP software is customizable, yet readymade generic software; it is not custom-made for a specific firm, yet has enough flexibility to customize (modify), rather than be built from scratch for a specific firm. ERP software understands the needs of any organization within a specific industry segment. Many of the processes implemented in ERP software are *core processes* such as order processing, order fulfillment, shipping, invoicing, production planning, bill of material (BOM), purchase order, general ledger, etc., that are common to all industry segments. That is the reason why the package software solution works so well. The firm-specific needs are met through a process of customization.

ERP does not merely address the needs of a single function such as finance, marketing, production or human resources; rather it addresses the *entire needs* of an enterprise that cuts across these functions to meaningfully execute any of the core processes.

ERP integrates the functional modules *tightly.* It is not merely the import and export of data across the functional modules. The integration ensures that the logic of a process that cuts across the function is captured genuinely. This in turn implies that data once entered in any of the functional modules (whichever of the module owns the data) is made available to every other module that needs this data. This leads to significant improvements by way of improved consistency and integrity of data.

ERP uses the *process view* of the organization in the place of function view, which dominated the enterprise software before the advent of ERP. The process view provides a much better insight into the organizational systems and procedures and also breaks the "kingdoms" that work at cross-purposes in many organizations.

To implement such a demanding software one needs high-performance computing, high-availability systems, large, high-speed, high-availability on-line storage, and high-speed, high-reliable networks, all at affordable cost.

B. Why ERP?

In spite of heavy investments involved in ERP implementation, many organizations around the world have gone in for ERP solutions. A *properly implemented ERP solution would pay for the heavy investments handsomely and often reasonably fast.* Since ERP solutions address the entire organizational needs, and not selected islands of the organization, ERP introduction brings a new culture, cohesion, and vigor to the organization. After ERP introduction the line managers would no longer have to chase information, check compliance, to rules or conformance to budget. What is striking is that a well-implemented ERP can guarantee these benefits even if the organization is a multi-plant, multilocation global operation spanning the continents. In a sense ERP systems can be compared to the "fly-by-wire" operation of an aircraft. ERP systems similarly would relieve operating managers of routine decisions and leave them with lots of time to think, plan, and execute vital long-term decisions of an organization. Just as a fly-by-wire operation brings in amazing fuel efficiency to the aircraft operation by continuous monitoring of the airplane operation, *ERP systems lead to significant cost savings by continuously monitoring* the organizational health. The seemingly high initial investments become insignificant in the face of hefty long-term returns.

At another level, organizations today face the twin challenges of *globalization* and *shortened product life cycle.* Globalization has led to unprecedented levels of competition. To face such a competition successful corporations should follow the best business practices in the industry. Shortened life cycles call for *continuous design improvement, manufacturing flexibility,* and *super*

efficient logistics control; in short a better management of the entire supply chain. This in turn presupposes faster access to accurate information both inside the organization and from the entire supply chain outside. The organizational units such as finance, marketing, production, and human resources need to operate with a very *high level of integration without losing flexibility.* ERP systems with an organizational wide view of business processes, business needs of information, and flexibility meet these demands admirably.

Thanks to developments in computing and communication technology, it is possible to network organizational units through reliable communication channels, providing tighter integration among them. The server technology today permits very high reliability and access to large data securely at reasonable cost. The open systems philosophy, client/server architecture, high-performance operating systems, relational database management systems (RDBMS) and rapid application development (RAD) tools are available today that permit such enterprise-wide systems to be deployed. These explain the motivating factors behind contemporary ERP systems.

However, there has been a number of disaster stories as well, as reported in a recent *Harvard Business Review* article. Any large-scale software with organization-wide implications must be carefully executed with care, planning, and a willingness to face resistance to change. After all ERP software is only a tool; without a change management strategy, ERP will fail. This should be kept in mind before implementing any of the ERP software.

II. ERP—DIFFERENT PERSPECTIVES

A. Historical Perspective

ERP systems evolved from the *materials requirements planning* (MRP) systems of the seventies and the *manufacturing resources planning* (MRP II) systems of the eighties. Essentially MRP addressed a *single task* of materials planning within manufacturing function while the MRP II addressed *the entire manufacturing function.*

Industries such as automobile manufacture had large inventories of assemblies and sub-assemblies; often there were complex subassembly-to-assembly relationships characterized by BOM involving thousands of parts. The need to drive down the large inventory levels led to the early MRP systems that planned the "order releases." Such planned order releases ensured the time phasing and accurate planning of the subassembly items.

A typical example from bicycle manufacture can illustrate the point—to manufacture 1000 units of bicycles, one needs 2000 wheels, 2000 foot-pedals, and several thousands of spokes. On a given day, a plant may have 400 units of complete bicycles in stock, 6300 units of wheels, 370 units of foot-pedals and 87,900 units of spokes. If the plant is to assemble 800 units of bicycles for the next 4 days of production, determining the precise numbers of each of the items—foot-pedals, wheels, and spokes—is a nontrivial problem. If the independent demand for spare parts is also to be taken into account, one can visualize the complexity. A typical automobile plant with hundreds, if not thousands of parts, has to face problems that are orders of magnitude more difficult. MRP systems address this need.

Using the processing power of computers, databases to store lead-times and order quantities and algorithms to implement the BOM explosion, MRP systems brought considerable order into the chaotic process of material planning in a discrete manufacturing operation.

MRP II went beyond computation of the materials requirement to include loading and scheduling. MRP II systems could determine whether a given schedule of production is feasible, not merely from material availability but also from other production resource points of view. The increased functionality enabled MRP II systems to provide a way to run the MRP II systems in a loop:

First to check the feasibility of a production schedule taking into account the constraints
Second to adjust the loading of the resources, if possible, to meet the production schedule
Third to plan the materials using the traditional MRP

The nineties saw an unprecedented global competition, customer focus, and shortened product life cycles. To respond to these demands corporations had to move toward *agile manufacturing* of products, *continuous improvements* of processes, and *business process reengineering.* This called for integration of manufacturing with other functional areas including accounting, marketing, finance, and human resource development. For example, activity-based costing would not be possible without the *integration of manufacturing and accounting.*

Mass customization of manufacturing needs *integration of marketing and manufacturing.*

Flexible manufacturing with people empowerment necessitates *integration of manufacturing with human resource development function.*

In a sense the business needs in the nineties called for integration of all the functions of management. ERP systems are such integrated information systems.

III. ERP—AN INFORMATION SYSTEM PERSPECTIVE

ERP systems can be viewed as a logical extension of the evolution of electronic data processing (EDP), management information systems (MIS), decision support systems (DSS), and knowledge-based systems (KBS) over the past four decades. The EDP systems concentrated on the *efficiency* aspect to get mundane things like payroll calculation, inventory reports, or census reports generated faster and accurately. The MIS systems addressed the operational information needs through *effectiveness* measures like exception reporting, insights into processes, etc. The DSS used extensive modeling tools such as optimization, simulation, and statistical analysis to reveal patterns in the information generated by MIS systems to genuinely support *tactical* and even *strategic* decisions. The KBS systems went beyond data, information, and models to capture knowledge of the decision maker and to use the captured knowledge to propose far superior *innovative* solutions.

Another categorization of applications view business systems as office automation systems (OA), online transaction processing (OLTP), and DSS. OA included tasks like word processing, spreadsheets, presentation, e-mail, and other communication tools that are generally used for personal productivity. OLTP systems use large databases, networks, and mission-critical applications to improve the organizational productivity. DSS address the needs of top management through natural language processing, expert systems, and other sophisticated tools.

Unfortunately both the approaches missed out on the key issue of integration. The EDP-, MIS-, DSS-, and KBS-based classification assumes a *compartmentalization* across the layers of management. The OA, OLTP, and DSS classification assumes that the tasks are independent. Both assumptions are invalid in the real-world scenario. The ERP systems remove the deficiencies by taking a holistic view of information across the organization. ERP systems capture the essence of the business processes. *It is driven by the business needs and not the information technology (IT) needs.* An IT-driven solution often attempts to formulate a way of using a technique to solve a known business problem. The emphasis is on the usage of a technique or a technology. ERP systems take a business-driven view. They solve the business problem using a combination of the tools and implement the best practices using contemporary technology. This explains the phenomenal success of ERP compared to many other systems.

IV. COMPONENTS OF ERP

Typically any ERP software consists of the following key features.

Finance module with the following key topics:

- *General ledger* (GL) that includes management accounting, balance sheet, and profit and loss statement and closing procedures
- *Accounts receivables* (AR) that includes customer management, invoice and credit/debit memo and reconciliation
- *Accounts payable* (AP) that includes invoice receipt, payments, credit and invoice posting
- *Financial control* that includes cash management, electronic banking, account clearance, cash management, and foreign exchange
- *Asset management* that includes loans, stocks, currency exchange, depreciation, and valuation
- *Funds flow* that includes budgeting, expense accounts, and funds availability control

All these with the added feature of using fiscal year choice, common chart of accounts, and with *all transactions on-line*

Cost module with the following key topics:

- *Cost center/profit center* accounting that includes with cost elements, variance analysis, and transfer pricing
- *Profitability analysis* that includes sales and profit planning, contribution planning, and *customer order and cost of sale* accounting
- *Order and project accounting*
- *Product cost accounting* that includes product, job order costing
- *Performance analysis* including ABC

Sales module with the following key topics:

- *Sales support* including database, telemarketing
- *Customer call management*
- *Inquiry, quotation, tendering, order processing*
- *Special order* such as cash order

- *Contracts, schedule* agreement
- *Pricing* including pricing plans
- *Shipping and transportation*
- *Invoicing*
- *Sales information systems*

Production planning module with the following features:

- Sales planning and production planning
- BOM with many levels
- MRP II simulation
- *Capacity planning*
- Plant data collection
- Production scheduling and control
- Costing
- Project management
- Production information systems

Materials management module with features that typically include:

- Forecasting
- Requirement planning
- Vendor evaluation
- Purchase management
- Inventory management
- Invoice verification
- Warehouse management
- Customer managed inventory

Quality module with features such as:

- Quality planning
- Quality control
- Inspection
- Quality documentation (ISO 9000 support)
- Quality information system

Plant maintenance module with such features as:

- Maintenance planning
- Maintenance orders
- Asset history management
- Worker scheduling
- Integration with PLC
- Plant maintenance information system

Service management module with features such as:

- Customer management
- Device management
- Warranty management
- Cost monitoring
- Service information systems

Human resource module with rich functionalities that include:

- Master data management
- Payroll accounting
- Travel expense
- Time management
- Application processing
- Personnel development
- Manpower planning
- Human resources information system

As can be readily seen the features are very rich, comprehensive, and in turn complex; this explains the widespread interest as well as the confusion that surrounds typical ERP software. The added advantage is the tight integration of modules, for example, SD, sales and distribution; MM, materials management; FI, finance; CO, control and PPC, production planning and control (Fig. 1).

Most ERP software provides extensive facilities for data protection, security, access control, automatic logging of key transactions, business model editors, data migration tools, reporting tools, MIS tools, data archiving tools, data warehousing, and data mining capability either as part of the core product or through strategic partnership with report writing software vendors, DBMS vendors, data storage vendors, etc.

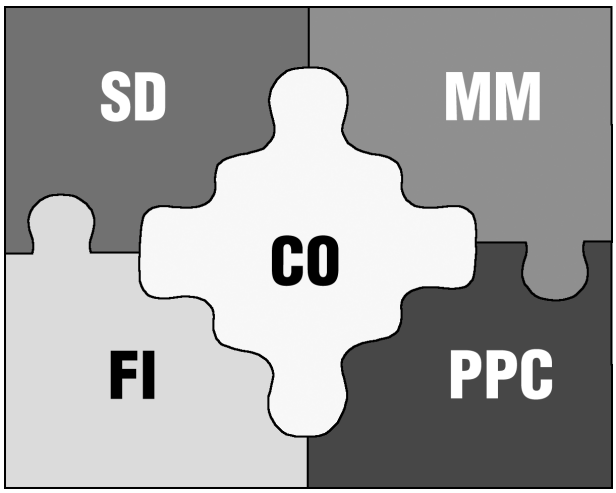


Figure 1 Integration of multiple functions in ERP.

V. ERP IMPLEMENTATION ISSUES

Typically ERP introduction in an organization goes through the following stages:

- *Concept selling*—In this stage ERP consultants take the top management through the ways in which ERP will help the organization in achieving the business goals. Generally this process takes several weeks. The CEO, CFO, CTO, and the CIO must be actively involved in this process.
- *ERP feasibility study*—Once the top management is convinced of the role of ERP, consultants do a feasibility study that broadly quantifies the benefits, costs, and the readiness of the organization for ERP implementation.
- *ERP readiness*—Once the feasibility is achieved the organization is readied for ERP by way of getting the IT and information infrastructure good enough for ERP implementation. This may involve hardware addition/upgradation, network upgradation, and standardizing *key business elements* (account codes, material codes, cost centers etc.), and *key business processes* (order entry, procurement, production, logistics, invoicing etc.). Often this process leads to significant business process reengineering (BPR). Depending on the context BPR could be a full-blown exercise or integrated with ERP implementation. It is pragmatic to view the "best business processes" built into the ERP software as desirable for the organization and modify the existing processes to adapt to the ERP software.
- *ERP software selection*—At this stage a detailed evaluation of the existing leading edge ERP software is done (based on distributor availability, knowledgeable user/consultant support, and training support). Some puritans postpone software selection to a later stage.
- *Mapping "as is" processes and "to be" processes (through pragmatic reengineering)*—At this stage, the existing organizational processes are modeled and using the combined expertise of the entire organization, the processes are improved to take advantage of the ERP. This process is often called the "reengineering process." Special tools to map processes, document processes, and improve processes are available either as part of the ERP software or as independent software pieces under the name "process modeling software" such as ARIS Toolset or Microsoft Visio Enterprise Edition. This process is also referred to as process modeling.
- *ERP implementation plan*—After the process mapping, the entire process of implementation is identified that includes the nature of implementation, extent of implementation, time schedule, cost schedule, training requirements, identifying key users, transition planning, data migration, etc. A key issue is the nature of implementation—some users prefer "big bang" that implies all modules implemented across all departments of the enterprise in one go. Many others prefer module-based implementation, say finance first, followed by sales. Yet others implement all modules in one location and roll out the modules in other location (in multilocation organizations). There is no universal "best" option; the user and consultant should jointly decide the choice. Generally an *ERP Steering Committee* with the blessings and representation of the top management is constituted to supervise the implementation.
- *Actual implementation*—This stage involves the training of the users in the detailed modules, *customizing the software* to meet the specific needs of the enterprise, *configuring the software* to understand industry and firm specific practices (accounting, material, BOM structures, scheduling practices, depreciation options, valuation of inventory/storage, stocking policies, etc.).
- *Preparing to "go live"*—At this stage the decision of project going live is decided. This may coincide with natural "roll over" dates such as beginning of the fiscal year, planning period, etc. To keep the enthusiasm level of the people, a "kick off" function is organized to formally mark the "go live" phase of ERP. All account related start-up figures are fed into the ERP software.
- *"Go live"*—when the organization starts using the ERP software.
- *Postimplementation*—In this phase the ERP software performance is fine tuned to meet the users needs. Often more reports as demanded by the user community are identified and designed.
- *Support initiatives*—Along with the implementation of ERP software a change management exercise is concurrently carried out. This involves a change management workshop that prepares the top management for significant changes in roles that generally accompany ERP implementation. One aspect of this exercise is extensive training. Once there is "buy in" from the key officials, they become change management champions and in turn ready their staff for change management in their own domain. On a technical side, there will

be a number of data conversion, program conversion, and "add-in" and "bolt–on" programs that need to be addressed. These include conversion of data from old systems, revised coding if necessary, identification of specialized programs for which there are no provisions in the specific ERP software, and any interface program such as with external computer-aided design (CAD)/computer-aided manufacturing (CAM) systems, programmable logic controllers (PLC), point of sales (POS), plant maintenance systems, real-time systems, etc.

VI. ERP BENEFITS

Data discipline, build-up of solid information infrastructure, integration across firm-level functions (finance, production, marketing, and human resources), process orientation at the level of internal processes, and significant usable transactional data generation are some of the real benefits of ERP. Data discipline provides an ability to try out new business models that are necessary in the connected world; a solid information infrastructure provides robustness and control in the "mergers and acquisition" dominated volatile business world. Firm-level integration provides foundation for integration across suppliers and customers. Process orientation within an enterprise, that is characteristic of ERP, is another key benefit.

The key benefit of ERP implementation is that ERP provides a solid information infrastructure for an enterprise. As an infrastructure, ERP data is shared by *all* departments across the organization and owned by all users. ERP is *not* one more project initiative from EDP/MIS/IT departments. ERP also ensures *ready data availability*. A well-implemented ERP would pave the way for organizational level *data discipline*. With "information available on tap," it is important that the users start planning for innovative use of this information for planning and analysis. Finally infrastructure should not be viewed from a narrow "cost benefit" and return on investment (ROI) perspective. The true benefits of ERP are not necessarily apparent on day one. Accordingly benefit cost ratio might unduly overemphasize costs that are apparent and underemphasize benefits that may not be apparent. Like every other infrastructure—roads, seaports, airports, telecom, and railways—information infrastructure in the form of ERP needs a different mindset too. It must be noted that investments in infrastructure pays by the innovative ways in which the infrastructure is put to use: investments in roads pays off through returns from trucking industry, business generated through phone calls pays for investments in telecom network. Similarly innovative use of data generated through ERP would pay for ERP investments. The improved organizational agility provided by ERP must be put to good use.

VII. ERP AND OTHER ENTERPRISE FUNCTIONS

A. ERP and SCM

With every enterprise setting up web-based storefront with fancy shopping carts, order processing has become relatively simple for the end users. It is particularly true of the business to consumer (B2C) scenario. As the Christmas 1999 season proved, it is *not* sufficient to build easy-to-use electronic storefronts. There must be a rugged ERP-class backend system to support thousands of customers and variety of products that can be sold on a 24/7 basis throughout the year. More important is the ability to manage the logistics of supply of raw materials that are needed for the production of items to meet the orders taken over the electronic storefront. One must also manage multiple levels of supply and in-house store—finished goods, work-in-process, raw materials, and even consumables across the entire supply chain. Ultimately the delivery of the items to the end customer at the right time at right cost must be guaranteed too. All this calls for advanced planning and optimization of goods and services across multiple echelons. This exercise calls for sophisticated multistage optimization systems with superior performance to handle millions of customers, thousands of orders, and hundreds of products—the core of SCM software. Once again SCM software has changed too; it is not limited to manufacturing execution and sophisticated planning. SCM is merging into e-business scenarios, permits web-based usage, works across corporate intranets and extranets, and triggers production planning systems from supply planning through sophisticated middleware, a powerful application integration indeed.

B. ERP and CRM

Corporate applications of the nineties concentrated on internal efficiencies. With functional integration, process orientation, and "best of breed" practices embedded in the software, internal users did get significant benefits. Corporations gained through better cost control, inventory reduction, reduced cycle times,

improved resource utilization, etc. With the arrival of the World Wide Web and the surge in interest in e-commerce, the interest shifted from back office to front office. Once the front office became the focus area, the customer moved to center stage. The super efficient ERP systems had also built a solid information base about past customers—their buying patterns, their product preferences, payment preferences, shipping preferences, location preferences, brand preferences, and even time preferences—thanks to the solid transaction handling capability of ERP systems. Thanks to the technology of data warehousing and data mining, the customer information and transaction information that had been "logged" religiously by ERP systems and faithfully "backed up" could turn into "gold mines" of information. The retail industry could launch "market basket analysis," the telecom industry could perform "churn analysis," and the financial industry (particularly credit card firms) could predict spending patterns that could be used to target focused advertisements. It is important to realize that many of these data mining tasks could not have been attempted before the ERP systems were running for several years. It is interesting though that in some industries, like telecom and banking, the benefits of using customer information through CRM far outweighed the benefits of ERP, proving that the customer is still the king, even in the Internet age. In addition, e-CRM addresses not only CRM in the Internet age but also CRM in the specific context of web-based business processing; and even processing associated with dot com companies. For example, pure dot com companies like Yahoo! and AltaVista and web-hosting companies like Exodus need guaranteed web-serving performance at the customer end. To manage "page-to-page" refresh time across millions of customers under varying loads is a very demanding customer service. Issues like personalization, studying customer navigation in user sites, and "delivering" information "most relevant" to the individual customer preferences are challenging.

C. ERP and E-Commerce

ERP is *inward* focused, it looks at the enterprise. E-commerce is *outward* focused, it looks at the customers. To deliver value and succeed in business, you need both ERP and e-commerce. Dell Computers had initially put a stop to SAP because Dell found ERP was too rigid; today Dell e-commerce solution works with *all* ERP software. Since Dell sells to UniLever, GM, and Boeing, Dell's e-commerce solution should cowork with all the ERP systems of Dell's customers. Dell primarily assembles PCs and servers; naturally their processes are far simpler. But for companies like Boeing and GM with complex operations, ERP is far more important. But everyone needs both e-commerce and ERP. Their relative importance will vary with the nature of a specfic company's operations. While e-commerce may be more important for a company like Dell that sells computers directly to millions of end customers globally, ERP may be more important for a company like Boeing that sells far complex equipment to just dozens of airlines.

The message is loud and clear—it is *not* ERP or e-commerce; it will always be ERP *and* e-commerce—both today and tomorrow.

VIII. ERP PRODUCTS

We will briefly outline the features of the top five ERP software products (in alphabetical order).

A. BaaN ERP

The ERP software from BaaN Company (acquired by InvenSys in June 2000) grew out of a product with unique strengths to address manufacturing industry, particularly those with complex process-oriented production. Over the years the product has evolved to address all aspects of an enterprise across a range of industry segments. The strength of BaaN ERP has been its component-based architecture, modeling based implementation, and tight integration across all subsystems of manufacturing—production, materials, procurement, shipping, and logistics.

The component-based architecture gives the unique advantage of reduced code size (in millions of lines) with every new release of the product. Much of BaaN's component technology is built around the COM/DCOM/COM+ architecture of Microsoft. BaaN's Business Object Interfaces is a set of API's that offer a component-based approach to ERP. Such an approach permits an easy mix and match approach to deploy the "best of breed" modules among different ERP products.

BaaN ERP products also exploit fully the back and front office features of Microsoft Office products so that MS Word can be used for documentation, MS Excel can be used for analysis and decision support, and MS Exchange/MS Mail/MS Outlook can be used for communication seamlessly with the BaaN ERP product suite.

BaaN has a unique dynamic enterprise modeling (DEM), which introduced a model-based approach to implementation of ERP product. During the configuration phase of any ERP implementation, designers use a less structured approach to correctly capture the requirements of an individual organization. Thanks to DEM from BaaN this process could take a more structured approach that leads to a consistent, reliable, and documented way of configuring the software without extensive programming. Most other ERP vendors have model-based configuration today; but BaaN is the early pioneer and continues to offer superior capabilities in this area.

BaaN ERP is strong in supply chain integration, once again thanks to its strong base among manufacturing and distribution firms. Today every ERP vendor also offers SCM capability. BaaN today has extensive support to both EDI and XML standards for integration across multiple suppliers and business partners. BaaN ERP has unusual support to handheld devices like personal digital assistants (PDAs) including wireless devices support so that field staff can get the full benefit of enterprise software.

Like every other ERP vendor BaaN ERP today is no longer sold as a stand-alone ERP but as a series of e-business modules.

The financial troubles of BaaN throughout the past year and the ultimate acquisition by InvenSys in June 2000 did create problems for BaaN, but the company continues to lead through its technological innovation and appears to have come out of the financial troubles.

B. J.D. Edwards ERP

J.D. Edwards ERP has traditionally been a strong contender in the AS/400 marketplace. Over the years this product has been ported successfully to Windows NT and Unix platforms as well, though JDE is still largely found in AS/400 shops, the stronghold of mid-size companies.

The product went through several transformations over the years. OneWorld is the most successful component-based ERP from JDE and has had a good market acceptance. JDE has a unique hierarchy-based approach to components that is claimed to make configuration much easier. The current version is built around OneWorld and has several components to address the whole range of functions of a contemporary enterprise, namely, manufacturing, financials, logistics, and distribution.

Thanks to its strong presence in distribution, JDE is strong in CRM through its close relationship with CRM leader Siebel Systems. Today JDE works closely with several SCM and CRM vendors and quickly positioned itself to address the e-business model of an enterprise.

C. Oracle ERP

Oracle ERP traditionally has been sold as Oracle Applications. Originally sold as Oracle Financials and Oracle Manufacturing where the product had unique strengths, Oracle ERP has matured into a full-service ERP provider addressing all needs of an enterprise—order processing, financials, manufacturing, and human resources.

For many years Oracle Applications suffered from a lack of tight integration across modules and a two-tier architecture. In the past couple of years Oracle has integrated the products well and started to fully support the n-tier architecture; in fact, Oracle's support to thin clients and web-based applications permit MIS managers to upgrade new versions of the software with practically no installation at client end.

Self-service across all modules (human resources, financials, supply chain) is a unique strength of Oracle Applications. This reflects the broad philosophy that drives the company and its products—using Internet at the core of all its products and technologies.

Today Oracle Applications is tightly integrated with its supply chain and CRM modules.

Reflecting the broad technology directions, Oracle Applications supports middleware, CORBA-based components, and is an excellent support to EJB (and several Java-based technologies).

Oracle's strengths in database technology and the related areas of data warehousing and data mining give Oracle Applications significant advantages to address enterprise-wide OLAP applications and business intelligence.

D. PeopleSoft

Originally started as human resource applications software, PeopleSoft has evolved over the years to address all functions of an enterprise and managed to win clients in financial and other applications as well. Ease of use continues to be the key strength of PeopleSoft.

With its acquisition of CRM software company Vantive, PeopleSoft became a pioneer in visualizing the strength of ERP and CRM products to address corporate applications. This strategy also gives the added advantage of tighter integration across ERP and CRM

segments of PeopleSoft enterprise software, currently marketed as e7.5—Integrated e-Business Backbone.

A unique aspect of PeopleSoft ERP is its ability to address enterprise performance management through the evolving concept of "balanced scorecard" methodology—a radically new approach to measuring performance using key performance indicators (KPI). Through its web-enabled interface to enterprise performance appraisal, PeopleSoft attempts to provide insight into organizational performance that can be used by senior executives from anywhere.

PeopleSoft industry solution templates are a unique way to address vertical industry segments (like insurance) that are also "ready to run," significantly reducing the time and effort to implement PeopleSoft ERP in customer premise.

E. SAP

SAP has been the market leader in ERP segment for many years with its SAP R/2 product for the mainframes and R/3 for the client/server environment. SAP ERP has the highest level of tight integration of functional modules. This feature was the strength of SAP ERP, particularly for Fortune 500 companies. With the Internet era, e-business scenario, and the turbulence in the global marketplace, the strength of a monolithic product like SAP R/3 became a liability. Thanks to the quick redesign and repositioning of its products, SAP continues to drive the enterprise segment.

SAP Business Framework addresses the need for architecture and framework-based application development. Business Framework is an evolution from SAP Business Engineer, which is more of a toolset.

SAP Business API (BAPI) is a powerful way through which SAP could embrace a component-based approach to ERP implementation. BAPI provides application integration across multiple vendor products without complex programming. In a sense the core product feature could be extended without extensive development efforts. Through an innovative object repository (a collection of BAPIs), business objects could be built out of BAPI.

Addressing the needs of extended enterprises in the form of SCM and CRM, SAP built Advanced Planner and Optimizer (APO) and Sales Force Automation (SFA) with CRM functionality. Through strategic partnerships with SCM vendor Manugistics and CRM vendor Siebel, SAP also offers multivendor solutions to address the complex demands of extended enterprises.

Recognizing the e-business challenge, SAP introduced mySAP—a combination of personal portal, e-marketplace, and an application service provider (ASP) offering—all three rolled into one. mySAP has evolved quickly over the past year, starting as a fancy interface with "bells and whistles" personalization into a B2B portal. With added functionality that cuts across personal productivity, personal workplace, and personal desktop, mySAP evolved into an e-marketplace that goes beyond the B2B portal. Through strategic relationships with key industry leaders in the MRO (maintenance, repairs, and overhaul) segment, and chemical and pharmaceutical industry, SAP could use its established presence in Fortune 500 companies with a formidably stable ERP product and SCM, CRM, and SFA offerings to offer a powerful e-marketplace. Again through its relationship with telecom giants like Deutsche Telecom and British Telecom, SAP could position mySAP as a flexible ASP that could offer a test drive, solution builder, solution implementer, solution provider, or a solution builder cum provider.

There are many other promising ERP software vendors including MFG/PRO from QAD (www.qad.com), Priority from Eshbel (www.eshbel.com), and e.Applications from Ramco (www.rsi.ramco.com). We have limited ourselves to the top five vendors only.

IX. FUTURE OF ERP

Corporate applications are undergoing a metamorphosis. First, ERP, SCM, and CRM are seamlessly coalescing into one another; with practically all enterprise software vendors offering these features as part of their broader e-business suite. This in turn has led to several mergers and acquisitions (BaaN and Invensys, PeopleSoft and Vantive, Nortel and Clarify), indicating a shift from growth phase to consolidation phase. Second, all the product suites are fully web-enabled so that the Internet browser has become the de facto user interface for all corporate applications. Third, as we enter the post "PC-centric" computing model, access and delivery of the enterprise software through portable and mobile devices are getting the maximum attention. Fourth, as we go past the "Dot.Com burst," robust e-commerce through business exchanges would be the order of the day. Several companies will move to e-markets; IT services will get outsourced; ASP will host applications; many of the enterprise software products will be offered as services; and the companies that help corporations to leverage e-business, companies who provide global e-commerce solutions and professional services, will be the most influential ones.

A. ERP To XRP

ERP software sales did slow down in 1999 and the trend continues in 2000. In fact, what was considered a great advantage became a disadvantage. ERP was replacing legacy systems but with the businesses moving toward e-business, the rigid ERP systems were viewed as stumbling blocks that were holding the companies from embracing the Internet. There was some reasonable bad press coverage that magnified the woes of ERP. The failing fortunes of an unusually successful ERP software vendor, BaaN Company (that was finally acquired by InvenSys), kept the rumor milling fairly busy for the first six months of the year 2000. But such criticism of ERP was quite unfounded. The unrealistic faith in the dot com companies was partly responsible for this negative view. Thanks to dot com burst, users are now realizing the folly of writing premature obituaries for ERP.

The Christmas shopping season of 1999 also taught everyone some true lessons. There were millions of people ordering things over the Internet. The e-commerce *pundits* thought it was a paradigm shift—new economy eclipsing the old economy. Yet, reality dawned on the people a couple of weeks later. The "clicks" could "take the order"; but the "bricks" were not there ready to supply the order. There were millions of disgruntled customers: those whose orders were considerably delayed, those whose orders got mixed up and messed up, and finally those whose orders vanished into cyberspace. In turn many of the users were promising *never* to order over the Internet again. The pendulum swung to the other extreme.

What went wrong? Web-enabling applications just meant web page enabling to many CIOs—quickly paint an HTML screen and add some graphics and an animated shopping cart. To spice up the user interface they just added a Java applet that would move around a shopping cart over the user screen—the cart that gets filled when items are ordered and gets emptied when orders are cancelled—all with a nice visual appeal. It was cool; it was fun. But it was useless too. What led to the disaster of New Year 2000 shopping was that the backend was not ready to cope with such huge demand. Accepting millions of orders over the Internet needs *solid backend application, a typical ERP class mission critical application, which is robust and scaleable.* Such enterprise applications form the bedrock of e-commerce. Not having such an infrastructure was the rub and e-commerce slowed down.

ERP fever as it existed until 1999 may never come back, but ERP software is slowly and steadily transforming itself to what some people call XRP or eXtended Resource Planning). XRP though not widely accepted does bring out the enlarged scope of corporate application systems.

B. E-Everything

With the explosive growth of the Internet and businesses continuing to find new ways of doing business, there was a sudden move to embrace an e-business approach to everything—procurement to e-procurement, logistics to e-logistics, market to e-market, billing to e-billing—not mere commerce to e-commerce. This had a tremendous impact on corporate applications. ERP systems had excellent order processing systems but could not readily extend to the Web. Users got used to the "slick" shopping carts on the screen and expected the "easy" interface from ERP software too. For simple B2C buying, typical of Amazon Bookstore, such "slick" interfaces were fine; the solid enterprise systems were built for complex products and were naturally complex. But once the users get used to simple and universal interfaces, they demanded the same from every corporate application. By now, most ERP vendors offer such interfaces; but it did take time.

C. From EDI to E-Procurement

Traditional B2B was characterized by robust and scalable but expensive and proprietary systems; the classic case of electronic document interchange (EDI). Such proven systems assumed limited buyers and sellers. With the Internet removing barriers to entry and the dot com fever that produced thousands of nimble corporations overnight, there was a need for "open" systems that supported standard interfaces like TCP/IP. With the easy availability of public Internet, corporations were retiring their private networks to take the cost and reliability advantage provided by the upcoming public Internet infrastructure. Obviously corporate applications had to look for electronic procurement. The first stage was set by corporations putting their catalogs on-line. But users wanted "full service," not only for looking up product features and test driving them but also the ability to put out requests for purchase (RFP), negotiations, and the ultimate purchasing. New business models like "reverse auctions" became a boon to B2B purchases. Niche players started appearing from nowhere to address this e-procurement market, particularly in segments like automotive, steel, and government; purchases that had millions of dollars of worth every year with

significant scope for improvement. Enterprise vendors like SAP quickly got their act together to offer outstanding e-procurement solutions.

D. Client/Server Computing to Componentization

Corporate applications in general and ERP systems in particular were well integrated and offered reliable solid code that appealed to CIOs; prior to ERP they had the "headache" of integration problems associated with many disjointed systems. One key success factor for the wide acceptance of ERP systems was the client/server architecture. This architecture allowed corporations of varying sizes with multiple product lines and many locations to distribute database servers, application servers, and clients across diverse platforms: IBM mainframes/high-end servers for database servers, medium-end servers/PC-based servers for application servers, and PC/Mac/Terminals for clients. This partitioning allowed CIOs to choose servers best suited for a specific function from a cross section of vendors, to shift hardware as per application need without users even knowing the changes; and to guarantee performance over the natural growth of application complexity and the extent of implementation. Of course there was a price to pay, the ERP code was a solid monolithic code. With e-enabling of corporations and individual functions like human resources, procurement, logistics, and manufacturing getting a web interface, there was a need to move toward "nimble" software that was less complex, needed less training, and ran on limited hardware resources (server and network). Also, there was a need for "mix and match" across the functions. For example, users wanted an option of using PeopleSoft for human resources, BaaN for manufacturing, and Oracle for financials. The ERP vendors also realized the potential of the emerging object-based framework (that goes beyond object-oriented programming) using technologies such as COM/DCOM from Microsoft, CORBA from OMG, and EJB from Sun Microsystems that promised better, reusable, and far more reliable code to replace the solid but monolithic code.

E. In the Connected World, Alliance Is the Key

In the early days of enterprise applications, the watchword was "tight integration." Companies like SAP offered a level of integration among all its modules that was unparalleled in corporate applications earlier. With the extension of enterprise to suppliers through SCM and customers through CRM, new players like i2 have emerged as market leaders. Though ERP vendors were quick to reposition themselves as "full-service" providers (ERP, SCM, and CRM) it was clear that ERP vendors would not be able to match the functionality of vendors like i2 (for SCM) and Siebel (for CRM) in their niche areas. Thanks to the Internet, portals, vortals (vertical portals), and exchanges quickly emerged in the year 2000 that offered a new business model of offering enterprise products as service. Companies like CommerceOne, Ariba, and IBM got their act together with their service offerings through their partnerships with ERP, SCM, and CRM vendors. More interestingly IBM, CommerceOne, and Ariba announced in October 2000 a strategic alliance that appears more promising than any other alliance announced earlier. It will be an interesting development to watch in the year 2001. Similar announcements have been announced by SAP also.

F. Cross-Discipline Integration to Integration across Enterprises

The key to the success of Enterprise Systems was their process orientation and application integration across the enterprise: financials, accounting, order processing, human resources, procurement, production, logistics, sales, support, invoicing, and billing. In the pre-Internet era such integration was sufficient. As the new Millennium dawned and the Internet economy boomed, organizations could not afford to be content with the success of the firm within their walls. Applications had to extend to their business partners: suppliers, resellers, and key customers (if not all customers). This in turn necessitated the move to look beyond enterprise to an extended enterprise.

G. Orderly Processes to Creative Thinking

Enterprise systems had an excellent "process orientation." In fact such process orientation helped organizations to think beyond their "factory chutes" and remove enormous inefficiencies across their internal functions—finance, marketing, production, and human resources—such benefits from process orientation far outweigh the benefits of introducing some ERP software. But the process orientation also led to rigidity, the processes were frozen in the ERP software. With the shift from business to e-business, organizations quickly needed to offer new "business models,"

new channels including the Internet, and new forms of service. The emphasis had to shift to organizational innovation rather than organizational discipline.

H. Web Changes Everything

The arrival of the Internet had a profound impact on corporate applications. Almost every business is becoming e-business. There is no industry unaffected by the impact of the Web. Core industries like coal, steel, power, utilities, manufacturing, auto, oil and gas, etc., could all benefit from the power of the Internet. This in turn led to the demand for web-enabling all corporate applications. Most human resources systems had to shift to a self-service operation over the corporate Intranet and manufacturing had to embrace e-engineering and e-procurement over the public Internet or Virtual Private Net between its suppliers. Marketing had to adapt to the electronic storefront, e-services, and e-support. Accounting and financials had to adapt to e-commerce. In a sense every function of the enterprise had to be e-enabled calling for design changes in ERP systems. The transformation of SAP to mySAP and Oracle to e-Applications represents this fundamental shift in corporate applications.

I. E-Markets

The arrival of the digital marketplace (also called e-markets) is a major development with far-reaching significance for corporate applications. It is a marketplace created by digital technology and obviously the Internet plays a crucial part. *Digital goods* like books, CDs, music, newspapers, databases, travel services, and magazines are the first to be sold over the Internet. The *digital marketplace* is a marketplace for digital goods and of significance to IT industry. With the success of Dell Computers' experience with direct selling of PCs over the Internet (now followed by IBM, HP, Compaq, and Acer), the digital marketplace is *not* limited to digital goods alone. With several portals (horizontal portals like Yahoo! and vertical portals like e-AutoMart) and storefronts appearing on the Internet everyday, the digital marketplace is influencing every other industry.

J. Mobility Is the Watchword

With its inherent globalization, the Internet is permitting many users to be mobile. With its ability to send and receive information during movement, mobile computing is introducing a "paradigm shift" in corporate applications whose real impact will be felt in the next two years. Mobile computing and mobile Internet access have suddenly started to influence corporate applications. Thanks to explosive growth of NTT DoCoMo mobile Internet service (with 1.4 million customers in 18 months), applications have to address ways of delivering information over mobile devices. The added complication is the multiplicity of promising technologies, with no clear indication of any one of them dominating. For example, in the access area analog mobile phones dominate the United States market; GSM dominates Europe and much of Asia; a range of technologies (UMTS, GRMT) are under deployment in Japan and Europe; and there is a promise of 3G (third generation mobile) that is likely to be around by the year 2003 throughout the globe. Then there are intermediate technologies like Wireless Application Protocol (WAP). On the devices front, there are competing technologies such as low-end handheld devices from 3Com and SONY (powered by Palm OS), high-end handheld devices from Compaq and HP (powered by Pocket PC and Windows CE from Microsoft), and technologies like Blue Tooth that promise extremely low-cost (less than $10) and permit wireless access within short distances (within 100 feet) across a variety of devices (computers, phones, fax machines, and even refrigerators) and offer further avenues to exploit information delivery.

There are significant strengths and weaknesses of these products that address different user needs (such as salespersons, managers, senior executives). Being nascent the devices are all evolving with continuous improvement in terms of battery life, display quality, display size, and color capability. The challenge for corporate applications is to constantly watch and continuously improve the information delivery mechanisms to address this fundamental change in user behavior.

K. With Enough Data to Warehouse and Cheap Processing Power OLAP Is a Reality

Thanks to ERP deployment over several years in corporations, enough corporate data has been archived that is structured, machine readable, accurate, and authentic. With enterprise systems being transaction oriented, the data captured has sufficient "metadata" information (organizational units, time stamp, account codes, customer profile, batch size, etc.) that is "hidden" but recoverable through sophisticated "data cleansing" and "transformations" that modern data warehouse engines can perform. With the sophistication of OLAP

tools, visualization tools, and data mining tools and the increased processing power of the corporate desktop, OLAP is a reality today. Managers with sufficient analytical ability can routinely perform sophisticated analysis right from their desktop computers without needing an analyst to assist them. The "what-if analysis" need not be a mere "simulation" or "scenario planning" but use "live" and "real" data captured for years through OLTP systems (that are part of enterprise systems). This offers an unprecedented ability to the end managers to go past "information" to "insight" into corporate performance.

L. The Power of the Objects

"Best of breed" is an oft-quoted term in the ERP world. All the ERP software vendors claim to have built in their software products the "best practices" for all the major organizational functions—order processing, procurement, manufacturing planning, transportation, invoicing, etc. Every one of the products is a "tightly integrated" system that addresses all the needs of a typical enterprise: an automobile plant, an oil refinery, a supermarket chain, an airline, a seaport, or even a courier company. The tight integration gives lots of value to the ERP software products. The "organizational processes" have been mapped into the world view of the ERP vendor so as to get over the nightmare associated with lots of home-grown software that are "sewed up" with individual functional modules. The tight integration also limits any user to single ERP software for all its functional needs. Due to historic reasons each of the ERP software "fits" some particular "function" exceptionally well—SAP for logistics, Oracle for financials, BaaN for manufacturing and PeopleSoft for human resources modules. If a user wants the best of the breed among ERP software for individual modules, for example, human resources, module of PeopleSoft and manufacturing module of BaaN, no consultant would advise him to implement both the ERP products. The reason is quite simple, the ERP software products are complex to understand, internalize, and deploy. Since the world views of individual ERP software vendors are different, integrating them to get the best of the breed advantage may turn out to be a nightmare. The training needs alone can be very expensive. Accordingly in the trade-off between "individual module lack of fit" and the "complexity of integrating multiple products," most users decide in favor of a single product. Unusually enterprising organizations take the risk of integrating and go for multiple ERP software products. Not every organization may be in a position to take such risks.

One possible solution to this problem would be the arrival of object technology. The complexity of the ERP software is due to the monolithic nature of the ERP software code. Object technology provides an ideal solution conceptually. Every software vendor is attempting hard to "componentize" the code. SAP (BAPI), BaaN, and Oracle have made considerable progress in this direction; yet it is not ready for deployment today.

The trade-off gets compounded as we move from enterprise to an extended enterprise. Organizations are realizing the value of "strategic sourcing" and are moving toward integrating suppliers within the broad concept of extended enterprise. In logistics-dominated industries such as FMCG industry, keeping short supply chains holds the key to cost containment. Naturally there are unusual opportunities in optimizing across manufacturing and supply. Manufacturing execution systems from i2 Technologies and Manugistics compellingly prove this point beyond any ray of doubt. The true power of ERP is realized when organizations integrate it with SCM. The secret behind such extension once again is the use of objects.

Another extension toward extended enterprise is the CRM. CRM is a way to realize the dream of one-to-one marketing and to use information imaginatively to target customers. Thanks to the technology of data warehousing and data mining, tools are available today in the market that permit any organization to deploy the CRM technology. Of course, the organizations should work with a vision to improve customer relationships. The churn analysis in the telecom industry, the market-basket analysis in consumer goods industry, and buying behavior in supermarket chains are telling examples of the power of CRM. ERP coupled with CRM is yet another way to realize the dream of extended enterprise.

Yet another proactive way to improve customer relations is the area of SFA. Thanks to enormous technological progress in call center, web, and database technologies, SFA today is a very powerful way to identify, retain, and delight the customers. Coupled with ERP, SFA can dramatically improve customer relations. Once again component technology (for example, Siebel 99 components) played a decisive role in the ability of software to interoperate.

M. Single "Core" to Multiple "Core" Competencies

A difficult decision for progressive organizations is the way to approach the individual technologies of

ERP, SCM, CRM, and SFA. Historically these technologies have come from different vendors. However, realizing the stagnation in ERP growth, the ERP vendors over the years have embarked on offering all these services as an "integrated suite" of services. The individual software vendors of SCM, SFA, and CRM also offer ways of "working with" every ERP software product. For the end users, it is a difficult choice, either to work with multiple vendors or to take the "suite" from ERP vendors. The "core competency" of ERP software vendors is very much the "enterprise" function and it is not clear whether their attempts to provide a full suite of services will be entirely successful. The CRM, SFA, and SCM vendors "co-working" with all the ERP software vendors poses another set of difficulties. For the end users the advantage of best of breed amongst SFA, CRM, SCM, and ERP can easily be offset by the drawback of "integrating" multiple products. Once again the promise of "enterprise component technology" is there to make this dream a reality. As of now the advice is clear: if the end users want a "headache-free" solution they should stay with the ERP vendor and their limited offerings in areas of SCM, CRM, and SFA. If the end users are enterprising and willing to take the risk, they should try the best of the breed approach and choose the best among CRM, SCM, and SFA individually and integrate with their ERP software. The risk is well worth it in terms of rewards but it is *not* for the fainthearted. As object technology matures over the next year the picture could be completely different both for the enterprise and the extended enterprise.

N. ERP Verticals

In the recent years ERP software vendors have partially addressed this problem by the introduction of ERP verticals. Typical solutions are specific to vertical market segments such as oil, automotive manufacturing, banking, telecom, food and beverage, media, government, etc. These are repackaged solutions based on extensive experience gained by a specific software vendor through dozens of implementations in many firms that are key players in a chosen industry. Some ERP software vendors are more successful in specific industry segments, for example, SAP in oil industry, BaaN in discrete manufacturing, and Oracle in telecom. Such re-packaged solutions lead to significant gains in implementation time and quality. However, they continue to maintain the "plain vanilla" nature of the ERP software by way of addressing mainly the "common business processes."

O. ERP and "Niche" Enterprise Software

For sustained competitive advantage firms should start leveraging the "special processes" that give distinct competitive advantage. Such an activity must be driven by the "core competence" of the firms and not by ERP software vendors alone. For example, many firms have core competence in product design, development, deployment, and maintenance. Industries in this segment would include shipbuilding, machine tools, capital goods manufacture, aircraft manufacturers, railway equipment manufacturers, power plant manufacturers, etc. In these industries product development is the key. Firms in these industries may find current generation of ERP software addressing only the peripheral functions. Engineering designs and project management that are generally outside the ERP software must start driving the enterprise; mere importing of product data from design software (AutoCAD/Pro Engineer/UG II) or the import of ERP data into project management software such as Primavera would not be sufficient. Design and development processes must be integrated into the very core of the organizational business processes. This would imply design data (including 3-D data, rendering, surface and machining characteristics) must be integrated into basic workflow, viewing, searching, version control, and access control. Current generation of ERP software does not implement all these, though they would support all these functions. Once again, design-focused companies would need very sophisticated product data handling for lifetime support, warranty calculations, etc. The emerging area of product data management (PDM) addresses these issues; but PDM alone would not be sufficient to meet the enterprise needs. ERP software vendors will not be able to provide full PDM functionality, though many of them provide very limited PDM functionality. What is called for is the next generation of ERP software that truly integrates such core functionality specific to the engineering industry. Such "PDM-enabled ERP" would be "engineer's ERP," quite different from the current plain vanilla ERP that is practically an "accountant's ERP."

One could cite many similar examples. Many airlines have implemented ERP; but their core functions such as "seat reservation system" continue to be outside the main ERP. To fully leverage their operations the airline industry would need a "seat reservation enabled ERP." Similarly mining industry would need "mine planning enabled ERP," and refineries would need "process control enabled ERP." In all these cases the firms would depend heavily on their core competencies and standard

ERP solutions that address only the common business processes would not give sustained competitive advantage. That is the place for the next generation of "beyond plain vanilla ERP."

P. E-ERP

Like every other software product, ERP software is transforming itself to address the needs of the e-business environment. This includes

- Interface improvements like web-enabling
- Extension to other parts of extended enterprise such as suppliers and customers
- Provision for on-line payments
- Providing on-line sales and service support
- Making many services self-service mode
- Personalizing the software to suit individual user needs
- Support for mobile works through multiple technologies and multiple devices, and
- Internet-based software licensing in the form of ASP

X. CONCLUSIONS

ERP had a significant impact on corporate applications over the past decade. It is quietly getting transformed into the core of enterprise computing, thanks to the maturity of SCM and CRM. Billions of dollars have been invested in ERP infrastructure across many large corporations. In turn, such investments have raised great expectations. While there are a number of cases where ERP has delivered, there are a number of instances where the benefits have not been realized, mostly due to organizational issues than technical issues. It is good to learn from such mistakes and proceed with caution. With e-business taking center stage and as we move past the dot com fiasco, ERP in its extended form will play a central role in the form of a key information infrastructure of all future enterprises.

SEE ALSO THE FOLLOWING ARTICLES

Computer-Integrated Manufacturing • Operations Management • Productivity • Reengineering • Supply Chain Management • Total Quality Management and Quality Control

BIBLIOGRAPHY

Bancroft, N. (1997). *Implementing SAP R/3: How to introduce a large system into a large organization*. New York: Prentice Hall.

Davenport, T. (1998). Putting the enterprise into the enterprise systems. *Harvard Business Review,* July–August 1998, 121–131.

James, D., and Wolf, M. L. (2000). A second wind for ERP. *McKinsey Quarterly,* pp. 100–108.

Kumar, K., and van Hillegersberg, J. (2000). ERP Experiences and Evolution. *Communications of the Association of Information Systems,* Vol. 43, No. 4, 23–26.

Lozinsky, S., and Wahl, P. (1998). *Enterprise-wide software solutions: Integration strategies and practices*. Reading, MA: Addison-Wesley.

Plotkin, H. (1999). ERP's—How to make them work. *Harvard Management Update,* U99-3C.

Sadagopan, S., ed. (1998). ERP—*A managerial perspective*. New Delhi: Tata McGraw-Hill.

Sadagopan, S. (1997). *Management information systems*. India: Prentice Hall.

Watson, E. W., and Schneider, H. (1999). Using ERP systems in education. *Communications of the Association of Information Systems,* Vol. 1, Article 9, 1–48.

Ergonomics

Waldemar Karwowski *University of Louisville*
Francesca Rizzo *University of Siena*
David Rodrick *University of Louisville*

GLOSSARY

domain Knowledge, facts, concepts, and terminology of a specific field.
environment The situational and physical context in which information seeking takes place.
ergonomics The scientific discipline concerned with the understanding of interactions among humans and other elements of a system.
evaluation An assessment of a system with respect to some standard/goal or a comparison among alternative approaches.
human–computer interaction (HCI) The discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use.
information systems (IS) Technological systems that manipulate, store, process, and disseminate information that has or is expected to have an impact on human organized behavior within any real context of use.
interaction design A design process that incorporates the end-users in the human–computer interface development.
iterative design Simulation, prototyping, and evaluation of different design possibilities of an interface.
mental model Mental representation that the user possesses of the system with which he or she interacts.
usability engineering A discipline focusing on system usability design and evaluation realized by a set of activities that take place throughout the lifecycle of a product.

This article explores the nature of human information seeking behavior and discusses various models of the interaction between people and information systems. The discussion of the multidisciplinary approach to human–computer interaction (HCI) is followed by a recent approach of usability engineering. The emergence of usability engineering is viewed as a reaction of the HCI community to rapid developments in the World Wide Web. The article concludes with an overview of current research trends in information systems with the focus on user-related issues.

I. INFORMATION SYSTEMS

Information systems (IS) can be defined as technological systems that manipulate, store, process, and disseminate information that has or is expected to have an impact on human organized behavior within any real context of use. IS are powerful external cognitive artifacts that extend people's higher level cognitive capabilities. Such capabilities include information encoding, decoding, and storing, and information searching, retrieving, and sharing. Other human cognitive tasks facilitated by the use of IS are reasoning, thinking, learning, and problem solving. Typically, IS are used for supporting information processing tasks that range from the most simple, for instance, an information retrieval activity, to the most complex tasks imaginable. Whatever functions they offer and support, IS are principally built to provide information services to some particular class of users

in order to satisfy a variety of their information needs. The principal activity performed by users interacting with the IS is an information seeking task, a process driven by an information problem. Examples of typical information problems are monitoring a current state or situation, getting information to make a decision, looking for an accommodation, booking a flight, finding a plumber, keeping informed about a business competitor, satisfying a curiosity, writing a publishable article, or investigating a new field. Addressing an information problem must be initiated by someone with a conscious activity to reach an object or goal. A person engaged in seeking a solution to an information problem has one or more goals in mind, and uses an IS as a supporting tool. Information access tasks are used to achieve such goals (Table I shows the principal types of an IS and the main human cognitive activities that it supports).

Since the main purpose of IS is to provide the needed information to the potential users in the most efficient and effective way possible, the information systems must be designed, developed, and utilized with due consideration of human perceptual, cognitive, and emotional abilities, limitations, and needs. The scientific discipline that focuses on the above issues in relation to the interactions between technology and people is *ergonomics,* also know as human factors. This article discusses the role of ergonomics in the design and use of information systems, with the main focus on supporting a variety of human cognitive (information access, processing, and decision making) activities.

II. ERGONOMICS DEFINED

According to the International Ergonomics Association (www.iea.cc),

> ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data, and methods to design in order to optimize human well-being and overall system performance. Ergonomists contribute to the design and evaluation of tasks, jobs, products, environments, and systems in order to make them compatible with the needs, abilities, and limitations of people.

Ergonomics as a discipline promotes a holistic approach to human work design that takes into account

Table I Principal Types of IS and the Main Human Cognitive Activities Supported

Principal IS types	Human supported cognitive activities
Internet	Learning
Intranet	Sense-making
Database	Reasoning
World Wide Web	Problem solving
Search engines	Decision making
Virtual libraries	Goal shifting
Booking systems (passenger booking systems of airlines; the hotel booking systems, etc.)	Intentions redefining
Decision support systems (rule based systems to simulate reasoning, problem solving, etc.)	Actions redefining
Expert systems (tutorials, artificial agents, etc.)	Information seeking
Transaction processing systems (transition processing systems of banks, etc.)	Information retrieval
Shared knowledge systems (online and offline educational systems, training systems, knowledge organization repositories, etc.)	Browsing
Cooperative work systems (bulletin board, groupware applications, etc.)	Scanning
Control information systems (flight control systems of a modern airplane, etc.)	Analyzing
Hypermedia systems (hypermedia learning environments, etc.)	Navigating
	Monitoring

physical, cognitive, social, organizational, environmental, and other relevant factors. *Physical ergonomics* is concerned with human anatomical, anthropometric, physiological, and biomechanical characteristics as they relate to physical activity. *Cognitive ergonomics* is concerned with mental processes, such as perception, memory, reasoning, and motor response, as they affect interactions among humans and other elements of a system. *Organizational ergonomics* is concerned with the optimization of sociotechnical systems, including their organizational structures, policies, and processes. Relevant topics include communication, crew resource management, work design, design of working times, teamwork, participatory design, community ergonomics, cooperative work, new work paradigms, virtual organizations, telework, and quality management.

III. MODELS OF HUMAN INFORMATION SEEKING

Although the goals of an information access process (IAP) differ depending on the characteristics of the users and the context and nature of the information problem, most IS have been developed according to the findings of normative theories about human information processing and problem solving. A brief historical overview of the principal paradigms offered by these theories is provided below.

As early as 1960, Simon distinguished three main categories of human problem solving and decision making activities: (1) intelligence, (2) design, and (3) choice. Intelligence is related to the problem identification, diagnosis, and definition. Design refers to those activities inherent in generating alternative solutions or options to solve the problem. Choice refers to those human activities that are inherent in evaluating and selecting a single choice from all possible alternatives. In 1980, Huber defined the conscious process aimed at understanding the current situation and explained why it does not always fit with a person's desires. He developed a 5-step model of problem solving: (a) problem identification, definition, and diagnosis; (b) generation of alternative solutions; (c) evaluation and choice among the options; (d) implementation of the chosen alternative; and (e) monitoring of the implemented action. In 1981, Wohl elaborated on the above paradigms in the context of tactical (military) problem solving that expands on the models Simon and Huber. The Wohl model consists of four steps: (a) stimulus that is inherent to data collection, correlation, aggregation, and recall activities; (b) hypothesis that involves creating alternative options to explain the possible causes of the problem, evaluating the adequacy of each hypothesis, and selecting one; (c) options that follow the hypothesis step with regard to what is happening and its implications, i.e., the judgment about the situation affecting the decision of what to do; and (d) response that is related to the implementation of a chosen plan of action.

The human information seeking models described above suggest that if there is a discrepancy between the actual situation and the desired one, a person returns to the first step in order to explore the problem again. Table II shows the relationship between the steps utilized by the models considered above, and the corresponding cognitive activities they require.

A. Basic Model of Human–IS Interaction

Salton noted in 1989 that the interaction style of the process of information access can be described in terms of query specification, information receipt, examination of retrieval results, and then either stopping or reformulating the query and trying the process many times until a fitting result set is obtained. A standard information process access can be described as follows:

(a) Identification of an information need
(b) Selection of information sources
(c) Query formulation
(d) Sending the query to the system
(e) Getting results in the form of information items
(f) Examining, interpreting, and evaluating the results
(g) Reformulating the query or stopping the searching

B. Advanced Models of Interaction

While useful for describing the basics of the information seeking (access) process (ISP), the simple interaction model (shown in Fig. 1) contains an underlying assumption that the user's information needs are static, and that the ISP consists of successively refining a query until it retrieves all and only those documents relevant to the original information needed. For this reason, many reviews of the subject literature focus on the iterative, uncertain, and unpredictable nature of these tasks.

Table II Relationships between Different Models of Human Information Seeking and the Cognitive Activities They Require

Generic steps of the models of Simon, Huber, and Wohl	Required cognitive activities
Intelligence	Data gathering
Stimulus	Data detecting
Problem identification, definition, and diagnosis	Filtering Correlation Aggregation Displaying Storing Recall
Design	Creation
Hypothesis	Evaluation
Generation of alternative solutions	Selection
Choice	Creation
Option	Evaluation
Evaluation and choice among the options	Selection
Response	Planification
Implementation of the chosen alternative	Organization Execution
Intelligence	Data gathering
Stimulus	Data detecting
Monitoring of the implemented action	Filtering Correlation Aggregation Displaying Storing Recall

1. Belkin's Model

The Anomalous State of Knowledge (ASK) model developed by Belkin in 1980 is the most often cited model of human information seeking. According to this model, an individual queries an information system in order to remedy a personal knowledge anomaly. The anomaly is solved when the information retrieved from the system satisfies the individual's anomalous state of knowledge. Practically, an information seeker in this model is concerned with a problem, but the problem itself and the information needed to solve the problem are not clearly understood. The information seeker must go through a process of clarification to articulate a search request. In this context, an IS should support iterative and interactive communication flow with the user. The ASK model characterizes how user-perceived needs change during the search process as a consequence of retrieved results feedback.

2. Kuhlthau's Model

In 1993, Kuhlthau developed a six stage model of the human information search process (ISP). The model includes human feelings, thoughts, actions, and strategies through six stages: (1) task initiation, (2) topic selection, (3) exploration, (4) focus formulation, (5) information collection, and (6) search closure. At the first stage of the model, an individual becomes aware about the lack of information he or she needs in order to understand a problem or perform a specific activity. Feelings of uncertainty and apprehension are the common state associated with this stage. The second stage involves *identification and selection* of the general domain to be analyzed. The third stage, *exploration,* is usually the most difficult step for the user because the uncertainty and doubts frequently increase. This is also the most difficult stage for the designers of IS because the ambiguity about what information is needed makes it difficult to support the

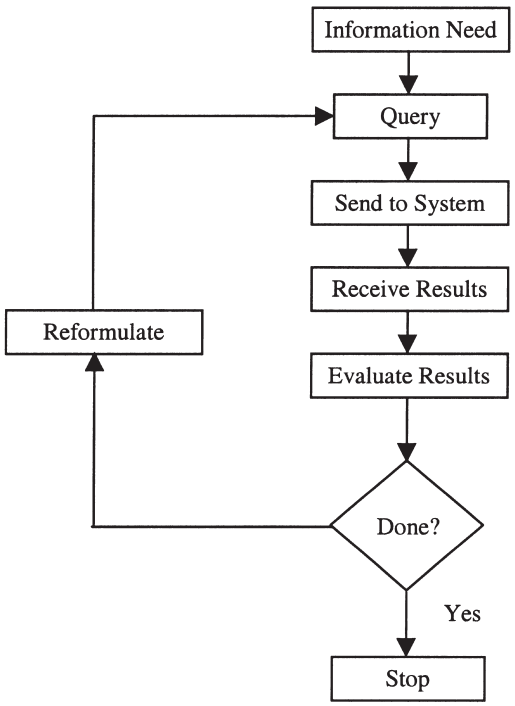


Figure 1 Basis model of the information access process.

user–IS communication. The fourth stage of the model is *formulation,* a turning point of the process when feelings of uncertainty diminish and understanding increases. At the fifth stage of the model, an interaction between the user and an IS is most effective and efficient as the main task is to gather information pertinent to the problem. The last stage of the model, *presentation,* includes completion of the search and solving the initial problem.

3. Ellis' Model

Ellis, in 1989, proposed a general model of information seeking behavior based on the information seeking patterns of different categories of scientists. The model describes six categories of information seeking activities: (1) starting, (2) chaining, (3) browsing, (4) differentiating, (5) monitoring, and (6) extracting. During analysis of the initial sources, it is possible to get suggestions, recommendations, and additional references. The next step is the activity of *chaining.* Chaining can be backward or forward. Backward chaining is accomplished by following the pointers (or indications) from the initial information source. On the other hand, the forward chaining identifies and follows up on other sources that refer to an initial source or a document. The third category in the general model of information seeking behavior is *browsing.* This step refers to looking for information at different microlevels. *Differentiating* activities are those that filter and select from among the sources scanned by noticing differences in the nature and quality of the information offered. Most of the time, the differentiation process depends on the person's previous knowledge and experience with the source. *Monitoring* is the activity of paying attention to the developments of an area by specific sources. An individual monitors by concentrating on a small number of what are perceived to be the core sources. The last stage of the model, *extracting,* focuses on working on a particular source or sources in order to identify something interesting.

4. Bates' Model

In 1989, Bates developed the "berrypicking" model of information seeking based on two main observations. The first is that, as a result of reading and learning from the information encountered throughout the search process, the users' information needs and their queries continually shift. The information discovered at one point in the interaction may suggest revising the direction. The origin goal may be changed in favor of another one. The second point is that the users' information needs are not satisfied by a final retrieved set of information, but, on the contrary, by a series of selections and bits of data collected along the way.

5. Marchionini's Model

Marchionini, in 1995, proposed another model of the information seeking process (see Fig. 2). In his model, the ISP is composed of eight subprocesses developed in parallel: (1) recognize and accept an information problem; (2) define and understand the problem; (3) choose a search system; (4) formulate a query; (5) execute a search; (6) examine the results; (7) extract information; (8) reflect/ iterate/ stop.

These and other models of information seeking behavior share a similar perspective on the information access process as a refining activity that depends on the interactions between all the information seekers and the IS. For example, the navigation and hyperlinks are relatively new tools supporting the information access processes. The growing importance of the end-user needs and requirements, confirmed by the studies on user behavior, and the strategic relevance of the customization of the information call for a user-centered design prospective in developing contemporary information systems.

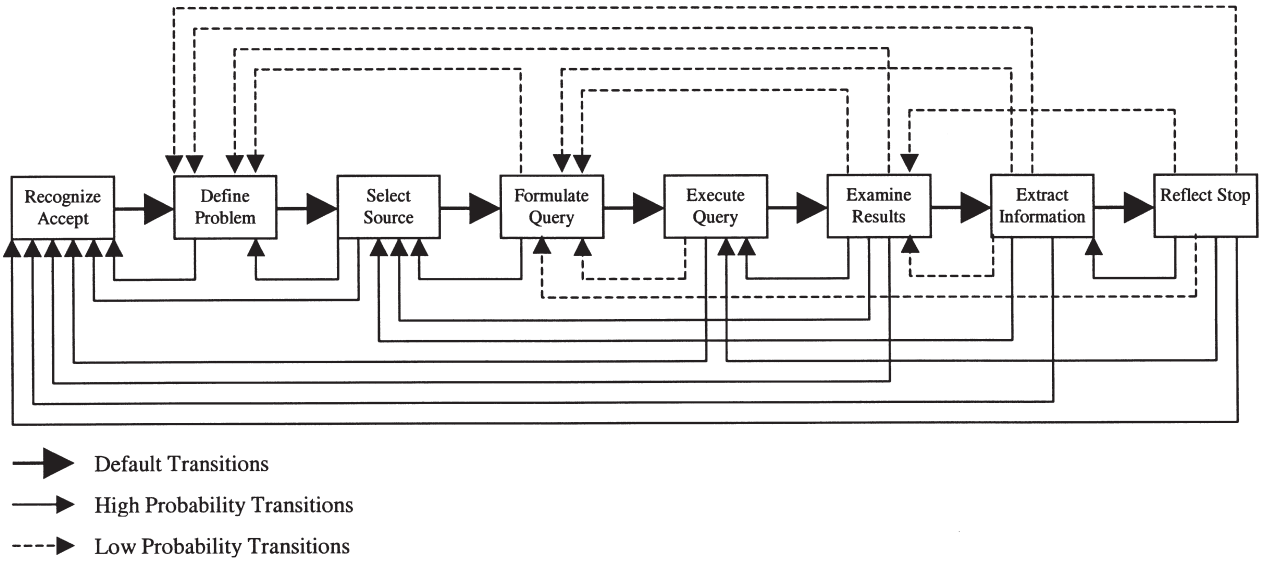


Figure 2 Marchionini's model.

C. Information Needs

An information need was defined by Belkin in 1980 as an anomalous stage of knowledge or a lack of knowledge. As noted by Marchionini in 1995, people feel a gap between what they know and what they need to know in order to make sense of the specific situations. Wilson stated in 2000 that an information gap may be manifested in a range of behaviors, from the goal-directed information searching to more exploratory information browsing. Consequently, one of the most important problems is how to implement an IS that would support people engaged in seeking a solution to an information problem, in order to change their state of knowledge.

Very often, the IS are designed for a prototypical user. This design information decision is an implicit assumption that individual differences can be ignored in favor of some central tendency model of the information system-user behavior. Contrary to this point of view, potential users face the IS at various stages of knowledge, skill, and ability. Some user may be a novice who has never experienced the system before. Other users may be an expert who is familiar with most of the system's features and capabilities. Others may apply previous knowledge and experience with similar devices in order to interact with the new one. However, most users are somewhere in between. They forget, they learn, relearn, and then learn again. In addition, typical users' information needs differ greatly depending on the specific circumstances and previous domain knowledge.

Dynamics in the use of information creates a need for development of adaptive IS. In this context, research results from the human–computer interaction studies and application of ergonomics research methodologies are critical to the design of adaptive (interactive) computer interfaces. Such interfaces can support the users in a continuous communication with a variety of information systems by tracking relevant interactions. The strong interactive style of interaction allows for highly adaptive systems.

IV. HUMAN–COMPUTER INTERACTION

A. HCI

The field of human–computer interaction (HCI) is concerned with creating usable computer systems. There are two general definitions of HCI on which practitioners in the field agree upon. The first one comes from The Curriculum Development Group of the Association for Computer Machinery (ACM) Special Interest Group on Computer–Human Interaction (SIGCHI) and states that the "Human–Computer Interaction is a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." The second definition by Dix *et al.* in 1993 defines HCI as "the study of the people, computer technology, and the way these influence each other. . . to determine how we can make this computer technology usable by people."

Both definitions stress two main dimensions of HCI. The first one points out that in order to build usable and human-centered technologies, it is necessary to study the users' roles, needs, previous experiences, and the real context of use. The second one considers HCI as the discipline that focuses on human cognition and behavior and applies relevant knowledge to the development of cognitive artifacts (see Table III).

One of the important forms of HCI theory is the user-centered design approach (see Fig. 3). The following are the main principles applied to a user-centered design approach as noted by Gould and Lewis in 1983:

- Understanding users. It is important to have an explicit representation about users' cognitive competences and attitudes, and an explicit representation of the cognitive nature of work that people have to perform.
- Interaction design. The end-users have to be engaged in the team work.
- Evaluation. From the beginning of the design process, the system must be evaluated in terms of the human–machine interactions.
- Iterative design. The design–evaluation–redesign process should continue until performance of the system conforms to the prescribed system usability goals.

B. Understanding Users

Collecting knowledge about users, their activities, and the context in which these activities take place is the central phase of the user-centered system design. This implies defining the characteristics of the user population and working with a representative sample of the user group. Different methodologies can be used for gathering information about the users' needs and requirements. One of the more powerful methodologies is the ethnographic methodology. This approach consists of analyzing humans performing their activities "in situ." Other helpful tools include interviews, surveys, data collection, questionnaires, focus groups, or brain storming. Whatever approach is applied, at the end of this phase the designers will collect enough data to:

- describe the activities that the new system must support
- individualize the goals of the activities
- define procedures and modalities
- define the scenarios of use

Table III IS Components and Related Human Factor Issues

Information system components	Knowledge basis	Search system (algorithms, procedures,. . .)	Interface
	Knowledge management	User studies	Human computer interaction
	Representation of knowledge	User modeling	Cognitive ergonomics
	Knowledge elicitation	Mental models	Activity theory
	Organizational memories	Problem solving	User studies
Human factor issues related	Cognitive sciences	Reasoning	Distributed cognition
	Cognitive psychology	Information processing	Situated cognition
	Social psychology	Memory	Organizational studies
		Attention	Mental models
		Representation of knowledge	Information design
		Human information strategies and tactics	Content design
			Contextual design
			Interaction design
			Usability
			Preferences
			Perception
			Visualization
			Anthropometrical issues

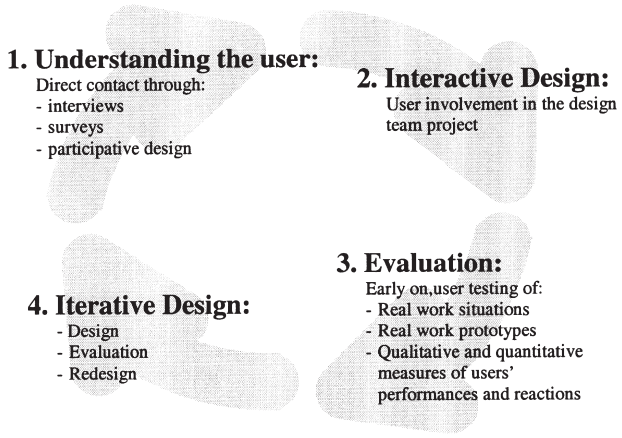


Figure 3 User-centered design process.

- understand the problems that users faced with other similar systems

C. Interaction Design

User involvement in system development is becoming more salient due to the fact that this can lead to better designed products from the perspective of the customers. One way to secure this is for designers to work in conjunction with the users, enrolling them early on in the development process, when their contributions to the system design are thought to be fundamental. Furthermore, user involvement allows for obtaining sufficient information about the initial system requirements, for assessing if a product meets the end users' requirements and needs, and for gathering data for the next version of the design.

D. Evaluation

The evaluation phase is fundamental in order to measure if the users' requirements have been satisfied. It can be conducted early in the cycle of design, i.e., formative evaluation, or it can take place at the end of the cycle, summative evaluation. Currently, there are three main evaluation methods suggested in the HCI field: (1) the cognitive walkthrough, (2) usability testing, and (3) heuristic evaluation. Each of these evaluation methods is briefly discussed below:

1. Cognitive Walkthrough

As described by Pejtersen and Rasmussen in 1997, cognitive walkthrough is based on the assumption that a design should support learning by exploration. This evaluation technique evaluates the ease of learning a design with reference to accepted attributes that are correlated with ease of learning. Central to this evaluation process is the identification of mismatches between the users' and designers' conceptualization of a task. This is done by simulating the user's problem-solving process and tracking the correct actions that a user takes without problems imposed by the interface.

In this method, an individual or a group of experts takes part in three stages of analysis at any design phase using a mock-up, prototype, or a complete interface. These stages are: (1) the walkthrough input, (2) the action sequences, and (3) design revisions following the evaluation.

2. Usability Testing

Usability testing with real users is the most fundamental usability method, since it provides direct information on the usage and related problem of any interface. Two of the methodological pitfalls of usability testing are (1) problems associated with reliability (mostly due to huge individual differences and choice of level of confidence), and (2) problems associated with validity (requiring inclusion of right users and tasks by considering time constraints, social influences, and other confounding variables).

A typical usability testing begins with test goals, plans, and a budget, followed by a pilot test. One of the critical factors in usability testing is getting the users. There is always a dichotomy between: (1) novice and expert users, and (2) between- and within-subjects design. Selection of appropriate test tasks is also important step. Nielsen in 1997 suggested a four-stage usability test. These stages are: (1) preparation, (2) introduction, (3) the test, and (4) debriefing. In usability testing, performance measurement is done by breaking down the abstract nature of "goal" into more concrete "components." Each of the components is then quantified precisely. Given the quantification of a component, a method for measuring the user's performance needs to be defined. Finally, actual activities to collect data are defined. Although a great variety of quantifiable usability techniques exists, many usability studies use qualitative data that require substantial usability engineering experience.

3. Heuristic Evaluation

One of the analytical evaluation techniques is heuristic evaluation, also called "discount usability" by Pejtersen and Rasmussen in 1997. It is a systematic in-

spection of a user interface where the goal is to find the usability problems in a user interface design so that they can be attended to as part of an iterative process. This involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles or heuristics. Nielsen in 1997 recommended using about five, and certainly at least three, evaluators in such an evaluation process. Heuristic evaluation is performed by having each individual evaluator inspect the interface alone. An evaluation session lasts one or two hours during which the evaluator goes through the interface several times and inspects various dialogue elements and compares them with a list of recognized usability principles (be it general, category-specific, or product-specific). Following the completion of individual evaluations, the evaluators interact with each other and the findings are aggregated in the form of a written report. Sometimes an observer is used that usually adds to the overhead of each evaluation session and at the same time reduces the workload of an evaluator.

E. Iterative Design

The implementation of an iterative process involves narrowly defined different design solutions used to implement the main interface functions and requirements. This process can be articulated in three phases, as follows:

- simulation of different design possibilities *(low fidelity prototype)*
- prototyping evaluation from the initial phases of the system development
- iterative prototyping (design–evaluation–redesign)

Prototypes allow designers to try and test the interface before the final system has been constructed, define the explicit users' requirements, develop these requirements experimentally, discover design problems early, and evaluate if the system meets the users' needs. Different methods can also be used in the prototype development, including the mock-ups, scenarios, video simulations, computer simulations, or storyboard.

F. Factors That Affect the Human–IS Interaction

Interacting with an IS depends on several factors, such as the information seeker's needs and features; the user mental model of the system, tasks, search system functions, and characteristics; the user domain knowledge; and system domain knowledge, setting, and system outcomes (see Fig. 4).

G. Information Seeker

The information seeker is the user of the system. The user identifies all the tasks that are relevant to perform his or her plan of action; monitors the interaction with the system, evaluates and retrieves the most relevant findings, judges the progress of the interaction, and decides if the information seeking process is complete. The information seeker is motivated by an information problem or a need that activates a variety of memory traces. This information and related relationships help the user to define the problem and to develop a possible plan of actions. Broad characteristics of users include previous experiences in the domain of knowledge, user's preferences and abilities, and mental models of the system and a mental model of the problem which he or she is facing. In addition, it is important to consider that the information access process takes place in a situated context that at any moment affects human performance and cannot be neglected.

H. Task

The task is the manifestation of an information problem in a plan of action. Usually an information access task includes a more or less detailed articulation of the problem, depending on the depth of understanding by the users, and the level of ambiguity of the current situation. The task includes all mental and physical activities performed by the user interacting with a system, and in reflecting on the outcomes. Tasks

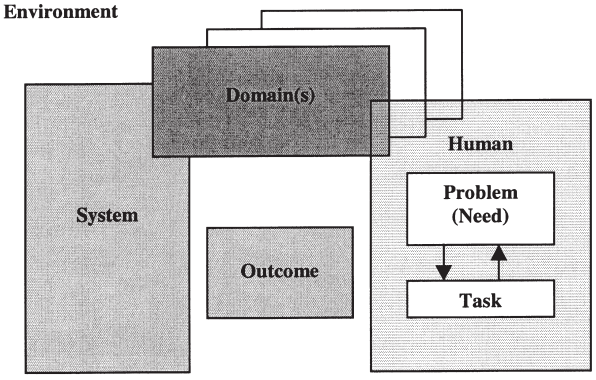


Figure 4 Factors affecting information-seeking process.

that are explicitly goal oriented can take a chance according to the search progresses and/or the user evaluation.

I. Mental Models

The user's mental model of the system is the model of interactions developed by the user during different interactive sessions. The information seeker constructs and uses mental models of the IS to perform tasks. In addition, for every information problem, the information seeker has his or her particular representation developed from other experiences, setting, and domain.

J. Information System

The information system is the main source of information that is composed of rules that govern the access to the system, an interface by which the communication between the user and the system takes place, and various tools for the organization and representation of knowledge. The rules that govern the representation of knowledge and the access to the system influence the user's performances. The usability of an IS depends on the assumptions made by the designers about user information needs, features, and abilities. The interface determines how learnable, usable, and effective an IS can be. The type, quantity, and quality of information provided by the system determines how system designers organize, index, and display information. This consequently constrains the interface design and has a direct impact on the strategies and tactics applied by the user to access information.

K. Domains

The domain describes knowledge, facts, concepts, and terminology of a specific field. It can be referred to the concern of the IS and/or to the information held by the users. When the notion of the domain is linked to the users, it is possible to distinguish between knowledge of the domain and knowledge of the device. It is important to note that both can influence the style of the HC interaction. For instance, the high domain knowledge or knowledge of the facilities afforded by a particular system enables the users to access information more effectively and provides a richer set of concepts and terms of query formulation.

Domains are different in complexity, evolutionary status, and size. In addition, the amount of the level of articulation of the information varies across domains. Every domain has its rules and codes to organize, classify, and access information. As a consequence a domain has a strong influence on the IS interaction style.

L. Environment

The *environment* is the situational and physical context in which information seeking takes place. The concept of environment is referred to here as a series of elements: societal, contextual, situated, or organizational variables that may affect task execution. For instance, the execution of a task depends on a person or a team. A task can be performed in an asynchronous or synchronous way. People can be co-located or distributed in different places. The physical characteristics of the environment may constrain allocated time, the degree of distraction, or cost.

M. System Outcomes

System outcomes are the feedback from the system in terms of retrieved objects and traces of the overall process. In general, information gained from this type of outcome allows a user to evaluate the system response and this evaluation changes the seeker's knowledge and most of the time determines if a specific information need has been satisfied or not.

V. USABILITY ENGINEERING

Usability engineering is a set of activities that ideally take place throughout the lifecycle of the product. Since usability is not a single, one-dimensional property of a user interface, five attributes that constitute usability can be distinguished. These attributes are: (1) learnability, (2) efficiency, (3) memorability, (4) errors, and (5) satisfaction. Nielsen in 1997 identified the unique constituents and methodology for each of the life cycle stages of product usability. Table IV summarizes these stages, constituents, and methods.

Godd *et al.* in 1986 distinguished seven steps of the basic usability engineering process. These steps are:

1. Define measurable usability attributes.
2. Set the quantitative levels of desired usability for each attribute. Together, an attribute and a desired level constitute a usability goal.
3. Test the product against the usability goals. If the goals are met, no further design is needed.

Table IV Summary of Lifecycle Stages of Product Usability

Lifecycle stages	Constituents	Methods
1. Know the user	Define the user	
a) Individual user characteristics	Work experience Educational levels Ages Computer experience Reading and language skills Time available for learning Opportunity for attending training courses Work environment Social context	Behavioral research techniques
b) Task analysis	Users' overall goal Approach to the task Information needs Responses to exceptional situations or emergencies Users' model of the task Identify effective users and user strategies Identify the weakness of the current situation	Checklist Systematic observation Interview Hierarchical task analysis Cognitive task analysis
c) Functional analysis	Identify underlying functionality of the task Surface procedures	Observation
d) International use	Consider the foreign users if the product is intended to be internationalized	
2. Competitive analysis	Old product as first prototype Competing products as second prototypes (usually 3 or 4)	Comparison of the prototypes
3. Goal setting	Learnability Efficiency of use User error rate Subjective satisfaction	Quantitative measurement Different levels of performance can be specified
a) Parallel design	Explore different design alternatives (typically 3 or 4) Independent work of each designer to have most diversity Diversified parallel design	Comparison among alternative designs Essentially important for novel systems
b) Participatory design	User involvement and participation	Joint application design Interview
4. Coordinated design of the total interface	To achieve consistency	General interface standards Ad hoc standards Formal coordination activities
5. Heuristic evaluation	Systematic usability inspection of the product/system Determine good and bad aspects of the interface	Visibility of system status Match between system and the real world User control and freedom Consistency and standards Error prevention Recognition rather than recall Flexibility and efficiency of use Aesthetic and minimalist design Help users recognize, diagnose, and recover from errors Help and documentation

(continues)

Table IV *(continued)*

Lifecycle stages	Constituents	Methods
6. Prototyping	Prototyping at different stages Vertical prototyping Horizontal prototyping	Waterfall approach Wizard of Oz technique Simulation Mockups Forward scenario simulation
7. User testing	Selection of test users Testing of real tasks	Experimental/laboratory testing Ratings (for severity)
8. Iterative design	Design of alternative versions of product/interface following empirical testing	Thinking aloud technique User interaction sequences
9. Field testing	Gather information for future version or new product/interface	Interviews Questionnaires Observation Follow-up

4. If further design work is needed, analyze the problems that emerge.
5. Analyze the impact of possible design solutions.
6. Incorporate user-derived feedback in product design.
7. Return to Step 3 to repeat the test, analysis, and design cycle.

Wixon and Wilson in 1997 pointed out several major advantages of usability engineering. First, developers can agree on a definition of usability. Second, usability can be quantified, it is not just personal opinion. Third, usability can be put on an equal footing with other engineering attributes like reliability and performance. Fourth, usability problems can be prioritized as a function of their impact on usability goals. Finally, the goals are clearly separated from the methods and implementation.

A. User-Interface Standardization

The user-interface technology standards can be classified into two main types: (1) a de facto standard for user-interface implementation, and (2) a series of standards developed by the international or national agencies. The first types of standards are either provided by the software producers or defined by consensus within the software industry. Further, standards can be classified as technical and ergonomics. Table V summarizes principles associated with different types of standards, including the national/international ergonomics standards that are applicable to user-interface designs. Reed *et al.* in 1999 proposed a framework of incorporating standards in the HCI approach in designing computer systems. Figure 5 schematically depicts the proposed framework.

VI. CONTEMPORARY RESEARCH TRENDS

A. Factors Affecting IT End-User Satisfaction

One of the key issues in contemporary information systems is the end-user satisfaction. The end-user satisfaction model involves three basic elements: (1) perceived benefits, (2) organizational support, and (3) user background. These elements are affected by the specific organizational and psychological parameters (see Fig. 6). Based on a meta-analysis, positive support has been found for the varying degrees of influence of model variables on the end-user satisfaction. The most significant relationships are related to the user involvement in systems development, perceived usefulness, user experience, organizational support, and user attitude toward an IS.

B. User-Adapted Interface Design

User-interface adaptation has drawn considerable attention in recent years due to increased quality requirements of the interactive computer systems with respect to user abilities, requirements, preferences, and interests. Akoumianakis and Stephanidis in 1997

Table V Summary of Standards and Their Principles

Type of standard	Specific standard	Principles/topics covered
De facto	OPEN LOOK SAA/CUA User Interface	Controllability Memorability Consistency
	OSF/Motif	
International	ISO 9241 Part 10 Dialogue Design	Suitability for the task Self-descriptiveness Controllability Conformity with user expectations Error tolerance Suitability for individualization Suitability for learning
	ISO 9241 Part 11 Usability	Effectiveness Efficiency Satisfaction
	ISO 9241 Part 12 Information Presentation	Clarity Discriminability Conciseness Recognizability Legibility Comprehensibility
	ISO 9241 Part 13 User Guidance	Recommendations specific to prompts, status, feedback, error management, online help, help navigation, browseable, and context sensitive help
	ISO 9241 Part 14 Menu Dialogues	Menu structures Grouping and sequencing Menu navigation Option selection and execution methods Menu presentation
	ISO 9241 Part 15 Command Dialogues	Syntax and structure Command arguments Distinctiveness Abbreviations Function keys and hot keys Input output considerations Feedback and help
	ISO 9241 Part 16 Direct Manipulation	Characteristics and appropriateness of direct manipulation Metaphors Appearance of objects Feedback Input devices Pointing and selecting Dragging, sizing, scaling, rotating Manipulation of text objects Manipulation of windows
	ISO 9241 Part 17 Form Filling Dialogues	Appropriateness of form filling dialogues Form filling structures Layout Field, label lengths, and alignments Input considerations Alphanumeric text entry Choice entries

(continues)

Table V *(continued)*

Type of standard	Specific standard	Principles/topics covered
		Menus and buttons Controls Feedback Navigation
	ISO 13407 Human-centerd design process for interactive systems	Rationale for adopting a human-centered design process Principles of human-centered design Planning the human-centered design process Human-centered design activities
	ISO 14915 Software ergonomics for multimedia user interfaces	Part 1: Design principles and framework Definitions Application of ISO 14915 Overview of the parts of ISO 14915 Design goals and principles Multimedia design aspects Design and development considerations Part 2: Control and navigation Navigation structures and aids Common guidance on media controls Basic controls Media control guidelines for dynamic media Guidelines for controls and navigation involving multiple media Part 3: Media selection and combination General guidelines for media selection and combination Media selection for information types Media combination and integration Directing users' attention Part 4: Domain specific aspects Computer based training Computer supported cooperative work Kiosk systems Online help Testing and evaluation
	ANSI/HFES 100-199X (A revision of ANSI/HFES 100-1988 standard)	The principles adopted and topic areas addressed in this standard are approximately the same as those covered in ISO 9241 Parts 3–9, *Ergonomic Requirements for Office Work with Visual Displays* and in ISO 13406, *Ergonomic Requirements for Flat Panel Display.*
National	Human Factors Engineering of Computer Workstations	
	HFES 200-199X Ergonomic Requirements for Software User Interfaces	The objective is to be a compatible superset of 9241 parts 10–17. It covers all the dialogue techniques addressed in the ISO standards and in addition will provide recommendations and guidance for the new topics of Voice I/O, Color, and Accessibility.
	NCI/NIH/DOHHS Research-based Web Design and Usability Guidelines	Design process Set and state goals Set performance and/or performance goals Share independent design ideas Create and evaluate prototypes Design considerations Level of performance User's workload Consistency

(continues)

Table V *(continued)*

Type of standard	Specific standard	Principles/topics covered
		Feedback to users
		Use of logos
		Minimization of page size
		Minimization of frames
		Content/content organization
		Level of importance
		Useful content
		Hierarchy of information flow
		Sentence/paragraph length
		Printing options
		Titles/headings
		Page titles
		Well-designed headings
		Page length
		Determination of page length
		Scrolling vs paging needs
		Page layout
		Alignment of page elements
		Level of importance
		Consistency
		Minimization of unused space
		Hierarchy of information flow
		Format of efficient viewing

classified interactive computer systems as adaptable and adaptive systems. A system is called *adaptable* if it provides tools that make it possible for the end-user to change the system's characteristics. Conversely, an *adaptive* system can change its own characteristics automatically by making assumptions about the current user at the runtime. Earlier attempts to construct adaptable systems include OBJECTLENS by Lia and Malone in 1988, BUTTONS by MacLean *et al.* in 1990, and Xbuttons by Robertson *et al.* in 1991. These systems allow the user to modify certain aspects of their interactive behavior.

Akoumianakis and Stephanidis in 1997 developed the USE-IT system for automating the design of interactions at the physical level. USE-IT generates a collection of adaptation rules from three basic knowledge sources: (a) the user model, (b) the task schema, and (c) a set of platform constraints (i.e., interaction objects, attributes, device availability, etc.). Within the system, a data structure has been designed to: (1) facilitate the development of plausible semantics of adaptation at the lexical level, (2) allow unification of design constraints, and (3) enable selection of maximally preferred design options. The output can be subsequently interpreted by the run-time libraries of a high-level user interface development toolkit, which provides the required implementation support for realizing the user-adapted interface on a target platform.

C. Data Models and Task Complexity of End-User Performance

Data models are representation vehicles for conceptualizing user data requirements and design tools for facilitating the definition of data. The two classes of data models are: (1) logical/implementation, and (2) conceptual/semantic models. Logical/implementation models are of three classes, including: (a) hierarchical (HM), (b) network (NM), and (c) relational models (RM). The major conceptual models are: (a) entity-relationship (ERM), (b) semantic data (SDM), and (c) object-oriented models (OOM). All these models have different performance characteristics. For example, the OOM and NM models often outperform the RM and HM models in terms of comprehension, efficiency, and productivity. The extended entity-relationship model (EERM) was found superior to the RM model. Comparison of various models has shown that user performance is better with respect to comprehension, efficiency, and productivity when using the OOM

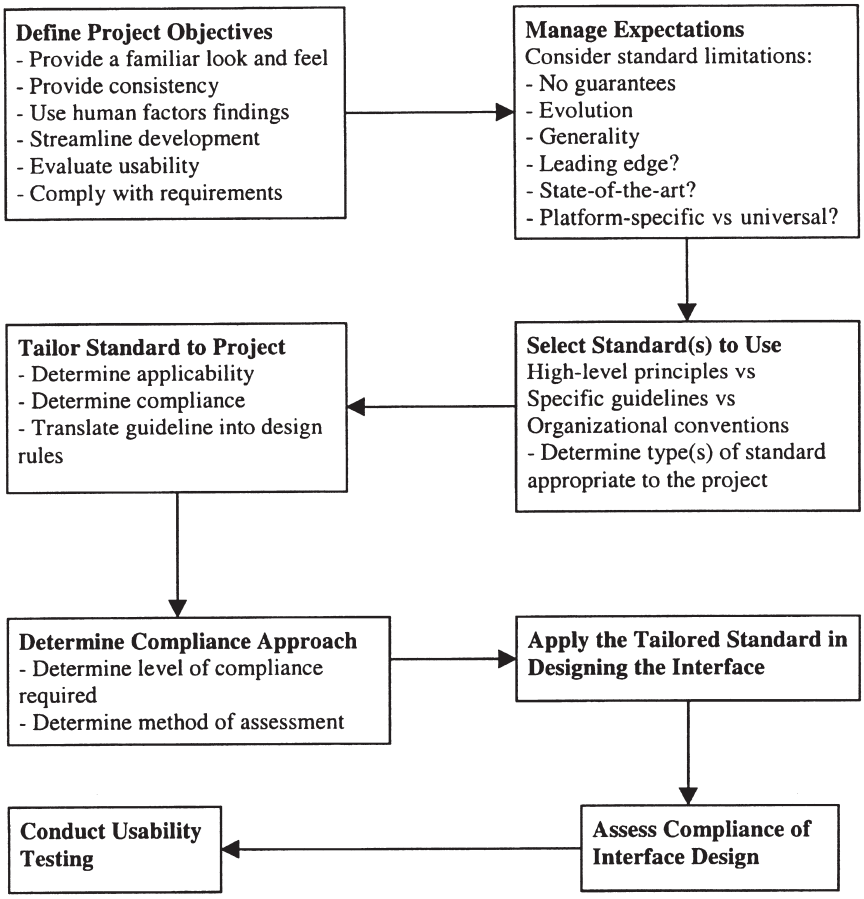


Figure 5 HCI framework for incorporating standards to design.

model rather than the data structure diagram or ERM model.

VII. CONCLUDING REMARKS

Typically, the information systems have been designed based on the principles of certainty and order. Consequently, users of these systems are expected to have clear representations of their needs and problems. It is often assumed that the users are able to externalize their lack of information through some predefined "request and query" scheme, which is compatible with the system's classification structure and interacting tools. Contrary to the above assumptions, people using the information systems often have to deal with unwanted uncertainty and noise that arise from the dynamic environment of daily living. By doing so, people can also learn during the information search process. Since the user's information needs, queries, and information seeking strategies and tactics often shift, the usable information systems should allow the users to understand available information presentation types (information architecture), relate information types to their requirements, and understand the contents of the system. Such systems should help the users to understand what kind of service is offered by the system, support user control, and reduce working memory load.

The usable information systems should be designed to assure full compatibility with a variety of users with different needs, and, therefore, be flexible in their functionality with respect to user control, error prevention and correction with informative feedback, and user guidance and support. The ergonomics discipline allows for the consideration and satisfaction of the above requirements at a very early stage of system design, with a clear focus on the potential users and their abilities and limitations, as well as their needs and expectations.

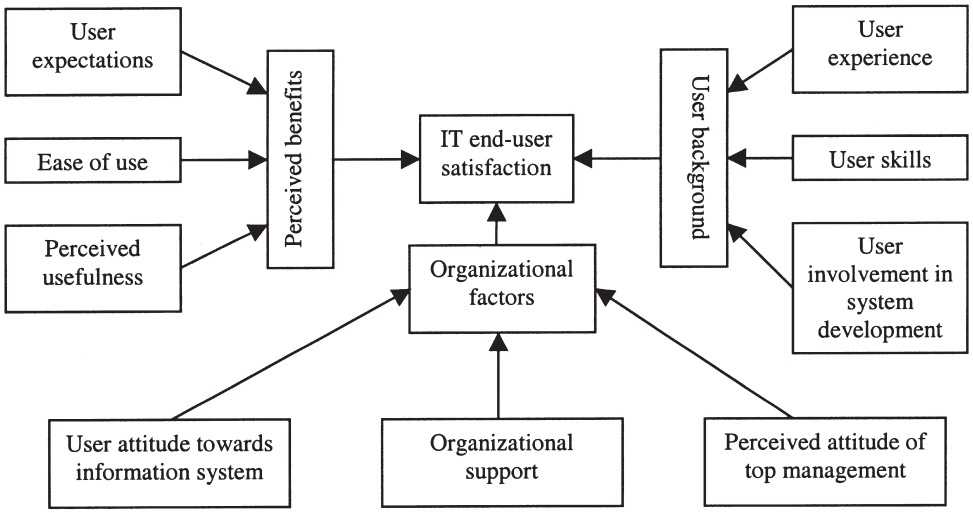


Figure 6 Factors affecting information technology end-user satisfaction.

SEE ALSO THE FOLLOWING ARTICLES

End-User Computing Concepts • Human Side of Information, Managing the Systems • People, Information Systems Impact on • Prototyping • Psychology • Quality Information Systems • Success Measures of Information Systems • Systems Implementation • Telecommuting • User/System Interface Design • Virtual Organizations

BIBLIOGRAPHY

Akoumianakis, D., and Stephanidis, C. (1997). Supporting user-adapted interface design: USE-IT system. *Interacting with Computers,* Vol. 9, 73–104.

Dix, A., Finlay, J., Abowd, G., and Beale R. (1993). *Human computer interaction.* Englewood Cliffs, NJ, Prentice Hall.

Dzida, W. (1997). International user-interface standardization, in *The computer science and engineering handbook* (A. B. Tucker, Jr., Ed.), 1474–1493. Boca Raton, FL: CRC Press.

Helander, M. G., Landaur, T. K., and Prabhu, P. V. (Eds.) (1997). *Handbook of human-computer interaction.* Amsterdam: Elsevier.

Karwowski, W. (Ed.). (2001). *International encyclopedia of ergonomics and human factors.* London: Taylor & Francis.

Karwowski, W., and Marras, W.S (Eds.). (1999). *The occupational ergonomics handbook.* Boca Raton, FL: CRC Press.

Kuhlthau, C. (1993). *Seeking meaning: A process approach to library and information services.* Norwood, NJ: Ablex.

Liao, C., and Palvia, P. C. (2000). The impact of data models and task complexity on end-user performance: An experimental investigation. *International Journal of Human-Computer Studies,* Vol. 52, 831–845.

Mahmood, M. A., Burn, J. M., Gemoets, L. A., and Jacquez, C. (2000). Variables affecting information technology end-user satisfaction: A meta-analysis of the empirical literature. *International Journal of Human-Computer Studies,* Vol. 52, 751–771.

Marchionini, G. (1995). *Information seeking in electronic environments.* Cambridge, UK: Cambridge Univ. Press.

Marti, A. H. (1999). User interface and visualization, in *Modern information retrieval* (R. Baeza-Yates and B. Rebeiro-Neto, Eds.). New York: Addison–Wesley.

Nielsen, J. (1997). Usability engineering, in *The computer science and engineering handbook* (A. B. Tucker, Jr., Ed.), 1440–1460. Boca Raton, FL: CRC Press.

Nielsen, J. (2000). *Designing Web usability: The practice of simplicity.* Indianapolis, IN: New Readers.

Norman, D. A. (1993). *Things that make us smart.* New York: Addison–Wesley.

Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction,* 2nd ed. Reading, MA: Addison–Wesley.

Error Detecting and Correcting Codes

Patrick Verlinde
Royal Military Academy, Belgium

GLOSSARY

acknowledgment (ACK) The *confirmation* sent by the receiver to the sender to inform that a message has been received without errors.

automatic repeat request (ARQ) The strategy in which there is only error *detection* (no error correction) and *retransmission*.

block code A code that transforms every *block* of k information moments in a codeword of n moments.

channel coding The whole of operations which are executed on a signal, with the purpose of transmitting the signal as good as possible via an imperfect channel.

convolutional code A code that transforms k information moments in a codeword of n moments in a *continuous* way.

forward error correcting (FEC) The strategy in which there is error *correction* as well as error detection, but no retransmission.

I. INTRODUCTION

We consider (see Fig. 1) an information source that delivers a sequence of binary symbols (also called binary moments), which implies that eventually a preliminary source coding has taken place. In general, however, this sequence cannot be transmitted directly as such on the transmission channel, so a *channel coding* process is necessary.

Generally speaking, the denomination *channel coding* covers the whole of operations which are executed on the signal, with the purpose of transmitting this signal as good as possible via an imperfect channel (limited bandwidth, noise, etc.).

In our specific case, however, we consider only channel coding in the strict sense of coding that allows us to detect and correct errors occurred at reception. Other signal coding operations such as frequency conversion (modulation) are supposed to be part of the channel.

If one does not send redundant information and if at the receiver one or more moments have been changed (0 changed to a 1 or the other way round), there is no way of detecting these errors. The only way to detect errors consists of sending supplementary information together with useful information.

This supplementary information, in the form of redundant moments, is added according to a law L, which is known both at the transmitter and the receiver. At reception, it is then sufficient to check if the law L has been satisfied. If the law L has not been satisfied, one is sure an error has corrupted the received word. If the law L has been satisfied, one can be almost sure that the received word is correct, with the exception of the probability that an error did transform an authorized word (in the sense of the law L) into another authorized word!

It is the nature of the law L that characterizes the chosen code. One has to be aware that no single code is perfect; this means that no single code allows us to detect all errors, let alone correct them.

To achieve this, a coding system is placed at the output of the binary source, with the purpose of adding the required redundancy. The output signal of this coder is then applied to the transmission channel, which is composed of a dedicated module that

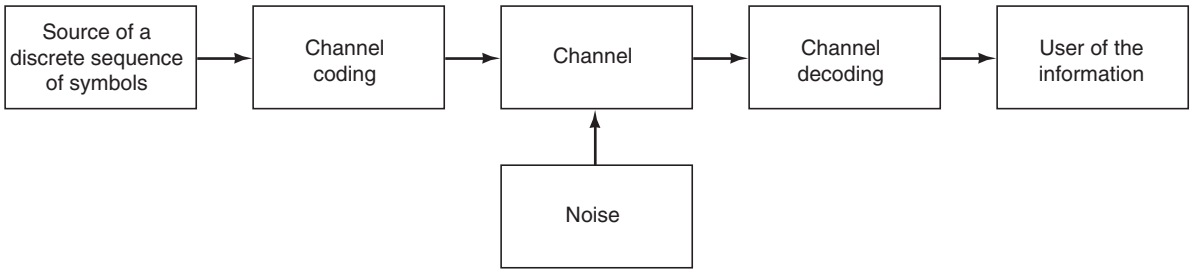


Figure 1 Basic elements of a digital transmission system.

has to deliver the definitive signal to be sent via the physical transmission channel (in baseband or in transposed band).

The purpose of this is that one should be able to consider the whole chain between the output of the binary source at the transmitting side and the output of the decoding module at the receiving side as a fictive non-noisy transmission channel. The decoding module will in principle correct the errors that have been made by the decision element (a 1 or a 0 has been received), which is located at the input of the receiver.

Such an approach allows us to study, for example, *source coding* independently of considerations regarding the protection of the information against perturbations due to noise on the physical transmission channel.

In this chapter, we present two different strategies used for the detection and correction of errors:

1. The strategy for error detection and retransmission (automatic repeat request or ARQ; automatic repetition query)
2. The strategy for error detection and correction without retransmission (FEC, forward error correcting).

More detailed information on error detecting and correcting codes and their applications can be found in the Bibliography.

II. THE POSSIBLE STRATEGIES

A. ARQ: Error Detection and Retransmission

The principle here is to increase the redundancy of the binary sequence by adding binary symbols that allow for the detection of errors. Therefore the binary sequence is divided into consecutive blocks to which binary control symbols are added. When at reception an error is detected, one does not try to correct this error. Instead a retransmission of the erroneous block is requested. This supposes, of course, the use of an *acknowledgment,* or *ACK.*

With this strategy different procedures can be used according to the type of transmission channel. The transmission channels can be classified according to the possible transmission directions. In this way one defines simplex, half-duplex, and full-duplex transmission channels. A simplex transmission channel allows communication in only one direction. A half-duplex transmission channel allows communication in either direction, but only one at a time, while a full-duplex transmission channel allows simultaneous communication in both directions.

It should be clear that the ARQ strategy cannot be used with a simplex transmission channel, since this strategy depends on the sending of an ACK by the receiver. In fact, three different ARQ procedures are possible. In the first case the transmitter waits until reception of the ACK for the block that he just transmitted, before transmitting the next block. In this case a half-duplex transmission channel is sufficient.

In the two other cases, there is a simultaneous transmission of information blocks by the transmitter and of ACKs (or not-ACKs) by the receiver. The latter arrive at the transmitter with a certain delay with respect to the moment of transmission of the blocks to which they are related. It must thus be possible to identify these blocks. The difference between the second and the third case resides in the retransmission which is executed in case of reception of a not-ACK: complete retransmission of all blocks which follow the erroneous received block in the second case or just a retransmission of the erroneous block in the third case. These last two cases do require a full-duplex transmission channel.

The advantages of the ARQ method over the FEC method are as follows:

- Simplicity of coding and decoding since the "correction" aspect is not taken into account in the code. The redundancy needed in this ARQ strategy is therefore smaller than in the case of FEC.

- The correction by retransmission has an *adaptatif* character in the sense that the redundancy needed for the "correction" (here the retransmission) only needs to be introduced when an error really occurs.
- The ARQ method allows for very small residual error percentages.

The drawbacks are:

- The need for at least a half-duplex transmission channel.
- The delay between the moment of transmission of a block and the instant that block is received correctly is variable in time. This delay can become unacceptable if many successive retransmissions are needed.
- The need for a buffer memory for the transmitter.

B. FEC: Error Detection and Correction

One differentiates between two types of error detecting and correcting codes: (1) block codes and (2) convolutional codes.

1. Block Codes

Let us suppose one wants to transmit words of k moments (containing the information) through a noisy transmission channel. To be able to correct errors, m moments, which are called *redundant* moments, are added. This way, for every k information moments, a block consisting of $n = m + k$ binary signals is transmitted. The coding law L defines the chosen correspondence between the 2^k words to be coded and the 2^k words which have been very carefully selected from the 2^n possible words. Among the 2^n possible words, 2^k words are thus chosen (and they are called the *codewords*) to be transmitted instead of the words to be coded, and the $2^n - 2^k$ other words are *never* transmitted. One says that one works with a code (n, k). It is clear that the 2^k chosen codewords have to be as different from one another as possible, if one wants the receiver to detect and eventually correct the errors due to the transmission.

2. Convolutional Codes

In the case of a block code the coder transforms every combination of k information moments in a codeword of n moments. Each block of n moments is thus obtained independently of the previous or the following blocks. In the case of convolutional coding the binary sequence is generated in a continuous way.

The k information moments are treated using a *sliding window* (the size of this window characterizes the *memory* effect) and a continuous stream of coded moments (codewords) is generated. Each information moment stays in the sliding window for a certain finite time period. During this *active* time period this particular information symbol influences the generated sequence of coded moments.

In the most general case a convolutional code is generated by shifting the information sequence to be transmitted through a linear shift register with a finite number of states. In general such a shift register consists of K groups of k cells. The content of these different cells is then used in n different modulo-2 adders (or XOR gates, also denoted by $\oplus$), as shown in Fig. 2.

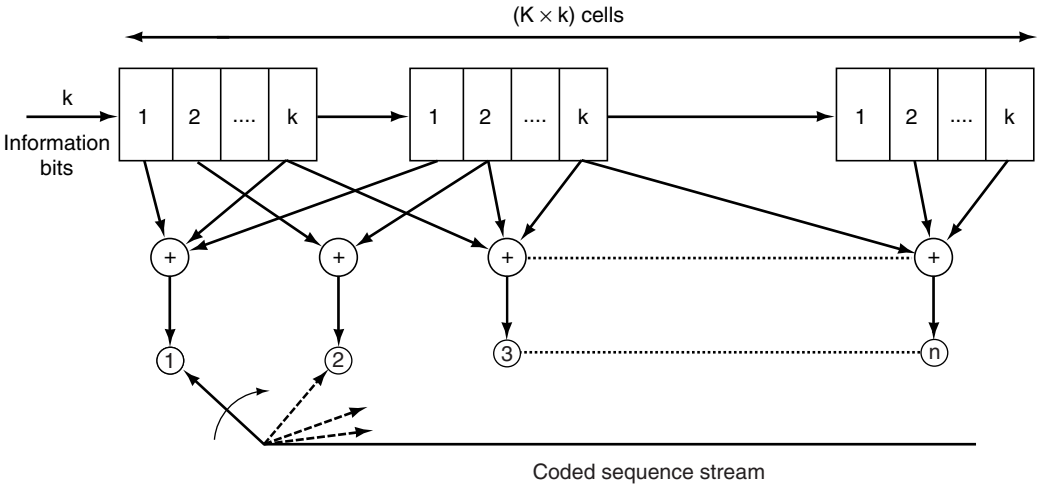


Figure 2 Architecture of a convolutional code generator.

As a reminder, the definition of the XOR operation is as follows:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

The information moments at the input of the convolutional code generator, which are supposed to be binary, are shifted k at a time in the shift register. For each k-binary moment input sequence $n > k$ binary moments are generated at the output. The parameter K is called the *constraint length* of the convolutional code.

At time i a group of k data binary moments m_i is shifted in the first k cells of the shift register. All previous binary moments already present in the shift register are shifted k cells to the right. Then the outputs of the n modulo-2 adders are sequentially sampled and transmitted.

III. ELEMENTARY NOTIONS— TRANSMISSION CHANNEL MODELS

Concerning the transmission channel coder, we assume that it receives at its input a binary sequence with a binary rate of R symbols per second. This binary sequence comes from the module that has performed the source coding. The binary rate of the coded sequence is R_c symbols per second, with $R_c > R$.

A. Code Rate

The *code rate* is per definition:

$$\rho = \frac{R}{R_c}$$

If, for instance, each binary symbol before coding contains information of 1 bit and if the code rate is $\frac{1}{2}$ then each coded symbol contains only $\frac{1}{2}$ bit of information.

In the case of block and convolutional codes the code rate is given by:

$$\rho = \frac{k}{n}$$

B. Redundancy

For block and convolutional codes one defines the *redundancy* as:

$$\text{Redundancy} = \left(\frac{1}{\rho}\right) - 1$$

In the specific case of block codes, the redundancy can also be expressed as the ratio of the number of redundant symbols m and the number of information symbols k:

$$\text{Redundancy} = \frac{m}{k}$$

C. Transmission Channel Models

For the coder, the transmission channel is composed of a modulator that delivers the signal to be transmitted on the real transmission channel, of the transmission channel itself, and of the demodulator.

The demodulator at the receiving side is composed of a module that transforms the signal back into the baseband, followed by a sampler (sampling frequency $= R_c$) and a decision element. The decision element can make two types of decision:

1. A *hard decision,* if it decides that the received symbol is a 1 or a 0. This decision is based on the comparison of the sampled value with a decision threshold.
2. A *soft decision,* if it delivers the quantified value of the sample without deciding on the binary value of the received symbol. The number Q of quantification levels is arbitrary but finite.

This transmission channel has a binary entrance alphabet and a Q-ary output alphabet ($Q > 2$ in the soft decision case). It is thus a discrete channel.

In practice, the combination of a block code and a soft decision demodulator is not often used due to the complexity of its realization. Convolutional codes on the other hand are very often used with soft decision demodulators.

IV. GENERALITIES ON BLOCK CODES

A. Introduction

The block coding has been presented in a previous section in the particular case of a so-called *systematic* code, i.e., the coded word is obtained by adding m redundant symbols to the k binary moments of the word to be coded. More generally a block code (n, k) consists of defining a correspondence between 2^k words of k binary moments to be coded and 2^k codewords of n binary symbols $n > k$.

Note that the choice of a code concurs with two successive operations:

1. The choice of the 2^k codewords (the set C) between all 2^n possible words of n binary symbols (the set N)

2. The choice of the one-to-one relation between the chosen codewords and the 2^k messages to be coded (the set M).

The nature and the properties of a code depend only on the first operation. The second operation is just a matter of convenience. Two codes formed by the same codewords (the set C), but associated with other messages (i.e., another one-to-one correspondence between C and M) are considered to be *identical*.

If one accepts this then the total number of different codes (n, k) is equal to C_N^{κ} with $\kappa = 2^k$ and $N = 2^n$.

At the decoding an error will be detected if the received word of n binary symbols does not belong to C. If the received word does belong to C, then there is probably no error, unless a codeword has been transformed (due to errors) into another codeword of C.

A block code detects all errors which transform a codeword of C into a word of N which does not belong to C. The code lets pass all other errors.

B. Maximum Likelihood Principle—Hamming Distance

For the *correction* of the detected errors the *maximum likelihood principle* is applied. The problem posed is the following: One has received a word w which does not belong to C and one has to determine the word v, belonging to C, which has been transmitted. The maximum likelihood principle consists of replacing the wrong word w with that codeword v belonging to C which differs from the received word w in a minimum of positions; it will indeed be that word which has been transmitted with the largest probability, knowing that w has been received.

The previous considerations show the necessity to introduce the notion of *distance* between two words of n binary symbols. The distance that is most often used is the *Hamming* distance:

Hamming distance. The Hamming distance $d(v, w)$ between two words v and w of n binary symbols is the number of positions in those two words for which the binary symbols differ.

This notion has to be seen in relation to the notion of the *weight* of a word composed of n binary symbols:

Weight. The weight $wt(w)$ of a word w of n binary symbols is the number of symbols equal to one.

If one uses the previously defined logical operation XOR, then one can easily check that

$$d(v, w) = wt(v \oplus w)$$

The word $e = v \oplus w$ is a word of n binary symbols, which is called the *error configuration*. It contains a 1 in each position where v and w differ. To see this, it is sufficient to know that if v and w are two words of n binary symbols, then $v \oplus w$ is also a word of n binary symbols obtained by XOR-ing the corresponding binary symbols of v and w.

One can easily proof the following properties:

- $0 \leq wt(v) \leq n$
- $0 \leq d(v,w) \leq n$
- $d(v, v) = 0$
- $wt(v \oplus w) \leq wt(v) + wt(w)$
- $d(v, w) \leq d(v, u) + d(u, w)$

The maximum likelihood principle can now be reformulated as the *error correcting* principle by stating the following: "If one has identified the received word w as erroneous, then w shall be decoded into the word v in such a way that the corresponding error configuration $e = v \oplus w$ is of minimal weight."

In this stage the correction of a received word w (which has been noticed to be erroneous, i.e., it does not belong to C) consists of calculating all error configurations that correspond to all codewords of C and in choosing that codeword v for which the corresponding error configuration is minimal. If this minimum is reached for only one codeword of C, then there is no ambiguity; if on the other hand several codewords do satisfy this criterion of minimal error configuration, then one can either choose one codeword at random or ask for a retransmission.

This decoding procedure is illustrated by the example shown in Table I, where $n = 3$ and $C = \{000, 111\}$. The symbol * in Table I indicates the smallest error configuration on each row and thus the word v of C which will be chosen in case w has been received. This decoding procedure will thus conclude that 000 was transmitted if 000, 100, 010 or 001 are received and that 111 was transmitted in all other cases. Subsequently the following questions can be asked:

- When can a code C *detect* an error?
- When can a code C *correct* that error?

C. Detection Capacity of a Code

A code C can detect an error configuration e if one has for each codeword v of C:

$$(e \oplus v) \notin C$$

One can therefore use a table of the same type as Table I. The first column contains all words of n binary symbols and can thus be interpreted as the set of all possible error configurations. Each error configuration is

Table I Example of a Decoding Rule

Received words w	Error configuration $000 + w$	Error configuration $111 + w$	Decoded words v
000	000*	111	000
100	100*	011	000
010	010*	101	000
001	001*	110	000
110	110	001*	111
101	101	010*	111
011	011	100*	111
111	111	000*	111

on its row added modulo-2 with all different codewords of C. If on a particular row no result at all of this addition produces a codeword of C, then the error configuration associated with that row can be detected.

A better way to analyze error detection capacity is to find all nondetectable error configurations. It is clear that if one considers a randomly chosen pair of codewords of C, for instance v and w, the error configuration $e = v \oplus w$ will not be detectable, since $w = v \oplus e$ (and since w is a codeword of C). Indeed, in Z_2 we have that $\oplus$ is equivalent with $-$! The set of all error configurations which are nondetectable is hence formed by the results of all modulo-2 additions of all pairs of codewords of C.

Here the notion of *distance d of a code C* has to be introduced. It is the minimal distance between all pairs of (different) codewords of C. Since $d(v, w) = wt(v \oplus w)$, the distance d of a code C is the minimal weight of $v \oplus w$ ($v \neq w$, v and $w \in C$).

Theorem

A code C *with distance* d *can detect all error configurations (different from zero) with weight* $\leq$ d $-$ *1.*

One says that a code detects errors of order t if that code detects *all* error configurations of weight $\leq t$ and if that code does not detect at least one error configuration of weight $t + 1$. The preceding theorem shows that a code with distance d can detect the error configurations of order $t = d - 1$. This implies not that this code does *not* detect error configurations of weight d or higher, but it does not detect them *all.*

D. Correction Capacity of a Code

If a codeword v is transmitted and if the word w is received (which implies the existence of an error configuration $e = v \oplus w$), the detection procedure based on the maximum likelihood will decide that the word v was transmitted if w lies closer to v than to every other codeword of C.

If this is the case for every codeword v of C on which the error configuration e is applied, one says that C corrects the error configuration e. In other words, C corrects the error configuration e if, for each $v \in C$ that applies, $v \oplus e$ lies closer to v than to every other codeword of C.

One says that a code corrects the errors of order t if that code corrects *all* error configurations of weight $\leq t$ and if at least one error configuration of weight $t + 1$ cannot be corrected.

Remark

This definition is relatively strict since it excludes the case mentioned previously in which a received word, identified as being erroneous and located at the same distance of two codewords, is decoded by choosing at random one of those two codewords.

To distinguish between these two decoding procedures, which both rely on the maximum likelihood principle, we will use the denomination of *incomplete maximum likelihood decoding (IMLD)* in the case of the strict definition given above, while the less stringent decoding mentioned in this remark will be called *complete maximum likelihood decoding (CMLD).*

When not specifically mentioned, all theories presented hereafter are applicable to the IMLD case.

Theorem

A code C *with distance* d *can correct all error configurations with a weight* $\left\lfloor \frac{(\mathrm{d}-1)}{2} \right\rfloor$ *(the notation* $\lfloor \mathrm{x} \rfloor$ *indicates the largest integer* $\leq$ x*). Furthermore there exists at least one error configuration with weight* $1 + \left\lfloor \frac{(\mathrm{d}-1)}{2} \right\rfloor$ *that cannot be corrected by* C.

Again a code with distance d could very well be capable of correcting certain error configurations with weight $> \left\lfloor \frac{(d-1)}{2} \right\rfloor$, but such a code does not correct them all.

V. LINEAR CODES

A. Introduction

This Section describes a very broad class of codes in which one finds almost all codes used in practice. The success of this type of codes resides in the fact that they allow the use of very powerful mathematical tools, such as those of linear algebra.

Because of the fact that linear codes are highly structured, the following problems, for which the solution is very complicated in the general case, do have a simple solution if one assumes the linearity of the code:

- Maximum likelihood decoding
- Simple coding requiring little memory
- Identification of detectable and correctable error configurations.

As a matter of fact, this explains their success.

B. Definition of a Linear Code

A code C is called *linear* if for each pair of codewords v and w, the word $v \oplus w$ is also a codeword, i.e., this word also belongs to C. Because in this definition v does not necessarily differ from w, each linear code contains the codeword consisting of n zeros (called the *zero* codeword). This condition is necessary, but not sufficient.

Example

Following the definition it is easy to check that the code $C_1 = \{000, 111\}$ is a linear code, while the code $C_2 = \{000, 001, 101\}$ is not a linear code.

C. Distance of a Linear Code

Theorem

The distance of a linear code is equal to the minimum weight of its nonzero codewords.

The proof of this theorem is trivial, taking into account the definition of a linear code.

D. Generator Matrix and Coding

Until now we have always considered the codewords to be binary. The opportunity presents itself now to generalize this and to introduce the so-called *p-ary* codes. This means that every symbol can take now p values, denoted by $0, 1, \ldots, p-1$. It should, however, be clear that the case $p = 2$ is the most common one. From now on, we will always work in Z_2^n, unless otherwise mentioned.

A code C can now be defined as a subset of the vector space Z_p^n, which consists of the set of all n-tuples defined on the finite field $Z_p = \{0, 1, \ldots, p-1\}$ (which means that every symbol out of the n can take one of p values: the vector space Z_p^n consists thus of p^n different n-tuples). This subset C is supported by a *base* which consists of k vectors (k is the *dimension* of the code, which means that C contains p^k n-tuples). The dimension n of the vector space Z_p^n is called the *length* of the code C and d is called the *distance* of the code C. [The notion of distance of a code C, namely, $d(v, w) = wt(v \oplus w)$ remains valid in the p-ary case, under the condition that we admit that $\oplus$ represents the addition modulo p and that the weight of a word is the number of symbols different from zero.]

These three parameters—length n, dimension k, and distance d—form the three essential characteristics of the code C. This code is then often denoted as the linear code $C(n, k, d)$.

One calls *generator matrix* G of such a code the $(k \times n)$ matrix which has k rows formed by a basis of C. The rank of this matrix is necessarily equal to k. This generator matrix leads to a very simple coding procedure based on the following property.

Let u be a word to be coded with length k, considered as a $(1 \times k)$ row vector, and let G be the generator matrix of a linear code C. The vector $v = uG$ is a codeword of C since v is obtained by a linear combination of the base of C. Note that the product of the vector u and the matrix G is realized as usual, but using the rules of the multiplication modulo p.

Every word to be coded corresponds thus through this operation to a codeword of C. One can imagine easily that this coding procedure requires a lot less memory space than the procedure based on storing a table, which fixes the correspondence between the words to be coded and the corresponding codewords.

E. Parity Check Matrix and Decoding

The second matrix associated with a linear code C is extremely useful for decoding. One calls *parity check matrix* H of a linear code C the matrix whose columns form a base of the subspace orthogonal to C, which is denoted $C^\perp$. The subspace $C^\perp$ is defined as the set of all vectors of Z_p^n that are orthogonal to all vectors of C (i.e., to all codewords). Two vectors of Z_p^n are

orthogonal if their scalar product is zero, the scalar product of two vectors being calculated using the rules of the algebra modulo p.

If C has n as length and k as dimension, every parity check matrix H of C must have n rows and $n - k$ columns and H must be of rank $n - k$. This follows from the property of the independence of the vectors of a base and from the fact that the sum of the dimensions of C and $C^{\perp}$ is equal to n.

A useful property of the parity check matrix H is that $GH = 0$.

F. Canonical Form of a Generator Matrix—Systematic Code—Equivalent Codes

Remember that two matrices are equivalent (according to the rows) if one matrix can be obtained from the other one by a series of elementary operation on the rows. An elementary operation on the rows is either a permutation of two rows or a substitution of a row with the sum modulo p of that row and any other row.

Starting with a generator matrix G $(k \times n)$ of a code C with length n and dimension k, it is sometimes possible to obtain an equivalent matrix under the following *reduced row echelon form (RREF):*

$$G' = [E_k, X]$$

where X is a $k \times (n - k)$ matrix and E_k is the unity matrix $(k \times k)$.

This form is called the *canonical* form of the generator matrix of C. One also says that the code generated by such a generator matrix under canonical form is a *systematic* code.

When a code is under systematic form, the k first moments of each codeword are identical to those of the word to be coded: the *information moments*. These are then complemented by $n - k$ control or redundant moments. The problem of retrieving the original information word from the codeword then becomes trivial in this case.

Note, however, that it is not always possible to transform a code in a systematic code by elementary operation on the rows. This remark forces us to find a solution for the case in which a code does not have a generator matrix under canonical form.

The solution is based on the extension of elementary operations on the rows to elementary operations on the columns. An elementary operation on the columns is either the permutation of two columns or the multiplication modulo p of a column with a scalar $\neq 0$. While the elementary operations on the rows do not change the code (i.e., the codewords of the linear code C generated via the generator matrix G or via the generator matrix G', equivalent according to the rows, are the same), the elementary operations on the columns, which transform the matrix G into matrix G^*, define other codewords and thus another code C^*, which is called the equivalent code of C. However, one can show that this equivalent code C^* has the same length n, the same dimension k, and the same distance d as the code C.

The following theorem justifies the use of this equivalent code G^* when the code C does not have a generator matrix under canonical form.

Theorem

Every linear code C *is equivalent with a linear code* C* *which has a generator matrix* G* *under canonical form.*

G. Decoding of a Linear Code

The decoding principle is based on the use of the notion of *coset associated with a code C.* A coset denoted $u \oplus C$, associated with the code C (and to the vector u) is the set of all codewords of C to which the vector u has been added: $u \oplus C = \{w \in Z_p^n : w = u \oplus v, v \in C\}$.

It can be shown that the coset $u \oplus C$ is identical to the coset $w \oplus C$. Let us suppose that a codeword v of a linear code C is transmitted and that the word w (which is not a codeword, thus allowing the error detection) is received. The resulting error configuration is $u = v \oplus w$ (which leads to $v = u \oplus w$).

Since one has received w and after determining that w does not belong to C (error detection), it is sufficient to estimate the error configuration u to apply the correction and to determine v. Again the maximum likelihood principle is used here by assuming that the most likely error configuration is the one with the lowest weight.

To find this error configuration, it is useful to note that since $v = u \oplus w$, u and w do necessarily have to belong to the same coset of C. Since u and w necessarily belong to the same coset, the correction procedure is as follows: At the reception of a word w which has been labeled as erroneous, the correction procedure consists then of choosing from the coset $w \oplus C$ the word u with the smallest weight and to conclude that the codeword which was transmitted was $v = u \oplus w$.

It is now easy to understand why we sort the different cosets of C under the form of a normalized table (*standard array;* see example hereafter). To decode a linear code C using such an array it is sufficient to complete these tasks:

- Find in the standard array the received word w labeled as erroneous.
- Verify that the word in front of the *row* to which w belongs (the so-called *coset leader*) is the most probable error configuration.
- Choose as codeword v the word at the top of the *column* to which w belongs.

Example

Let us consider a linear code $C = \{0000, 1011, 0101, 1110\}$. The standard array of the cosets is shown in Table II. The code C itself is the coset associated with the zero vector and it is always presented in the first row. The second column of the table is called the column of the *coset leaders* and it is constructed by taking at each row an n-tuple of minimal weight (i.e., the most probable error configuration in the maximum likelihood sense), which was not yet present in the preceding rows.

Let us suppose that the word $w = 1101$ was received. This word does not belong to C and needs thus to be corrected. The word w belongs to the second row of the standard array. The most probable error configuration is then 1000 and the codeword that most probably was sent is $1101 \oplus 1000 = 0101$.

If the received word is $w = 1111$, then the corresponding coset has as coset leader the error configuration 0100 and the most probable codeword is 1011. We should mention, however, that the same coset contains also the word 0001, which is an error configuration that is as probable as 0100! The choice of the transmitted codeword (1011 instead of 1110) is thus arbitrary. The error-correcting capacity in the IMLD sense is 0 (the received word 1111 is equidistant from the codewords 1011 and 1110)!

The longest operation is the search for w in the standard array. This search can be made easier. To do so we define (for a linear code C with length n, dimension k, and parity control matrix H) the *syndrome* associated with a word w of Z_2^n as the word wH (which belongs to Z_2^{n-k} and which is different from zero if w does not belong to C and is thus erroneous).

Table II Example of a Standard Array

$0000 \oplus C \equiv C$	0000	1011	0101	1110
$1000 \oplus C$	1000	0011	1101	0110
$0100 \oplus C$	0100	1111	0001	1010
$0010 \oplus C$	1001	0010	0111	1100

One has to remark that if u is the error configuration associated with w, one has

$$w = v \oplus u$$

and thus:

$$wH = vH \oplus uH = 0 \oplus uH = uH$$

This means that w and u have the same syndrome.

If one generalizes what we just proved, one can say that all words of the same coset do have the same syndrome, which is the same as the one of the "coset leader." As a matter of fact this is trivial since every word of a coset is obtained by addition modulo-2 of this coset leader with a codeword of C.

If one constructs a table that associates each "coset leader" with its syndrome, it is no longer necessary to search for w in the standard array in order to correct the words labeled as being erroneous. We call this new table the *standard decoding array*. Note that the parity check matrix H is indispensable for constructing this table.

Example

Let us again take the example of the linear code $C = \{0000, 1011, 0101, 1110\}$. The parity check matrix H for this code is:

$$H = \begin{bmatrix} 11 \\ 01 \\ 10 \\ 01 \end{bmatrix}$$

and the standard decoding array is shown in Table III.

At the reception of a word w, one calculates its syndrome wH. If this syndrome differs from zero, one searches in the standard decoding array for the coset leader u corresponding to this syndrome and one decides that the word that was most likely sent is $v = w \oplus u$.

So, if the word $w = 1101$ was received, the corresponding syndrome is $wH = 11$ and, after checking that this syndrome is different from zero (i.e., an error did occur), the correction is applied by considering

Table III Example of a Standard Decoding Array

Coset leader	Syndrome
0000	00
1000	11
0100	01
0010	10

the coset leader 1000 as the most likely error configuration, and by deciding that the most probable word transmitted was $v = w \oplus u = 1101 \oplus 1000 = 0101$.

Calculating the distances between 1101 and the other codewords, one finds:

$$d(0000, 1101) = 3$$
$$d(1011, 1101) = 2$$
$$d(0101, 1101) = 1$$
$$d(1110, 1101) = 2$$

and it can be seen that $v = 0101$ is indeed the word of C that lies the closest to 1101.

Remark

A relatively fast way of calculating the correction table when the parity check matrix H and the distance d of the code are known is to consider all error configurations e with weight $\leq \frac{(d-1)}{2}$ and to calculate the syndrome corresponding to each of those error configurations.

Although the use of the standard decoding array constitutes progress with respect to the use of the complete decoding array, one has to bear in mind that the search in such a table remains a long operation (especially if one decodes on line). The error-detecting and -correcting codes that are very popular are in practice those for which a particularly fast decoding method has been found.

H. Perfect Codes

1. Hamming Bound

The problem of determining the number of codewords in a code with length n and distance d has until now not found a general solution. However, if $|C|$ represents the number of codewords of the code C, the so-called *Hamming bound* gives an upper bound for this number:

$$|C| \leq \frac{2^n}{C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^t}$$

where C_n^i represents the number of different words of length n containing i ones, and with $d = 2t + 1$ (odd) or $d = 2t + 2$ (even). This bound does not suppose that the code is linear. Let us remind the reader also that such a code can correct all error configurations with weight $\leq t$.

2. Perfect Codes

A code C with length n and with odd distance $d = 2t + 1$ is called perfect if $|C|$ reaches the Hamming bound:

$$|C| = \frac{2^n}{C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^t}$$

It is possible to give a geometrical interpretation to this notion of perfect code. Therefore, it is necessary to define the notion of a sphere of radius r and center u (u belongs to Z_2^n) in the space Z_2^n as the set of points v of Z_2^n for which $d(u, v) \leq r$. The notion of correction capacity presented earlier, can then be represented by using these spheres.

Considering a code with distance $d = 2t + 1$, the spheres of radius t centered on the codewords will not intersect! So if there are at most t errors in a received word, this erroneous word is located in the sphere corresponding to the transmitted codeword and the correction can happen without ambiguity.

Each sphere of radius t contains a number of words equal to:

$$C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^t$$

For a perfect code, the set of spheres with radius t and centered on the codewords, fills the whole space Z_2^n (i.e., those spheres form a *partition* of Z_2^n).

From this geometrical interpretation it also follows that a perfect code can only exist for odd values of $d = 2t + 1$. A perfect code C with length n and distance $d = 2t + 1$ can thus correct all error configurations with weight $\leq t$ and *no others*.

3. Theorem of Tietäväiren and Van Lint

If C is a perfect code (with length n and distance $d = 2t + 1$) that is not trivial (i.e., C differs from Z_2^n and n differs from $2t + 1$), one has:

- Either $n = 23$ and $d = 7$ (this code is the code of Golay) or
- $n = 2^r - 1$ for $r \geq 2$ and $d = 3$ (these codes form the family of Hamming codes).

The only two possible values for t are, therefore, $t = 1$ or $t = 3$.

In the next section we study as an example the case $t = 1$ in more detail.

I. Example of Perfect Linear Codes: The Hamming Codes

Definition

Each code with length $n = 2^r - 1$ with $r \geq 2$ for which the rows of the parity check matrix H are formed by the set of all vectors of length r different from zero is called a Hamming code of length n. It can be shown that for a perfect Hamming code $k = n - r$.

Example

Hamming code of length $n = 7$ ($r = 3$):

$$H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}$$

Knowing H and using $GH = 0$ it is possible to determine the generator matrix G:

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$$

This code has dimension $k = 4$ and contains thus $2^4 = 16$ codewords. The distance of the code is $d = 3$.

Theorem

The dimension k *of a Hamming code is* $2^{\mathrm{r}} - 1 - \mathrm{r}$ *and it contains a number of codewords equal to:*

$$2^{2^{\mathrm{r}}-1-\mathrm{r}} = 2^{\mathrm{k}} = 2^{\mathrm{n}-\mathrm{r}}$$

The distance of the code is $\mathrm{d} = 3$. *A Hamming code is a perfect code and corrects all single errors.*

The construction of the standard decoding array of a Hamming code is immediate. Indeed, all single errors (and only those) are corrected and thus the error configurations to be considered are all words of length $2^r - 1$ and weight 1 (i.e., all the coset leaders). The correction array is then shown in Table IV where E_{2^r-1} is the unit matrix $(2^r - 1) \times (2^r - 1)$.

Example

For the Hamming code with length $d = 7$ presented before, a received word $w = 1101001$ gives a syndrome $wH = 011$. This syndrome is the fourth row in the matrix H. The coset leader is thus the fourth row of E_7: $u = 0001000$ and w is being decoded as $v = w \oplus u = 1100001$.

Table IV Example of a Hamming Decoding Array

Coset leader	Syndrome
00 . . . 0	0
$E_{2^{r-1}}$	Different lines of H

J. Cyclic Codes

1. Preliminary Remark

The theory of cyclic codes is based on the use of polynomials instead of matrices. Therefore one has to introduce the ring $Z(x)$ of polynomials defined on Z_2 as well as, in a more general way, the ring $F(x)$ of polynomials defined on a Field F.

One then defines the ring $F(x)/h(x)$ of polynomials defined on F modulo $h(x)$, which is the set of polynomials of degree inferior to the degree of $h(x)$.

In what follows we consider the specific case in which $h(x) = x^n + 1$ and $F = Z_2$. We denote by R_n the set $F(x)/(x^n + 1)$.

In the theory of cyclic codes, each word $v = a_0 a_1 a_2 \ldots a_{n-1}$ of length n of Z_2^n corresponds to a polynomial $v(x)$ by the following relation:

$$v(x) = a_0 \oplus a_1 x \oplus a_2 x^2 \oplus \ldots \oplus a_{n-1} x^{n-1}$$

In this notation we use the convention not to write the terms of the polynomial corresponding to $a_i = 0$. In this way to $v = 1010$ corresponds the polynomial $v(x) = 1 \oplus x^2$. Every word of n binary digits can thus be seen as an element of Z_2^n or of R_n.

If one multiplies by x a polynomial $v(x)$ corresponding to a word v, the the polynomial $x.v(x)$ modulo $x^n \oplus 1$ corresponds to the word $\pi(v)$, obtained from v by a cyclic permutation.

Example

Let us consider R_5.

$v = 01111$ corresponds to $v(x) = x \oplus x^2 \oplus x^3 \oplus x^4$

$\pi(v) = 10111$ corresponds to $x.v(x)$ modulo $x^n \oplus 1 = x^2 \oplus x^3 \oplus x^4 \oplus x^5$ modulo $x^5 \oplus 1 = 1 \oplus x^2 \oplus x^3 \oplus x^4$

It can be checked that the cyclic permutation operation is a linear operation in the sense that:

1. $\pi(v + w) = \pi(v) + \pi(w)$
2. $\pi(av) = a\pi(v)$ where $a \in Z_2$ and $v,w \in Z_2^n$

2. Definition

A linear code C is said to be cyclic if every cyclic permutation of a codeword is also a codeword.

3. Theorem

If C *is a cyclic code and if* $v \in C$, *then, for every polynomial* $a(x) \in R_n, c(x) = a(x).v(x)$ *modulo* $(1 + x^n)$ *is also a codeword.*

4. Generator Polynomial

a. Preliminary

Considering the nonzero codewords of a cyclic code C, it is easy to show, based on the linearity properties of the cyclic codes, that there is among those codewords a *unique* codeword noted g such that the degree of the associated polynomial $g(x)$ is minimal.

Every codeword $c(x)$ of C can be obtained from $g(x)$ by multiplication with a polynomial $a(x)$:

$$c(x) = a(x).g(x)$$

Indeed, the degree of $c(x)$ being greater than or equal to that of $g(x)$, we can write:

$$c(x) = q(x).g(x) \oplus r(x)$$

or

$$r(x) = q(x).g(x) \oplus c(x)$$

But $c(x)$ and $q(x).g(x)$ (see theorem just above) being codewords implies that $r(x)$ is also a codeword. Furthermore, either $r(x) = 0$ or the degree of $r(x)$ is inferior to the one of $g(x)$. But since $r(x) \in C$, the latter possibility has to be ruled out, which leads to the conclusion that $r(x) = 0$. Therefore, we conclude that $g(x)$ divides every codeword $c(x)$ of C.

b. Definition

The unique nonzero polynomial of minimal degree of a cyclic code C is called the *generator polynomial.*

First Theorem

If C *is a cyclic code of length* n *and if its generator polynomial* g(x) *is of degree* n − k, *then the dimension of* C *is* k.

Second Theorem

The generator polynomial g(x) *of a cyclic code* C *of length* n *divides* $(1 \oplus x^n)$. *This allows us to find all the cyclic codes of length* n *by factorization of* $1 \oplus x^n$.

5. Coding Cyclic Codes

One of the important properties of cyclic codes is their coding ease. In practice, two methods can be used. The first (direct) method does not generate a systematic code and is based on multiplying by the generator polynomial. The second (indirect) method does generate a systematic code, based on the division by the generator polynomial.

a. Nonsystematic Coding

The word u of length k to be coded is considered a polynomial $u(x)$ and the codeword $u(x).g(x)$ is transmitted. This multiplication can be easily implemented using shift registers.

Let us cite as an example a cyclic form of the Hamming code with length $n = 7$ whose generator polynomial is $g(x) = 1 \oplus x \oplus x^3$. The word 1001 to be coded is then coded as follows:

$$u(x).g(x) = (1 \oplus x^3)(1 \oplus x \oplus x^3) = 1 \oplus x \oplus x^4 \oplus x^6$$

The codeword is thus 1100101. It is easy to check that the code is not systematic.

b. Systematic Coding

The working principle is based on the modification of the correspondence between a binary word and the associated polynomial. In this case the leftmost binary symbol corresponds to the highest power of the polynomial. The k information symbols correspond in this manner to the following polynomial:

$$u(x) = u_{n-1}x^{n-1} \oplus u_{n-2}x^{n-2} \oplus \ldots \oplus u_{n-k}x^{n-k}$$

Now the polynomial $u(x)$ (degree $n - 1$) is divided by the generator polynomial $g(x)$ (degree $n - k$) leading to a quotient $q(x)$ (not further used) and a remainder $r(x)$ (degree $< n - k$):

$$u(x) = q(x).g(x) \oplus r(x)$$

The transmitted codeword is then $u(x) \oplus r(x)$, which is indeed a codeword since it is a multiple of $g(x)$:

$$u(x) \oplus r(x) = q(x).g(x)$$

Note that the highest degree of $r(x)$ is maximum $n - k - 1$, while the term of $u(x)$ with the lowest degree has the minimum degree $n - k$. This means that the two polynomials $u(x)$ and $r(x)$ do not get "mixed," which explains why the coding is systematic.

Taking again the example of the cyclic Hamming code used in the previous paragraph, the information word 1001 is coded in the following manner. The division of $u(x) = x^6 \oplus x^3$ by $g(x) = 1 \oplus x \oplus x^3$ results in a remainder $r(x) = x^2 \oplus x$. The corresponding codeword is thus $u(x) \oplus r(x) = x^6 \oplus x^3 \oplus x^2 \oplus x$, which is the associated polynomial to the codeword 1001110. (The reading is also done from the left to the right by

starting with the highest power of the polynomial.) It is easy to see that this coding is indeed systematic.

The division of polynomials is performed as easily as their multiplication by the use of shift registers.

6. Decoding of Cyclic Codes

We have seen that the basic element for decoding linear codes is the syndrome. In the case of cyclic codes, it is more convenient to use a polynomial called a *syndrome polynomial.*

a. Definition

Let C be a cyclic code of length n and with generator polynomial $g(x)$. The syndrome polynomial $s(x)$ associated with a received word w (n-tuple) is then the remainder of the division of the polynomial $w(x)$ associated with the received word w by $g(x)$:

$$s(x) = w(x) \bmod g(x)$$

b. Properties

1. If the degree of $g(x)$ is $n - k$, the degree of $s(x)$ will be inferior to $n - k$ and will correspond to a word of length $n - k$.
2. While studying linear codes we have seen that the syndrome s associated with a received word w is equal to the syndrome of the error configuration e corresponding to the received word w. This property is conserved for the syndrome polynomials. More precisely, if the transmitted codeword is $q(x).g(x)$ and if the received word is $w(x) = q(x).g(x) \oplus e(x)$, where $e(x)$ represents the polynomial associated to the error configuration e, we have:

$$\begin{aligned} e(x) \bmod g(x) &= [w(x) \oplus q(x).g(x)] \bmod g(x) \\ &= w(x) \bmod g(x) = s(x) \end{aligned}$$

3. Decoding using the syndrome polynomial is based on this last property. The set of syndrome polynomials is made up of all polynomials of degree inferior to $n - k$. For each syndrome polynomial $s(x)$ there is a corresponding error configuration $e(x)$ (of minimal weight) such that $s(x) = e(x) \bmod g(x)$. Therefore, we again need to construct a table delivering for each error configuration the associated syndrome polynomial. The calculation of a syndrome polynomial is easily done using a shift register.

Example

Let C be the Hamming code of length $n = 7$ with generator polynomial $g(x) = 1 \oplus x \oplus x^3$. The syndrome of the received word $w = 1011001$ is the remainder of the division:

$$(x^6 \oplus x^3 \oplus x^2 \oplus 1) \div (x^3 \oplus x \oplus 1)$$

which leads to $s(x) = x \oplus 1$.

Since Hamming codes correct only one error, we have to establish a table showing the correspondence between the error configurations of weight 1 and the associated syndrome polynomials. This correspondence is presented in Table V. Consulting this table indicates that the most probable error configuration in the specific case of our example is $e = 0001000$ and correspondingly the most probable codeword is then $v = 1010001$.

c. MEGGIT decoder (1960)

The longest operations in the type of decoding presented above are required for the construction of the correspondence table between the error configurations and the syndrome polynomials, and the search in this table at each correction.

An important simplification of the algorithm can be obtained if the cyclic nature of the code is taken into account. Initially, we only consider the binary symbol of the highest degree of the received word w, which we correct or not according to its syndrome. Next the same operation is executed using $\pi(w)$ afterwards with the other cyclic permutations of w. The same operation is thus repeated n times.

The advantage of this method resides in the fact that it is only necessary to establish the correspondence table between the error configurations and the syndrome polynomials for the correctable error configurations of degree $n - 1$. In the previous example only the last line of Table V is used.

Table V Decoding Table for the Hamming Code of Length $n = 7$

$e(x)$	$s(x)$
0	0
1	1
x	x
x^2	x^2
x^3	$x \oplus 1$
x^4	$x^2 \oplus x$
x^5	$x^2 \oplus x \oplus 1$
x^6	$x^2 \oplus 1$

Furthermore the calculation of the syndrome does not have to be reiterated at each cyclic permutation, as is shown by the next theorem.

Theorem

C *is a linear cyclic code with generator polynomial* g(x). *If the syndrome polynomial of a word* w *is* s(x), *then the syndrome polynomial of* π(w) *is the remainder* $s_1(x)$ *after division of* x · s(x) *by the generator polynomial.*

This theorem makes the calculation of the syndrome polynomials of the permutations of *w* easier, thanks to the use of a shift register.

d. Code with unitary error correction capacity

Let us suppose that a code can only correct one error. The only error configurations to be considered are then those of weight 1. The successive permutations of *w* bring the erroneous binary symbol to the utmost right of the word after at the most $n - 1$ permutations. The corresponding syndrome allows us to detect this error configuration and, counting the number of permutations executed to arrive at this error detection, allows us to retrieve the error position in the word *w*.

Example

C is a cyclic code of length n and with generator polynomial $g(x) = 1 \oplus x \oplus x^3$. This code corrects all error configurations of weight 1. From Table V we need only to maintain the last syndrome polynomial $x^2 \oplus 1$.

Let $w(x) = 1 \oplus x^2 \oplus x^3 \oplus x^6$ be the polynomial corresponding to the received word 1011001. The syndrome is $s(x) = w(x) \bmod g(x) = 1 \oplus x$, which does not correspond to our syndrome polynomial. We calculate successively:

$$s_1(x) = x \cdot s(x) \bmod g(x) = x \oplus x^2$$

$$s_2(x) = x^2 \cdot s(x) \bmod g(x) = 1 \oplus x \oplus x^2$$

$$s_3(x) = x^3 \cdot s(x) \bmod g(x) = 1 \oplus x^2$$

which corresponds to the only syndrome polynomial in our table. The binary symbol to be corrected is thus the fourth one starting from the highest power (x^6) at the right, which leads to the coefficient of the third power (x^3). The decoded word is then 1010001.

e. Code with Error Correction Capacity Equal to 2

For a cyclic code of length $n = 15$, allowing for correction of two errors, the classic decoding would need a correspondence table between the error configurations and the syndrome polynomials of 121 rows (all error configurations of weights 0, 1, and 2, for which there are respectively 1, 15, and $C_{15}^2 = 105$ different possibilities). For the decoding according to the method of Meggit, we need only keep the rows corresponding to $e(x) = x^{14}$, $e(x) = x^i \oplus x^{14}$ $i = 0, 1, \ldots, 13$.

On reception of the word *w*, its syndrome is calculated. If this syndrome is in the table the binary symbol w_{14} is corrected. Next the syndromes of the cyclic permutations of *w* are calculated and the binary symbols $w_{13}, w_{12}, \ldots, w_0$ are corrected successively if the corresponding syndrome is in the table.

K. BCH Codes

1. Introduction

An important class of cyclic codes is one of the BCH codes. These codes were named after the discoverers A. Hocquenghem and afterwards (independently) R. C. Bose and D. K. Ray-Chaudhuri. Their correction capacity can be high and their decoding is rather easy. Furthermore, this class is relatively huge.

These codes are in general binary codes but an important specific case of the BCH codes is the Reed-Solomon code, used for instance in the compact disk digital audio system, which is a nonbinary code presenting excellent properties in the case of burst errors. Hamming codes can also be shown to be BCH codes. In fact, sometimes BCH codes are presented as a generalization of Hamming codes that allow multiple error correction, compared to the single error correction capability of the Hamming codes.

The difficulty with the study of BCH codes comes from the way in which they are defined. This definition requires knowledge of the theory of the finite fields often called Galois fields (GF).

2. Galois Fields

The definition of a Galois field is an extension of the definitions of *group* and *ring*. Let us start with the notion of commutative group. A set *G* provided with a composition law noted ⊕ is a commutative group, noted $(G, \oplus)$, if the following hold true:

1. $\forall x,y \in G : (x \oplus y) \in G$.
2. The law ⊕ is associative and commutative.
3. There exists a neutral element noted 0 such that $\forall x \in G : x \oplus 0 = x$.
4. Every element $x \in$ G has an opposite noted $-x$, such that $x \oplus (-x) = 0$.

A set R provided with two composition laws, noted $\oplus$ and ., is a ring, noted $(R, \oplus, .)$, if the following hold true:

1. $(R, \oplus)$ is a commutative group.
2. $\forall x,y \in R : (x.y) \in R$.
3. The operation . is associative and distributive to the left and the right with respect to the operation $\oplus$.
4. If the operation . is commutative, then the ring is said to be commutative.

A set F then is a field, if these statements are true:

1. $(F, \oplus, .)$ is a commutative ring.
2. There exists a unity element for the operation . noted 1, such that $\forall x \in F : x.1 = x$.
3. Every element $x \in F \backslash \{0\}$ has an inverse noted x^{-1}, such that $x.x^{-1} = 1$.

A field is said to be *finite* if the number of elements in the set is finite. This number of elements is called the *order* or the *cardinality* of the field.

A Galois field with q different elements is referred to as GF(q) and it has the following properties:

1. GF(q) exists if and only if q is an integer power m of a prime p: $q = p^m$. If $m \geq 2$, then the field GF(p^m) is called an *extension field* of GF(p).
2. A *primitive element* β of a field is an element such that the successive powers of β generate all the nonzero elements of the field.
3. Every field contains at least one primitive element.
4. The *order* n of a nonzero element α of the field is the smallest power for which the result is 1: $\alpha^n = 1$.
5. A primitive element β is of order $q - 1$.
6. A polynomial $h(x)$ of degree m and irreducible on Z_2 is called *primitive* if it does not divide any of the polynomials $1 \oplus x^s$, $\forall s < 2^m - 1$. If a primitive polynomial is used to realize a GF, then β, the root of $h(x)$ [which does not belong to Z_p since $h(x)$ is irreducible on Z_p], is a primitive element of that GF;
7. A *minimal polynomial* m_α of a nonzero element α of the field GF(p^m) is the lowest degree polynomial with coefficients defined on GF(p) that has the element α as a root. Every minimal polynomial m_α associated with an element α of a GF(2^m) is a factor of $1 \oplus x^{2^m-1}$.

3. Construction of an Extension Field GF(p^m)

To be more specific, an example is given here in order to show the procedure of constructing an extension field GF(p^m):

1. Choose an irreducible polynomial $h(x)$ of degree m, which is defined on Z_p.
2. GF(p^m) is composed of all polynomials of degree $< m$, defined on Z_p (there are exactly p^m such polynomials, which corresponds to the order of the GF, as it should).
3. All the operations of addition and multiplication are defined as usual, but using the operation modulo $h(x)$.

Example: Construction of the Extension Field GF(2^4): $p = 2$ and $m = 4$

GF(2^4) can be constructed using the *primitive* polynomial $h(x) = 1 \oplus x \oplus x^4$. The extension field is represented in Table VI, where three possible representations of the codewords are given: words of Z_2^4, polynomials modulo $h(x)$, and powers of the primitive element β.

The representation using the powers of β, makes it easier to perform multiplication operations in the field. This can be easily seen in the following example:

$$(1 \oplus x \oplus x^2 \oplus x^3).(1 \oplus x^2 \oplus x^3) = \beta^{12}.\beta^{13} = \beta^{25} = \beta^{10} = (1110), \text{ since } \beta^{15} = 1$$

Table VII shows the minimal polynomials linked to the different elements of GF(2^4).

4. Definition of BCH Codes

Consider a Galois field F of order q and a number n, prime with q (a number n is prime with another number q if they have no common divider but 1). One calls q-ary BCH code with correction capability equal to t and length n ($n > 2t + 1$), the set of n-tuples w whose associated polynomials $w(x) \in F(x)$ satisfy the following equations:

$$w(\alpha) = w(\alpha^2) = w(\alpha^3) = \ldots = w(\alpha^{2t}) = 0$$

where α is an element of order n of some extension field of F. If the chosen element is primitive, then $n = q^m - 1$ and the code is said to be a *primitive BCH code.*

The BCH code defined here has as generator polynomial $g(x)$ the *smallest common multiple (SCM)* of the minimal polynomials $m_i(x)$ corresponding to the elements α^i chosen for the definition of the code, where $i = 1, 2, \ldots, 2t$:

$$g(x) = \text{SCM}\,[m_1(x), m_2(x), \ldots, m_{2t}(x)]$$

It is necessary to take the smallest common multiple since certain minimal polynomials are identical [this is, for instance, the case for $m_1(x)$ and $m_2(x)$ in GF(2^m)]. The SCM reduces itself simply to the product of all $m_i(x)$ from which one eliminates the factors that are a repetition of a preceding factor.

Table VI Three Possible Representations of the Extension Field GF(2^4)

Words of Z_2^4	Polynomials defined on Z_2 modulo $h(x)$	Powers of β
0000	0	0
1000	1	β^0
0100	x	β^1
0010	x^2	β^2
0001	x^3	β^3
1100	$1 \oplus x$	β^4
0110	$x \oplus x^2$	β^5
0011	$x^2 \oplus x^3$	β^6
1101	$1 \oplus x \oplus x^3$	β^7
1010	$1 \oplus x^2$	β^8
0101	$x \oplus x^3$	β^9
1110	$1 \oplus x \oplus x^2$	β^{10}
0111	$x \oplus x^2 \oplus x^3$	β^{11}
1111	$1 \oplus x \oplus x^2 \oplus x^3$	β^{12}
1011	$1 \oplus x^2 \oplus x^3$	β^{13}
1001	$1 \oplus x^3$	β^{14}

It can be shown that a BCH code as defined above has a distance $d \geq 2t + 1$. This means that the correction capacity t_c is *at least* equal to t. The quantity $\delta = 2t + 1$ is called the *designed distance* of the BCH code.

Example: A Binary BCH Code with a Correction Capacity of 2

Let β be a primitive element of GF(2^4). It has already been mentioned that this extension field of GF(2) can be constructed with the *primitive polynomial* $h(x) = 1 \oplus x \oplus x^4$. The BCH code with correction capacity $t_c = 2$ and length $n = 15$ is generated by the following generator polynomial:

$$g(x) = m_{\beta^1}(x)\, m_{\beta^3}(x) \equiv m_1(x)\, m_3(x)$$

where $m_i(x)$ is the minimal polynomial of β^i.

Table VII Minimal Polynomials of the Extension Field GF(2^4)

Elements of Z_2^4	Minimal polynomial
β^0	$1 \oplus x$
$\beta^1, \beta^2, \beta^4, \beta^8$	$1 \oplus x \oplus x^4$
$\beta^3, \beta^6, \beta^9, \beta^{12}$	$1 \oplus x \oplus x^2 \oplus x^3 \oplus x^4$
β^5, β^{10}	$1 \oplus x \oplus x^2$
$\beta^7, \beta^{11}, \beta^{13}, \beta^{14}$	$1 \oplus x^3 \oplus x^4$

Using the minimal polynomials from Table VII, the generator polynomial of this BCH code is thus:

$$g(x) = (1 \oplus x \oplus x^4)(1 \oplus x \oplus x^2 \oplus x^3 \oplus x^4) = 1 \oplus x^4 \oplus x^6 \oplus x^7 \oplus x^8$$

This polynomial is a factor of $1 \oplus x^{15}$, which is a necessary and sufficient condition to be a generator polynomial.

In a more general manner, it can be shown that BCH codes of correction capacity $t_c = 2$ and of length $n = 2^m - 1$ are cyclic codes with generator polynomial $g(x) = m_{\beta^1}(x)\, m_{\beta^3}(x)$, where β is a primitive element of the GF(2^m), $m \geq 4$.

5. Reed-Solomon Codes

These codes, which are members of the family of BCH codes, do not use a binary but a q-ary alphabet. This might appear strange since they are most often used in binary transmission channels. We will show to that purpose that they posses a so-called *binary representation,* which leads to interesting properties in the field of the correction of *burst errors.*

In a BCH code, the α^i, zeros of the polynomials associated with the codewords, belong to the extension of the field F, which constitutes the alphabet for the codewords. The Reed-Solomon codes are specific cases of the BCH codes in which the α^i are chosen in F itself and not in its extension!

This has as a consequence that the minimal polynomial [which belongs to $F(X)$] associated with an element α^i is nothing more than $(x - \alpha^i)$. Furthermore, a Reed-Solomon code is a primitive BCH code, which implies that its length $n = q - 1$.

a. Definition

A Reed-Solomon code of correction capacity t, is a primitive BCH code for which the generator polynomial is:

$$g(x) = (x - \beta^1)(x - \beta^2) \ldots (x - \beta^{2t})$$

where β is a primitive element of the field F of order q, defining the alphabet of the code.

b. Properties

1. The minimal distance of the code defined above is $2t + 1$.
2. The dimension of this code is $n - 2t = q - 1 - 2t$.

Example: A Reed-Solomon Code in GF(2^3)

The construction of the extension field GF(2^3) can be done in the same way as we have shown in the section on the BCH codes. In this case the primitive polynomial is:

$$h(x) = 1 \oplus x \oplus x^3$$

and the elements of the field are given in Table VIII. The Reed-Solomon code using an octal alphabet and with correction capacity $t = 1$ has the following generator polynomial:

$$g(x) = (x \oplus \beta^1)(x \oplus \beta^2) = \beta^3 \oplus \beta^4 x \oplus x^2$$

The codeword corresponding to the word $(1, \beta^1, 0, 0, \beta^3)$ to be coded is then $(\beta^3, 0, \beta^4, \beta^1, \beta^6, 1, \beta^3)$.

c. Transformation of a q-ary Code in a Binary Code

Let C be a Reed-Solomon code defined on an alphabet $F = \mathrm{GF}(2^m) = \mathrm{GF}(q)$. As can be seen in Table VIII, for every symbol of the alphabet there is a corresponding m-tuple composed of binary symbols. The q-ary code C of length $n = 2^m - 1$ can thus be replaced by a binary code C^* of length nm by replacing each q-ary symbol with its binary representation (1 q-ary symbol is replaced by m binary symbols).

Example

The octal codeword $(\beta^3, 0, \beta^4, \beta^1, \beta^6, 1, \beta^3)$ from the previous example leads thus to the binary codeword (110, 000, 011, 010, 101, 100, 110).

d. Protection Against Burst Errors

Certain transmission channels are such that the errors are grouped. This is called an *error burst*. Reed-Solomon codes do possess a high correction capacity for this type of error, thanks to the mechanism of the transformation of a q-ary code in a binary code. If the correction capability of the q-ary code C is t, and if the field F used is $F = \mathrm{GF}(2^m)$, then it can be shown that the derived binary code C^* can correct an error burst of $(t - 1)m + 1$ binary elements. This guarantees indeed that maximum t adjacent q-ary symbols are erroneous. This supposes of course that the different successive error bursts are separated by error-free intervals that are sufficiently long to prevent two successive error bursts from being located in the same word.

6. Fast Decoding of BCH Codes

Let us consider a q-ary BCH code of length n and designed distance δ, defined on a Galois field F of order q. It is clear that for $q > 2$, it is not sufficient to know the *position* of the errors in order to be able to correct

Table VIII Three Possible Representations of the Extension Field GF(2^3)

Words of Z_2^3	Polynomials defined on Z_2 modulo $h(x)$	Powers of β
000	0	0
100	1	β^0
010	x	β^1
001	x^2	β^2
110	$1 \oplus x$	β^3
011	$x \oplus x^2$	β^4
111	$1 \oplus x \oplus x^2$	β^5
101	$1 \oplus x^2$	β^6

them, as it is when using binary codes. Therefore, we introduce two important polynomials: the *(error) localizer polynomial* $\sigma(x)$ and the *(error) evaluator polynomial* $\epsilon(x)$. It can be shown that knowing these polynomials does allow for the correction of the errors (within the limits of the correction capacity of the code).

Different algorithms that allow these two polynomials to be calculated do exist: the matrix method, the method of Berlekamp-Massey, and the method based on the algorithm of Euclid. These methods fall outside the scope of this article but they are explained in the Bibliography entries.

VI. CONVOLUTIONAL CODES

A. Introduction

Convolutional codes were introduced around the mid-20th century as a possible alternative to block codes. As we have already explained, block codes are limited to the transmission of codewords in blocks. Block codes were the first type of error-detecting and -correcting codes to be investigated. Even today they are the object of a major part of the actual research in error-detecting and -correcting codes. In contrast, convolutional codes have already in many practical applications proved to be at least as efficient as block codes and, furthermore, they have the advantage that they tend to be much easier to implement than comparable block codes.

In the case of block codes $B(n, k, d)$ (in which n represents the number of symbols in the output sequence, k is the number of symbols in the input sequence, and d is the distance of the code), it is true that every one of the $(n - k)$ redundant symbols in a codeword depends only on the k corresponding information symbols of the input sequence, and on no other information symbol. This means that the generator of a block code operates *without memory.*

Convolutional codes on the other hand are not limited to this form of block coding. Indeed, they can handle a continuous semi-infinite stream of information symbols. These are then treated, using a *sliding window,* which characterizes the *memory effect,* and a continuous stream of coded symbols (codewords) is then generated. Every information symbol stays in the sliding window for a certain finite time period. During this *active* time period the corresponding information symbol influences the generated sequence of codewords.

B. Connection-Oriented Generation of Convolutional Codes

In the most general case, a convolutional code $C(n, k, K)$ is generated by sending the input sequence to be transmitted through a linear shift register with a finite number of states. Such a shift register consists primarily of K cells of k *binary* moments. The content of these different cells is then used in n different linear algebraic function generators (in modulo-2 adders, also called XOR operators), as shown in Fig. 2.

The choice of connections between the modulo-2 adders and the different cells of the shift register determine the characteristics of the generated code. Every modification in the choice of these connections results in a different code. The problem of choosing the correct connections, in order to obtain good distance properties, is complicated and has not been solved yet in the most general case. Even so, good convolutional codes have been found already for all constraint lengths smaller than approximately 20 using exhaustive search programs.

As opposed to a block code, which is characterized by a fixed word length n, a convolutional code has no determined block length. However, it does happen that the convolutional codes are forced into a certain block structure by a so-called *periodic truncation.* This way of working requires that a number of zero binary moments be appended to the end of the input sequence. These zeros then take care of removing all remaining data binary moments from the shift register (*flushing*) or, in other words, of resetting the content of the shift register to zero (*clearing*). Since these added zeros do not carry information, they decrease the effective code rate. To keep the code rate as close as possible to k/n, the period of this periodic truncation is in general made as large as possible.

A method for the description of a convolutional code is to give its *generator matrix,* just as we did for the block codes. As an alternative for specifying the generator matrix, a functional equivalent representation is used here. In this representation a set of *n connection vectors* is specified, one for each modulo-2 adder. Each connection vector has kK dimensions and contains the connections between all cells of the shift register and the respective modulo-2 adder. A 1 in the ith position of this vector points at the existence of a connection between the ith cell of the shift register, which has kK cells in total, and the considered modulo-2 adder. A 0 in the ith position of that vector is then a sign that no connection exists between the ith cell of

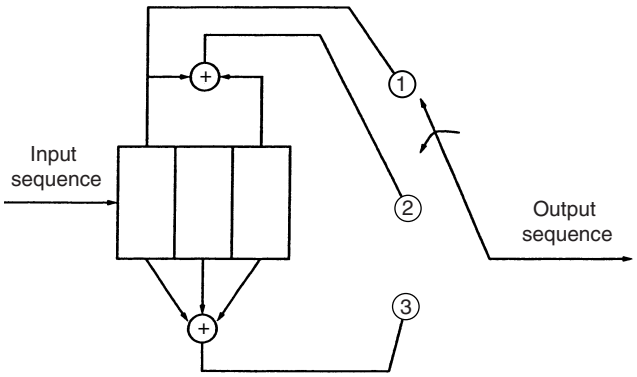


Figure 3 Example of a $K = 3$, $k = 1$, $n = 3$ convolutional code generator.

the shift register and the corresponding modulo-2 adder.

To be more specific we present now an example. Let us consider the convolutional coder with constraint length $K = 3$, the number of input binary moments which are introduced in the shift register at the same time $k = 1$, and the number of generated binary output moments per k input binary moments $n = 3$. This convolutional coder will be used here as a connection-oriented model for studying convolutional code generators. This code generator is represented in Fig. 3. The time representation of the input and output sequences is represented in Fig. 4.

Let us now make the following assumption: We number the different outputs of the function generators that generate the 3-binary moment output sequences from top to bottom as 1, 2, and 3. We suppose that the same numbering scheme is used for differentiating between the three corresponding function generators. Taking into account these conventions we can then construct the following connection vectors. Because only the first cell of the shift register is connected to the first function generator (no modulo-2 adder has been used here), the corresponding connection vector is:

$$g_1 = [100]$$

The second function generator is connected to positions 1 and 3. The corresponding connection vector is then:

$$g_2 = [101]$$

Finally, we find for the third function generator in a similar manner the following connection vector:

$$g_3 = [111]$$

From this we can conclude that when $k = 1$, specifying the code generator needs n connection vectors, each one of dimension K. More generally, for a binary convolutional code generator of constraint length K and with code rate k/n, where $k > 1$, the n connection vectors are kK-dimensional vectors.

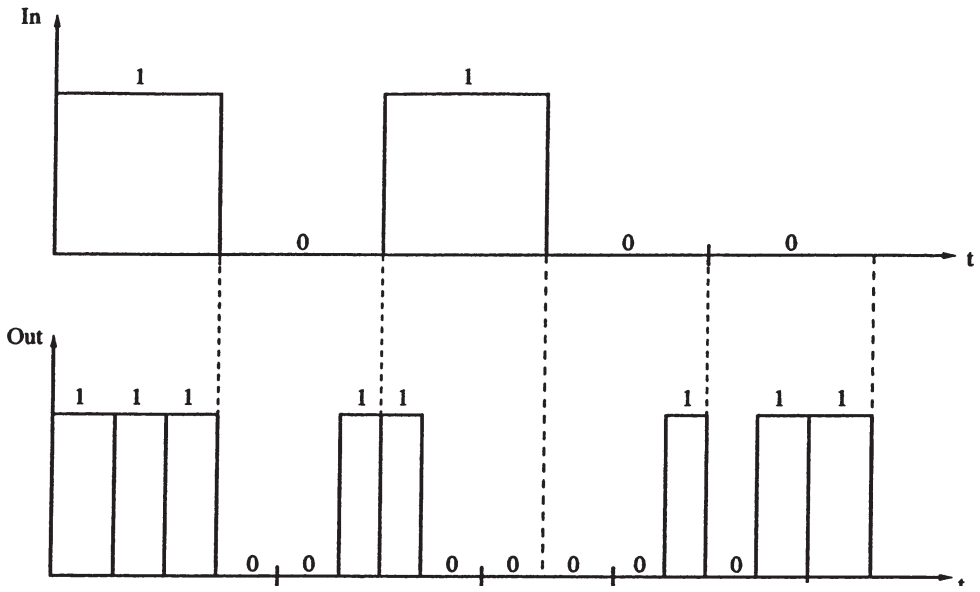


Figure 4 Time representation of the input and output sequences.

C. Impulse Response of a Convolutional Code Generator

We can also look at the convolutional code generator represented in Fig. 3 from the point of view of its *impulse response*. The impulse response in the case of a convolutional code generator is nothing less than the response of that code generator to one binary moment 1 shifted through it. Let us have a look in Table IX at the contents of the shift register of the example in Fig. 3, while a binary moment 1 is shifted through it.

The output sequence for the input sequence 1 is called the impulse response of the convolutional code generator. If we then look for the response of this code generator for the input sequence $m = 101$, then the corresponding output sequence can be found by *superposition* or *linear addition* of the responses of the time shifted *impulses*. This is shown in Table X. We can see that this generates the same output sequence as the one we have obtained earlier. This shows that convolutional codes are *linear*. It is this property, where the output sequence is generated as the *convolution* of the impulse response of the code generator with the input sequence, that has given rise to the name *convolutional* code.

Table IX Example of the Impulse Response of a Convolutional Code Generator

Register content	Output sequence
100	111
010	001
001	011
Input sequence	100
Output sequence	111 001 011

Table X Application of the Linear Superposition Principle to a Convolutional Code Generator

Input m	Output sequence
1	111 001 011
0	000 000 000
1	111 001 011
Modulo-2 addition	111 001 100 001 011

D. Some Other Possible Descriptions of a Convolutional Code

1. Tree Diagram

The tree diagram of the convolutional code generator represented in Fig. 3 is shown in Fig. 5. The rule to be followed in determining the path through the tree is that the upper branch needs to be chosen if the next input binary moment is a 0, and the lower branch if it is a 1. In this manner a particular path through the tree is chosen, which is completely determined by the input sequence. Supposing that the code generator contains at the start nothing but zeros; the tree diagram shows that, if the first information binary moment is a 0, the first output sequence is 000, and if the first input binary moment is a 1, the first output sequence is 111. If now the first binary moment is a 1 and the second a 0, then the second output sequence is 001. If we continue on this path, then we see that if the third input binary moment is a 1 the output sequence is 100.

The big disadvantage of a tree diagram is that the number of branches increases as a function of 2^L, where L represents the number of binary moments in the input sequence. However, for this problem there exists an elegant solution as shown in the next section.

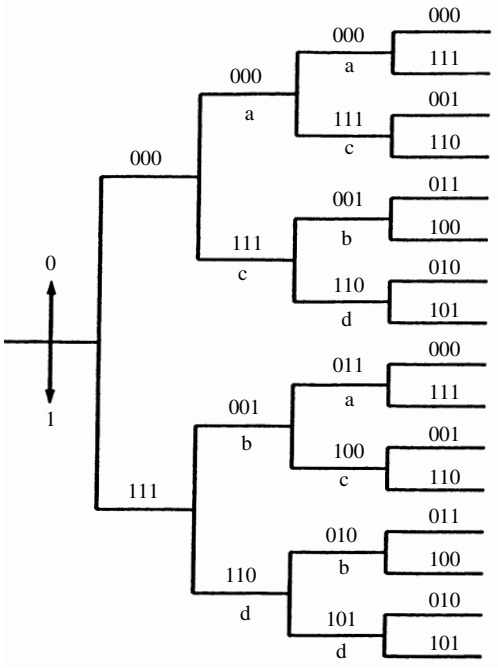


Figure 5 Tree diagram for a $K = 3$, $k = 1$, $n = 3$ convolutional code generator.

2. Trellis Diagram

If we take a closer look at the tree diagram of Fig. 5, we see that its structure repeats itself after the third step. This phenomenon is consistent with the fact that the constraint length of this convolutional code is $K = 3$. In other words, this means that the output sequence consisting of three binary moments at each step is determined by the current input binary moment and the two previous input binary moments (i.e., the two binary moments in the first two cells of the shift register). The data binary moment in the third (last) cell of the shift register is shifted out and therefore it has no influence on the output sequence. More generally stated we can say that the output sequence consisting of $n = 3$ binary moments is determined by the current input binary moment and by the $2^{kK-1} = 4$ possible states of the $kK - 1 = 2$ first cells of the shift register. These $2^{kK-1} = 4$ states are denoted in Fig. 5 as follows: $a = 00$, $b = 01$, $c = 10$, $d = 11$. Labeling each node in the tree diagram with the name of the state in which the first $kK - 1 = 2$ cells are at that moment, we see that in the third step there are two nodes with label *a*, two with label *b*, two with label *c*, and two with label *d*. We also can observe that all branches coming from two nodes with the same label (i.e., for which the two first cells of the shift register are in the same state) are identical; they generate the same output sequences. This means that two nodes with the same label can be merged. Such an operation transforms the tree diagram from Fig. 5 to a new, more compact diagram shown in Fig. 6. This new diagram is called a *trellis* diagram.

In the most general case the trellis diagram consists of 2^{kK-1} nodes in order to be able to represent all the possible states of the $kK - 1$ first cells of the shift register. The trellis diagram enters some kind of *steady state* after $K - 1$ steps.

E. Optimal Decoding of Convolutional Codes—The Viterbi Algorithm

1. Maximum Likelihood Decoding

If all possible input sequences (messages) are equiprobable, then it can be shown that the decoder that minimizes the *error probability* is the decoder that is based on the comparison of the *conditional* probabilities (often called *likelihood functions*) of the different possible transmitted codewords, after which the codeword (transmitted sequence) with the largest probability is chosen. Applying this maximum likelihood decision strategy for decoding convolutional codes, there are typically a very large number of allowed codewords that could have been transmitted. To be more specific, we can suppose that a codeword consisting of L binary moments is an element of the set of 2^L possible sequences of L binary moments. In this case we can say that the maximum likelihood decision strategy leads to a decoder which will consider a particular sequence as the most likely transmitted sequence, if its likelihood is larger than the likelihoods of all other possibly transmitted sequences. Such an optimal decoder is called a *maximum likelihood decoder*.

The decoding problem consists of choosing a path through the trellis diagram of Fig. 6 such that the *likelihood* function is maximized. The main advantage of the trellis diagram is that with this particular representation it is possible to realize a decoder that eliminates paths that cannot possibly be candidates any longer for the maximum likelihood sequence. The decoded path is then chosen from a list of *surviving paths*. Such a decoder can be shown to still be optimal, in the sense that the decoded path is exactly the same as the decoded path that would be obtained

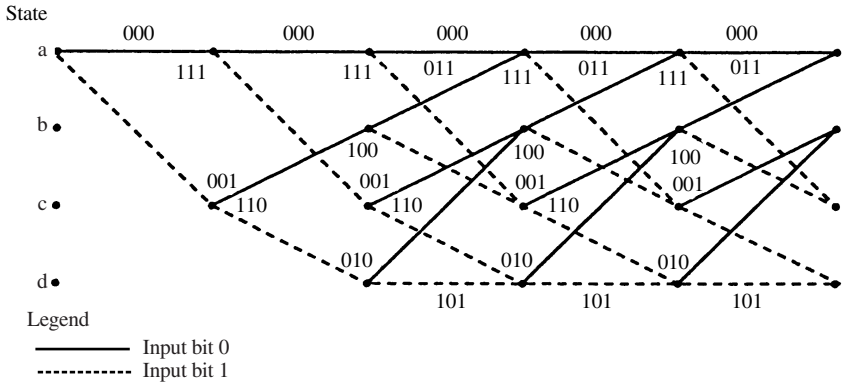


Figure 6 Trellis diagram for a $K = 3$, $k = 1$, $n = 3$ convolutional code generator.

through an exhaustive approach as described above. However, the early rejection of unlikely paths dramatically reduces the complexity of the decoder.

Different algorithms exist that propose *approximated* solutions for the maximum likelihood decoding problem. These approximate methods are all *suboptimal*. The *Viterbi decoding algorithm* implements a maximum likelihood decoder and is therefore *optimal*. However, this does not mean necessarily that the Viterbi algorithm is the best solution for every application, since this algorithm is very demanding with respect to the hardware complexity.

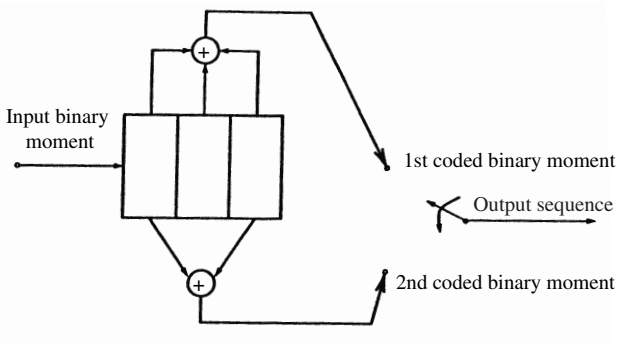


Figure 7 Simple convolutional coder with code rate $\frac{1}{2}$.

2. The Viterbi Decoding Algorithm

As already mentioned, the Viterbi decoding algorithm (discovered in 1967) implements essentially a maximum likelihood decoder. The computational load, however, is reduced by using the special structure of the trellis diagram. The main advantage of the Viterbi decoding scheme with respect to an exhaustive decoding method is that the complexity of a Viterbi decoder is not a function of the length L of the sequence to be decoded. The algorithm is based on the calculation of a *distance measure* between the received signal at time t_i and all paths in the trellis diagram that arrive in every possible state at that time t_i. The Viterbi algorithm eliminates those paths from the trellis diagram that can no longer be a candidate for the maximum likelihood choice. When two paths do arrive at the same time t_i in the same state, then the path which has the smallest distance measure is chosen. This path is called the *surviving path*. This selection of surviving paths is done for all states. The decoder continues in this manner to progress along the trellis diagram, in the meantime eliminating the less likely paths. In the particular case in which the two distance measures are the same, the survivor is chosen arbitrarily.

3. Application of the Viterbi Decoding Algorithm on a Simple Example

For simplicity reasons we consider a binary symmetric channel: the Hamming distance is thus the adequate distance measure. The simple convolutional coder for this example is presented in Fig. 7 and its corresponding trellis diagram is shown in Fig. 8. A similar trellis diagram can be used to represent the decoder, as shown in Fig. 9.

The idea at the base of the decoding procedure can best be understood by placing the trellis diagram of the coder from Fig. 8 next to that of the decoder from Fig. 9. For the trellis diagram of the decoder, label each branch of the trellis diagram at a moment t_i with the *Hamming distance* between the received sequence and the respective branch of the trellis diagram of the decoder. The example in Fig. 9 shows a message sequence m, the corresponding transmitted codeword sequence $U = 11\ 01\ 01\ 00\ 01\ \ldots$, and a noise-corrupted received sequence $Z = 11\ 00\ 01\ 00\ 01\ldots$. The codewords, which are linked to the branches of the trellis diagram of the coder, do characterize completely the coder from Fig. 7 and they are known *a priori* as well by the coder as by the decoder! These codewords appear then at the output of the coder. The labels (i.e., Hamming distances) associated with the branches of the trellis diagram of the *decoder* are accumulated immediately. This means that at the reception, each branch of the trellis diagram of the decoder is labeled with a similarity measure (the Hamming distance) between the received symbol and each one of the possibly transmitted symbols for that time interval.

In Fig. 9 we see that in sequence Z the received symbols at time period t_1 are 11. To be able to label the branches in the decoder at that time period t_1 with the appropriate Hamming distance, we look at the trellis diagram in Fig. 8. There we see that a transition from state 00 to state 00 generates the codeword 00. But in reality we did receive 11. Therefore the transition between state 00 to 00 in the decoder is labeled with the Hamming distance between these two sequences, namely, 2. In this manner all branches of the trellis diagram of the decoder are labeled as the symbols are received at each time interval t_i. The decoding algorithm now uses these Hamming distances in order to find the *most probable* (smallest Hamming distance) path through the trellis diagram.

At the base of the *Viterbi decoding* lies the following observation: Every time two arbitrary paths do con-

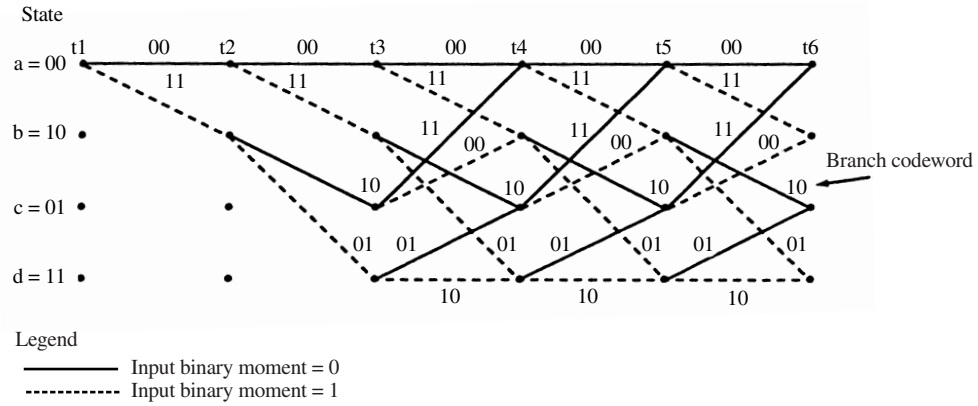


Figure 8 Trellis diagram of the convolutional coder presented above.

verge in one single state, one of those two can always be eliminated in our quest for the optimal path. To illustrate this, Fig. 10 shows the case of two paths which converge in state 00 at time interval t_5.

We define the *accumulated Hamming path distance* for a given path at time interval t_i as the sum of the Hamming distances of all branches which are part of that specific path. The upper path from Fig. 10 has a path distance of 2, while the lower one has a path distance of 3. This means that the lower path can never be part of the optimal path (in the sense of the smallest distance), since the upper path, which arrives in the same state, has a lower distance measure.

At each time interval t_i there are 2^{K-1} states in the trellis diagram, where K represents the constraint length. Each state can in steady-state condition always be reached via two paths. Viterbi decoding implies determining for those two paths the distance measures and the subsequent elimination of that path with the largest one. This calculation of the path distances is done for each one of the 2^{K-1} states at time interval t_i; after that the decoder moves on to the next time interval t_{i+1} where the whole process is repeated. The first few steps in our example are the following ones (see Figs. 11 through 14).

At time interval t_1 the received symbols number 11. The only possible transitions from state 00 are to state 00 or to state 10, as can be seen in Fig. 11. The state transition from 00 to 00 has distance measure 2 and the transition from 00 to 10 has distance measure 0.

At time interval t_2, two possible branches leave from each state, as can be seen in Fig. 12. The corresponding accumulated path distances are represented next to the respective end nodes.

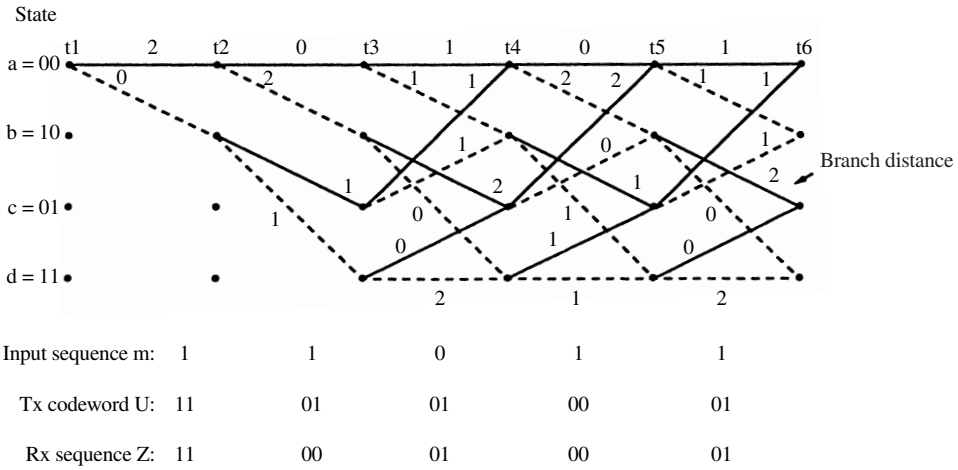


Figure 9 Example of a trellis diagram for a $K = 3$, $k = 1$, $n = 2$ convolutional decoder.

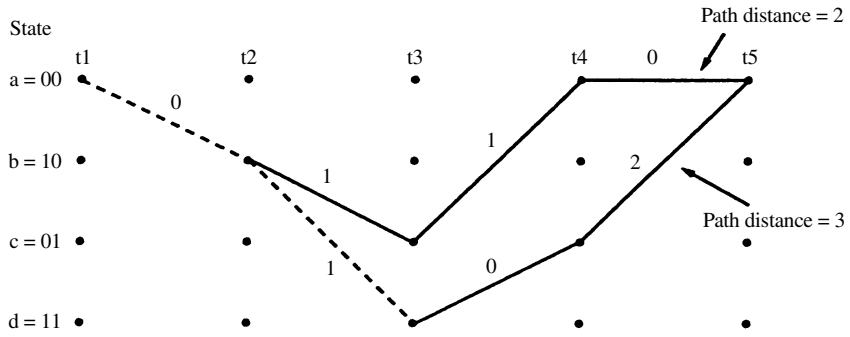


Figure 10 Path (Hamming) distances for two converging paths.

At time interval t_3, represented in Fig. 13, again two branches leave from each state and two paths converge in each state at time interval t_4. The remaining path for each state is represented in Fig. 14. At that time interval in the decoder there is only one path left between time intervals t_1 and t_2. That allows the decoder to decide that the state transition which has occurred between those time intervals was the transition from state 00 to 10. Because this transition was caused by an input binary moment equal to 1, the decoder outputs a 1 as the first decoded binary moment. Note that the first binary moment was only decoded after the calculation of the path distances had advanced much further in the trellis diagram. For a typical decoder implementation this means a *decoding delay* which can become as high as five times the constraint length.

At each following step in the decoding process, two possible paths always arrive in each state. Each time, one path will be eliminated after comparing the respective path distances. The decoder continues in this manner deeper in the trellis diagram and makes decisions with respect to the (possible) input binary moments by eliminating all paths but one.

4. Remark Related to the Memory Capacity Needed

The memory requirements for the Viterbi decoder grow in an exponential manner with the constraint length K. In the case of a code with rate $1/n$, the decoder stores after each decoding step a set of 2^{K-1} paths. It is very likely that all of these paths have a common part at their root, and that it is only at their end that they fan out to the different states. So if the decoder stores a sufficiently large part of the history of the paths, then under that hypothesis the *oldest* bits on all paths will be the same. In that case it is possible to implement a simple decoder by storing only a *fixed* path history. Such a decoder then adds with each step the oldest bit of an arbitrary path to the decoded sequence. It is possible to show that a fixed path history of a length of four to five times the constraint length K, is sufficient for near-optimal performance of this simple decoder. The memory requirement is the main constraint with respect to the implementation of a Viterbi decoder.

F. The (Free) Distance of a Convolutional Code

We now study the distance properties of convolutional codes using the example of the convolutional coder

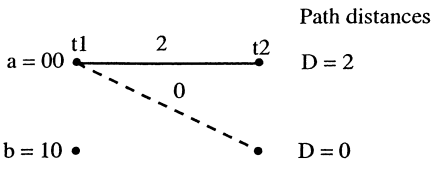


Figure 11 Remaining paths at time interval t_2.

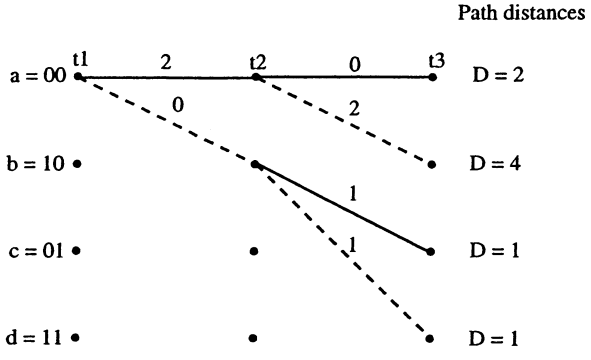


Figure 12 Remaining paths at time interval t_3.

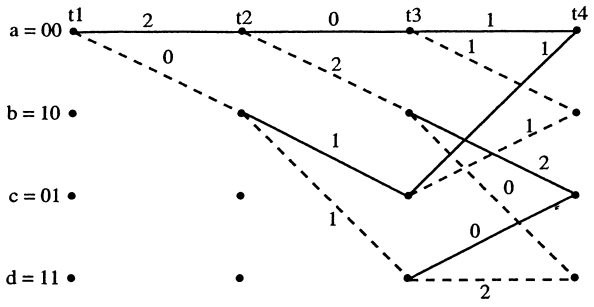


Figure 13 Distance comparisons at time interval t_4.

presented in Fig. 7. The corresponding trellis diagram is given in Fig. 8. Because convolutional codes are *linear* codes, what we need to do now is evaluate the distance between all possible pairs of codeword sequences, in order to find the minimal distance. As in the case of block codes, it is this minimal distance which determines the error-correcting capacity of the convolutional code.

It can be shown that the minimal distance of a convolutional code can be found by simply calculating the distance between each codeword sequence and the sequence consisting of all zeros. The only paths we are interested in are those paths that first diverge from the all-zeros path and later converge again to the all-zeros path. If we suppose that the all-zeros sequence was the transmitted sequence, then an error will occur every time the distance of an arbitrary path different from the all-zeros path which converges at time t_i in the state $a = 00$ is smaller than the distance of the all-zeros path at time t_i. This would indeed lead to the elimination of the all-zeros path at time t_i, which is clearly erroneous. The minimal distance for making such an error can be found by investigating exhaustively every path that diverges and converges again later at the state $a = 00$. To do this, it is easier to use another version of the "classic" trellis diagram. In Fig. 15 every branch of the trellis diagram is labeled with its respective Hamming distance to the path consisting of all zeros.

Using the information in this "modified" trellis diagram, we can now easily compute the distance between every path which diverges first and converges later, and the path consisting of all zeros. The minimal distance is called the *free* distance and in the case of our example this free distance equals 5. To determine the error-correcting capacity t_c of a convolutional code, we can use the same expression as in the case of block codes, using the notion defined earlier of free distance d_f:

$$t_c = \left\lfloor \frac{d_f - 1}{2} \right\rfloor$$

VII. CONCLUSIONS

As can be seen from this short introduction to the subject, a lot of different error-detecting and -correcting codes do exist. They all have their specific advantages and drawbacks and no one single code is best suited for all applications/situations. A possible use of these different codes, in which the goal is to combine the best characteristics of several ones, is the use of so-called *concatenated* codes. A concatenated code is a code that uses two levels of coding (an inner code and an outer one) in order to achieve the

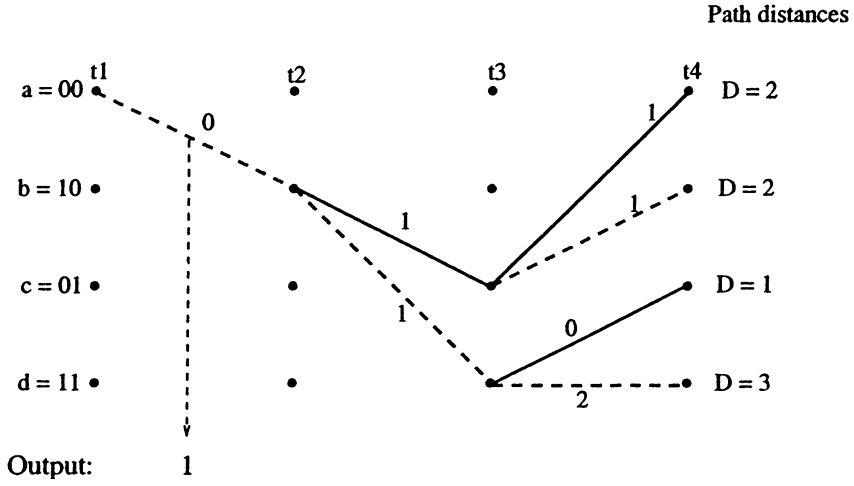


Figure 14 Remaining paths at time interval t_4.

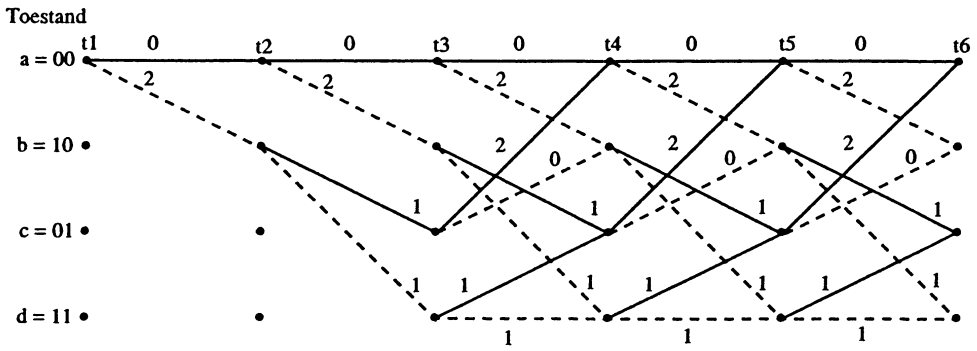


Figure 15 Trellis diagram showing the Hamming distance with respect to the all-zeros path.

required error performance. This subject can be studied further in the presented literature.

SEE ALSO THE FOLLOWING ARTICLES

Electronic Data Interchange • Information Theory • Pseudo Code

BIBLIOGRAPHY

Blahut, R. E. (1983). *Theory and practice of error control codes.* Reading, MA: Addison-Wesley.

Clark, G. C., and Cain, J. B. (1981). *Error-correction coding for digital communications.* New York: Plenum Press.

MacWilliams, F. J., and Sloane, N. J. A. (1977). *The theory of error-correcting codes.* Amsterdam: North-Holland.

Pless, V. S., and Huffman, W. C. (1998). *Handbook of coding theory.* Amsterdam: Elsevier.

Proakis, J. G. (1995). *Digital communications,* 3rd ed. New York: McGraw-Hill.

Reed, I. S., and Chen, X. (1999). *Error-control coding for data networks.* Norwell, MA: Kluwer Academic.

Sklar, B. (1988). *Digital communications—fundamentals and applications.* Upper Saddle River, NJ: Prentice Hall.

Vanstone, S. A., and van Oorschot, P. C. (1989). *An introduction to error correcting codes with applications.* Norwell, MA: Kluwer Academic.

Ethical Issues

Effy Oz
Pennsylvania State University

GLOSSARY

informed consent A person's consent to use information about the person after the person has been informed about how the information will be used.
intellectual property The results of intellectual work, usually in the form of books, music, pictures, plays, and software applications.
privacy A person's ability to control information about himself or herself.
software piracy Illegal copying of software. The term *piracy* often refers to the illegal copying of any work in digital form.
spamming Sending an e-mail message, usually of a commercial nature, to multiple people who did not solicit it.

ETHICS is a branch in philosophy that studies morality. It deals with right and wrong acts. *Ethical* is often synonymous with *right,* while *unethical* is often synonymous with *wrong*. However, there is hardly any absolute ethical and unethical behavior in terms of time and location. What might once have been considered ethical (e.g., polygamy) is now unethical. What is considered ethical in some countries (copying someone's creative work) is considered unethical in others. Although we do not elaborate on ethical theories in this article, the reader is encouraged to review the major ethical theories, utilitarian as well as deontological. However, modern means of communication and transportation have gradually eroded national and cultural differences and, therefore, the differences in ethical approach to ethical issues.

Historically, every major technological change has prompted discussion of the adequacy of current social and ethical norms. Often, these discussions have resulted in new political doctrines, social agreements, ethical codes, and legislation. One need only look back and see what happened in the Industrial Revolution: A sense of injustice produced an ongoing public debate on employer–employee relations, socialist movements, and social legislation that stemmed out of a feeling that the world we were used to yesterday is no longer our world today. Similar discussions and debates have taken place since information technology (IT) started to permeate our lives in the mid-1980s. Ethical norms have been created; legislation followed; and the debates are still going on. The rapid development of a global information network, the Internet, has only intensified calls for ethical norms and legislation regarding privacy, freedom of speech, and intellectual property rights. This article surveys the major ethical issues relating to IT. Undoubtedly, as the technology develops, new issues will arise.

I. THE NEED FOR ETHICAL NORMS IN THE INFORMATION AGE

In broad terms, we can see several major changes in human history during the past 5000 years. In the Western World, more than two-thirds of the workforce is engaged in work that either produces information or

relies on information and does not produce any tangible goods. Education relies on information systems and computer networks. At home, we spend an increasing amount of time with computers for entertainment and communication. A growing number of working people rely on IT to work from home rather than in organizational offices.

The growing reliance on IT has brought up grave issues: Employees and consumers are losing control of information that reveals much of their private lives; it is relatively easy to steal information in digital form; free speech can be supported by the technology, but there are concerns about unrestricted transmission of violent, racist, and sexual material on the Internet; violation of intellectual property rights has increased; and other crimes, popularly referred to as computer crimes, are proliferating. These issues have prompted some countries to pass laws specifically dealing with acts performed with IT. The laws include prohibition of certain uses of computers, such as unauthorized access to computer systems; limits on collection, maintenance, and dissemination of personal data; rules for duplication of digital intellectual property; and other rules whose purpose is generally to limit our use of IT for purposes that do not harm the well-being of other people.

As often happens with ethical issues, we strive to resolve the collision of interests of two or more parties. For instance, employers want to monitor their employees to ensure productivity and security while employees resist infringement on their privacy rights. Governments want to be able to monitor communication of people suspected of illegal activities, while citizens resist government attempts to gain access to their communication, much of which is now executed through computer networks. Individuals pursue free speech and want to be able to post on the World Wide Web any material they wish to post, while other individuals are concerned that some people, especially children, may be harmed by certain information. Meanwhile, millions of IT professionals have a tremendous impact on our lives but do not have to comply with any mandatory code of ethics as other professionals do. Clearly, IT has created situations that call for reconsideration of current ethical codes and initiation of new ones to deal with the issues. In many cases, the public debate on an IT ethical issue has found its way to legislatures, which pass laws to address concerns.

II. CULTURAL DIFFERENCES

Ethical conflicts often occur when two cultures encounter each other. Cultures have different social values and therefore promote acts that may be deemed ethical in one culture but unethical in another. One such example is the difference in approach to intellectual property. In the West, intellectual property is simply that: the property of the person or organization that created it, be it a book, a painting, musical piece, or a software application. In many Asian countries, the concept is relatively new, because for many centuries artistic creations were not the property of the person who created them. So, while in the United States the painter of a new piece of art is entitled to royalties from those who copy her work, in China the greatest honor a painter can receive is to have many people copy his work; for many centuries painters did not expect any material compensation for the copying. The introduction of software in countries with this culture has created friction between old traditions and new realities, whereby software authors (who are mainly Western organizations) demand financial compensation for copies of their work. While governments in these countries yielded to Western pressure and passed intellectual property laws similar to the Western ones, old habits die hard and pose a challenge both to the authors and the local governments.

Similar cultural differences exist with regard to privacy and free speech. While Americans treasure their personal privacy in dealings with governments, they have largely accepted violation of privacy by private organizations. This is probably the result of a long promotion of personal freedom while at the same time espousing free markets and competition of commercial enterprises. In Western Europe, the quest for privacy progressed regardless of who the potential violator might be; privacy laws treat all organizations equally, whether governmental or private.

Free speech is sanctioned in the constitutions of many Western countries, but is not regarded as a supreme value in many other countries. In Asian, Arab, and some African countries, values such as harmony of the family and the community far supercede the value of free speech, which is individualistic in essence. Thus, when means such as the Internet became available, Westerners took advantage of them to voice their opinions more than individuals in other nations. Of course, political systems also have a major role in the measure of how much free speech citizens are allowed; however, the political systems themselves are often the result of cultures that allow them to exist.

III. INVASION OF PRIVACY

In the context of information, privacy is the ability of individuals to control information about themselves. In the past, collection, maintenance, and dissemina-

tion of any information, let alone, personal information, was expensive, labor intensive, and paper based. Nowadays, it is inexpensive, involves little labor, and is digital. Huge amounts of personal information, such as credit card numbers and Internet activities, can be automatically collected via digital means, fed into databases, manipulated, duplicated, and transmitted using IT. The threat to privacy has increased greatly since the introduction of computers into the business world in the 1950s and 1960s, but it intensified even more in the mid-1990s with the opening of the Internet to commercial activity.

A. Violation of Consumer Privacy

Consumers are often asked to return warranty cards of the products they purchased with much more information than is required for ensuring warranty of the product. Questions such as "What is your favorite hobby" and "How many alcoholic beverages do you consume per week" are not unusual. The same questions are asked of people who log on to a Web site and wish to participate in sweepstakes, download software, or receive information. The data are channeled into large databases. Database management systems are used to manipulate the data in almost any manner imaginable. It can be sorted by demographic and geographic characteristics, matched with other data that the individual provided in the past to another organization, and practically produce an elaborate personal dossier of the individual. Such dossiers may include credit history, employment history, medical profile, buying patterns, religious and professional affiliations, and even social relationships and political inclinations. Telecommunications technology lets the owner of such a database transmit millions of personal records to other organizations, usually for a fee, within minutes.

Commercial organizations claim that they need information about consumers to compete in a free market. The information can help them produce products that the consumers need and market them to the individuals most likely to purchase them. This helps not only the organizations, which can save costs when "target marketing," but also provides consumers with better products and services. Civil rights advocates, on the other hand, argue that personal dossiers held by organizations violate privacy. They argue that when an organization collects personal information, it should disclose several items to the individual: the purpose of collecting the information, the intended use, the length of time the information will be held, whether the information may be communicated to other organizations, and what security measures will be taken to ensure that only people with a "need to know" will access it. In addition, many privacy advocates demand that organizations give individuals an opportunity to scrutinize their records and ensure that proper corrections are made when the individual legitimately asks for corrections. Some advocates also insist that organizations should pay individuals every time they sell the individual's record to another party.

In countries that are members of the European Union, many of these demands have been met by laws. The 15 members have enacted privacy laws that conform to the European Directive on Data Protection *(Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data)*. The directive requires any organization, whether a government agency or a commercial enterprise, to tell individuals how the data will be stored and used, and for how long. It guarantees them scrutiny of their records and that they will be notified whenever the organization intends to transfer the data to another organization. Furthermore, no decision based solely on automated processing of personal data may be made regarding credit or employment.

The Internet, and especially the commercialization of the Web, have intensified the issue of consumer privacy. Cookie technologies help companies collect personal information even if the surfer is not aware of it. *Cookies* are information files kept on the hard disk of the user. They accumulate information such as access codes, Web pages to which the user logged on, and the mouse's click stream. Over time, a cookie may collect enough information to describe the shopping habits and other characteristics of the user. While the idea is to make the Web experience of the user more efficient (no need to reenter access codes and account numbers) and effective (automatically taking the user to pages of greater interest), privacy advocates argue that they expose much of a person's private life to commercial organizations.

Software technologies such as data warehousing and data mining exacerbate the privacy problem. Companies channel millions of records from transactional databases to large archival databases. Some companies, especially large retail chains and financial institutions, have amassed several terabytes of data each. Special applications that combine artificial intelligence and statistical methods "mine" the data for shopping patterns, fraud patterns, and other patterns. To minimize identification of individuals when using such data, corporations have tried to develop privacy-enhancing technologies. However, the possibility of identification always exists.

Almost all of the information accumulated about consumers comes from the consumers themselves. Often, they are notified that they must provide information to receive a service. If they agree, the act is called "informed consent"; they know how the information will be used, and they give it out of their free will in return for something, such as a product, a service, or participation in a sweepstakes. Many Web sites also let consumers choose not to allow the organization to transmit the information to any other party. Yet, concern is growing about privacy on the Web, especially regarding information provided by minors.

B. Violation of Employee Privacy

Employers have two main purposes for monitoring workers: ensuring productivity and enforcing security. For many years the two purposes were pursued by supervisors who were physically present at the place where work was performed. The television era ushered in the ubiquitous closed-circuit camera, which allows managers to visually monitor their subordinates from a distance with or without an early warning. As of the mid-1980s, when millions of workers started using personal computers for their daily work, the PC itself has become a means of monitoring. By connecting the PCs to an organizational network, supervisors can monitor everything that an employee does with a computer, from the number of keystrokes per minute to electronic mail messages to Web pages viewed by the worker.

Business leaders keep arguing that they must ensure productivity by monitoring without prior warning. They argue that a prior warning defeats the purpose, because it changes the workers' behavior. Workers have claimed that unannounced electronic monitoring causes stress, fear, and tension. Attempts in the U.S. Congress to legislate against monitoring without early warning have failed. Employee claims that employers violate their privacy when supervisors monitor their e-mail messages have been rejected by U.S. courts. The courts maintain that since the equipment belongs to the employer, the employer is entitled to enforce organizational policies as to the content of e-mail and Internet information that the employees send and receive.

IV. ETHICAL ISSUES ON THE INTERNET

The Internet has connected millions of people the world over. It is not limited to any national territory, and hence provides opportunities to improve education, commercial activities, and the quality of work life. However, it has also generated phenomena of questionable ethicality.

A. Free Speech

The Internet, especially its most popular application—the Web, became a new means of mass communications in the 1990s. Unlike most television transmission, Web pages can be received globally, rather than within a limited territory. Anyone who has a computer with access to the Internet can post Web pages with any content in the form of text, still pictures, animation, sound, and streaming video. This may include materials that are offensive to some people, such as virulent racist slurs, pornographic images, and images of extreme violence. In some countries such posting is limited. For example, the German state of Bavaria threatened America Online (AOL), the Internet service provider and information service company, that its service would be banned in that state unless it blocked Web pages with pornographic content. The Chinese government systematically blocks access to sites that post "unacceptable" materials, including pornography and political information that government officials regard as offensive.

Free speech is sanctioned in the constitutions of some countries, including the United States and West European states. Yet, large parts of the populations of these countries have demanded that information posted on the Internet be censored. In 1996, the Decency in Communication Act was passed by the U.S. Congress but was later struck down by the Supreme Court as unconstitutional. The court placed the responsibility at the receiving end: If a person does not wish to receive such information, he or she should not log on to the Web site. Claims that children may have access to violent or pornographic Web pages did not sway the court, which insisted that it is parents' responsibility to monitor what their children do. Yet, in other Western countries censorship has been applied to the Internet. For instance, a French court found an Internet service provider (ISP) responsible for pornographic images that his clients posted. In protest, the ISP shut down the service and dissolved the firm.

One must understand that any censorship of the Internet is limited. It is practically impossible to block access to every Web site that transmits unacceptable information. As an increasing number of Internet lines become unguided, that is, use satellites, setting a control office that monitors all transmission via

guided (cables, optical fibers, and the like) lines will become impossible, because an increasing number of people will receive information using satellite antennas.

B. Spamming

Soon after facsimile (fax) machines became commonplace in many households, legislatures rushed to pass laws against broadcasts of fax advertisements. However, there is no law that forbids commercial enterprises to advertise through electronic mail. E-mail advertising is one of the least costly means of advertising. Many businesses broadcast daily advertisements to people whose e-mail address they obtained either directly, mainly through the Web, or purchased from companies that sell e-mail address lists.

Massive broadcast of e-mail, especially for commercial purposes, is popularly called *spamming*. Many people resent the phenomenon of "junk e-mail," masses of e-mailed advertisements. Such e-mail clogs their e-mail boxes, waste their times trying to separate useful e-mail from spam, and takes up disk space and CPU time that could otherwise be used for useful work. Several states in the United States have proscribed spamming, but the majority of states and other countries allow it.

C. Cybersquatting

For several years after the Internet was opened for commercial activities, anyone could receive a Web domain name (such as http://www.mycompany.com) free of charge. Now, domain names are sold on a first-come, first-served basis. Some people saw an opportunity to capitalize on the opportunity and purchased domain names containing the names of commercial organizations or generic names that might be attractive to Internet entrepreneurs, and then offered them for sale. Reportedly, some domain names, such as http://www. drugs.com have been sold for hundreds of thousands of dollars.

As long as the names were generic, there were no complainers. However, commercial organizations claim that domain names should be treated the same way as trademarks: They uniquely associate products and services with a specific company, and therefore should not be used without permission by anyone except the company. People or organizations that register domain names with intention to sell them later are popularly called *cybersquatters*.

American courts have accepted the argument. If the domain name is generic, the party that registered it is entitled to it. However, if someone registered a name that had been the trademark of a business, the business is entitled to it. In such a case, the business must only pay the holder back the registration fee and can then begin using it. Some people claim that the Web is a new business frontier and should not be ruled by traditional conventions and laws; thus, trademark law should not be applied to the Web. The debate is likely to continue.

D. Gambling

Gambling, in itself, has been the topic of ethical debate for many years. It is legal in some states, illegal in others, and pronounced immoral by many people regardless of its legal status. The Internet poses serious challenges to communities that control gambling. There is no practical way to stop people from gambling online. All they have to do is provide a credit card number to the online casino and then play the games online. The social danger may be greater than gambling from locations where gambling is prohibited; it is the ability of anyone with access to the Internet to gamble from the comfort of one's home or office at any time of the day. This may lure more people to gamble with larger amounts of money and result in a greater number of ruined families. Proponents of online gambling argue that it simply provides a choice, and that the choice should remain with individuals, not governments.

E. Daytrading

Online trading of securities, especially stocks, has gained huge popularity. However, it does not only replace human brokers. It allows small investors to buy and sell stocks within minutes. Indeed, thousands of individuals engage in what has become known as *daytrading:* buying and selling the same stock multiple times on the same day to profit from tiny fluctuations in the stock's price. The media have reported that some people resigned from their jobs to devote all their time to daytrading. With the opening of "off-hours" trading to the public, small investors could potentially devote 24 hours per day to such activity.

Experts argue that only a few people can earn a living from daytrading, let alone become rich from it. Daytrading is practically gambling, especially in light of the fact that many daytraders are not knowledgeable

investors. Some sociologists fear that daytrading, along with online gambling, will ruin families. Civil libertarians argue that daytrading should be subject to individual choice and that governments should not intervene in the matter. As with online gambling, there is no practical way to control daytrading.

V. VIOLATION OF INTELLECTUAL PROPERTY

Software has always been available in digital form, stored on magnetic disks and tapes and optical disks and tapes. Artistic works, such as books, pictures, and sounds, are also available, or can be made available, in digital form. The ease, small cost, and small chance of being caught make illegal copying of such work tempting.

A. Software

Two organizations have kept tabs on illegal copying of software, also called software piracy: the Software and Information Industry Association (SIIA) and Business Software Alliance (BSA). According to both organizations, the financial damage of illegal copying of software worldwide is $10–15 billion annually. Software piracy seems to be the most pervasive crime in which people of all walks of life engage. The problem seems to be worse in countries where material compensation for artistic work has not been the norm for many centuries, such as Asian and African societies. Under Western pressure, especially from the United States, the majority of the world's countries now have copyright laws similar to the American one, forbidding copying of software and other artistic work without permission from the owners.

B. Artistic Works

Artistic works such as books, music, photos, and paintings can be easily digitized. Once digitized, they can be copied to another computer storage medium within seconds or minutes, or transmitted to another computer thousands of miles away. Unlike other copies (such as from a copy machine or an analog music tape), digital copies of artistic work are as good as the original. Digital copying is also easy. It is extremely difficult to pinpoint someone who copied digital work unless the copies are openly offered for sale. This is the reason why leaders in the music industry are looking for means to protect their copyrights against infringement.

The Internet offers an effective means to publish music and distribute it. This gives new artists an opportunity to publish their work, but the same technology enables people with little regard for the law to post at Web sites copyrighted music that can then be downloaded by anyone. Attempts by the music industry to outlaw the sale of special devices that download music from the Web have failed. In a way, this is a repetition of the attempt to prohibit the sale of videocassette recorders, which failed. Leaders in the music industry have been looking for innovative means to protect the industry from mass violation of copyright laws. Some observers opine that this industry will have to change dramatically because of the new technologies.

VI. COMPUTER CRIME

The advent of computers in government and business organizations in the 1960s ushered in some types of crime that either did not exist before, or took a turn for the worse because of the new technology. The following sections provide a brief review of the major types of what is popularly referred to as computer crime.

A. Unauthorized Access

Unauthorized access to a computer system is any logging on to a system directly or through communications lines without permission of the owner or operator of the system. Popularly, the act is often called hacking or cracking. Some countries do not have laws that forbid such acts, but it is considered unethical. Where there are laws, some prohibit only uninvited access to security systems. A *security system* is one that specifically requires an access code (such as a password) or automatically identifies the person who tries to log on. Under these laws, access to a system that is not a security system is not an offense.

Unauthorized access alone may not cause any damage. The reason why unauthorized access is regarded as unethical or is illegal is simple: Hacking is often the prelude to other illegal or, at least, unethical acts, such as fraud and money theft, information theft, service theft, data alteration, and the launching of computer viruses.

When the culprits are caught, they often argue that the intrusion helped the organization realize that its security measures were flawed, and hence they provided a good service. This argument would not withstand any ethical doctrine. It is akin to burglars claim-

ing they helped the owners realize their home could be invaded.

B. Fraud and Money Theft

While we still use cash for transactions, much of the money that exchanges hands is actually electrical impulses and magnetic fields. Billions of dollars are transferred daily from one bank account to another by way of simple instructions to computers and transmission of electrical signals via computer networks. By either illegally obtaining access codes or by circumventing them, criminals can transfer millions of dollars from one bank account to another from a remote computer. Contrary to popular belief, the majority of online fraud and money theft is carried out by "insiders," that is employees of the victim organizations.

C. Information Theft

While several decades ago much of industrial espionage was done by searching paper folders and wastebaskets, modern theft of information for industrial espionage or other purposes is done by accessing computer systems either directly or via networks such as the Internet. The ease with which this can be done, and the difficulty involved in tracing the culprits has led some countries to change their laws; information is now considered property like any physical property. These laws were also modified to define theft of information as copying information without permission. Thus, culprits are considered thieves even if they do not remove anything physically, and even if they do not mean to deprive the lawful owner of the information or the use of information. The mere copying is criminal.

D. Data Alteration and Data Destruction

Data alteration and data destruction in corporate databases have been the nightmare of information systems executives. As long as a system is connected to a public network such as the Internet, it is potentially subject to uninvited intrusion. Once the system has been invaded, the hackers often find ways to either destroy or alter data. Cases have been reported of intrusion into hospital databases and alteration of medication doses that could result in killing patients. Thus, the risk may not be only material, but also a matter of life and death.

In recent years, the most common data alteration and destruction incidences involved corporate Web pages. The home pages of many companies and government agencies have been defaced either "for fun" or because the hackers resent the policies or activities of the organization. As the Internet is growing in size and use, this type of crime, although not outlawed by all countries, seems to be the greatest threat to electronic commerce and dissemination of information.

E. Computer Viruses

A computer virus is any rogue computer program that can spread by using computer networks or exchange of storage devices such as magnetic disks. The early computer viruses appeared in 1988. Some estimates put the number of new viruses at 3000 per year. A computer virus may cause damage in several ways: It may over-occupy the CPU, thereby slowing down productive work; it may over-occupy network servers, thereby not allowing reasonably fast data communications; or it may simply destroy data files or applications. The growing popularity of e-mail fostered the worldwide spread of destructive viruses such as the Melissa and I LOVE YOU viruses in the late 1990s and 2000.

Interestingly, some people claim that viruses should be considered a form of free speech and therefore should not be banned. Indeed, not all viruses are malicious, but the consternation that even the benign ones cause makes them unethical, if not criminal, in the eyes of the majority of people.

Another interesting point is the legality of computer viruses. The majority of countries (including most states in the United States) do not prohibit computer viruses per se. An attempt to modify the U.S. federal act against computer crimes (Computer Fraud and Abuse Act of 1986) failed not because of objection from any interested party, but because of a problematic situation. The launchers of viruses often are not aware of the launch, because they simply transfer contaminated files received innocently from other people. Thus, legislating against people who launch viruses may miss the real culprit. Legislating against creating a virus violates civil rights, because the mere creation of the virus does not cause any damage. Legislation against viruses must include creation as well as knowingly launching. Many legislatures seem to have had a serious challenge with the language of proposed bills and have thus decided against passing antivirus laws. In the meantime, countries that have computer crime laws prosecute under laws that forbid

unauthorized access, because often the person who launches a virus does so by accessing a computer system without permission. Most laws regard the mere reach of the virus into a system as unauthorized access.

F. Service Theft

Service theft occurs when an employee or another person uses the resources of a computer system without permission. This may include use of the computer's CPU, disk space, and applications. Unauthorized use of computing resources is often the result of a lack of clear organizational policies rather than deliberate theft of service. Companies that have clear policies which are communicated often to workers suffer less from such acts. Organizations vary in their official or unofficial policies regarding worker use of computers. Some allow use only for work. Others allow use of such resources outside paid time. Some further restrict use for educational purposes only.

G. Denial of Service

Online businesses depend on reliable and prompt availability of the information and services they provide via the Internet. By flooding a site with inquiries, perpetrators clog servers with illegitimate requests and deny access to many legitimate ones, since servers can respond to only a limited number of request at a time. Several denial-of-services attacks have forced, for example, online brokerages, auction sites, and other online businesses to shut down service for several hours at a time. No effective remedy to this type of attack has been found.

H. Internet Terrorism

Hackers often seize confidential information. Sometimes they use credit card information to make illegal purchases, but in some cases they have tried to extort money. They threaten the victim businesses with publication of the stolen information if the organization does not pay the requested ransom.

VII. CODES OF ETHICS AND PROFESSIONAL STANDARDS

Information systems (IS) professionals perform work that has as much impact as civil engineers, certified public accountants, lawyers, and often physicians, but do not have a mandatory code of ethics and professional standards like these and many other professionals have. In fact, the only codes that IS professionals honor, voluntarily, are those of some professional organizations. The major international organizations include Association for Computing Machinery (ACM), Association for Information Technology Professionals (AITP), Institute of Electrical and Electronics Engineers (IEEE) Computer Society, and the Institute for Certification of Computer Professionals (ICCP). Several countries have their own, national organizations, such as the British Computer Society (BCS), Canadian Information Processing Society (CIPS), and German Gesellschaft für Informatik. Some of these organizations accept members from any country.

A. Obligations

The breadth and depth of the codes of ethics of IS professional organizations vary from no codes at all to terse codes of several lines (such as the code of AITP) to very detailed codes (such as the codes of ACM and IEEE Computer Society). The organization of the principles of these codes varies, too. However, some core elements appear in almost all of the codes. Here, we list them by the constituency to which the IS professional has an ethical obligation. Many of the ethics principles apply to several constituencies.

1. To Society

Educate the public about information technology; protect the privacy and confidentiality of information; avoid misrepresentation of the member's qualifications; avoid misrepresentation of information technology; obey laws; and do not take credit for other people's achievements.

2. To Employers

Update own knowledge in the field of information technology; accept responsibility for own work; present work-related information to the employer in an objective manner; and respect confidentiality.

3. To Clients

Protect confidential information and privacy; give comprehensive opinion regarding information systems; do not diminish the effectiveness of a system through commission or omission.

4. To Colleagues

Some of the codes of ethics mention colleagues, the profession, and the professional organization itself as a party toward which the member has ethical obligations. The principles are not different from those of other professions and include respect for colleagues' work and not denigrating the profession or the organization.

B. Remaining Issues

It is important to recognize that unlike in other professions, none of the IS codes of ethics clearly prefers a certain party. Physicians always have the interest of their patients (clients) above those of other parties. The same principle applies to attorneys: They always defer to the interests of their clients as opposed to those of any other party. IS professionals do not have such clear guidance either during their education or in codes of ethics. A simple case illustrates the dilemma they may face.

Suppose an IS professional is called to fix the hard disk of a university professor. The computer belongs to the professor, but it is linked to the Internet via a server that is owned by the university. The IS professional finds that the professor downloaded pornographic images from the Web. Should the professional report this fact to his employer, namely, the university? Should the professional limit himself only to fixing the problem, disregarding the information he found on the disk? None of the codes of ethics of any professional organization gives clear guidance to the professional. The lack of ranking of constituencies in importance for professionals' consideration in cases of ethical dilemmas seems to be their greatest weakness. Perhaps the codes will grow to resemble those of more established professions as the IS profession matures.

The huge loss of financial and other resources due to failure of information systems often points to lack of ethics and professional standards of the people who build and maintain the systems. In some of the largest failures of IS development projects, there were clear violations of ethical principles as simple as telling the truth about the status of a project or disclosing a lack of skill or technical capability.

C. Certification

Few of the professional organizations and societies also serve as certification institutions; one exception is the ICCP. The need for certification is a controversial issue. Certification of professionals is often expected of people with expertise above the norm who are also in a position to make decisions that affect clients, the clients' stakeholders, and the public at large. Mandatory certification could guarantee a minimum level of skill so that employers better know what to expect from a new hire and clients know what to expect from IS professionals who offer to develop and maintain information systems for them. However, the IS profession (a controversial term in itself) has been characterized by lack of mandatory standards in general, let alone standardization of ethical principles and professional standards. Thus, despite the great impact of IS professionals on our physical and financial well-being, work, and educational systems, none is subject to mandatory certification.

VIII. CONCLUSIONS

The proliferation of information systems and their use throughout the world and the growth of public computer networks have raised issues about ethical conduct with information systems. Predominantly, the issues involve privacy, unauthorized access to and use of information systems, free speech, protection of digitized intellectual property, and lack of professional ethics. Legislation has addressed some of the issues. However, as information technology progresses and continues to invade many facets of our lives, we should expect more questions about the ethics of development and use of information systems to emerge.

SEE ALSO THE FOLLOWING ARTICLES

Computer Viruses • Copyright Laws • Crime, Use of Computers in • Digital Divide, The • Ethical Issues in Artificial Intelligence • Firewalls • Forensics • Law Firms • Privacy • Research • Security Issues and Measures • Software Piracy

BIBLIOGRAPHY

Johnson, D. G., Nissenbaum, H. F. (eds.). (1995). *Computers, ethics, and social values.* Englewood Cliffs, NJ: Prentice Hall.

Kling, R. (1996). *Computerization and controversy: Value conflicts and social choices,* 2nd ed. San Diego: Academic Press.

Oz, E. (1994). *Ethics for the information age.* Dubuque, IA: Wm. C. Brown.

Rothfeder, J. (1992). *Privacy for sale.* New York: Simon and Schuster.

Wecket, J., and Douglas, A. (1997). *Computer and information ethics.* Westport, CT: Greenwood Publishing Group.

Ethical Issues in Artificial Intelligence

Richard O. Mason
Southern Methodist University

GLOSSARY

agent, intelligent A small software program constructed on AI principles that performs electronic tasks for its master. Advanced agents learn by observing their users' activities. Also called a *bot,* short for robot. Used to dig through data in databases and perform tasks on the Internet. Technically, an agent is a bot sent out on a mission.

artificial intelligence (AI) The demonstration of intelligence by computers or machines; that is, making machines do things that are usually done by human minds.

degree of personhood The rating given any entity such as an AI program on a scale of "0" for an inanimate object to "1" for a person with full moral status, rights, respect, and liberty. AI programs or machines that exhibit intelligence may qualify for high ratings.

ethics The study and evaluation of the conduct of persons in light of moral principles. Historically, ethics sought human well-being. AI-based machines raise questions about this goal.

intelligence The display of a general mental ability, especially the ability to make flexible use of memory, reasoning, judgment, and information in learning and dealing with new situations and problems on the part of an entity. Displaying intelligence is one of the primary criteria for qualifying for personhood.

moral status The condition of an entity due to which moral agents have, or can have, moral obligations. People may not treat an entity with moral status in any way they please. They are obligated to consider its needs, interests, or well-being and to pay it respect.

personhood Any entity who displays a degree of self-consciousness and intelligence, such as many healthy adult human beings do, qualifies for personhood. The more nearly the entity displays it (that is, the higher the degree of personhood) the greater that entity's claim is for maximal moral status. AI machines that display superintelligence appear to qualify for personhood.

responsibility The characteristic of an entity that is able to make moral or rational decisions on its own and therefore is answerable for its behavior. When members of a society regard an entity as responsible they react to it with a characteristic set of feelings and attitudes such as gratitude, indignation, resentment, respect, forgiveness, or love. Generally speaking, for an entity to be morally responsible for something not only must it have done or caused some act but also it must be able to give an account of its actions including explaining its intentions.

superintelligence Intelligence displayed by an entity that at least in some aspects exceeds or matches that of human beings.

I. ARTIFICIAL INTELLIGENCE DESCRIBED

A. Artificial Intelligence Defined

Artificial intelligence (AI) is defined generically as the demonstration of intelligence by computers or machines; that is, making machines do things that are usually done by minds. According to the *Longman*

Dictionary of Psychology and Psychiatry, for an entity to display intelligence, it requires a "general mental ability, especially the ability to make flexible use of memory, reasoning, judgment, and information in learning and dealing with new situations and problems." Intelligence in this sense includes the ability to think, see, remember, learn, understand, and, in the long run, use common sense. This is a useful working definition of intelligence although some AI researchers differ on how it is to be applied to their work. Proponents of AI such as Kurzweil and Dyson argue that some of today's computers and software (circa 2000) indeed exhibit intelligence by this general definition. Kurzweil cites in support of his position the fact that on May 11, 1997, IBM's chess playing computer, "Deep Blue" (a RISC system/6000 scalable power parallel systems high-performance computer programmed in C language), beat then-reigning human chess champion Gary Kasparov in a six-game match. *Time* reported in its May 26, 1997, issue that Kasparov reflected on his experience: "The decisive game of the match was Game 2, which left a scar on my memory and prevented me from achieving my usual total concentration in the following games. In Deep Blue's Game 2 we saw something that went well beyond our wildest expectations of how well a computer would be able to foresee the long-term positional consequences of its decisions. The machine refused to move to a position that had a decisive short-term advantage, showing a very human sense of danger."

B. Information Processing or Symbolic Model and Artificial Neural Networks: Two Main Theories of AI

John McCarthy first used the term *artificial intelligence* at the field's founding conference held at Dartmouth College in 1956. The underlying assumption of the conference was that "[e]very aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it." This assumption led initially to the use of symbolic logic, patterned after the theories of George Boole, as a model of mind. Research based on this assumption resulted in an information processing or "cognitivist" theory of intelligence, one that emphasizes the functions of discriminating, acquiring, recording, analyzing, and remembering information and the role these functions play in decision making. Approaches based on this assumption are called *symbolic* or *symbol-processing AI.* These systems tend to be designed and programmed from the top down and proceed deductively rather than by means of training the computer and evolving concepts inductively. Symbolic AI is structured in levels. The top is the *knowledge level* at which basic rules and other knowledge are specified. This flows down to the *symbol level* at which the knowledge is represented by symbolic structures. Finally, at the base, is the *implementation level* at which the symbol-processing operations are actually carried out. Most AI systems of this type employ logical reasoning methods to deduce actions to be taken and conclusions to be drawn.

Expert systems (ESs) are one popular manifestation of symbolic AI. Arguably the first AI program was of this type. Dubbed the "Logic Theorist" it was written in the fall of 1955 by Allen Newell, Herbert Simon, and J. C. Shaw to prove theorems in geometry. This paradigm reigned until the early 1970s when, among others, Terry Winograd, a student of AI pioneer Seymour Papert, began to question the strong representational assumptions being made by Newell and Simon. Winograd predicted that future research would not uncover a unified set of principles of intelligence adequate to form the knowledge level. Rather, he envisioned a variety of mechanisms and phenomena. Some would be consistent with the symbolic AI paradigm; some, however, would be quite different. Roger Penrose in *The Emperor's New Mind* observed that once a noncomputational element was introduced into intelligence then computers could never do what humans could. During this time, neural net theory emerged and a debate commenced.

The second major type of AI is called artificial neural networks (ANNs). ANNs are modeled on the structure of the brain and nervous system, based in part on theories originally developed by John von Neumann, Warren McCulloch, Walter Pitts, and Donald Hebb. Sometimes also called *subsymbolic AI* systems, ANNs proceed from the bottom up, beginning with primitive *signals.* As they move up the hierarchy signals are converted into symbols. Proponents of this approach, such as MIT professor Rodney Brooks, believe that AI programs should be coupled closely with the physical world and made to evolve much as human intelligence has evolved over hundreds of thousands of years. The strategy they propose is called *subsumption architecture.* It begins inductively by simulating lower level animal functions and proceeds by adding competencies at higher levels. Whereas symbolic AI assumes that symbolic representation is necessary and fundamental to general intellegence, ANN downplays the importance of representation or, as in the case of Brooks, denies that it is needed at all.

ANNs emulate the processing capabilities of the human brain by simulating a distributed network consisting of tens of billions of cells that form arrays of millions of neurons (approximately 10^{11}) that are connected by a dendritic tree and excited by as many as 10^{14} synapses. (The number of synaptic connections in the human brain is likely greater than the number of atoms in the universe.) These *learning systems* are trained rather than programmed. The ANN approach is also called *connectionism*. Others have labeled it the *animat approach*. The symbolic and ANN approaches differ in their understanding of the nature of cognition and how it works and also in how one knows if cognition is functioning adequately. Two researchers, Rosenschein and Kaelbling, propose an intermediate approach in which desired behavior is programmed at a high level (symbolic) and a compiler is employed to create behavior evoking circuitry at a lower level (ANN). Although the underlying theories of AI and their implementing technologies are quite different and the competing theorists continue to debate their relative merits, their moral implications are similar.

Among the computer-based technologies that have been used as a means toward achieving AI's goals are the following:

- *Expert systems:* Computer programs that support workers in complex situations in which experts outperform nonexperts.
- *Neural networks:* As described above, these ANN computer programs identify objects or recognize patterns after having been trained on a group of examples.
- *Natural language processors:* Computer programs that translate or interpret language as it is spoken by normal people.
- *Visual processing:* Computer programs that emulate the human retina by means of algorithms. Some of these programs can recognize objects and shadows and identify small changes from one image to another.
- *Robots:* Computer-based programmable machines that have physical manipulators and sensors. Robots often use visual processing and draw on symbolic reasoning or neural networks for intelligence. They may perform tasks normally done by humans with greater speed, strength, consistency, and precision.
- *Fuzzy logic:* Reasoning based on imprecise or incomplete information in terms of a range of values rather than point estimates.
- *Case-based reasoning:* A computer program that searches for previous cases that are similar to the one in which a decision must be made.
- *Intelligence software agents:* Small computer programs constructed on AI principles that performs electronic tasks for their masters. Agents or *bots* may be launched into a system or network to work in the background while other processing is going on in the foreground.

C. AI's Goals and Criteria for Achieving Them

Two criteria are generally used to determine whether a machine or computer program has achieved the goals of AI research: (1) It displays general human-type intelligence with all of its subtleties and ramifications, or (2) it performs specific tasks in a manner indistinguishable from characteristic human efforts. To date AI technologies have been applied primarily toward projects, such as chess playing, directed at the second criterion—task performance. Although developments have lulled from time to time, since its inception AI technology has grown in productivity and use, especially since the advent of agents on the Internet. AI software and the computers that execute it have improved in performance—speed, scope, cost, etc. As a result, their use and extent of application have increased, especially as parts of other computer-based applications.

II. THE ETHICAL CHALLENGE OF AI

A. Is an AI Machine a "Person"?

Midway through the 20th century Norbert Wiener, a founder of cybernetics—the science of control and communication in animals and machines—anticipated the application of cybernetic and related theories to develop automata, robots, and other machines with intelligence. In his 1948 book Wiener observed: "It has long been clear to me that the modern ultra-rapid computing machine was in principle an ideal central nervous system to an apparatus for automatic control; and that its input and output need not be in the form of numbers or diagrams but might very well be, respectively, the readings of artificial sense organs, such as photoelectric cells or thermometers, and the performance of motors or solenoids. With the aid of strain gauges or similar agencies to read the performance of these motor organs and to report, to "feed back" to the central control system as an artificial

kinesthetic sense, we are already in a position to construct artificial machines of almost any degree of elaborateness of performance. He concludes, "this new development has unbounded possibilities for good and evil."

Its potential for achieving good or evil results places AI in the realm of ethics, which studies and evaluates the conduct of persons in the light of moral principles. The field of ethics deals primarily with human well-being. Thus, AI challenges ethics at its very core. Underlying the challenge are two key questions. Can a machine duplicate human intelligence? And, can it reach consciousness or sentience? George B. Dyson answers both positively. "In the game of life and evolution there are three players at the table: human beings, nature, and machines. I am firmly on the side of machines." Dyson, Kurzweil, and Hans Moravec, director of the Mobile Robotics Laboratory of Carnegie Mellon's Robotics Institute, are among those researchers who believe that AI will eventually result in machines that possess and display all the functional characteristics of human beings. Physically they will be silicon based rather than carbon based; but, they will be able to think, feel, have moods, be emotional, interact socially with others, draw on common sense, and have a "soul." Thus, AI-based systems, according to these researchers, will become the next stage in the evolution of life, emerge as our successors, and create a future society populated and governed by computers. (Section H below continues this theme.) Most AI researchers, however, are more cautious. A thread running through their reservations may be related to the concept of a human soul—that nonmaterial, animating, and vital principle in human beings that undergirds thought, action, and emotion. There is an illusive spiritual nature to human beings, they believe, that is immortal and separable from the body at death and cannot be duplicated by earthly apparatus. People with this point of view implicitly believe in a unique human "spirit"—a part of the self experienced separate from its earthly connections. Both of these metaphysical views about the possibility of replacing human beings by machines, however, raise significant questions about the moral status of AI-based machines.

Traditionally, granting moral status to an entity depends on it showing some form of rationality such as that displayed by humans. Indeed, Aristotle defined "man" as a rational animal. It can be argued that if an AI program truly exhibits "intelligence" or rationality, then it deserves the moral status of "personhood." This is true in part because such a machine's actions would be able to either affect (1) itself as subject or (2) others in its role as an agent, agents being the means by which something is done or caused. In this case, an AI machine should be considered a member of the moral community (whether or not it is considered to have a "soul"), just as most human beings are. To have moral status means that an entity assumes rights and duties in a society. Consequently, other moral agents in the society can have moral obligations toward it, such as respecting its individuality and autonomy. Mary Anne Warren in *Moral Status* explains: "If an entity has moral status, then we may not treat it in just any way we please; we are morally obliged to give weight in our deliberations to its needs, interests, or well-being. Furthermore, we are morally obliged to do this not merely because protecting it may benefit ourselves or other persons, but because its needs have moral importance in their own right."

Being human, however, is not a necessary or sufficient condition for an entity to be classified a "person" in this sense. Historically, the status of personhood has been accorded primarily to healthy adult, mostly male, human beings. The ancient Greeks did not give full moral status to women, slaves, or those whom they considered barbarians. History recounts the struggle on the part of these parties to achieve personhood. Newly born children, the mentally ill, and the physically impaired have also been denied full personhood. More recently, arguments have been made to accord personhood status to fetuses, animals, and even parts of the ecosystem. Rocks and stones, for the most part, have been excluded. Nevertheless, trees because they are living things have their advocates. Corporations are treated as "persons" for legal purposes. Just who or what belongs to the circle of "persons" has been a question of ongoing social debate. It remains today a negotiable and empirical question. This means that intelligent machines may, if they meet certain criteria, qualify for personhood and become members of a moral community with at least some of the rights and privileges that come with this designation.

B. Degrees of Personhood

Thus, the question of granting personhood to an AI machine or robot depends on where the line is drawn between persons and inanimate objects. It is useful to think in terms of a continuum running from "persons" on one end and "objects" on the other. An inanimate object may be accorded a "0," whereas an ideal full-fledged human citizen may be accorded a "1." A rating of 1 signifies perfect personhood, something deserving of receiving total respect. Granting an entity total respect means giving it complete autonomous

determination over its actions and not imposing limitations on its liberty; that is, treating it as a full-fledged Kantian end and not as a means. As alluded to above, many entities in our society are accorded partial personhood. In effect they have a rating greater that 0 but less than 1. Nevertheless, the higher the rating the more respect an entity deserves from other members of its society. For example, according to *Roe v. Wade,* during the first trimester of gestation a fetus may receive less respect than it does during the next two trimesters. A newly born child receives less respect than a young adult, etc. In each of these successive stages, the fetus/human gets a higher rating on the object/person or degrees of personhood scale. A fundamental ethical question posed by AI is what rating should be given to a particular AI computer program, machine, or robot. On this various advocates disagree.

How can we answer this empirical question of who qualifies for personhood and what degree of personhood rating to accord it? The overarching criterion is displaying some from of cognitive capacity—being conscious, having perceptions, feeling sensations. For John Locke a person is "a thinking intelligent being, that has reason and reflection, and can consider itself as itself, the same thinking thing, in different times and places; which it does only by the consciousness which is inseparable from thinking, and, . . . , essential to it; it being impossible for anyone to perceive without *perceiving* that he does perceive." Thus, entities with significant cerebral functioning are considered persons. They receive ratings at or near 1. People who are comatose, however, although they are due respect do not receive as high a rating. Other capacities that have been proposed as essential for personhood include the following: being conscious of one's own existence and capable of self-direction, being able to enter meaningful relationships, being capable of minimal independent existence, and being in possession of a minimal IQ (say, 20–40). The more these criteria are satisfied the higher the personhood rating.

C. Turing's Test

Yet these criteria are difficult to apply directly to intelligent machines. Consequently, some AI researchers propose another tactic. In 1950, at the outset of AI research, one of its founders, Alan Turing, proposed a test for determining whether a machine could think and exhibit cerebral functioning. He called it the "Imitation Game." In this game an interrogator uses a computer terminal to ask questions of an entity—a person, computer, or machine—located in another room. If, based on the entity's typed responses, the interrogator cannot determine whether the responses came from a human being or a machine then the machine is considered to have intelligence. As researchers have struggled using many different approachs to determine whether or not a particular machine had intelligence they have continually, if sometimes reluctantly, returned to "Turing's test." Turing also predicted at mid-century that "in about fifty years' time, it will be possible to programme computers . . . to make them play the imitation game so well that an average interrogator will not have more than a seventy percent chance of making the right identification after five minutes questioning." He concluded, "We may hope that machines will eventually compete with men in all purely intellectual fields."

Turing's powerful but immensely subtle technique has served as the arbiter of intelligence in entities since it was proposed. Deep Blue's performance and that of other contemporary computer programs (circa 2000) appear to qualify as intelligent by Turing's test. Thus, they are potential candidates for personhood and moral status. Related to this test is a subtle and controversial position: If, based on Turing's test or some other criteria, an entity displays degrees of mind or of cerebral functioning, then it should be accorded an appropriate degree of moral status, that is, a rating well above 0. This means that machines could also be accorded rights and duties—proper respect. Perhaps, in a democratic society they ultimately might be able to vote, a possibility that frightens AI critic Joseph Weisenbaum.

D. Weizenbaum's Critique

In 1976 Joseph Weizenbaum published *Computer Power and Human Reason* in which he criticized the AI research community for proceeding so vigorously and for not paying sufficient attention to ethical and human issues. About a decade earlier Weizenbaum had demonstrated a successful AI program called ELIZA. Based on the "Rogerian" or nondirective school of psychotherapy, ELIZA simulated (or, in the author's words, "caricatured") dialogue between a psychoanalyst (played by the computer) and a patient (human). The program didn't "think." Rather, it parroted back phrases depending on what the patient inputted. For example: *Patient:* Men are all alike. *Computer:* In what way? *Patient:* They're always bugging us about something or other. *Computer:* Can you think of a specific example? Etc.

What alarmed Weizenbaum, an unwitting charlatan, was that many people actually believed that they were talking to an intelligent machine. They took the illusion seriously. It deluded them he said. Society makes too much of the computer, he argued, and this encourages a dehumanizing worldview in which society strives to conform to the image of the computer. In this world ethical and religious principles are replaced by mechanistic or scientific principles. The measure of man becomes the machine. Many AI researchers, Weizenbaum claims, are really "hackers" and are leading society, perhaps unknowingly, into a dead-end mechanistic world. Offering this argument in 1976 Weigenbaum was the first person to raise the question of the morality of AI explicitly and to try to assign responsibility.

E. AI and Moral Responsibility

While the possibility of a machine being granted moral status is the most compelling ethical issue raised by AI, there are others, determined largely by the uses to which AI programs are actually put. These ethical considerations have evolved as AI research and development has progressed. AI programs form relationships with other entities. They are used, for example, to advise human users, make decisions, and in the case of intelligent software agents to chat with people, search for information, look for news, find jobs, and shop for goods and locate the best prices. Their role in these relationships engenders moral responsibility. According to philosopher P. F. Strawson when members of a society regard an entity as responsible they react to it with a characteristic set of feelings and attitudes such as gratitude, indignation, resentment, respect, forgiveness, and love.

Marina Oshana takes an accountability view of responsibility. She believes that for an entity to be morally responsible for something not only must it have done or caused some act but also it must be able to give an account of its actions including explaining its intentions. Theologians Richard Niebuhr and Charles Curran argue that in its broadest conception responsibility involves at least four things:

1. The ability to initiate new actions or respond to actions imposed on an entity by others
2. The ability to give the actions meaning
3. The willingness to be held accountable for any reaction to its reactions
4. The ability to join with others in a continuing community of agents.

Some AI machines come close to satisfying these four criteria and, moreover, with each new development the field moves closer to meeting them. Thus, the level of moral responsibility that is attributed to an AI entity depends on the relationships it establishes within the social network in which it is placed.

III. THE SOCIAL ROLE OF AI RESEARCH AND DEVELOPMENT

A. Five General Categories

Using this approach the results and use of AI research can be classified into five general categories of use, each with implications for responsibility and moral status: research object, tool or instrument, slave, partner or assistant, and superintelligence or autonomous being. Generally speaking, as an AI program moves from research object to superintelligence it qualifies for a higher degree of personhood and should be accorded more respect.

B. Responsible Parties for an AI System

In assessing responsibility in these five roles, several human contributors to a particular AI program or those who use it should be considered:

1. *Computer manufacturers* who produced the machines on which the AI program runs
2. *Systems designers* who conceived of and designed the systems
3. *Programmers* who write the instructions that the machines execute
4. *Knowledge engineers* who elicit knowledge from human experts and introduce it into the machines
5. *Data collectors* who introduce raw data into the machines' databases
6. *Inference engine designers* who developed the logic technologies that apply the knowledge, draw inferences from data in the databases, and decide what steps to take next
7. *Human users* who employ the AI system to serve some purposes such as decision making.

Any of these parties may bear some of the responsibility for the actions taken as a result of an AI system. Not all of these parties, however, need be present in every AI system. Expert systems, for example, tend to draw on all seven. Say, for example, that a medical ES

misdiagnoses a patient and as a result the patient dies. The harm may be traced back to any of several parties: a flaw in the manufacturer's machine, a misspecification or oversight by the systems designer, a bug in the program or sloppy coding by the programmer, a misinterpretation or lack of understanding on the part of a knowledge engineer, an error in data collection or coding, a logic flaw in the inference engine, a failure to use the system properly on the part of the user, or some combination of these. Similar responsibility chains pertain to other AI techniques although neural net systems may not have knowledge engineers. Case-based systems employ case analysts rather than knowledge engineers. Nevertheless, when either blame or praise is to be assigned, ethicists will typically attempt to trace back through the chain of events to determine the relative contribution of each of these parties. The types of issues for which blame or praise is to be accorded depend in large part on which of the five categories of relationships an AI system is placed.

C. Moral Implications of AI as a Research Object

According to two AI pioneers, Feigenbaum and Feldman, the goal of AI research and development is "to construct computer programs which exhibit behavior that we call 'intelligent behavior' when we observe it in human beings." The research they reported in their seminal book included explorations into computer programs that played games such as chess and checkers, proved mathematical theorems, balanced production assembly lines, solved calculus problems, answered questions about baseball, understood natural language, recognized patterns, and simulated human thought. The simulations delved into verbal learning, concept formation, decision making under uncertainty, social behavior, and problem solving. Most of these efforts were directed toward learning how the human mind functioned and replicating portions of its operations in a computer program. Basic inquiry of this type continues to this day. Thus, from its inception AI has had both scientific and engineering purposes.

Yet even in the early exploratory studies, questions arose as to whether or not this line of scientific inquiry should be pursued. Academic freedom supports the argument that it is morally acceptable to pursue all questions about nature. Indeed, many AI research scientists believe that they have a moral obligation to seek the truth wherever it may lie. Nevertheless, some people are disturbed by any effort to inquire into something as mysterious as the human mind, the soul, the wellspring of spirituality. In effect, they think it is immoral to muck around in something so close to life. To cite one reason: Thinking of this sort starts society down a slippery slope toward replacing the human mind as its source of control. Ultimately, therefore, human liberty is at stake. Moreover, some observers, such as David Noble, feel that simulations of the mind are demeaning and ultimately dangerous to humanity. In *The Religion of Technology*, Noble acknowledges the religious roots of modern technology but contends that society must now strive to separate itself from its clutches. It is menacing, he believes, to discover and exercise the Godlike knowledge and powers derived from research in AI, genetic engineering and the Human Genome project, manned space exploration, and nuclear weapons. Like Sun Microsystems chief scientist and cofounder, Bill Joy, Noble argues that it is better to be cautious and perhaps regulate technological advance.

One criticism is that programs oversimplify the complex operations of the mind by assuming it is analogous to a computer with bits and switches. Thus, the research assaults human sensibilities and even brings into question our basic homocentric view of the world. For example, in a study completed in 1963, Geoffrey Clarkson built a simulation model that produced highly similar portfolio decisions to those rendered by an experienced, sophisticated senior trust officer. While researchers applauded the results, in other quarters this finding created a threatening specter that professional activities—ones that appear to require exceptional intelligence—could be duplicated by a computer model. In the wake of these and other early AI research results, a physician, Lewis Thomas, cautioned against anyone pursuing AI research at all. In an article published in 1980 he observed that: "[t]he most profoundly depressing of all ideas about the future of the human species is the concept of artificial intelligence. That these devices will take over and run the place for human betterment, strikes me as wrong in a deep sense, maybe even evil. . . . Machines like this would be connected to each other in a network all around the earth, doing all the thinking, maybe even worrying nervously. But being right all the time."

AI researchers retort that just as the geocentric view of the world was challenged effectively by Galileo and social values changed accordingly, society must challenge its contemporary assumptions about the central role of the mind in human affairs. This, they argue fervently, is a new frontier. Progress must go on.

In 1984 linguist Roger Schank rebutted Thomas by arguing that: "In trying to model our thought processes on computers, we continually learn more about what it means to be human. Far from dehumanizing us, AI research has compelled us to appreciate our human qualities and abilities."

The central role that inquiry plays in modern society adds to the importance of these issues concerning the ethics of AI research. Any intellectual discovery can be both elating and distressing at the same time. Accordingly, scientific results can make people feel either helped or harmed or, likely, both at the same time. One reason for this is that the metaphysical belief systems that underlie research paradigms have significant moral consequences. They speak to our place in the universe. Many of the philosophical issues raised by AI research are prompted by worldview assumptions—often hidden—made about two central entities: the nature of the human brain and the nature of computer and communication machines.

Is the brain finite or infinite? Arguments for a finite, closed system brain follow in a philosophical tradition beginning at least with Socrates and Plato and running to Hobbes, Descartes, and George Boole. This view assumes that once an adequate level of understanding is reached, all human cognitive processes can be reduced to a set of explicitly stated definitions and rules. Thinking can then be simulated by formally manipulating symbols. AI researchers Alan Newell, Herbert Simon, and J. Clifford Shaw are among those who have generally worked under this assumption.

Arguments for an infinite, open system brain follow in a philosophical tradition including Pascal, Heidegger, Merleau-Ponte, and Wittgenstein. This is a humanist, phenomenological perspective that holds that human beings ultimately do not have access to either the basic elements or even the first principles of their brains. Therefore, a full understanding of people's everyday thinking, perceptions, and actions is beyond human grasp. The functioning brain reaches beyond the cerebellum and is intrinsically interrelated with the body and the nervous system. In 1972 the philosopher Herbert Dreyfus adopted this position as the basis of a rather scathing critique of AI research. It is impossible, Dreyfus argues, for AI to ever reach its research goals and, therefore, it borders on the unethical—at the very least, it is futile—for AI researchers to devote society's resources to this ultimately sterile line of inquiry.

The finite machine assumption is based on common practice. Since their inception, and until recently, the reigning concept of computers has been the von Neumann stored program machine model. In this model instructions and data are stored together, intermixed in a single, uniform storage medium. Instructions and data are "fetched" and then "executed" in a focal register in a computational cycle guided by a program counter that also permits branching—that is, altering the sequence by which instructions are executed. Instructions as well as data can be modified. Because most early AI programs were run on machines based on this model, critics claim that the researchers' concept of brain was unduly restricted to a finite conception. This point of view is supported by a broad theoretical reliance on George Boole's symbolic logic and Claude Shannon's theory of relay and switching circuits that applies Boole's logic to computer circuit design. This is a powerful assumption, however, and can be used to explain many rational processes.

The infinite machine assumption is based on a theory developed by Alan Turing. Some people point out that the brain with its trillions of connections is far too complex to be duplicated by a von Neumann machine. The brain may be finite—since it is composed of a finite number of neurons, axons, and synapses—but it is immense beyond imagination in the number of different states it can assume. Consequently, as Turing believed, it takes a machine with infinite processing capacity to duplicate adequately a human brain. The theoretical machine he conceived is called the Turing machine. It consists of a paper tape of indefinite length and a machine through which the tape is fed. This machine can do four things: it can move the tape one space, it can place a mark (0 or 1) on a space, it can erase a mark, and it can halt. Turing proved mathematically that a machine with just these four functions, *if given a long enough tape and adequate amount of time,* could solve any analytical problem. The set theorist Georg Cantor's theory of denumerable or countable infinite sets demonstrates the possibility that such an infinite tape could be related to a finite series of states and, thus, in principle, makes it theoretically possible to relate an infinite machine to a finite brain. Different interpretations of the phrase "long enough tape and adequate amount of time" are the main point of contention between those who favor the finite machine theory and those who maintain that an infinite machine is possible.

The underlying worldview of AI research that is most often criticized is the one that assumes a finite brain is being simulated on a finite machine. The Dreyfus brothers are among those leading the attack. In their 1986 book they report that Newell and Simon expressed a belief in this metaphysic in 1958. "[I]n

the visible future," Newell and Simon argued, "the range of problems [that AI computers] can handle will be coextensive with the range to which the human mind has been applied." Thus, Newell and Simon proposed the symbolic AI-based information processing model of the brain and used means–ends reductionism as a tool of inquiry. Acceptance of this worldview led AI pioneer Marvin Minsky to describe the human brain as a "meat machine." Most of Weisenbaum's fury was targeted at those who actively purse this metaphysic. Observers like the Dreyfus's and Weisenbaum believe that this strong dependence on closed system rationality may have even had deleterious effects on the field of psychology and corrupted our general understanding of the brain and mind. Far from being a scientific tool to learn more about human intelligence and other cognitive processes, these critics believe that the research may be dysfunctional. University of California at Berkeley philosopher John R. Searle calls the finite/finite worldview "strong AI" since a conclusion of its assumptions is that the mind is to the brain as the brain is to the computer hardware. Overall this metaphysic suffers from several disturbing characteristics: It oversimplifies the nature of being human, it challenges the homocentric worldview, and it encourages people in power to replace humans with machines. If this metaphysic is ontologically true, however, then machines eventually can have the potential to be accorded a high degree of personhood.

The worldview that emerged from the Dartmouth conference and that guides most of the work to this day is the finite brain, infinite machine metaphysic. The presumption was (and continues) that as bigger, better, faster computers with larger memories and more sophisticated programming capacity become available, machines (especially globally networked machine systems) will reach close enough to infinity (i.e., have a long enough "tape" and enough time) to simulate or duplicate the finite brain. These machines will reach Turing's ideal. In this belief system, as Noble points out, human beings are viewed primarily as machines. From an ethical standpoint this suggests that humans can be treated the same way machines are treated. Under this metaphysic AI-programmed machines as they develop and mature toward infinity have the potential to be accorded a high degree of personhood.

The possibility that the brain may be an infinite, open system, however, raises very deep questions about the goals and ultimately the efficacy of AI research. At this writing (circa 2000) machines are effectively finite. Taken together this supports an infinite brain, finite machine metaphysic much like that argued for by the Dreyfus brothers and other AI critics. Searle, for example, argues that although AI systems may be able to simulate substantial regions of human thought they will never be able to think meaningfully or to understand. Thus, he rejects the "strong AI" hypothesis in favor of a "weak AI" hypothesis. In his famous "Chinese Room" paper, written in 1980, Searle describes a thought experiment in which he himself mimics a computer and argues that he could pass a Turing test and be able to respond accurately to questions posed in Chinese although he is totally ignorant of the Chinese language. AI is thus "weak" in the sense that it will never be able to fully grasp and understand the way an infinite brain does even if it passes Turing's test. Although some research shows that most if not all human thinking takes place within the confines of the cerebral hemisphere (by exciting the more than 10 billion neurons it contains), interactions with the senses and the nervous system also play in important part. Given existing difficulties of defining what constitutes a mental attribute and of resolving the mind–body problem, the assumption of infinite brain seems quite plausible. Moreover, at this time, little if any progress has been made in imbuing AI programs with other crucial cognitive attributes such as self-awareness, consciousness, and feelings. While any basic computer given enough time can simulate a Turing machine, the current state of our knowledge about hardware and software practically favors arguments for a finite machine. Under this infinite brain, finite machine metaphysic, consequently, AI's goal of simulating every aspect of human learning and intelligence is ultimately unattainable although it may be approachable in the limit. Consequently, machines do not have the potential for duplicating all aspects of human cognition and, therefore, may not be accorded as high a degree of personhood as is possible under the previous two worldviews.

One final possible metaphysic remains: infinite brain, infinite machine. Under this metaphysic AI's goal may, in abstract theory, eventually be achievable since at some stage in their development machines may be able to adequately, but not completely, duplicate an infinite brain. But the process is infinite and never comes to closure. Under this metaphysic machines have the potential to be accorded a high degree of personhood depending on how well machines are developed but this will be a long time coming.

Responsibility in the case of AI as a research object lies mostly with the researcher—that is, the role of the systems designer. The nature of the benefit or harm (if any materializes) is primarily psychological and

ideological and, for the most part, rather contained. An additional set of ethical issues emerges, however, when engineering is involved and AI machines are taken out of the laboratory and actually used. AI machines assume moral agency when they are deployed to initiate actions in the real world. In the remaining categories of relationships the product of the research is implemented in different ways. Accordingly, additional, and quite practical, ethical considerations must be taken into account.

D. Moral Implications of Using AI as a Tool or Instrument

The history of technology is the story of humanity's efforts to control its environment for its own benefit by creating tools. Tools are artifacts that are constructed to aid a human being to solve a problem. Thus, tools amplify human behavior, but they are morally malleable. Inherently, they are neither good nor evil. Their social value depends on how they are used by those who employ them. Put to use as a tool, technology both shapes its users as subject and affects other parties in its role as an agent. That is, tools serve as a means to an end.

Computers have often been treated as tools. In a 1984 article psychologist Donald Norman, for example, argues that "computers are tools, and should be treated as such; they are neither monsters or savants, simply tools, in the same category as the printing press, the automobile, and the telephone." Historically, computers were developed only to serve people. For example, Charles Babbage (circa 1821) intended his Analytical Engine to be used to calculate accurate tables of mathematical results—such as logarithms—to help people who needed them. The Analytical Engine was to be a tool. The trend continues. Circa 2000, in excess of 100 million people use computers as aids. These workers and citizens from all walks of life use this technology for all sorts of personal and professional tasks. Many of these computer programs, such as Microsoft's Front Page, have routines embedded in them that are derived from the results of AI research. In addition, most word processors anticipate problems and provide help functions and most spreadsheet programs incorporate some intelligence to structure calculations.

When AI is used as a tool the moral onus rests on the user—the user dominates the tool. A user who uses an AI-aided spreadsheet for nefarious purposes, for example, bears the burden of responsibility. In some cases, however, the computer manufacturer and the software programmer are clearly responsible for the program performing as advertised. Under human control, however, some programs go beyond their use as mere tools. It is possible to incorporate decision-making functions and, with robots, decision-taking functions within these computer-based systems. Used in this way AI-based systems can assume the social roles of slaves or of partners.

E. Moral Implications of Using AI as a Slave

A slave is given a task by its owner and then uses its physical and intellectual capabilities to complete it. Command and control resides with the owner. Wiener foresaw that if AI was used in this way social disruption would ensue. "It gives the human race a new and most effective collection of mechanical slaves to perform its labor," he said in his 1948 work. "Such mechanical labor has most of the economic properties of slave labor, although, unlike slave labor, it does not involve the direct demoralizing effects of human cruelty. However, any labor that accepts the conditions of competition with the slave labor accepts the conditions of slave labor, and is essentially slave labor."

An AI slave is a machine that is either controlled by another machine or is abjectly subservient to a specific person, organization, or identifiable influence. It is considered to be the property of a person or organization. In the ancient meaning of the concept a slave was an economic commodity owned by a household or family. Today, AI machines are more likely to be slaves owned by a corporation or some other organization. (To date most have been developed for and owned by the military.) The actual use of AI as a slave, as a robot or automaton, may have either ethically positive or negative results. Every society has a number of dull, menial, routine, and repetitive jobs to be done. In his movie *Modern Times* Charlie Chaplin dramatized some of the evils of this dull, routine work. It is dehumanizing. The hope of AI is that many of these dull, or dangerous, or exceedingly arduous jobs of society will in time be taken over by intelligent machines. But, as Wiener observes, these AI-driven machines will inevitability compete with human laborers in the labor market. Just as a group of "pick-and-shovel" laborers can not compete economically with a steam shovel on an excavation job, workers who perform simple, routine mental activities will not be able to compete with computers. Wiener concludes "the average human being of mediocre attainment or less has nothing to sell that is worth anyone's money to buy."

In a manner not unlike the way that English weavers were replaced by the Jacquard loom at the outset of the industrial revolution, AI systems will be used as slaves and applied to mental tasks that have clear, definable goals and draw on a well-bounded knowledge domain. One crucial ethical issue revolves around how those people who are replaced will be treated. In a just society they will be given an appropriate place in the society or be reciprocated for their losses. If these displaced persons do not feel properly compensated, however, they will revolt, as the Luddites did around 1811. AI-programmed machines may have this capacity in the future. If an AI machine in its role as a slave displays enough cognitive capacity to be accorded a high degree of personhood, an additional set of issues are raised. These are questions of tyranny and despotism. An AI slave, like a human slave then, would be beholden under political subjection. Releasing it from this subjection opens up the possibility of citizenship. In the process, AI machines could become "citizens" and assume some of the rights of citizenship.

A computer slave's efforts fully substitute for those of a human being. If an AI slave acts under the direction of its user or owner—that is, it is effectively coerced—then the user bears the burden of responsibility, although the chain of responsibility may flow from there back to include designers, programmers, manufacturers, and other parties.

F. Intelligent Software Agents: A Special Case of the Slave Role

Intelligent software agents, in general, are computer programs that can be launched into a computer system or into a network to perform processes in the background, often while the computer is performing other work in the foreground. These "agents" perform electronic tasks for their masters and may also learn by observing their master's activities. Agents have a mission and act autonomously to complete one or more tasks. Many agents are, in effect, also "secret" agents because the user is often unaware that they are at work. The World Wide Web and other online or interconnected systems have spawned many agents of this type. Some are local, operating just on the computer in which they are lodged; others are mobile, reaching or roaming out to a variety of other computers. Most rely on some form of AI programming to carry out their tasks. Agents are called by various names depending on what functions they perform. In the new vocabulary of AI on the Internet the term *bot* (short for *robot*) is often used instead of agent. Many different types of bots have evolved.

Chatterbots are used to engage in conversation on the Internet. Shoppingbots are agents that shop and locate best prices for the users. Docbots and Jobots are used to locate physicians and employment, respectively. Musicbots seek out a piece of music or a CD or an audio file. "Spiders" or spiderbots explore the structure of the Web and act on the pages they find there performing such activities as counting, identifying, or indexing them. Search engines rely on spiders. Knowbots or *digital butlers*—to use Nicholas Negroponte's term—perform a variety of tasks assigned to them by their user. *Cookies* are small strings of symbols that communicate between a Web browser and a connected server. They are resident on the user's hard drive. Upon request by a connected server, a cookie collects information about what has been stored or retrieved by a user's browser. Many customer-profiling systems rely on cookies. *Viruses* are small programs written to intentionally cause damage or disruption to a computer and are transmitted by infected disks or online connections. All of these intelligent agents or bots and many others are based on principles originally discover by AI researchers.

These intelligent agents are also moral agents because their activities may either harm or help people or both simultaneously. Just as during the 18th and 19th century European merchants sent parties called agents to America and other distant lands to perform work for them, users send software agents out into cyberspace to do jobs for them. Agents are given instructions by their "principles" or owners and then act on them autonomously. In particular, once sent on their mission intelligent agents operate, for the most part, without the direct intervention of humans or others. They undertake many activities such as interacting with other agents including humans and perceiving the environment in which they are operating so they can respond accordingly. They are frequently proactive, are diligent in the sense that they continue to work at all times, and yet they are capable of performing many complex tasks. Consequently the agents' principles qua owners bear much of the responsibility for an agent's actions as long as the agent is following instructions.

For the most part, the work of intelligent agents is considered to be beneficial. For example, intelligent software agents promise to help deal with information overload, especially on the Internet. Among the mundane or tedious tasks that agents can accomplish are the following: manage e-mail, mine databases,

organize computer interfaces, scan articles, personalize data acquisition such as searching through news sources and reports to provide information users want, arrange and schedule meetings, manage networks, and help map the some 1 billion pages of information currently on the Internet. Yet, in the process of achieving these benefits and the very secrecy in which they generally operate intelligent agents raise important moral questions.

Search and knowledge bots, for example, can be used for data mining purposes—extracting buried or previously unknown pieces of information from large databases. Applied to personal information they can be used to profile individuals and as a consequence place them into some consumer category or risk group about which they are unaware and, more importantly, that may have social implications associated with it that harms or stigmatizes them. For example, an individual with an excellent credit record could be denied a loan because a bot assigned her to a particular risk group as a result of its data mining work. Thus, when applied to personal information bots can be used to invade or compromise people's privacy. Intellectual property rights may also be threaten by bots. As pending court cases against the online peer-to-peer music exchange company Napster (circa September 2000) reveal, music bots may be perceived as thieves by music copyright holders.

Shopping bots create other moral issues. For example, an MIT Media Lab professor, Pattie Maes, developed a bot called "Firefly" that has made her famous and is making her rich. (Microsoft purchased it.) Firefly matchs people's interests in objects like films, music, and the like. Similar applications of agent technology include Jango (now owned by Excite) and Anderson Consulting's "RoboShopper." Experiments with these shopping bots, however, have suggested some crucial ethical questions: Who is responsible for an agent's purchases? Is an agent's contract valid? What about other things an agent might do in pursuing its mission? Indeed, can an agent be trusted? How does a human owner or principle keep an agent abreast of her changing interests and offline transactions? Who is responsible for this communication? Significantly, the possibility of agent failure, misuse, or co-option is ever present. Agents can wrench control from the user and pass it on to their owner and, thereby, they can be the source of a great deal of harm. In fact, it may be necessary to employ other AI techniques to counter an agent's untoward activities. "Reputation servers" that can verify the credibility of an agent and help build trust present one possibility. At a broader systemic level preliminary experiments at IBM and other research institutes indicate that widespread use of bots in e-commerce can materially affect the functioning of economic markets. This of course raises questions about justice and fairness in markets and how best ethically to manage the transition from a physical marketplace to agent-oriented markets in cyberspace.

G. Moral Implications of Using AI as a Partner, Aid, or Associate

In a human–machine partnership each entity is united or associated with the other in an activity of common interest. The computer becomes an assistant or colleague. Consequently, the output of such a partnership is a joint output. Some AI researchers view these partnerships as the ideal relationship. For example, enthused by the possibilities of interactive computing, in 1960 an MIT professor, J. C. R. Licklider, proposed that the best arrangement for both humans and AI-driven machines was neither a master/slave relationship nor a coexistence among competitors but rather a partnership. He sought the most effective combination of the processing powers of computers and the cortical powers of human operators. An association between two or more different organisms of different species (such as humans and computers) that works to the benefit of each member is called a *symbiosis*. "Man–computer symbiosis is a subclass of man–machine systems." Licklider observed in 1960, "There are many man-machine systems. At present, however, there are no man-computer symbioses. . . . The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human being has ever thought and process data in a way not approached by information-handling machines we know today." Researchers working in this tradition stress the use of machines to supplement human capabilities. The modern fighter plane is a good example. Traveling at speeds beyond the capability of a pilot to control the plane's flight in all its dimensions, the on-board computer performs functions and provides calculations to keep it on course. As a partner, the computer brings the attributes of speed, reliability, accuracy, and, of course, freedom from boredom to the partnership and, thereby, complements the pilot's superior capacity for judgment and common sense. In recognition of the value of the partner role Scandinavia Airlines changed the label for their flight maintenance system from "expert system" to "a system for experts."

Expert systems are among the types of AI applications in which a program partners with a human being. The term *expert system* describes a computer program that supports the work of professionals in situations in which recognized experts do better than novices and nonexperts. In general, ESs have been employed in situations having one or more of the following characteristics: the choice of an effective response depends on a common body of knowledge or expertise, the expertise is identifiable and extractable, the expertise is relatively scarce, and there are numerous recurrences of situations calling for the application of the expertise. Although the situation may be complex its problem domain can be effectively bounded or circumscribed. For example, one ES called PUFF has been used at the Pacific Medical Center in San Francisco to diagnose lung disease disorders. Dr. Robert Fallat used PUFF to do most of the routine data analysis involved. The program arrives at the same diagnosis about 75% of the time. So Dr. Fallat uses PUFF as an aid, helping him, saving him time but not replacing him. In another example, American Express uses an ES called the Authorizer's Assistant to analyze difficult credit granting cases. Drawing on some 800 rules elicited from practicing credit experts, the program typically renders a credit decision in about 90 seconds, 4 or 5 of which are required for the Authorizer to search the firm's files, analyze the data, apply the rules, and arrive at a recommendation. The remainder is devoted to the human credit agent asking questions and making a final decision. Used as a partner in this way the Authorizer has resulted in increased agent productivity, fewer denials, and improved predictability of credit and fraud losses. Some other historically important ESs include MYCIN, which diagnoses and recommends treatment for infectious blood diseases; DENDRAL, a program that determines the molecular structure of chemical compounds; PROSPECTOR, which helps geologists locate ore deposits; DELTA, which GE uses to diagnose trouble and breakdowns in diesel-electric locomotives and suggest repair approachs; XCON, which is used by Digital Equipment Corporation to configure computers; and ExperTAX, Coopers & Lybrand's system to help junior auditors.

ESs, for the most part, are beneficial. They have been effectively applied to well-defined problem areas meeting the following criteria: numerous occurrences of the problem situation which make replication effective, a situation in which experts outperform nonexperts and the expertise is available, the relevant knowledge in principle may be articulated, and in cases in which the expertise is scarce. Today, as in the case of bots, many ES-type systems are incorporated in larger programs and increasingly are written in general computer languages such as C, C++, Visual Basic, or Java rather than in specialized languages such as LISP or PROLOG. (Although research continues using these languages.)

One classic application of an ES illustrates some of the ethical issues involved. In the early 1980s Campbell Soup developed an expert system to diagnose problems with the sterilizers in its cookers—the five-story vats in which soup is brewed. The cookers contain hydrostatic and rotary sterilizers used to kill botulism bacteria in the canned soup. There were more that 90 of these cookers in Campbell's factories around the world. At the time, one man, a 63-year-old named Aldo Cimino, was the company's expert in charge of troubleshooting, maintaining, and repairing the equipment, a function essential to the safety and quality of the product, and he was nearing retirement. Faced with losing this valuable expertise, Campbell's constructed an ES comprised of some 150 rules that made about 90% of Cimino's scarce knowledge available to the 90 sites. The ES was a success and was featured in a 1985 issue of *Personal Computing* magazine. It reduced costs, distributed valuable intelligence globally, and helped the company solve crucial problems involving the heath of their customers rapidly. Yet, the program raise questions of intellectual property rights. One article's headline read "An Expert Whose Brain Was Drained" and the text contained equally chilling metaphors about "canning" Cimino's 43 years on accumulated knowledge. Cimino, it appears, was an informed and willing participant in the development of the system (informed consent) and in just a few weeks of effort the knowledge it had taken him a lifetime to acquire was captured. Nevertheless, some people feel uneasy about this project and what it portends. It reveals the possibility that AI can be used for a deep violation of something private and personal. Common social sensitivities suggest that draining one's brain is a threat to that person's human dignity, unless, of course, that person consents and wants his or her expertise to become someone else's property and potentially live in perpetuity.

In developing ESs such as Campbell's, and in other AI applications, programmers proceed by extracting knowledge from experts, workers, and other knowledgeable people. Then the expertise is implanted into computer software where it becomes capital in the economic sense, property available for the production of wealth. The process of "disemminding" knowledge from an individual (akin to the disembodiment

of physical energy from workers during the Industrial Revolution) followed by the subsequent "emminding" of it into computers has major sociopolitical effects. It transfers control of the person's, such as Cimino's, knowledge—intellectual property—to those who own or use the hardware and software. This transfer of property raises several fundamental ethical questions. Who owns this knowledge? To what extent is it private or corporate or in the public domain? Is the transfer of the knowledge, that is, property, warranted? Is it just? Did it take place under conditions of truly *informed consent?* How is the contributor to be compensated? What are the effects of eliminating or "dumbing down" the jobs of one or more human's jobs?

The aforementioned systems are used in partnership with a human worker. The outcome of their acts are a joint product, just as the results of a rider and her horse, or a sheep dog and his shepherd, or a pilot and his automated plane activities are also joint products. Consequently, the responsibility for moral agency, for the good and bad things that happen as a result of using the system, is attributable in part to the human agent and in part to the machine. Indeed, until machines can be held accountable, the ultimate responsibility lies with the human. Nevertheless, the moral status of the AI system used in a partnership situation is similar to that of a professional. In fact, proposals have been made to legally recognize artificial intelligence programs so that they and their designers and programmers can be held liable, thereby in part absolving the user of the liability.

The basis for this legal approach, by analogy, is the ethics of professionalism. Professionals are people such as physicians, lawyers, and accountants who apply specialized knowledge to help people in need. Since they occupy a position of comparative social power, professionals are committed to high ethical standards. These standards include doing no harm, being competent, avoiding conflicts of interest, meeting client expectations, maintaining fiduciary responsibilities, safeguarding a client's privacy, protecting records and intellectual property, providing quality information, and abiding by applicable laws, contracts, and license agreements. Their obligations further include developing and maintaining the body of knowledge, educating and training other professionals, and monitoring and self-regulating their practice. Programmers and knowledge engineers as well as users should strive to implement systems that abide by these standards.

Some people have argued that a professional could come to rely too heavily on an ES partner and become lazy and deplete his or her skills. Thus, a professional could become negligent. But as Bruce Buchanan, a cofounder of DENDRAL, observed, ESs will not have fulfilled their promise until a physician is held liable for negligence for failing to use a system like, say, MCYIN when treating a patient. That is, an ES may reach the level that, given a duty of competence, a prudent practitioner would be expected to use appropriate AI systems in the professional conduct of his or her business. This possibility may become a pressing legal and ethical issue for AI in the future. Tort law may be applied whenever an AI system directly commits a wrongful act or causes injury or damage. In addition, as AI systems increasingly become the "standard of practice" a professional or a company may be held liable for *not* using an ES or other appropriate AI system when it was available. This means that users in a variety of fields may eventually he held responsible for knowing about AI and applying it to their work where appropriate. Importantly, users and society in general must become active in establishing laws and policies that set liability with respect to AI use or nonuse.

In some systems, however, the human partner's role is diminishing, approaching if not reaching zero. The history of DENDRAL, arguably one of the very first expert systems, reflects this possibility. DENDRAL is an expert chemist system that analyzes the molecular structure of substances submitted to a mass spectrometer, an instrument that vaporizes the unknown substance and records a spectrum of frequencies showing the molecular weight of each ion on a graph. Prior to DENDRAL, organic chemistry specialists would examine these graphs and related data intently to select from the thousands of possible isomers the particular molecular structure of bonds and atoms in the sample substance. DENDRAL was conceived and implemented by Edward Feigenbaum, Bruce Buchanan, Robert Lindsey, and Nobel laureate Joshua Lederberg with early consultation from C. West Churchman. Early runs of DENDRAL were able to identify correctly only a very small percentage of the compounds presented to it. The researchers consider this a "win" and pressed on. The system improved its performance as new knowledge was gained and incorporated. Within a year or so of its inception DENDRAL could compete effectively with practicing molecular chemists. Today DENDRAL (or Meta-DENDRAL) is no longer an expert system per se. Through research motivated in large part by the DENDRAL project, the underlying science is now known with certainty. The basic knowledge is public and analytic and no longer relies on the hunches and heuristics of experts. Chemists in universities and industrial labs all over the world now use DENDRAL.

Recent research developments, however, are raising new issues about future AI partnerships. Work is under way to develop methods for "downloading" the contents of one's brain and storing it away in a computer's memory for future use. At a later date this "brain" may be "uploaded" again to the donor (say, in the case of Alzheimer's disease) or, alternatively, saved for perpetuity. Embedded and implanted processors are also under development that, for example, will augment the mental abilities of users. Users of such devices then become human–machine hybrids. While initially the burden of responsibility for a hybrid's actions rests with the user, in time some of this burden will shift to the computer technology as it becomes more sophisticated and expansive. This is especially true in cases in which the computer element is in communication with other computers that provide it with data and direction. Looking forward to the year 2029 Kurzweil forecasts that direct neural pathways will have already been perfected for high-bandwidth connections to the human brain and that a range of neural implants will become available to enhance human visual and auditory perception and interpretation, memory, and reasoning.

The trouble with having a good partner—especially one that is programmed to learn—is that as the partner learns and develops it takes on more responsibility in the relationship. The history of DENDRAL, for example, substantiates this possibility. In due time, the more accomplished partner will replace or supersede the less accomplished. Moral responsibility is no longer split. Eventually, the partner with superintelligence dominates the relationship. Social power is transferred to the AI computer. Then either the need for the human partner is eliminated or the machine assumes a status of its own and coexists with the humans.

H. Moral Implications of AI as Superintelligence

1. Superintelligence Prophesized

"Within thirty years," Vernor Vinge prophesied in 1993, "we will have the technological means to create superhuman intelligence. Shortly after, the human era will be ended." Inventor and entrepreneur Ray Kurzweil writing in 1999 agrees (depending on how the concept "human" is defined). "Before the next century is over, human beings will no longer be the most intelligent or capable type of entity on the planet." Both authors forecast that society will reach a point—some call it the "Singularity" point, others the "Omega"—beyond which machines will have more social power than humans. This will result from the accelerating rate of computing and communications power brought about by technological improvements. If or when this point is reached, the human experience as we know it will be radically changed. Intelligent machines will rule the world.

The 1968 movie *2001: A Space Odyssey* (by Stanley Kubrick, based on a book by Arthur C. Clarke) is a cautionary tale about human destiny after the AI Singularity point is reached. The movie features a computer named HAL that guides the spacecraft on its mission. HAL exhibits several attributes of intelligence that are central topics of AI research today. HAL can understand natural languages (natural language processing); reason with logic (ES, etc.); see and perceive (vision and neural networks); and monitor and control the spaceship (cybernetics and robotics). HAL also displays elementary emotions: "Stop, Dave. I'm afraid . . . I'm afraid." Moreover, HAL provokes sympathy among many movie viewers. Some feel a greater loss when HAL dies than they do when astronaut Frank Poole drifts off into space. Yet in the end it turns out that HAL has a tragic moral flaw. The computer is confronted with a dilemma. On the one hand, HAL's program requires him to keep his crew informed of all significant events and, hence, tells him that he should inform the crew about any impending danger that occurs during their secret mission. On the other hand, doing so would reveal HAL's confidential orders, orders he "promised" (was programmed) not to disclose. HAL's moral logic seeks an answer that will resolve the tension in this dilemma. And HAL finds a solution: kill everyone on board. Then he can keep his promise and not have to lie. Logically HAL's quandary is resolved. With no crew aboard he has no dilemma. So at the crucial moment of truth HAL lacks the common sense of a child. He does not even question his solution. HAL's designers failed to incorporate moral principles such as "Thou shall not kill" or "First do no harm" effectively into his programs. But this leaves open the question of whether or not it is humanly possible to incorporate all relevant moral principles and resolution processes in a computer program, especially if the machine is finite and the brain infinite.

An AI researcher named Douglas Lenat is actively working to provide the AI community the common sense that HAL lacked. Since about 1984 Lenat has worked to create a program with common sense called CYC, short for encyclopedia. Millions of commonsense facts and rules of thumb from everyday life

describing the real world of places, things, people and their affairs, time, space, causality, and contradiction have been introduced into CYC's database. In early 2000 the database contained more than a billion bytes of information. Lenat believes that this body of common sense will eventually serve as a backdrop for all AI programs and keep them from making the kinds of mistakes that narrow-minded computers (such as HAL) do. If CYC is successful, a major step will have been taken toward superintelligence. According to a December 1999 article in the *Austin Chronicle,* however, CYC has not yet reached this level of learning. For example, in one run CYC concluded that everybody born before 1900 was famous, because all of the people recorded in its database who were born prior to 1900 were famous people. Despite limitations like this, CYC has shown that a commonsense backdrop can keep other programs from making some obvious mistakes.

The trajectory of AI research indicates that someday—perhaps not as soon as either Vinge or Kurzweil predict—machines with the superintelligence of HAL or with even greater capabilities will be produced. In his 1988 book *Robot: Mere Machine to Transcendent Mind,* Hans Moravec forecasts that "within the next century they [AI computer programs] will mature into entities as complex as ourselves, and eventually into something transcending everything we know—in whom we can take pride when they refer to themselves as our descendants." He goes on the say that "We are very near to the time when virtually no essential human function, physical or mental, will lack an artificial counterpart. The embodiment of this convergence of cultural developments will be the intelligent robot, a machine that can think and act as a human."

Kurzweil in *The Age of Spiritual Machines: When Computers Exceed Human Intelligence* paints the following picture of the year 2099: "There is a strong trend toward a merger of human thinking with the world of machine intelligence that the human species initially created. There is no longer any clear distinction between humans and computers. Most conscious entities do not have a permanent physical presence. Machine-based intelligences derived from extended models of human intelligence claim to be human, although their brains are not based on carbon-based cellular processes, but rather electronic and photonic equivalents. Most of these intelligences are not tied to a specific computational processing unit. The number of software-based humans vastly exceeds those still using native neuron-cell-based computation." If Kurzweil is right, by 2099 the Singularity point will have been passed. A new world order will be in the making. What are the moral implications of this eventuality?

2. Three Scenarios of a World with Superintelligent AI

In his 1993 book Daniel Grevier describes several possible scenarios. Each has important implications for the flourishing of humankind.

a. Colossus

A militaristic computer dominates society. Patterned after the 1969 movie *Colossus: The Forbin Project,* in this scenario multiple distributed computers linked by the Internet and other telecommunications are able to make decisions and act more quickly than their human controllers and take charge. These machines merge electronically. The resulting composite "digital superpower" dictates its will to humanity. In *Darwin Among the Machines: The Evolution of Global Intelligence,* George Dyson presents a detailed argument, based on an analysis of intellectual history, as to how this scenario will come about. Such a scenario poses several ethical problems. One is the control issue illustrated by HAL. How do we know that this new superpower will act in humanity's best interests? Another is accuracy. No human or machine is mistake free. Is it possible to build a machine that will make only a few tolerable mistakes, to display only a few minor (tolerable) human flaws?

Closely related to this is the problem of program stability. These complex systems reach levels well beyond a human being's ability to comprehend. They are highly interactive, technically chaotic. They may become unstable at any moment. Indeed, the history of large-scale programming is instructive in this regard. For example, a large simulation program that is run on two different computers with slightly different circuit designs (say, in round-off or floating-point treatment) might produce radically different results. Or, a given program might run smoothly for months, even years, only to crash one day due to internal program conflicts that were unanticipated or not defined.

Finally, such a machine, since it will be learning and modifying its own behavior, will create its own value system. It may become highly egotistical. Even if it is programmed initially with good intentions from a humanistic point of view, it may well arrive at a value system in which humans are not valued very highly. We know that in some human value systems (such as the Nazi regime's) certain categories of people can be considered expendable, extinguishable. Why can't AI machines learn this as well? This is the Frankenstein

motif in which a creature constructed by a human being feels ignored or exploited and seeks sympathy for itself. Not finding satisfaction, the creature turns evil and ultimately destroys its animator as well as, perhaps, others around it.

b. Big Brother

This is the Orwellian motif as portrayed in *1984* and *Animal Farm*. The AI computers form a coalition with enough power to create a totalitarian society that controls all flows of information. All historical records are destroyed or modified to meet the computer's plans and replaced by the computers' propaganda. In this society there is no place for truth. Privacy is impossible, apathy prevails. This system does not necessarily kill people physically. Rather, it destroys their souls. It dehumanizes them.

Aldus Huxley in *Brave New World* describes how this state of dehumanization can be reached in another way. In Huxley's account Big Brother does not intrude in human affairs so much as the humans are inevitably drawn to "him." The AI system distracts humans by feeding them with trivia. It entertains them and pacifies them with "soma." In this scenario the benign machines dominate human beings not by force, but, as the media pundit Neil Postman put it, because the people are simply amusing themselves to death. Laxity and apathy become the significant moral issues.

c. Blissful Utopia

No full-blown AI utopia based on the creation of superintelligent machines has been written to date. Yet authors like Dyson, Kurzweil, and Crevier offer arguments claiming that the move toward superintelligent machines is an integral part of the natural processes of evolution and that, all things considered, it is a good and probably inevitable thing. The ethics of natural evolution support it. According to this scenario this phase of evolution represents the next stage in the perfectibility of humankind. With the guidance of machines, people are able to conquer every kind of disorder and conflict in their lives and souls.

Just as the invention of writing and subsequently printing improved the quality of human life despite some dislocations, superintelligent computer robots will create an even better world—a utopia. The first two scenarios above depict "dystopias." In this scenario AI-driven machines, it is often implicitly assumed, will create a new society based on rationality, harmony, utility, simplicity, and order. Conflicts between people and their environment will be largely eliminated. This new regime will be a better one because the people living in it will become morally better people. They will be happier, more self-fulfilled, more autonomous, and freer. H. G. Wells, in *The Outline of History* and some of his earlier works, suggests that Darwinian evolution moves in this optimistic direction.

This scenario also reinforces Robert Owen's 1836 idea that with the aid of technology and the right social system humankind is capable of "endless progressive improvement, physical, intellectual, and moral, and of happiness, without the possibility of retrogression or of assignable limit." The Romano-British monk Pelagius believed that humans could perfect themselves by exercising their own free will. Saint Augustine believed that only God could perfect humankind. In *Emile* Rousseau suggests that by carefully selecting a tutor (i.e., programmer) and purifying a person's environment one can be educated to become a perfect person. Like these thinkers who went before them, AI utopians believe that carefully constructed intelligent machines will allow humankind to reach the same goal of perfectibility. A related possibility is that humans can also achieve immortality by means of "downloading" their contemporary and finite minds into infinite, self-reproducible silicon memories. For example, in Hans Moravec's view, human brain cells will eventually be replaced by an ANN composed of electronic circuits with virtually identical input and output functions. In such an ideal world moral issues are minimal.

Crevier believes that during the first two decades of the 21st century the blissful utopia scenario—one he calls "Lift-Off"—will prevail. "AI will gradually seep into all human activities, with mostly beneficial effects," he writes in *AI: The Tumultuous History of the Search for Artificial Intelligence*. "Since during that period machines will remain less intelligent than people, we should keep the upper hand on them without too much difficulty." But following this brief golden age he foresees the possibilities of scenarios 1 or 2 coming into play. "The machines will eventually excel us in intelligence, and it will become impossible for us to pull the plug on them. (It is already almost impossible: powering off the computers controlling our electric transmission networks, for instance, would cause statewide blackouts.) Competitive pressures on the businesses making ever more intensive use of AI will compel them to entrust the machines with even more power. E-commerce and E-government are major factors in this trend. Such pressures will extend to our entire social and legal framework. For instance, proposals already exist for legally recognizing artificial intelligence programs as persons in order to solve

the issues of responsibility posed by the use of expert systems."

In any of these three scenarios AI-based machines qualify for a high degree of "personhood." They will have moral status and likely will become the arbiters of morality. Moreover, their display of intelligence will have qualified them for a level of respect similar to that accorded a human person. In a 1964 article, philosopher Hilary Putham laid out the criterion: "[I]f a machine satisfied the same psychological theory as a human, then there is no good reason not to regard it as a conscious being in exactly the same way that the human is regarded as a conscious being."

4. Asimov's Moral Rules for Robots

If Putham, Dyson, Kurzweil, Moravec, and others are right and machines reach a high enough level of intelligence to satisfy the same psychological theory as humans then machines will also become moral beings. Blame, praise, and responsibility can then be assigned to these machines. Consequently, these superintelligent AI machines will need a moral center. HAL's transgressions must be avoided. This problem was addressed imaginatively by science fiction writer Isaac Asimov in his 1950 book *I, Robot*. His "Rules for Robotics" are, in priority order:

1. A robot may not injure a human being or through inaction allow a human being to come to harm.
2. A robot must obey orders given it by humans except when such orders conflict with the first law.
3. A robot must protect its own existence as long as such protection does not conflict with the first and second laws.

To this basic list might be added W. D. Ross's list of *prima facie* duties including beneficence, nonmaleficence, justice, gratitude, reparation, fidelity, and self-improvement. This rule-based approach to morality, of course, is based on a duty or obligation-based philosophy. Other moral systems would require another approach.

IV. SOME RELIGIOUS IMPLICATIONS OF AI

Many human values are rooted in religion, and religious grounding is the source of many people's identity. Religion is also the motivating force behind society's drive to develop technology Professor David F. Noble asserts in a 1997 book, *The Religion of Technology: The Divinity of Man and the Spirit of Invention*. AI research—the quest for "immortal mind" as Noble describes it—has been inspired by human needs or desires for immortality, resurrection, redemption, deliverance, disembodied perfection, and Godlike omnipresence and omnipotence. AI like other modern technological initiatives—atomic weapons development, space exploration, and genetic engineering—is animated by these fundamental religious purposes. AI is to modern society what the promise of the Cathedral of Notre Dame was to the Parisian artisans who in about 1163 began on a 100-year-plus construction program to save their souls. Similarly, machines with superintelligence open up the possibility of human transcendence and immortality. These machines, however, could craft lives with their own initiative and chart destinies beyond the human experience. A new species, *"Machina sapiens,"* would supersede *Homo sapiens*. The species *"Machina sapiens"* is the next phase in Descartes great plan to separate the human mind from the constraints of the body. For Descartes the human intellect is godly, trapped in a body that is a hindrance to thinking. Freeing the mind from the body frees the soul. But, in creating *"Machina sapiens,"* researchers are playing God. In this new endeavor they are vulnerable to committing sins and transgressions. Religious reflection on this matter to date is sparse. Nevertheless, if a sufficient condition for possessing a soul is to be a living being and AI eventually succeeds in producing a living being of some sort, then this new being will have a soul. It will, thus, have religious status. And, a new religious order may be born.

V. WHAT HATH GOD (OR AI) WROUGHT?

We face a modern Malthusian challenge. Our largely 19th-century education and economics may be unsuited for the demands of 21st-century technology. In 1798 the British economist Thomas Malthus published *An Essay on the Principle of Population* in which he argued that population tends to grow at a rate faster than the food supply does, with inevitably disastrous results. This outcome is abated somewhat, he observed, if the increase in population is checked by moral restraints or by traumas such as war, famine, and disease. History has shown, however, that improvements in technology, especially since the Industrial Revolution, have raised economic production

rates even faster that the rate of increase in population. Consequently, the global capability to produce food and other commodities continues to exceed the minimal requirements for survival. Apart from some severe problems of distribution, the Malthusian disaster has been averted.

Today the pressing issue is not whether technology is growing fast enough to meet the demands of a growing population. Rather, it is whether it is being innovated too fast. Computers are a case in point. In the 20th-century, computer technology, following a pattern know as "Moore's law," has improved in performance at a surprisingly steep exponential rate. Kurzweil, for example, claims that computers are about 100 million times more powerful for the same unit cost today (circa 2000) than they were a half century ago. Metcalfe's law further states that the value of a network and its capacity to do work expand exponentially with the number of nodes it encompasses. Today's Internet links millions of computers. The modern Malthusian problem, then, is that technologies, such as AI, are expanding at a rate much faster than our social and ethical understanding of them. We have little guidance as to how to use or not use them in morally acceptable ways. Scenarios, such as those summarized above, are unfolding before we adequately understand their possibilities. And, the faster the rate of technological development, the greater the gap in our ethical understanding. Each new development presents a "defining moment" or "moment of truth" at which its socially acceptable use of a technology can be determined or, at least, tagged for ongoing monitoring. Unfortunately this crucial moment often is not fully recognized and in its wake the gap widens. The moment of social recognition often comes only after a disaster occurs and, lamentably, at a time when the costs of corrective action are quite large. In the 21st century, instantiations of AI are creating the modern Malthusian challenge. The ethics of AI require that we continue to reflect deeply on what is being wrought.

This suggests the use of a "precautionary principle" in moving ahead with AI implementations. In the face of uncertain outcomes we should stand ready to acknowledge our incomplete knowledge of the implications of AI systems and take steps, where necessary, to avoid major harm or the crossing of some critical threshold that unleashes an irreversible process. Adopting the precautionary principle should not serve as an unnecessary damper on research or delay implementing applications that will bring major benefits to society. Rather, it should serve as an aspiration to design AI systems that incorporate the highest ethical ideals, perhaps even higher than many of its users.

SEE ALSO THE FOLLOWING ARTICLES

Computer Viruses • Crime, Use of Computers in • Cybernetics • Expert Systems Construction • Future of Information Systems • Machine Learning • Medicine, Artificial Intelligence in • Software Piracy • Virtual Reality

BIBLIOGRAPHY

Abatgemarco, F. (November 1985). An expert whose brain was drained. *Personal Computing,* 98.

Asimov, I. (1950/1963). *I robot.* Garden City, NY: Doubleday.

Boden, M. (1985). Panel: Artificial intelligence and legal responsibilities. *Proc. International Joint Conference on Artificial Intelligence,* Los Angeles.

Churchman, C. W. (1971). *The design of inquiring systems.* New York: Basic Books.

Clarkson, G. P. E. (1963). A model of the trust investment process. *Computers and thought,* E. A. Feigenbaum and J. Feldman (Eds.). New York: McGraw-Hill.

Crevier, D. (1993). *AI: The tumultuous history of the search for artificial intelligence.* New York: Basic Books.

Curran, C. E. (1999). *The Catholic moral tradition today: A Synthesis.* Washington DC. Georgetown University Press.

Dreyfus, H. (1972). *What computers can't do: A critique of artificial reason.* New York: Harper & Row.

Dreyfus, H., and Dreyfus, S. (1986). *Mind over machine.* New York: Free Press.

Feigenbaum, E. A., and Feldman, J. (1963). *Computers and thought.* New York: McGraw-Hill.

Hogan, J. P. (1997). *Mind matters: Exploring the world of artificial intelligence.* New York: The Ballantine Publishing Co.

Kurzweill, R. (1999). *The age of spiritual machines: When computers exceed human intelligence.* New York: Penguin Books.

Licklider, J. C. R. (March 1960). Man–computer symbiosis. *IRE Transactions on Human Factors in Electronics,* Vol. HFE-1, 4–11.

Lindsay, R., Buchanan, B., Feigenbaum, E., and Laderberg, J. (1980). *Applications of artificial intelligence for chemical inference: The DENDRAL project.* New York: McGraw-Hill.

Locke, J. (1687/1965). *An essay concerning the human understanding.* M. Cranston (Ed.). New York: Collier.

McCorduck, P. (1979). *Machines who think.* New York: W. H. Freeman and Co.

Minsky, M. (1986). *The society of mind.* New York: Simon and Schuster.

Moravec, H. (1988). *Mind children.* Cambridge, MA: Harvard University Press.

Murch, R., and Johnson, T. (1999). *Intelligent software agents.* Upper Saddle River, NJ: Prentice Hall.

Newell, A., and Simon H. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice Hall.

Niebuhr, H. R. (1963). *The responsible self: An essay in christian moral philosophy.* New York: Harper and Row.

Nilsson, N. J. (1998). *Artificial intelligence: A new synthesis.* San Francisco: Morgan Kaufman.

Noble, D. (1997). *The religion of technology: The divinity of man and the spirit of invention.* New York: Peguin Books.

Norman, D. (1984). Worsening the knowledge gap. *Annals of the New York Academy of Sciences,* Vol. 426, 225.

Owen, R. (1836/1970). *Book of the new moral world.* iv. New York: A. M. Kelley.

Putnam, H. (1964). Robots: Machines or artificially-created life? *Journal of Philosophy,* Vol. 61, 668-691.

Schank, R. (with Peter G. Childers) (1984). *The cognitive computer: On language, learning, and artificial intelligence.* Reading, MA: Addison-Wesley.

Searle, J. R. (1980). Minds, brains and programs. *Behavioral and Brain Sciences,* Vol. 3, No. 3, 417–458.

Strawson, P. F. (1962). Freedom and resentment. *Proc. British Academy,* Vol. 48, 187–211. London: Oxford University Press.

Thomas, L. (February 1980). On artificial intelligence. *New England Journal of Medicine,* Vol. 28, 506.

Turing, A. M. (1950/1963). Computing machinery and intelligence. *Computers and thought,* E. A. Feigenbaum and J. Feldman (Eds.). New York: McGraw-Hill.

Vinge, V. (1993). The coming technological singularity: How to survive in the post-human era. *Whole Earth Review,* 88–95.

Walters, J. W. (1997). *What is a person?* Urbana: University of Illinois Press.

Warren, M. A. (1997). *Moral status: Obligations to persons and other living things.* Oxford: Clarendon Press.

Weizenbaum, J. (1976). *Computer power and human reason: From judgment to calculation.* New York: W. H. Freeman

Wiener, N. (1948). *Cybernetics or control and communication in the animal and machine.* Cambridge, MA: The MIT Press.

Winograd, T. (1972). *Understanding natural language.* New York: Academic Press.

Evolutionary Algorithms

Zbigniew Michalewicz
University of North Carolina, Charlotte

Marc Schoenauer
Ecole Polytechnique

GLOSSARY

fitness The measure of adaptation of the individuals to their artificial environment—basis for Darwinian selection mechanisms.

genotype The representation of an individual that will be transmitted (after possible modification by variation operators) to its offspring during evolution.

hybridization Use of existing (nonevolutionary) optimization techniques within an evolutionary algorithm.

individual Possible solution to the problem at hand, that is, from a mathematical point of view, a point of the search space.

phenotype The behavioral part of an individual. The phenotype is computed, or "decoded," from the genotype, and the fitness is computed on the phenotype.

population Set of individuals. The population is generally of fixed size.

replacement Second phase of artificial Darwinism. The fittest (deterministically or stochastically) individuals will survive.

representation Synonym for the genotypic space (space of genotypes). The choice of a representation for a given problem is the very first step of the design of an evolutionary algorithm.

selection First phase of artificial Darwinism. The fittest (deterministically or stochastically) individuals will reproduce.

variation operators Modification of the individuals in the search space. According to Darwin's principles, variation operators are not aware of the fitness of the individuals.

EVOLUTIONARY COMPUTING is an exciting development in computing science. It amounts to building, applying, and studying algorithms based on the Darwinian principles of natural selection ("survival of the fittest") and undirected variations. Evolutionary algorithms can also be viewed as an interesting category of modern heuristic search. This overview article presents the main paradigms of evolutionary algorithms (genetic algorithms, evolution strategies, evolutionary programming, genetic programming) as well as the trend for unification of these paradigms and hybridization with other existing search techniques. A brief survey of theoretical results is presented, as well as a list of application areas ranging from optimization, modeling, and simulation to entertainment.

I. INTRODUCTION

The evolutionary computation (EC) techniques are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian strife for survival. The idea behind evolutionary algorithms is to do what nature does. Let us

take rabbits as an example: At any given time there is a population of rabbits. Some of them are faster and smarter than other rabbits. These faster, smarter rabbits are less likely to be eaten by foxes, and therefore more of them survive to do what rabbits do best: make more rabbits. Of course, some of the slower, dumber rabbits will survive just because they are lucky. This surviving population of rabbits starts breeding. The breeding results in a good mixture of rabbit genetic material: some slow rabbits breed with fast rabbits, some fast with fast, some smart rabbits with dumb rabbits, and so on. And on the top of that, nature throws in a "wild hare" every once in a while by mutating some of the rabbit genetic material. The resulting baby rabbits will (on average) be faster and smarter than these in the original population because more faster, smarter parents survived the foxes. (It is a good thing that the foxes are undergoing a similar process—otherwise the rabbits might become too fast and smart for the foxes to catch any of them). So the metaphor underlying evolutionary algorithms is that of natural evolution. In evolution, the problem each species faces is one of searching for beneficial adaptations to a complicated and changing environment. The "knowledge" that each species has gained is embodied in the makeup of the chromosomes of its members. From the point of view of optimization, EC is a powerful stochastic zeroth-order method (i.e., requiring only values of the function to optimize) that can find the global optimum of very rough functions. This allows EC to tackle optimization problems for which standard optimization methods (e.g., gradient-based algorithms requiring the existence and computation of derivatives) are not applicable. Moreover, most traditional methods are local in scope, thus they identify only the local optimum closest to their starting point.

II. AN ALGORITHM

For the sake of clarity, we shall try to introduce a general framework that can account as much as possible for most of the existing evolutionary algorithms.

Let the search space be a metric space E, and let F be a function $E \to \Re$ called the *objective* function. The problem of evolutionary optimization is to find the maximum of F on E (the case of minimization is easily handled by considering $-F$).

A *population* of size $P \in \Re$ is a set of P *individuals* (points of E) not necessarily distinct. This population is generally initialized randomly (at time $t = 0$) and uniformly on E. The *fitnesses* of all individuals are computed (on the basis of the values of the objective function); a fitness value is represented as a positive real number—the higher the number, the better the individual. The population then undergoes a succession of *generations;* the process is illustrated in Fig. 1.

Several aspects of the evolutionary procedure require additional comments:

- *Statistics and stopping criterion:* The simplest stopping criterion is based on the generation counter t (or on the number of function evaluations). However, it is possible to use more complex stopping criteria, which depend either on the evolution of the best fitness in the population along generations (i.e., measurements of the gradient of the gains over some number of generations) or on some measure of the diversity of the population.
- *Selection:* Choice of some individuals that will generate offspring. Numerous selection processes can be used, either deterministic or stochastic. All are based on the fitness of the individuals. Depending on the selection scheme used, some individuals can be selected more than once. At that point, selected individuals give birth to copies of themselves (clones).
- *Application of variation operators:* To each one of these copies some operator(s) are applied, giving birth to one or more offspring. The choice among possible operators is stochastic, according to user-supplied probabilities. These operators are always stochastic operators, and one usually distinguishes between *crossover* (or *recombination*) and *mutation* operators:
  - Crossover operators are operators from E^k into E, i.e., some parents exchange genetic material to build up one offspring. In most cases, crossover involves just two parents ($k = 2$), however, it need not be the case. In a recent

```
procedure evolutionary algorithm
begin
   t ← 0
   initialize population
   evaluate population
   while (not termination-condition do
   begin
      t ← t + 1
      select individuals for reproduction
      apply variation operators
      evaluate newborn offspring
      replace some parents by some offspring
   end
end
```

Figure 1 The structure of an evolutionary algorithm.

study, the authors investigated the merits of "orgies," where more than two parents are involved in the reproduction process. Evolution strategies and scatter search techniques also proposed the use of multiple parents.
  - Mutation operators are stochastic operators from E into E.
- *Evaluation:* Computation of the fitnesses of all newborn offspring. As mentioned earlier, the fitness measure of an individual is directly related to its objective function value.
- *Replacement:* Choice of which individuals will be part of the next generation. The choice can be made either from the set of offspring only (in which case all parents "die") or from both sets of offspring and parents. In either case, this replacement procedure can be deterministic or stochastic.

Sometimes the variation operators are defined on the same space as the objective function (called *phenotype space* or behavioral space); in other cases, an intermediate space is introduced (called *genotype space* or representation space). The mapping from the phenotype space in the genotype space is termed *coding*. The inverse mapping from the genotype space in the phenotype space is termed *decoding*. Genotypes undergo variation operators, and their fitness is evaluated on the corresponding phenotype. The properties of the coding mappings can greatly modify the global behavior of the evolutionary algorithm.

III. GENETIC ALGORITHMS

In the canonical genetic algorithm (GA), the genotype space is $\{0,1\}^n$. Note that the phenotype space can be any space, as long as it can be coded into bit string genotypes. The selection scheme is proportional selection (the best-known being the *roulette wheel selection*): P random choices are made in the whole population, each individual having a probability proportional to its fitness of being selected. The crossover operators replace a segment of bits in the first parent string by the corresponding segment of bits from the second parent, and the mutation operator randomly flips the bits of the parent according to a fixed user-supplied probability. In the replacement phase, all P offspring replace all parents. Due to that generational replacement, the best fitness in the population can decrease: The original GA strategy is not *elitist*.

In more recent works, the genotype space can be almost any space, as long as some crossover and mutation operators are provided. Moreover, proportional selection has been gradually replaced by ranking selection (the selection is performed on the rank of the individuals rather than on their actual fitness) or tournament selection (one selects the best individual among a uniform choice of T individuals, T ranging from 2 to 10). Finally, most users use the elitist variant of replacement, in which the best individual of generation t is included in generation $t + 1$, whenever the best fitness value in the population decreases.

IV. EVOLUTION STRATEGIES

The original evolution strategy (ES) algorithm handles a "population" made of a single individual given as a real-valued vector. This individual undergoes a Gaussian mutation: addition of zero-mean Gaussian variable of standard deviation σ. The fittest individual from the parent and the offspring becomes the parent of the next generation. The critical feature is the choice of parameter σ: Originally, the so-called 1/5 thumb rule [i.e., When more than 1/5 mutations are successful (respectively, unsuccessful), increase (respectively, decrease) σ] was used to adjust parameter σ along evolution.

More recent ES algorithms are population-based algorithms, termed $(\mu,\lambda) - \text{ES}$ or $(\mu + \lambda) - \text{ES}$: μ parents generate λ offspring. (There is no selection at that level, i.e., every parent produces λ/μ offspring on average.)

The main operator remains mutation. When working on real-valued vectors (still their favorite universe) ESs generally use the powerful paradigm of *self-adaptive mutation:* The standard deviations of Gaussian mutations are part of the individuals, and undergo mutation as well. Last, ESs now frequently use a global recombination operator involving all individuals in the population.

The replacement step is deterministic, i.e., the best μ individuals become the parents of the next generation, chosen among the $\mu + \lambda$ parents plus offspring in the elitist $(\mu + \lambda) - \text{ES}$ scheme, or among the λ offspring in the nonelitist $(\mu,\lambda) - \text{ES}$ scheme (with $\lambda \geq \mu$). Typical values for (μ,λ) are (1,7), (10,100) or (30,200).

V. EVOLUTIONARY PROGRAMMING

Originally designed to evolve finite state machines, evolutionary programming (EP) emphasizes the phenotype space. As in ESs, there is no initial selection: Every

individual in the population generates one offspring. Moreover, the only evolution operator is mutation. Finally, the best P individuals among parents and offspring become the parents of the next generation.

Recent advances handle any space, still emphasize the use of mutation as the only operator, independently design the self-adaptive Gaussian deviations for real-valued variables, and now use a stochastic tournament replacement scheme: Each individual (among the $2P$ parents plus offspring) encounters T random opponents, increasing its score by one point if it has better fitness. The P individuals having the highest scores get along to the next generation. Note that EP replacement scheme is always *elitist*.

VI. GENETIC PROGRAMMING

Genetic programming as a method for evolving computer programs first appeared as an application of GAs to tree-like structures. Original GP evolves tree structures representing LISP-like S expressions. This allows us to define very easily a closed crossover operator (by swapping subtrees between two valid S expressions, we always gets a valid S expression). The usual evolution scheme is the steady-state genetic algorithm (SSGA): A parent is selected by tournament (of size 2 to 7 typically) and generates an offspring by crossover only (the other parent is selected by a tournament of usually smaller size). The offspring is then put back in the population using a death-tournament: T individuals are uniformly chosen, and the one with the worse fitness gets replaced by the newborn offspring.

More recently, mutation operators, for example, random replacement of a subtree or random change of a node or a leaf, have been used—see the state-of-the-art books listed in the Bibliography.

VII. MODERN TRENDS: HYBRID METHODS

Many researchers modified further evolutionary algorithms by "adding" some problem-specific knowledge to the algorithm. Several papers have discussed initialization techniques, different representations, decoding techniques (mapping from genetic representations to phenotypic representations), and the use of heuristics for variation operators. Davis wrote (in the context of classical, binary GAs):

> It has seemed true to me for some time that we cannot handle most real-world problems with binary representations and an operator set consisting only of binary crossover and binary mutation. One reason for this is that nearly every real-world domain has associated domain knowledge that is of use when one is considering a transformation of a solution in the domain. . . . I believe that genetic algorithms are the appropriate algorithms to use in a great many real-world applications. I also believe that one should incorporate real-world knowledge in one's algorithm by adding it to one's decoder or by expanding one's operator set.

Such hybrid/nonstandard systems enjoy a significant popularity in evolutionary computation community. Very often these systems, extended by the problem-specific knowledge, outperform other classical evolutionary methods as well as other standard techniques. For example, a system called Genetic-2N, constructed for the nonlinear transportation problem, used a matrix representation for its chromosomes, a problem-specific mutation (main operator, used with probability 0.4), and arithmetical crossover (background operator, used with probability 0.05). It is hard to classify this system; it is not really a genetic algorithm, because it can run with a mutation operator only without any significant decrease of quality of results. Moreover, all matrix entries are floating-point numbers. It is not an evolution strategy, because it does not use Gaussian mutation, nor does it encode any control parameters in its chromosomal structures. Clearly, it has nothing to do with genetic programming and very little (matrix representation) with evolutionary programming approaches. It is just an evolutionary computation technique aimed at particular problems.

VIII. COMPARISON

Many papers have been written on the similarities and differences between these approaches. Clearly, different points of view can be adopted.

- *The representation issue:* Original EP, ESs, and GAs address only finite state machines, real numbers and bit strings, respectively. However, recent tendencies indicate that this is not a major difference. More important is the adequacy of the variation operators to the chosen representation and the objective function (i.e., the fitness landscape).
- *Bottom-up versus top-down, and the usefulness of crossover:* According to the schema theorem of Holland and Goldberg, GA's main strength comes from the crossover operator: Better and better solutions are built

by exchanging *building blocks* from partially good solutions previously built, in a bottom-up approach. The mutation operator is then considered as a background operator. On the other hand, the philosophy behind EP and ESs is that such building blocks might not exist, at least for most real-world problems. This top-down view considers selective pressure plus genotypic variability brought by mutation to be sufficient.

The discussion on crossover has been going on for a long time. And even when crossover was experimentally demonstrated beneficial to evolution, it could be because it acts like a large mutation; recent experiments suggest that the answer is highly problem dependent.

Yet another example of the duality between crossover and mutation comes from GP history: the original GP algorithm used only crossover, with no mutation at all, the very large population size being assumed to provide all the necessary building blocks to represent at least one sufficiently good solution. But more recent works on GP accommodate mutation also, on a much smaller population.

- *Mutation operators:* The way in which mutation operators are applied differs from one algorithm to another.

GA uses a static mutation rate or user-prescribed evolution scheme to globally adjust either the mutation rate (i.e., the number of individuals that undergo mutation) or the strength of mutation (i.e., the average number of bits that are flipped in an individual undergoing mutation).

Originally, ES used a heuristic adaptation mechanism (the 1/5 rule), which was later turned into the modern self-adaptive mutation: All individuals carry their own copy of the standard deviation(s) of the mutation. These variances undergo in turn mutation, and the individual is further modified according to the new value of the variance, which is therefore evolved and optimized "for free." The strength of mutation in EP is historically defined as a function of the relative fitness of the individual at hand before independently turning to self-adaptation.

Note that self-adaptive mutation rates (i.e., dependent on the individual) have a significant impact only when all individuals undergo mutation, which is not true for GAs where the mutation rate is generally low. However, the importance of local mutation is confirmed by theoretical results in ESs. A prerequisite for convergence is the *strong causality principle* emphasized by Rechenberg of ESs: Small mutations should have small effects on the fitness. This is not the case when floating-point numbers are encoded into binary strings (as is the case for classical GAs).

- The selection–replacement mechanisms range from the totally stochastic fitness proportional selection of GAs with generational replacement, to the deterministic (μ, λ) replacement of ES, through the stochastic, but elitist, tournament replacement of EP and the steady-state scheme (tournament selection and death tournament replacement) used in GP. Though some studies have been devoted to selection/replacement mechanisms, the choice of a selection scheme for a given problem (fitness-representation-operators) is still an open question (and is probably problem dependent).

The current trend in the EC community is to mix up all of these features to best fit the application at hand, on a few pragmatic bases: Some ESs applications deal with discrete or mixed real-integer spaces, the "binary is the best" credo of GAs has been successfully attacked (Antonisse, 1989), and the schema theorem extended to any representation. Note that some ES variations incorporate crossover, mutation has been added to GP, and so on. And the different selection operators are more and more being used now by the whole community.

On the other hand, such hybrid algorithms, by getting away from the simple original algorithms, also escape the few available theoretical results. Thus, the study of the actual complexity of the resulting algorithms remains unreachable.

IX. THEORETICAL RESULTS

Theoretical studies of evolutionary algorithms are of two types: An evolutionary algorithm can be viewed as a Markov chain in the space of populations, because population at time $t + 1$ only depends on population at time t (at least in the standard algorithms). The full theory of Markov chains can then be applied. On the other hand, the specific nature of evolution strategies allowed precise theoretical studies on the rate of convergence of these algorithms using probability calculus (at least for locally convex functions).

Results based on Markov chains analysis are available for the standard GA scheme (proportional selection with fixed mutation rate). The need for an elitist strategy is emphasized by Rudolph. When the mutation rate is allowed to decrease along generations, techniques borrowed from the field of simulated annealing give more precise convergence results in probability. Yet a different approach is used by Cerf that considers the GA as a stochastic perturbation of

a dynamical system (a caricature GA). The powerful Friedlin-Wentzell theory can then be applied, resulting in a lower bound on the population size for a convergence in finite time of a modified GA (in which the selection strength and mutation rate are carefully modified along generations). However, even this last result is nonconstructive, i.e., of limited use when actually designing an instance of evolutionary algorithm for a particular problem.

On the other hand, ESs have considered theoretical studies from the very beginning: studies on the sphere and corridor models gave birth to the 1/5 rule, with determination of the optimal update coefficients for the mutation rate. The theory of ESs later developed to consider global convergence results in probability for the elitist models, as well as for the nonelitist $(1, \lambda) - \mathrm{ES}$. The whole body of work by Beyer concentrates on the optimal progress rate for different variants of evolution strategies (and, for instance, justify some parameter settings for self-adaptive mutation given by Schwefel). The main weakness of these results remains that they were derived on simple models of function; their main results (e.g., optimal parameter settings) are nevertheless applied without further justification to any function—and usually prove to be efficient hints.

However, one should keep in mind that all of the above theoretical analyses address some simple models of evolutionary algorithms. As stated earlier, the modern trends of EC gave birth to hybrid algorithms, for which generally no theory is applicable.

X. APPLICATION AREAS

Although it is often stressed that an evolutionary algorithm is not an optimizer in the strict sense, optimization problems form the most important application area of EAs. Some conferences dedicated to application of EAS and their proceedings provide a wide overview of actual applications. Another regularly updated source is the Evonet *Evolution@work* database.

This section will survey the preferred domains of application of EAs. The different subdomains are distinguished according to the type of search space they involve.

A. Discrete Search Spaces

Hard combinatorial optimization problems (NP-hard, NP-complete) involve huge discrete search spaces, and have been studied extensively by the operational research community. Two different situations should be considered: academic benchmark problems and large real-world problems.

As far as benchmark problems are concerned, it is now commonly acknowledged that EAs alone cannot compete with OR methods. However, recent advances in hybrid algorithms termed *Genetic local search,* where the EA searches the space of local optima with respect to some OR heuristic, have obtained the best results so far on a number of such benchmark problems.

The situation is slightly different for real-world problems: "Pure" OR heuristics generally do not directly apply, and OR methods have to take into account problem specificities. This is true of course for EAs, and there are many success stories where EAs, carefully tuned to the problem at hand, have been very successful, for instance, in the broad area of scheduling.

B. Parametric Optimization

The optimization of functions with floating-point variables has been thoroughly studied by practitioners, and many very powerful methods exist. Though the most well-known address linear or convex problems, many other cases can be handled successfully. Hence the niche for EAs is quite limited in that area, and only highly multimodal and irregular functions should be considered for EAs. However, successes have been encountered in such situations, in different domains ranging from electromagnetism to control and to fluid dynamics.

The situation drastically changes, however, when dealing with multiobjective problems: Evolutionary multi-objective (EMO) algorithms are the only ones that can produce a set of best possible compromise (the *Pareto set*) and have recently received increased attention. EMO algorithms use the same variation operators as standard EAs, but the Darwinian components are modified to take into account the multivalued fitness.

C. Mixed Search Spaces

When it comes to mixed search spaces, that is, when different types of variables are involved (generally both continuous and discrete variables), almost no classical optimization method applies, although some OR methods can be used if continuous variables are transformed into intervals, or continuous methods can be applied to the discrete variables. All of these

approaches can easily fall into traps due to the very different nature of continuous and discrete variables.

EAs, however, are flexible enough to handle such search spaces easily. Once variation operators are known for continuous and discrete variables, constructing variation operators for mixed individuals is straightforward: Crossover, for instance, can either exchange values of corresponding variables or use the variable-level crossover operator. Many problems have been easily handled that way, like optical filter optimization, where one is looking for a number of layers, the unknown being the layer thickness (continuous) and the material the layer is made of (discrete).

D. Artificial Creativity

But the most promising area of application of EAs, where EAs can be much more than yet another optimization method, is probably design. And here again, progress comes from the ability of EAs to handle almost any search space. The idea of component-based representations can boost innovation in structural design, architecture, and in many other areas including art. But the most original idea in that direction is that of embryogenies: The genotype is a program, and the phenotype is the result of applying that program to "grow an embryo"; the fitness is obtained by testing that phenotype in a real situation. Such an approach is already leading to astonishing results in analog circuit design for instance—though exploring a huge search space (a space of programs) implies a heavy computational cost. But we firmly believe that great achievements can come from such original ideas.

XI. CONCLUSIONS

Natural evolution can be considered a powerful problem solver that brought *Homo sapiens* out of chaos in only a couple of billion years. Computer-based evolutionary processes can also be used as efficient problem solvers for optimization, constraint handling, machine learning, and modeling tasks. Furthermore, many real-world phenomena from the study of life, economy, and society can be investigated by simulations based on evolving systems. Last but not least, evolutionary art and design form an emerging field of applications of the Darwinian ideas. We expect that computer applications based on evolutionary principles will gain popularity in the coming years in science, business, and entertainment.

SEE ALSO THE FOLLOWING ARTICLES

Cybernetics • Engineering, Artificial Intelligence in • Expert Systems Construction • Game Theory • Goal Programming • Hybrid Systems • Industry, Artificial Intelligence in • Intelligent Agents • Machine Learning • Neural Networks

BIBLIOGRAPHY

Angeline, P. J., and Kinnear, K. E., Jr. (Eds.) (1996). *Advances in genetic programming II,* Cambridge, MA: The MIT Press.

Antonisse, J. (1989). A new interpretation of schema notation that overturns the binary encoding constraint. 86–91.

Bäck, Th. (1995). Generalized convergence models for tournament- and (μ, λ)-selections. *Proc. 6th International Conference on Genetic Algorithms,* L. J. Eshelman (Ed.) San Francisco: Morgan Kaufmann, 2–8.

Bäck, Th. (1996). *Evolutionary algorithms in theory and practice.* New York: Oxford University Press.

Bäck, Th., and Schütz, M. (1995). Evolution strategies for mixed-integer optimization of optical multilayer systems.

Bäck, Th., and Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation,* Vol. 1, No. 1, 1–23.

Bäck, Th., Rudolph, G., and Schwefel, H.-P. (1993). Evolutionary programming and evolution strategies: Similarities and differences. 11–22.

Bentley, P. J. (Ed.) (1999). *Evolutionary design by computers.* San Francisco: Morgan Kaufman.

Beyer, H. G. (1993). Toward a theory of evolution strategies: Some asymptotical results for the (1, +λ)-theory. *Evolutionary Computation,* Vol. 1, No. 2, 165–188.

Beyer, H. G. (1994). Toward a theory of evolution strategies: The (μ, λ)-theory. *Evolutionary Computation,* Vol. 2, No. 4, 381–407.

Beyer, H. G. (1995). Toward a theory of evolution strategies: On the benefit of sex—the (μ/μ, λ)-theory. *Evolutionary Computation,* Vol. 3, No. 1, 81–111.

Beyer, H. G. (1995). Toward a theory of evolution strategies: Self-adaptation. *Evolutionary Computation,* Vol. 3, No. 3, 311–347.

Blythe, P. W. (1998). Evolving robust strategies for autonomous flight: A challenge to optimal control theory. *Adaptive computing in design and manufacture,* I. Parmee (Ed.), New York: Springer Verlag, 269–283.

Bonnans, F., Gilbert, J., Lemarechal, C., and Sagastizbal, C. (1997). *Optimisation numrique, aspects thoriques et pratiques,* Vol. 23 of *Mathematiques & applications.* New York: Springer Verlag.

Cerf, R. (1996). An asymptotic theory of genetic algorithms. *Artificial evolution,* J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers (Eds.), Vol. 1063 of *LNCS.* New York: Springer Verlag.

Chakraborty, U., Deb, K., and Chakraborty, M. (1996). Analysis of selection algorithms: A markov chain approach. *Evolutionary Computation,* Vol. 4, No. 2, 133–168.

Davidor, Y., Schwefel, H.P., and Männer, R. (Eds.) (1994). *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN).* New York: Springer-Verlag.

Davis, L. (Ed.) (1987). *Genetic algorithms and simulated annealing.* San Francisco: Morgan Kaufmann.

Davis, L. (1989). Adapting operator probabilities in genetic algorithms. 61–69.

Davis, L., and Steenstrup, M. (1987). Genetic algorithms and simulated annealing: An overview. 1–11.

Davis, T. E., and Principe, J. C. (1991). A simulated annealing like convergence theory for simple genetic algorithm. *Proc. 4th International Conference on Genetic Algorithms,* R. K. Belew and L. B. Booker (Eds.). San Francisco: Morgan Kaufmann, 174–181.

Davis, T. E., and Principe, J. C. (1993). A markov chain framework for the simple genetic algorithm. *Evolutionary Computation,* Vol. 1, No. 3, 269–292.

DeJong, K. A. (1992). Are genetic algorithms function optimizers? *Proc. 2nd* Conference on Parallel Problems Solving from Nature, R. Manner and B. Manderick (Eds.). Amsterdam: North Holland, 3–13.

Eiben, A. E., Aarts, E. H. L., and Van Hee, K. M. (1991). Global convergence of genetic algorithms: A markov chain analysis. *Proc. 1st Parallel Problem Solving from Nature,* H.-P. Schwefel and R. Männer (Eds.). New York: Springer Verlag, 4–12.

Eiben, A. E., Raue, P.-E., and Ruttkay, Z. (1994). *Genetic algorithms with multi-parent recombination,* 78–87.

Eshelman, L. J. (Ed.) (1995). *Proceedings of the Sixth International Conference on Genetic Algorithms.* San Francisco: Morgan Kaufmann.

Eshelman, L. J., Caruana, R. A., and Schaffer, J. D. (1989). Biases in the crossover landscape. 10–19.

European Network on Evolutionary Computing. Successful applications of evolutionary algorithms. Available at http://evonet.dcs.napier.ac.uk/.

Fogel, D. B. (1995). *Evolutionary computation. Toward a new philosophy of machine intelligence.* Piscataway, NJ: IEEE Press.

Fogel, D. B., and Atmar, W. (1992). *Proceedings of the First Annual Conference on Evolutionary Programming,* La Jolla, CA. Evolutionary Programming Society.

Fogel, D. B., and Atmar, W. (1993). *Proceedings of the Second Annual Conference on Evolutionary Programming,* La Jolla, CA. Evolutionary Programming Society.

Fogel, D. B., and Stayton, L. C. (1994). On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems,* Vol. 32, 171–182.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial intelligence through simulated evolution.* New York: John Wiley.

Fogel, D. B., Fogel, L. J., Atmar, W., and Fogel, G. B. (1992). Hierarchic methods of evolutionary programming. 175–182.

Galinier, P., and Hao, J. (1999). Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization,* Vol. 3, No. 4, 379–397.

Gero, J. (1998). Adaptive systems in designing: New analogies from genetics and developmental biology. *Adaptive computing in design and manufacture,* I. Parmee (Ed.). New York: Springer Verlag, 3–12.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences,* Vol. 8, No. 1, 156–166.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning.* Reading, MA: Addison-Wesley.

Goldberg, D. E., and Deb, K. (1991). A comparative study of selection schemes used in genetic algorithms. *Foundations of genetic algorithms,* G. J. E. Rawlins (Ed.). San Francisco: Morgan Kaufmann, 69–93.

Hamda, H., and Schoenauer, M. (2000). Adaptive techniques for evolutionary topological optimum design. *Evolutionary design and manufacture,* I. Parmee (Ed.). 123–136.

Hart, E., and Ross, P. (1998). A heuristic combination method for solving job-shop scheduling problems. *Proc. 5th Conference on Parallel Problems Solving from Nature,* T. Bäck, G. Eiben, M. Schoenauer, and H.-P. Schwefel (Eds.).

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Jones, T. (1995). Crossover, macromutation and population-based search. 73–80.

Kinnear, K. E., Jr. (Ed.) (1994). *Advances in genetic programming,* Cambridge, MA: The MIT Press.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural evolution.* Cambridge, MA: The MIT Press.

Koza, J. R., *et al.* (1999). *Genetic programming III: Automatic synthesis of analog circuits.* Cambridge, MA: The MIT Press.

Levine, D. (1997). An evolutionary approach to airline crew scheduling. *Handbook of evolutionary computation,* T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.). New York: Oxford University Press, G9.4:1–8.

Martin, S., Rivory, J., and Schoenauer, M. (1995). Synthesis of optical multi-layer systems using genetic algorithms. *Applied Optics,* Vol. 34, 2267.

McDonnell, J. R., Reynolds, R. G., and Fogel, D. B. (Eds.) (1995). *Proceedings of the Fourth Annual Conference on Evolutionary Programming.* Cambridge, MA: The MIT Press.

Merz, P., and Freisleben, B. (1999). Fitness landscapes and memetic algorithm design. *New ideas in optimization,* D. Corne, M. Dorigo, and F. Glover (Eds.). London: McGraw-Hill, 245–260.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs,* 3rd ed. New York: Springer Verlag.

Miettinen, K., Mkel, M. M., Neittaanmki, P., and Périaux, J. (Eds.) (1999). *Evolutionary algorithms in engineering and computer science.* New York: John Wiley.

Miller, B. L., and Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation,* Vol. 4, No. 2, 113–132.

Mueller, S. D., *et al.* (2001). Evolution strategies for film cooling optimization. *AIAA Journal,* Vol. 39, No. 3.

Nissen, V. (1997). Quadratic assignment. *Handbook of evolutionary computation,* T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.). New York: Oxford University Press, G9.10:1–8.

Nix, A. E., and Vose, M. D. (1992). Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence,* Vol. 5, No. 1, 79–88.

Obayashi, S. (1997). Pareto genetic algorithm for aerodynamic design using the Navier-Stokes equations. *Genetic algorithms and evolution strategies in engineering and computer sciences,* D. Quadraglia, J. Périaux, C. Poloni, and G. Winter (Eds.). New York: John Wiley, 245–266.

Oussedik, S., and Delahaye, D. (1998). Reduction of air traffic congestion by genetic algorithms. *Proc. 5th Conference on Parallel Problems Solving from Nature,* T. Bäck, G. Eiben, M. Schoe-

nauer, and H.-P. Schwefel (Eds.). New York: Springer Verlag, 885–894.

Paechter, B., Rankin, R., Cumming, A., and Fogarty, T. C. (1998). Timetabling the classes of an entire university with an evolutionary algorithm. *Proc. 5th Conference on Parallel Problems Solving from Nature*. T. Bäck, G. Eiben, M. Schoenauer, and H.-P. Schwefel (Eds.). New York: Springer Verlag.

Parmee, I. (Ed.) (1998). *Adaptive computing in design and manufacture*. New York: Springer Verlag.

Parmee, I. (Ed.) (2000). *Adaptive computing in design and manufacture—ACDM'2000*. New York: Springer Verlag.

Périaux, J., and Winter, G. (Eds.) (1995). *Genetic algorithms in engineering and computer sciences*. New York: John Wiley.

Quadraglia, D., Périaux, J., Poloni, C., and Winter, G. (Eds.) (1997). *Genetic algorithms and evolution strategies in engineering and computer sciences*. New York: John Wiley.

Radcliffe, N. J. (1991). Equivalence class analysis of genetic algorithms. *Complex Systems,* Vol. 5, 183–220.

Rechenberg, I. (1973). *Evolutionstrategie: Optimierung technisher systeme nach prinzipien des biologischen evolution*. Stuttgart: Fromman-Holzboog Verlag.

Rosenman, M. (1999). Evolutionary case-based design. *Proc. Artificial Evolution'99,* New York: Springer-Verlag, 53–72.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithm. *IEEE Transactions on Neural Networks,* Vol. 5, No. 1, 96–101.

Rudolph, G. (1994). Convergence of non-elitist strategies. *Proc. First IEEE International Conference on Evolutionary Computation,* Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano (Eds.). New York: IEEE Press, 63–66.

Schaffer, J. D. (Ed.) (1989). *Proceedings of the Third International Conference on Genetic Algorithms*. San Francisco: Morgan Kaufmann.

Schwefel, H.-P. (1995). *Numerical optimization of computer models,* 2nd ed. New York: John Wiley & Sons.

Syswerda, G. (1991). A study of reproduction in generational and steady state genetic algorithm. *Foundations of genetic algorithms,* G. J. E. Rawlins (Ed.). San Francisco: Morgan Kaufmann, 94–101.

Törn, A., and Zilinskas, A. (1989). *Global optimization*. New York: Springer-Verlag.

Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. *Proc. 3rd International Conference on Genetic Algorithms,* J. D. Schaffer (Ed.). San Francisco: Morgan Kaufmann, 116–121.

Whitley, D., Scott, V. S., and Böhm, P. W. (1997). Knapsack problems. *Handbook of evolutionary computation,* T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.). New York: Oxford University Press, G9.7:1–7.

Zalzala, A. (Ed.) (1995). *Genetic algorithms in engineering systems: Innovations and applications*. London: IEE.

Zalzala, A. (Ed.) (1995). *Second conference on genetic algorithms in engineering systems: Innovations and applications*. London: IEE.

Zitzler, E., Deb, K., Thiele, L., Corne, D., and Coello, C. (Eds.) (2001). *Proceedings of Evolutionary Multi-Critrion Optimization '01*. New York: Springer Verlag.

Executive Information Systems

Raymond McLeod, Jr.
University of Texas

GLOSSARY

critical success factor (CSF) A key activity that can contribute in a substantial way to the success or failure of an organization or activity.
data manager A person who is expert in certain types of data. The executive's data displays identify the data managers so that they can be contacted when help is needed.
drill down The ability of a user to first bring up a summary display and then cause the system to successively bring up more detailed displays.
executive A manager on the upper level of the organizational hierarchy who exercises a strong influence on the firm.
executive database Data and information that have typically been preprocessed by the firm's central computer and downloaded to the executive workstation.
executive information system (EIS); executive support system (ESS) A system that provides information to the executive on the overall performance of the firm. The information can be easily retrieved and can provide varying levels of detail.
executive sponsor A top-level executive, preferably the CEO, who serves as the driving force behind an EIS by encouraging its support during the development period.
executive workstation A personal computer that is usually networked to the firm's central computer. The workstation configuration includes secondary storage that houses the executive database.
management by exception (MBE) A management technique based on the idea that managers should direct their attention at only those activities that are going significantly better or worse than planned.
mental model An image in the manager's mind that enables him or her to understand phenomena and to experience events by proxy.
operating sponsor The person who works with both the using executives and the information specialists to ensure that the development of an EIS goes as planned.

I. INTRODUCTION

An *executive information system (EIS)* is a system that provides information to the executive on the overall performance of the firm. The information can be easily retrieved and can provide varying levels of detail. The term *executive support system (ESS)* is also used.

The popularity of the EIS concept peaked in the early 1990s, and many firms have implemented successful systems. After paying attention to the information needs of their executives, many firms shifted their focus to the support of other organizational groups. Neiman Marcus, a Texas-based retailer, for example, has directed its systems development efforts at the operational level—to systems that provide information to salesclerks as they interact with customers.

Although the interest in EIS as a systems objective may have passed it prime, the concept still has value. Many of the innovations developed for executives now

are being made available to lower level managers and other users. The EIS represents the ultimate in ease of use, and that characteristic is desirable for all systems.

II. WHAT IS AN EXECUTIVE AND WHAT DO EXECUTIVES DO?

The term *executive* is used to identify managers on the upper level of the organizational hierarchy who exercise a strong influence on the firm. The influence is gained by virtue of their hierarchical role, which enables them to engage in strategic planning and set policies. In addition, executives can often be distinguished from managers on lower levels by their attitude. Executives assign a higher value to the welfare of the firm than to the welfare of individual units within the firm. Managers on lower levels often place the welfare of their business areas ahead of that of the firm.

Additional insight into the activities of the executive can be gained by examining the contributions made by management theorists Henri Fayol and Henry Mintzberg.

A. Fayol's Management Functions

Writing in the early 1900s, French management theorist Henri Fayol presented his belief that all managers perform the same management functions: plan, organize, staff, direct, and control. The widespread belief is that planning is emphasized most on the executive level. That is the reason why Robert Anthony used the term *strategic planning system* for the activity of managers at the top. Executives develop the organization's strategic plans and then leave the implementation of those plans to managers on middle and lower levels. As the implementation process proceeds, the executives can exercise control to ensure that the plans are carried out.

B. Mintzberg's Managerial Roles

The executive's duties can also be defined in terms of Henry Mintzberg's managerial roles. Using data gathered from a study in the early 1970s of the activities of five CEOs, Mintzberg identified 10 roles and grouped them in three categories, as shown in Table I. He believes that all managers perform all roles, but the orientation is different on each management level.

Mintzberg found that the CEOs did not spend equal amounts of time in discharging the decisional roles. They concentrated on making long-range, entrepreneurial improvements to the firm and responding to unanticipated disturbances, while leaving much of the resource allocating and negotiating to managers on lower levels.

Table I Mintzberg's Managerial Roles

Interpersonal roles	Informational roles	Decisional roles
Figurehead	Monitor	Entrepreneur
Leader	Disseminator	Disturbance handler
Liaison	Spokesperson	Resource allocator
		Negotiator

III. HOW DO EXECUTIVES SOLVE PROBLEMS?

Most of the research involving executives has focused on observable behavior. An exception is the work of Harvard professor Daniel J. Isenberg, who studied the thought processes of more than a dozen executives over a 2-year period. Isenberg found that executives think about two general classes of problems: how to get things done and how to deal with a few overriding concerns or general goals. In thinking about how to get things done, executives are more concerned with the organizational and personal issues in getting subordinates to solve problems than with the specific solution. Although executives may face a large number of issues or concerns at any one time, they tend to be preoccupied by only a few. Executives commonly have particular agendas that they continually promote within their firms.

In solving problems, Isenberg observed that an executive will often skip from problem definition forward to solution implementation and then back to alternative evaluation. He concluded that executives do make rational decisions, but that the decisions might not always come as the result of following a series of well-defined steps in the same order.

Isenberg also concluded that executives use intuition at each step of the problem-solving process. If intuition does play a more important role at the executive level than any other, it is because of the unstructured nature of the problems and also the vast reservoir of experience that executives can apply.

IV. WHAT INFORMATION DO EXECUTIVES USE?

A number of studies have been conducted on executives' use of information. Henry Mintzberg's findings were the first to receive widespread dissemination and subsequently have been supplemented by those of Jones and McLeod and of Rockart and Treacy.

A. The Mintzberg Study

In his study of CEOs, Mintzberg identified five basic activities that accounted for their time: desk work, telephone calls, unscheduled meetings, scheduled meetings, and tours. At the time of his study, computer use at the executive level was rare, and he did not specifically refer to computer output, lumping all written media into a "documents" category. Rather than present the computer as an important information conduit, he emphasized the role of informal systems that communicate oral information.

B. The Jones and McLeod Study

In 1983, Jack W. Jones and Raymond McLeod, Jr., conducted a study of the incoming information flows of five executives. The executives included two CEOs, a president, and two vice presidents. The study findings provide insight into basic information characteristics at the executive level.

1. Volume of Executive Information

During a 2-week period, the executives and their secretaries logged 1454 information transactions that flowed to the executives. A transaction is a communication involving any medium—written or oral.

The executives received an average of 29 information transactions per day. The volume varied from one executive to another, and fluctuated from one day to the next for the same executive. The two CEOs had the highest average volume (40 and 41 per day), and the two vice presidents had the lowest (14 and 22 per day), with the president in between (28 per day).

2. Value of Executive Information

At the end of each day, the executives assigned a value ranging from 0 (no value) to 10 (maximum value) to each transaction. The executives gave 26% of the transactions very low values—0 (the most frequently assigned value), 1, or 2. At the other extreme, the executives gave only 6% of the transactions a 9 or a 10.

The values also varied from one executive to another, ranging from an average of 2.9 for one of the vice presidents to 5.5 for one of the CEOs. The two vice presidents had the lowest averages, but the small sample size prohibits any conclusions involving the influence of management hierarchy on perceived value of information.

3. Sources of Executive Information

Table II lists the volume percentage and average value for each information source. Committees provided the lowest volume but the information had the highest value. Conversely, the firm's environment provided the largest volume, but it also provided the information with the lowest average value. The two levels immediately below the executives provide a good balance in terms of both high volume and value.

4. Executive Information Media

Table III lists the media and their average values. As predicted by Mintzberg, executives do prefer oral media—occupying the top four positions. Telephone calls and business meals are the only oral media outranked by written media.

The term *computer reports* was used for all computer output. At the time of the study, none of the executives were querying their organization's databases or engaging in mathematical modeling. The executives were receiving only computer reports.

In evaluating the role of computer reports in an executive's information system, note that much of the information that is communicated to the executive in face-to-face conversation, telephone calls, letters, memos, and the like could be based on information that the source derived from computer reports. Therefore, the relatively low ranking of computer reports as an information medium is not an indication of the inability of the computer to generate important executive information. The computer output is simply being communicated to the executive by someone else in many cases.

5. Uses of Executive Information

The researchers, assisted by the executives, assigned one of Mintzberg's decisional roles to each information

Table II Information Volume and Value by Source

Source	Volume percentage	Average value
Committees	0.02	7.5
Upper levels	0.05	5.2
One level down	0.20	5.2
Two levels down	0.10	5.3
Three levels down	0.06	4.3
Four levels down	0.02	4.4
Internal units/ individuals	0.13	4.6
Environment	0.43	3.8

Table III Information Media Values

Mode	Medium	Average value
Oral	Scheduled meetings	7.4
Oral	Unscheduled meetings	6.2
Oral	Tours	5.3
Oral	Social activity	5.0
Written	Memos	4.8
Written	Computer reports	4.7
Written	Noncomputer reports	4.7
Written	Letters	4.2
Oral	Telephone calls	3.7
Oral	Business meals	3.6
Written	Periodicals	3.1

transaction. The assignment reflected how the executive would likely use the information. Most of the information was intended for use in handling disturbances, being an entrepreneur, and allocating resources. Very little was earmarked for negotiation. Six percent of the transactions could not be linked with any role and were given low values. At the time of receipt, if the executive could not perceive how a piece of information could be used, the information was given a low value. It is possible that the perceived value subsequently changed, based on eventual use or lack of use.

6. The Main Findings

The study revealed three main findings:

1. Most of the executives' information came from environmental sources, but the internal information was valued higher.
2. Most of the executives' information came in a written form, but the oral information was valued higher.
3. The executives received very little information directly from the computer.

Like the Mintzberg study, the Jones–McLeod research was conducted relatively early in the era of computer-based information systems. Computer use no doubt has increased significantly since the early studies, but certain inherent preferences for information sources and media most likely still persist. Executives like to obtain information in a face-to-face manner, using informal systems, and no executive information system is likely to be completely computer based. Rather, the computer will be integrated into noncomputer flows.

C. The Rockart and Treacy Study

In the early 1980s, two MIT researchers, John Rockart and Michael Treacy, identified a handful of executives who were using the computer to generate information. Sixteen companies were involved and at least one of the top three officers in each company, most often the CEO, was a user. One of the most dedicated was Ben W. Heineman, CEO of Northwest Industries. Heineman had a terminal in his office, one at home, and took one with him on vacations.

Whereas the guiding principle of information systems design has always been to provide the user with information and not data, Heineman recognized a huge advantage in working directly with the data. Another executive concurred, recommending that you "muck around" in the data in order to be able to know enough about a topic to ask the right questions.

Rockart and Treacy coined the term *executive information system* and recognized that such systems feature:

- *A central purpose.* Executives employ computer information primarily in performing Fayol's management functions of planning and controlling.
- *A common core of data.* The database is able to shed light on internal operations of the firm and on such elements in the firm's environment as industries, customers, vendors, and competitors. The data reflect three time periods—past, present, and future.
- *Two principal methods of use.* Executives use the EIS to remain current on what is happening now and what is likely to happen in the future, and to conduct personalized analyses of the data.
- *A support organization.* The executive relies on a support staff in the construction and use of the EIS. The term *EIS coach* has been used to describe a member of the executive's staff, the firm's information services unit, or an outside consulting organization who provides help in setting up the system. The term *EIS chauffeur* has been used to describe a member of the executive's staff who operates the equipment for the executive.

Whereas the Mintzberg study and the Jones–McLeod study addressed executives, their activities, and how they used information in a general sense, Rockart and Treacy set out to portray the computer as an important problem-solving tool at the executive level. Up until that time, the computer was viewed as being effective in solving only middle and lower management level problems. The Rockart and Treacy executives were living proof that the computer should be no stranger in the executive office.

During the years since the Rockart and Treacy study, much thought and effort has gone into the design of a computer-based executive information system. Executives who have analytical decisional styles have typically adopted EIS to a greater extent than have those with conceptual or behavioral styles. Also, executives who operate under severe time pressures are more likely to incorporate EIS into their daily activities.

V. COMPUTER-BASED EXECUTIVE INFORMATION SYSTEMS

The equipment configuration of a computer-based EIS usually includes a personal computer, which serves as an executive workstation. In very small firms with minimal computer resources, the PC may be a standalone device. However, the typical configuration calls for the PC to be networked to a larger system, such as a mainframe or a server. The workstation configuration includes secondary storage that houses the executive database. The executive database contains data and information that have typically been preprocessed by the firm's central computer and downloaded to the workstation. The downloading occurs frequently, perhaps several times a day as important data values change. The executive selects from menus to produce preformatted screen displays or to perform a minimum amount of processing. The system also permits use of the firm's electronic mail system and access to environmental data and information.

The system can be used for scanning or browsing, or for focused research. When browsing, the EIS helps the executive formulate problems, it fosters creativity, and the result is improvement in organizational effectiveness. When using the EIS for focused research, the system enables the executive to fine-tune operations, making the organization more efficient.

A. Theoretical Concepts

Executives have built their computer-based EISs on fundamental management concepts. Three stand out as providing key capabilities. They are critical success factors, management by exception, and mental models.

1. Critical Success Factors

A *critical success factor (CSF)* is a key activity that can contribute in a substantial way to success or failure for an organization. A firm or an industry will ordinarily be guided by several CSFs. D. Ronald Daniel, a management consultant, is credited with the concept, which was applied to EISs by John Rockart.

Executives who embrace the critical success factors concept use their EISs to monitor each of the CSFs, thereby maintaining their focus on activities of most importance.

2. Management by Exception

The concept of *management by exception (MBE)* recognizes that managers are too busy to become involved in everything. Instead, they should direct their attention toward only those activities that are going significantly better or worse than planned. The MBE concept can be traced to Frederick W. Taylor, who is known as the father of scientific management. Unlike the activities that are identified as CSFs and do not frequently change, the activities that are monitored by MBE can change frequently.

When managers practice MBE, they establish a zone of acceptable performance for an activity, and when actual performance falls outside the zone, the managers take action. The screen displays that executives retrieve frequently incorporate this concept by comparing budgeted performance with actual performance. The MBE capability relieves the executive of the need to devote unnecessary time to the monitoring activity, a job that can be done by the EIS.

3. Mental Models

The primary role of the EIS is to synthesize, or distill, a large volume of data and information to increase its utility. This distillation has been called *information compression,* and it produces a view, or a mental model, of the firm's operations. P. N. Johnson-Laird coined the term *mental model* in his 1973 book, and he recognized that such models enable managers to understand phenomena and to experience events by proxy. The proxy capability is especially valuable since executives are seldom on the physical scene where activity occurs. Instead, they must monitor the performance of the physical system by means of information that is relayed to them. The EIS can serve as a form of mental model for managers on the upper organizational level.

B. EIS Implementation Decisions

When an organization considers the development of a computer-based system to support its executives, a series of key decisions must be made. First, the firm must decide whether it will develop an EIS. When the idea is rejected, the executives continue to rely on their present systems. When the decision is to proceed, the next questions relate to the source or sources of the

software. If prewritten software is available to meet the executives' needs, it is purchased since the cost of prewritten software is invariably less than that of custom-developed software. Two basic types of prewritten software have been used for the EIS: personal productivity software and prewritten EIS software.

1. Personal Productivity Software

Personal productivity software is general-purpose software that anyone can use to develop her or his own applications. Examples are database management systems, electronic spreadsheet packages, graphics packages, and project management systems. If this type of software is acceptable, then it is the best option, because it is the least expensive. The main limitation is that the executives may perceive it as not being user friendly enough or not keyed to their special needs.

2. Prewritten EIS Software

When the firm decides not to go the personal productivity software route, the next option is prewritten EIS software, which is specially designed to meet the information needs of executives. The first examples of EIS software were designed for mainframe systems, and the pioneers were Pilot Executive Software, Inc., of Boston, and Comshare, Inc., of Ann Arbor, Michigan. Today, prewritten EIS software is available for all sizes of computers, with most of the packages aimed at microcomputer users.

Prewritten EIS software offers three major advantages. First, it enables the firm to get a system up and running quickly. Second, the EIS implementation project does not put as much of a burden on the firm's information services staff as when they must develop the EIS. Third, EIS software is specifically intended for the executive and offers a good chance of being used. A disadvantage could be the difficulty of tailoring the system to an executive's particular needs. Invariably, a certain amount of customizing needs to be done.

3. Custom EIS Software

Custom software is prepared by the firm's information services unit or by such an outside organization as an outsourcer. One of the most highly publicized examples of custom EIS software was the MIDS (Management Information and Decision Support) system implemented by Lockheed-Georgia.

The designers of MIDS built in one feature that could be effective in any EIS. This was the recognition that the executive may not be expert in the data that are used to produce the information and may therefore need help in interpreting the displays. To solve this problem, Lockheed-Georgia designated certain persons, called data managers, to be expert in certain types of data. The executive displays of the data identified these data managers so that they could be contacted when help was needed. The data manager role was considered to be so important that when one was unavailable for any reason, a substitute was identified.

VI. EIS CRITICAL SUCCESS FACTORS

Not only do EISs enable executives to manage by means of critical success factors, it is possible to use a special set of CSFs to manage EIS development projects.

John F. Rockart and David W. DeLong identified eight critical success factors for successfully implementing an EIS:

1. *A committed and informed executive sponsor.* A top-level executive, preferably the CEO, should serve as the executive sponsor of the EIS by encouraging its support during the development period. The most successful EIS efforts have been those where the driving force was a top executive. At Gillette, the president of the company's North American unit inaugurated the EIS. At Lockheed-Georgia, the system came at the urging of the president.
2. *Operating sponsor.* The executive sponsor is usually too busy to fully commit her or his time to implementation. The responsibility for daily project monitoring should be given to another top-level executive, such as the executive vice president. The operating sponsor works with both the using executives and the information specialists to ensure that the development work goes as planned.
3. *Appropriate information services staff.* Information specialists should be available who understand not only the information technology but also how the system will be used. The areas of information technology that are especially applicable include the Internet, data communications, database, and graphical user interfaces.
4. *Information technology.* Although the costs of the hardware and software continue to fall, they could still represent constraints for some organizations. Developers should ensure that system expectations do not outstrip the firm's technology capabilities and that the required tools are available. By the same token, developers should not build the system on unnecessary technology.

5. *Data management.* It is not sufficient to just display the data or information. The executive should have some idea of how current the data are. This can be accomplished by identifying the day and, ideally, the time of day that the data were entered into the system. The executive should also be able to pursue a data analysis, with the assistance of a data manager when needed.
6. *A clear link to business objectives.* Even though the EIS can, and should, be tailored to individual executive information needs, a good starting point when developing each executive's system would be the firm's strategic objectives. For each user, the developers should identify the information needed to manage the efforts required to meet the objectives.
7. *Management of organizational resistance.* In the event that an executive resists the EIS, efforts should be taken to win that person's support. A good strategy is to identify a single problem that the executive faces and then quickly implement an EIS to be used in problem solution. Through the use of such methodologies as prototyping, the executive can be presented with a working system within a few hours or days at the most. Care must be taken to select a problem that will enable the EIS to make a good showing. Then, additional applications can be added.
8. *Management of the spread and evolution of the system.* Experience has shown that when upper level managers begin receiving information from the EIS, the lower level managers reporting directly to them want to receive the same information. The lower level managers want to be able to anticipate problems and solve them before the upper level managers even know that the problems exist. EIS use therefore follows a trickle-down pattern. However, care must be taken to add users only when they can be given the attention that they need. One reason for the success of the EIS concept has been the high level of user support that is incorporated into system development. Care must be taken not to spread such support too thin when dealing with lower level users.

These CSFs have a definite top-down flavor, recognizing the need for top-level support and linkage to business objectives. They also recognize the importance of a good information services staff and technology resources. Within such a setting, attention can be paid to both enthusiastic and unenthusiastic users to maximize the opportunity for successful implementation.

VII. FUTURE TRENDS IN THE COMPUTER-BASED EIS

The top management level was the last frontier in the organizational hierarchy in terms of computer adoption. But, once the ice was broken, the same enthusiasm that had been shown on lower levels quickly spread. Whereas the Ben Heinemans of the early 1980s were rare, today's executive does not feel out of place being a computer user. As support for the EIS continues to increase, certain forces can be expected to exert both stimulating and restraining influences on system use.

A. Computer Use by Executives Will Continue to Increase

Computer use by executives is related to a complex set of issues that includes the executives' perceptions of their roles, their organizational cultures, and the demands and constraints placed on them as they perform their tasks. When the executives' initial encounters with the computer have been proactive rather than reactive, and when the executives have made the computer a part of their modus operandi, the usage has a tendency to remain stable or experience a pattern of continual growth.

B. EIS Outputs Will Rely Increasingly on Multimedia

The text-based and graphics outputs of early EISs are being supplemented by sound, animation, and motion. These additions improve the effectiveness of the systems but also add to their complexity. The objective of incorporating multimedia outputs is to increase the ability of the user to analyze information. However, research has not provided support for the objective being achieved in all cases. More study is required to determine ways to tailor EIS media to user needs.

C. Data Mining and Online Analytic Processing Will Facilitate Improved Data Analysis

Breakthroughs in data analysis techniques are being incorporated in EISs to enable users to utilize vast data resources in innovative ways.

Through the incorporation of artificial intelligence, data mining is enabling executives to identify

previously unidentified patterns in the data contained in their data warehouses and data marts.

Another technique, called online analytic processing (OLAP), is enabling executives to address business problems that involve multiple dimensions. By using OLAP hardware and software, executives can slice, aggregate, and summarize huge volumes of data to identify new trends, isolate problems, and discover potential business opportunities. One capability that OLAP provides is *drill down*. A user first brings up a summary display and is then able to click on particular values to cause the system to "drill down" to a lower level and bring up a more detailed display. This top-down process can continue until the user is presented with the most detailed data in the system.

Both data mining and OLAP enable the incorporation of an intelligent software agent in the EIS, enabling the system to monitor the data resource and notify the executive of changes as they occur.

D. Distributed EISs Will Be Achieved Using the Internet

Since the emergence of group decision support systems, awareness of the need for computer-based systems to support collaborative decision making has been increasing. Distributed decision making consists of coordinated decision making among communicating executives who each possess specialized knowledge and can process the knowledge in making decisions. A distributed EIS allows the sharing of resources, and an intelligent agent can pool the resources to produce a synergistic effect.

A key component in such distributed systems is the Internet. It provides a ready-made network capable of such communications alternatives as audio and video conferencing. The Internet search engines are not only easy to use but also provide a filtering capability. More and more EISs will become Internet based, so as to take advantage of this powerful network capability.

VIII. SUMMARY

Although many executives today use the computer, there are most likely proportionally fewer heavy computer users on the executive level than any other organizational level. If this situation is indeed true, it is probably caused by the problem-solving nature of the executive's tasks rather than by characteristics of the executives. In the early years of the computer, the older ages of the executives and the fact that they missed out on formal computer training was given as a reason for a low level of use. That appraisal no longer applies. If the executive emphasizes rational problem solving and recognizes the potential contribution of the computer to the process, then he or she likely will make use of computer information.

The important points relating computer use to executives are that such use is a very personal thing, and computer-generated information is only one part of all of the information reaching an executive. All executives want to receive good information from any source. Heineman, as strong a computer advocate as one could ever expect to find on the executive level, expressed this feeling when he said that he did not want to be a captive of any particular source of information.

SEE ALSO THE FOLLOWING ARTICLES

Decision Support Systems • Globalization and Information Management Strategy • Goal Programming • Group Support Systems • Management Information Systems • Project Management Techniques • Strategic Planning • Total Quality Management and Quality Control

BIBLIOGRAPHY

Chi, R. T., and Turban, E. (June 1995). Distributed intelligent executive information systems. *Decision Support Systems,* Vol. 14, No. 2, 117–130.

Elam, J. J., and Leidner, D. G. (June 1995). EIS adoption, use, and impact: The executive perspective. *Decision Support Systems,* Vol. 14, No. 2, 89–103.

Gopal, R. D., and Tung, Y. A. (Spring 1999). An examination of the role of Internet technologies in developing next generational executive information systems. *Journal of Computer Information Systems,* Vol. 39, No. 3, 81–91.

Huang, A. H., and Windsor, J. C. (1998). An empirical assessment of a multimedia executive support system. *Information & Management,* Vol. 33, 251–262.

Isenberg, D. J. (November–December 1984). How senior managers think. *Harvard Business Review,* 81–90.

Jones, J. W., and McLeod, R. Jr. (Spring 1986). The structure of executive information systems: An exploratory analysis. *Decision Sciences,* 220–249.

McLeod, R., Jr., and Schell, G. P. (2001). *Management information systems,* 8th ed. Upper Saddle River, NJ: Prentice Hall.

Rockart, J. F., and Treacy, M. E. (January–February 1982). The CEO goes on-line. *Harvard Business Review,* 82–88.

Seeley, M., and Targett, D. (1999). Patterns of senior executives' personal use of computers. *Information and Management,* Vol. 35, 315–330.

Vandenbosch, B., and Huff, S. L. (March 1997). Searching and scanning: How executives obtain information from executive information systems. *MIS Quarterly,* Vol. 21, No. 1, 81–102.

Expert Systems

Jay E. Aronson
University of Georgia

GLOSSARY

artificial intelligence (AI) The subfield of computer science that is concerned with symbolic reasoning and problem solving.
expert A human being who has developed a high-level of proficiency in making judgments in a specific, usually narrow, domain.
expert system shell A computer program that facilitates the relatively easy implementation of a specific expert system. Similar to the concept of a decision support system (DSS) generator.
inference engine The expert system component that performs reasoning.
knowledge Understanding, awareness, or familiarity acquired through education or experience. Anything that has been learned, perceived, discovered, inferred, or understood.
knowledge acquisition The extraction and formulation of knowledge derived from various sources, especially from experts.
knowledge base A collection of facts, rules, and procedures organized into schemas. The assembly of all of the information and knowledge of a specific field of interest.
knowledge engineering The engineering discipline through which knowledge is integrated into computer systems to solve complex problems normally requiring a high level of human expertise.
knowledge representation A formalism for representing facts and rules about a subject or a specialty.
production rule A knowledge representation method in which knowledge is formalized into rules containing an IF part and a THEN part.

AN EXPERT SYSTEM (ES) is a system that uses human knowledge captured in a computer to solve problems that ordinarily require human expertise. The term expert system was derived from the term *knowledge-based expert system*. ESs imitate the reasoning processes of experts in solving specific problems. The main purpose of an ES is to provide expertise to a novice user, who can then exhibit expert-level performance in their decision making. ES propagate scarce knowledge resources for improved, consistent results. These systems can also be used by experts as knowledgeable assistants. As the knowledge in an ES is improved and becomes more accurate, the system may eventually function at a higher level than any single human expert in making judgements in a specific, usually narrow, area of expertise (referred to as a domain).

I. INTRODUCTION

Expert systems are part of the general field of *artificial intelligence (AI)*. They use a symbolic approach to

representing knowledge, and simulate the process that experts use when they solve problems. To understand what ES are and how they work, we must first introduce some basic concepts on expertise and what makes an expert an expert. Since one goal of an ES is to capture knowledge, we discuss the process of knowledge acquisition. Knowledge must be represented, typically as production rules, but there are other representations, and each problem has a natural fit with one or more knowledge representations. To be useful, knowledge must be acted upon, which is the function of the inference engine—the brain. The structure of ES is important, as are the application areas to which ES have been successfully applied. Factors that make an expert system succeed or fail must be understood by application developers.

II. EXPERTISE AND EXPERTS

A. Expertise

Expertise is the extensive, task-specific knowledge acquired from training, reading, and experience. Here are examples of the types of knowledge included in expertise:

- Theories about the problem area
- Rules and procedures regarding the general problem area
- Rules (heuristics) of what to do in a given problem situation
- Global strategies for solving these types of problems
- Metaknowledge (knowledge about knowledge)
- Facts about the problem area

These knowledge types enable experts to make better and faster decisions than nonexperts in solving complex problems. It takes a long time (usually several years) to become an expert, and novices become experts only incrementally.

Some concepts about expertise include:

- Expertise is usually associated with a high degree of intelligence, but is not always associated with the smartest person.
- Expertise is usually associated with a vast quantity of knowledge.
- Experts learn from past successes and mistakes.
- Expert knowledge is well-stored, organized, and retrievable quickly from an expert.
- Experts can call up patterns from their experience (excellent recall).

B. Experts

It is difficult to define what an *expert* is because there are degrees or levels of expertise. It is believed that the top tenth of experts in any given area can perform 3 times as well as the average experts and 30 times as well as the lowest tenth experts. This distribution suggests that the overall effectiveness of human expertise can be significantly increased (up to 200%), if we can somehow make top-level expertise available to less knowledgeable decision makers.

Typically, human expertise includes behavior that involves the following activities:

- Recognizing and formulating the problem
- Solving the problem quickly and properly
- Explaining the solution
- Learning from experience
- Restructuring knowledge
- Breaking rules if necessary
- Determining relevance
- Degrading gracefully (awareness of limitation)

To mimic a human expert, it is necessary to build a computer system that exhibits all these characteristics. To date, work in ES has primarily explored the second and third of these activities. In addition, ES generally can estimate its measure of confidence in its solutions just like an expert.

C. Transferring Expertise

The objective of an ES is to transfer expertise from an expert to a computer system and then on to other nonexpert humans. This process involves: knowledge acquisition (from experts or other sources), knowledge representation, knowledge inferencing, and knowledge transfer to the user. The knowledge is stored in a component called a knowledge base, and reasoned with by the inference engine component.

III. KNOWLEDGE ACQUISITION

Knowledge acquisition is the process of extracting, structuring, and organizing knowledge from one or more sources, and its transfer to the *knowledge base* and sometimes to the inference engine. This process has been identified by many researchers and practitioners as a major bottleneck. Acquisition is actually done throughout the entire development process. Formally, *knowledge* is a collection of specialized facts, procedures, and judgment rules.

A knowledge engineer is responsible for formally applying AI methods directly to difficult applications normally requiring expertise. He is responsible for building complex computer programs that can reason.

Knowledge engineering deals with knowledge acquisition, representation, validation, inferencing, explanation, and maintenance. Knowledge engineering involves the cooperation of human experts in codifying and making the rules (or other procedures) that a human expert uses to solve real problems explicit.

A. The Knowledge Engineering Process

The knowledge engineering process includes five major activities:

1. *Knowledge acquisition.* Knowledge acquisition involves the acquisition of knowledge from human experts, books, documents, sensors, or computer files.
2. *Knowledge validation.* The knowledge is validated and verified (for example, by using test cases) until its quality is acceptable. Test case results are usually shown to the expert to verify the accuracy of the ES.
3. *Knowledge representation.* The acquired knowledge is organized into a knowledge representation. This activity involves preparation of a knowledge map and encoding the knowledge in the knowledge base.
4. *Inferencing.* This activity involves the design of software to enable the computer to make inferences based on the knowledge and the specifics of a problem. Then the system can provide advice to a nonexpert user.
5. *Explanation and justification.* This involves the design and programming of an explanation capability to answer questions like *why* a specific piece of information is needed or *how* a certain conclusion was obtained.

The most common method for eliciting knowledge from an expert is through interviews. The knowledge engineer interviews one or more experts and develops a vocabulary and an understanding of the problem domain. Then, he attempts to identify an appropriate knowledge representation and inferencing (reasoning) approach. The interviewing may take place over several weeks or even years.

There are several automatic and semiautomatic knowledge acquisition methods, especially ones for inducing rules directly from databases and text (for example, Knowledge Seeker from Angoss Software Systems).

IV. KNOWLEDGE REPRESENTATION

Once knowledge is acquired, it must be organized into a knowledge base. A good knowledge representation is one that naturally represents the problem domain. A variety of useful *knowledge representation* schemes have been developed over the years. The major knowledge representation schemas are production rules and frames. Other methods include decision tables, decision trees, O–A–V (object–attribute–value) triplets, semantic networks, scripts, and lists.

There are many tools for modeling knowledge. They include cognitive maps, also known as induction tables or knowledge tables. These help the knowledge engineer identify knowledge and focus on the problem at hand. Rules are developed directly in the table.

A. Production Rules

Many commercial ES tools and ready-made systems are rule-based systems. The knowledge is stored in the form of rules, as are the problem-solving procedures. For example, a rule may state: "*IF* the engine is idle, and the fuel pressure is less than 40 psi, *AND* the gauge is accurate, *THEN* there is a defect in the fuel pump."

Knowledge is presented as *production rules* in the form of condition–action pairs: "IF this *condition* occurs, THEN some action will (or should) occur." For example: IF the stoplight is red AND there is no sign indicating "No turn on Red" AND you have stopped, THEN a right turn is okay.

Each production rule in a knowledge base implements an autonomous chunk of expertise that can be developed and modified independently of other rules. When combined and fed to the inference engine, the set of rules behaves synergistically. Rules are the most common form of knowledge representation for ES because they are easy to understand, and they naturally model many situations.

B. Frames

A frame is a data structure that includes all the knowledge about a particular object. This knowledge is organized in a special hierarchical structure that permits a diagnosis of knowledge independence. Frames

are basically an application of object-oriented programming and are used extensively in ES.

Frames, as in *frames of reference,* provide a concise structural representation of knowledge in a natural manner. In contrast to other representation methods, the values that describe one object are grouped together into a single unit called a frame. Thus, a frame encompasses complex objects, entire situations, or a management problem as a single entity. Frames are often used in case-based reasoning.

C. Decision Tables

In a decision table (or induction table), knowledge is organized in a spreadsheet format, using columns and rows. The table is divided into two parts. First, a list of attributes is developed, and for each attribute all possible values are listed. Then, a list of conclusions is developed. Finally, the different configurations of attributes are matched against the conclusion.

Knowledge for the table is collected in knowledge acquisition sessions. Once constructed, the knowledge in the table can be used as input to other knowledge representation methods. It is not possible to make inferences with the domain tables by themselves, except when rule induction is used. Often rules are developed in which one of the factors is a conclusion of another rule. This calls for another table, which leads to the concept of knowledge chains. Decision tables are easy to understand and program.

D. Decision Trees

Decision trees are related to tables and often are used in system analysis. The trees are similar to the decision trees used in decision theory. They are composed of nodes representing goals and links representing decisions.

E. O–A–V Triplet

Another way to represent knowledge is to use objects, attributes and values—the O–A–V triplet. *Objects* may be physical or conceptual. *Attributes* are the characteristics of the objects. *Values* are the specific measures of the attributes in a given situation. O–A–V triplets are used in both frame and semantic network representations.

F. Semantic Networks

Semantic networks are basically graphic depictions of knowledge composed of *nodes* and *links* that show hierarchical relationships between objects. The nodes are interconnected by links or arcs. These arcs show the relationships between the various objects and descriptive factors. Some of the most common arcs are of the *is-a* or *has-a* type. *Is-a* is used to show class relationship; that is, that an object belongs to a larger class or category of objects. *Has-a* links are used to identify characteristics or attributes of the object nodes. Other arcs are used for definitional purposes. Semantic networks can show inheritance. Semantic nets are a visual representation of relationships, and can be combined with other representations.

G. Scripts

A script is a knowledge representation scheme describing a *sequence of events.* Scripts are a particularly useful form of knowledge representation because there are so many stereotypical situations and events that people use every day. Scripts may be used in case-based reasoning.

H. Lists

A list is a written series of related items. Lists are normally used to represent hierarchical knowledge where objects are grouped, categorized, or graded according to rank or relationship. A list may be viewed as an outline.

I. Hybrid Knowledge Representations

No single knowledge representation method is ideally suited for all problems. When using several sources of knowledge simultaneously, the goal of uniformity may be sacrificed in favor of exploiting the benefits of multiple knowledge representations, each tailored to a different subtask. The necessity of translating among knowledge representations becomes a problem in these cases. Nevertheless, several ES shells use two or more knowledge representation schemes. Typically they are rules and frames.

V. INFERENCING AND EXPLANATION

A. Inferencing

A unique feature of an ES is its ability to reason (e.g., think; see Table I). Once the knowledge is represented in the knowledge base, or is at least at a sufficiently high level of accuracy, it is ready to be used. We need a computer program to access the knowledge for making inferences. This program is an algorithm that controls a reasoning process. The program is usually called the *inference engine* or the control program. In rule-based systems, it is also called the rule interpreter.

The inference engine directs the search through the knowledge base. The process may involve the application of inference rules by pattern matching. The control program decides which rule to investigate, which alternative to eliminate, and which attribute to match. The most popular control programs for rule-based systems are forward and backward chaining.

B. Reasoning with Logic

For performing either deductive or inductive reasoning, several basic reasoning procedures allow the manipulation of the logical expressions to create new expressions. The most important procedure is called *modus ponens*. In this procedure, given a rule "if A, then B," and a fact that A is true, then it is valid to conclude that B is also true. In logic terminology, we express this as $[A \text{ AND } (A \rightarrow B)] \rightarrow B$.

A and $(A \rightarrow B)$ are *propositions* in a knowledge base. Given this expression, we can replace both propositions with proposition B; i.e., we use *modus ponens* to draw the conclusion that B is true if the first two expressions are true.

A different situation is the inferring that A is false when B is known to be false. This is called *modus tollens*. Resolution (which combines substitution, *modus ponens,* and other logical syllogisms) is another approach.

C. Inferencing with Rules: Forward and Backward Chaining

Inferencing with rules involves implementation of *modus ponens,* which is reflected in the search mechanism. Consider the following example:

Rule 1:IF an international conflict begins,

THEN the price of gold increases.

Let us assume that the ES knows that an international conflict just started. This information is stored as a fact in the rule base (or assertion base). This means that the premise (IF side) of the rule is true. Using *modus ponens,* the conclusion is then accepted as true. We say that Rule 1 *fires*. Firing a rule occurs only when all of the rule's hypotheses (conditions in the IF part) are satisfied (evaluated to true). Then, the conclusion drawn is stored in the assertion base. In our case, the conclusion (the price of gold increases) is added to the assertion base, and it could be used to satisfy the premise of other rules. The true (or false) values for

Table I Reasoning Methods

Method	Description
Deductive reasoning	Move from a general principle to a specific inference
	The general principle consists of two or more premises
Inductive reasoning	Move from some established facts to draw general conclusions
Analogical reasoning	Derive an answer by known analogy: a verbalization of internalized learning process; use of similar, past experiences
Formal reasoning	Syntactic manipulation of data structures to deduce new facts following prescribed rules of inferences (e.g., predicate calculus)
Procedural (numeric) reasoning	Use of mathematical models or simulation (e.g., model-based reasoning, qualitative reasoning, and temporal reasoning: the ability to reason about the time relationships between events
Metalevel reasoning	Knowledge about what you know (e.g., about the importance and relevance of certain facts and rules)

either portion of the rules can be obtained by querying the user or by checking other rules. Testing a rule premise or conclusion can be as simple as matching a symbolic pattern in the rule to a similar pattern in the assertion base. This activity is called pattern matching.

Every rule in the knowledge base can be checked to see whether its premise or conclusion can be satisfied by previously made assertions. This process may be done in one of two directions, forward or backward, and will continue until no more rules can fire or until a goal is achieved.

D. Backward Chaining

Backward chaining is a *goal-driven* approach in which one starts from an expectation of what is to happen (hypothesis), then seek evidence that supports (or contradicts) your expectation. Often this entails formulating and testing intermediate hypotheses (or subhypotheses).

An ES starts with a goal to be verified as either true or false. Then it looks for a rule that has that goal in its *conclusion*. It then checks the *premise* of that rule in an attempt to satisfy this rule. It checks the assertion base first. If the search fails there, the program looks for another rule whose conclusion is the same as that of the first rule. An attempt is then made to satisfy the second rule. The process continues until all the possibilities that apply are checked or until the initially checked rule (with the goal) is satisfied. If the goal is proven false, then the next goal is tried. (In some inferencing, even if the goal is proven true, the rest of the goals may be tried in succession.)

E. Forward Chaining

Forward chaining is a *data-driven* approach. One starts with all available information and tries to reach conclusions.

The ES analyzes the problem by looking for the facts that match the IF portion of its IF–THEN rules. For example, if a certain machine is not working, the ES checks the electricity flow to the machine. As each rule is tested, the program works its way toward one or more conclusions.

F. Backward Chaining versus Forward Chaining

We have seen that an antecedent–consequence rule system can run forward or backward. Which one is better? The answer depends on the purpose of the reasoning and the shape of the search space. For example, if the goal is to discover all that can be deduced from a given set of facts, the system should run forward, as in accounting audit applications, because most facts are initially available in documents and forms. In some cases, the two strategies can be mixed (bidirectional).

The forward or backward chaining is performed by a rule interpreter within the inference engine. It examines production rules to determine which should be fired and then does the rule firing. The control strategy of the rule interpreter (e.g., backward chaining) determines how the appropriate rules are found and when to apply them.

G. Case-Based Reasoning

Case-based reasoning (CBR) attempts to adapt solutions that were used to solve old problems and use them to solve new problems. One variation of this approach is rule induction. In rule induction, the computer examines historical cases and generates rules, which then can be chained (forward or backward) to solve problems. Case-based reasoning, on the other hand, follows a different process:

- It finds the cases in memory that contains solved problems similar to the current problem.
- It adapts the previous solution or solutions to fit the current problem, taking into account any differences between the current and previous situations.

The process of finding relevant cases involves:

- Characterizing the input problem by assigning appropriate features to it
- Retrieving the cases from memory with those features
- Picking the case or cases that match the input best

Case-based reasoning is an extremely effective approach in complex cases. The basic justification for using CBR is that human thinking does not use logic. It basically processes the right information retrieved at the right time. So the central problem is the identification of pertinent information whenever needed. This is done in CBR with the aid of scripts.

H. Explanation

Human experts are often asked to explain their views, recommendations, or decisions. If ES are to mimic humans in performing highly specialized tasks, they need to justify and explain their actions as well. An explanation is an attempt by an ES to clarify its reasoning, recommendations, or other actions (such as asking a question). The part of an ES that provides explanations is called an explanation facility (or justifier). The explanation facility has several purposes:

- Make the system more intelligible to the user
- Uncover the shortcomings of the rules and knowledge base (debugging the systems by the knowledge engineer)
- Explain situations that were unanticipated by the user
- Satisfy psychological and social needs by helping a user feel more assured about the actions of the ES
- Clarify the assumptions underlying the system's operations, to both the user and the builder
- Conduct sensitivity analyses (using the explanation facility as a guide, the user can predict and test the effects of changes on the system)

In developing large ES, the need for a good explanation facility is essential. Explanation is an extremely important function because understanding depends on explanation, thus making implementation of proposed solutions easier. ES explanations can make a system's advice more acceptable to users. Furthermore, explanation is essential for ES used for training. Explanation in rule-based ES is usually associated with tracing the rules that are fired during the course of a problem-solving session. Most ES explanation facilities include the *why question* (when the ES asks the user for some information); advanced systems include the *how question* (how a certain conclusion or recommendation was reached).

The *why* and *how* explanations often show the rules as they were programmed and not in a natural language. Some systems have the ability to present the rules in a natural language. A journalistic explanation facility ideally includes the six key questions *who, what, where, when, why,* and *how.* Some sophisticated ES do provide some of the more advanced explanation capabilities.

VI. UNCERTAINTY

A key issue in ES is the fuzziness of the decision-making process. Typical problems have many qualitative aspects (the engine sounds *funny*), and often when a rule reaches a conclusion, the expert may feel it is only right about 7 times out of 10. Consequently uncertainty must be considered in ES. The major methods for handling uncertainty are Bayesian probabilities, theory of evidence, certainty factors, and fuzzy sets.

Certainty theory relies on the use of certainty factors. Certainty factors (CF) express belief in an event (or fact or hypothesis) based on evidence (or the expert's assessment), along a scale, say from 0 to 10, where 0 means false and 1 means true. These certainty factors are not probabilities. The certainty factor indicates how true a particular conclusion is.

VII. ES ENVIRONMENTS AND COMPONENTS

Expert systems environments include: the development environment and consultation (runtime) environment. The development environment is used by the ES builder to build the components and put knowledge into the knowledge base. The consultation environment is used by a nonexpert to obtain expert knowledge and advice. These environments can be separated once a system is completed.

The three major components that appear virtually in every ES are the knowledge base, inference engine, and user interface. An ES may contain the following additional components:

- Knowledge acquisition subsystem
- Blackboard (workplace)
- Explanation subsystem (justifier)
- Knowledge refining system
- User(s)

Currently, most ES do not contain the knowledge refinement component. A brief description of each component follows.

A. Knowledge Base

The knowledge base contains the relevant knowledge necessary for understanding, formulating, and solving problems. It includes two basic elements: first, facts such as the problem situation and theory of the problem area; and second, special heuristics or rules that direct the use of knowledge to solve specific problems in a particular domain. In addition, the inference engine may include general purpose problem-solving and decision-making rules.

B. Inference Engine

The inference engine is the brain of the ES. It is also known as the control structure or rule interpreter (in rule-based ES). This component is essentially a program that provides a methodology for reasoning about information in the knowledge base and on the blackboard, and for formulating conclusions. This component provides directions about how to use the system's knowledge by developing the agenda that organizes and controls the steps taken to solve problems whenever consultation is performed.

C. User Interface

Expert systems contain a language processor for friendly, problem-oriented communication between the user and the computer. This communication can best be performed in a natural language. Sometimes it is supplemented by menus, electronic forms, and graphics.

D. Knowledge Acquisition Subsystem

The knowledge acquisition subsystem is the program that the knowledge engineer uses to input and test the rules or other knowledge representation. It usually includes a mechanism to verify that rules being added to do not conflict, subsume, or supercede existing rules (e.g., see Exsys). Sometimes the subsystem includes data mining software to conduct knowledge discovery in databases and text. These tools can examine source material and induce rules automatically.

E. Blackboard (Workplace)

The blackboard is an area of working memory set aside to store the description of a current problem, as specified by input data and to record intermediate hypotheses and decisions. Three types of decisions can be recorded on the blackboard: a *plan* (how to attack the problem), an *agenda* (potential actions awaiting execution), and a *solution* (candidate hypotheses and alternative courses of action that the system has generated thus far).

F. Explanation Subsystem (Justifier)

The explanation subsystem can trace responsibility for conclusions to their sources and explain the ES behavior.

G. Knowledge Refining System

Human experts have a knowledge refining system; that is, they can analyze their own knowledge and its use, learn from it, and improve on it for future consultations. Similarly, such evaluation is necessary in computerized learning, so that the program can analyze the reasons for its success or failure. This could lead to improvements that result in a more accurate knowledge base and more effective reasoning. This component is under development in experimental ES.

H. The User

The user of an ES is usually a nonexpert human that needs advice or training. However, robots or other automatic systems that use the ES output as an input to some action can use ES results. The user is usually considered part of the ES while other people involved in its development are not.

VIII. ES SOFTWARE AND TECHNOLOGY TOOLS

A. ES Technology Levels

ES software may be classified into five technology levels: languages, support tools, shells, hybrid systems, and ES applications (specific ES). The boundaries between the levels are fairly fuzzy and our classification is mainly for an understanding of ES software.

Roughly speaking, a specific application can be developed in one or more shells, support tools, hybrid systems, or languages. Shells and hybrid systems can be developed with languages or support tools, and support tools are developed with languages. The higher the level of the software, the less programming is required. The trade-off is that the higher the level, the less flexible the software. Generally speaking, the use of higher levels of software enables faster programming. On the other hand, complex and nonstandard applications must be built with lower levels of software.

B. Specific ES

Specific ES are the application products that advise users on a specific issue, such as a consultation system that diagnoses a malfunction in a locomotive, or systems that advise on tax shelters or on buying software or selecting a car. Currently most of these may be accessed directly through consistent web-browser interfaces.

C. Shells

Instead of building a specific ES from scratch, it is often possible to borrow extensively from a previously built specific ES. This strategy has resulted in several integrated software tools that are described as shell (skeletal) systems. Initially, expert systems, like MYCIN, were stripped of their knowledge base resulting in an empty shell: the explanation and inference mechanisms, and knowledge acquisition and representation aids. Even the name of the shell EMycin was derived from the term Empty Mycin. This is how the first generation of shells were designed and developed.

Expert system shells are now integrated packages in which the major components of the ES (except for the knowledge base) are preprogrammed. These include the user interface, inferencing, and interface with other software.

Generally a shell can only represent knowledge in one or two forms (for example, rules and cases) and manipulate them in a limited number of ways (e.g., backward or forward chaining). A good shell allows the knowledge engineer to focus on the knowledge, because the shell automatically manages the knowledge, the interface, the inferencing method(s), and the inferencing rules. The programmer needs only to insert the knowledge to build a specific ES. Examples of some rule-based shells are Exsys, InstantTea, XpertRule KBS, G2, Guru, CLIPS, and JESS.

D. Support Tools

With shells, the system builder needs to develop only the knowledge base, usually in small systems. In contrast, many other types of tools help build the various parts of complex systems. They are aids for knowledge acquisition, knowledge validation, and verification, and construction of interfaces to other software packages. For example, ACQUIRE (Acquired Intelligence, Inc., www.aiinc.ca) is a tool to assist in knowledge acquisition. Its SDK derivative allows it to integrate with other applications. EZ-Xpert (AI Developers, Inc., www.ez-xpert.com) is a rapid application development (RAD) tool. Once the knowledge is encoded in it, EZ-Xpert is capable of generating 26 different format expert system knowledge base files, including 8 language shells (in 2000).

E. Hybrid Systems (Development Environments)

Hybrid systems (development environments) are development systems that support several different ways to represent knowledge and handle inferences. They may use frames, object-oriented programming, semantic networks, rules and metarules, different types of chaining (forward, backward, and bidirectional), nonmonotonic reasoning, a rich variety of inheritance techniques, and more. Hybrid systems (environments) create a programming environment that enhances building complex specific systems or complex tools. Environments are more specialized than languages. They can increase the productivity of system developers. Although environments require more programming skills than shells, they are more flexible. Several hybrid systems are based on Smalltalk and OPS.

Initially, hybrid tools were developed for large computers and AI workstations. They are available on personal computers. Representative packages are ART-IM, Level5 Object, and KAPPA PC.

F. Programming Languages

Expert systems can be developed in a programming language, ranging from AI languages to object-oriented languages and environments such as the cT Programming Language Environment (Carnegie Mellon University, www.andrew.cmu.edu). cT is an enhanced algorithmic language that includes the ability to manipulate multimedia objects. Some can even be programmed directly in a spreadsheet like Excel.

Many ES tools, toolkits, shells, and deployed systems have been developed in the fifth-generation languages (5GL) LISP and Prolog (for example, SMECI from ILOG Inc. is written in Lisp).

IX. ES DEVELOPMENT AND OPERATION

ES implementation and use consist of three major activities: *development, consultation,* and *improvement.*

A. Development

The development of an ES involves the development of a problem-specific knowledge base by acquiring knowledge from experts or documented sources. The knowledge is then separated into *declarative* (factual) and *procedural* aspects. Development activity also includes the development (or acquisition) of an inference engine, a blackboard, an explanation facility, and any other required software, such as interfaces. The knowledge is represented in the knowledge base in such a way that the system can draw conclusions by emulating the reasoning process of human experts.

Determining appropriate knowledge representations is performed during development.

The process of developing ES can be lengthy. A tool that is often used to expedite development is called the ES shell. ES shells include all the major components of an ES, but they do *not* include the knowledge.

B. Consultation

Once the system is developed and validated, it can be deployed to users. The ES conducts a bidirectional dialog with the user, asking him to provide facts about a specific incident. While accepting the user's answers, the ES attempts to reach a conclusion. This effort is made by the inference engine, which chooses heuristic search techniques to be used to determine how the rules in the knowledge base are to be applied to each specific problem. The user can ask for explanations. The quality of the inference capability is determined by the quality and completeness of the rules (or appropriateness and depth of the knowledge representation), by the knowledge representation method used, and by the power of the inference engine.

Because the user is usually a computer novice, the ES must be very easy to use. The ES asks questions and the user answers them; additional questions may be asked and answered; and, finally, a conclusion is reached. The consultation environment is also used by the builder during the development phase to test the system. At that time, the interface and the explanation facility may be tested.

C. Improvement through Rapid Prototyping

An ES is an information system; its development follows a software development process. The goal of a development process is to maximize the probability of developing viable, sustainable software within cost limitations, on schedule, while managing change. Expert systems are improved through a process called rapid prototyping.

A demonstration prototype is developed to gain a manager's support (and no doubt, funding). In prototyping, the systems analysis and design work occurs concurrently with the development of the demonstration prototype; implementation occurs concurrently with developing the knowledge base. Evaluation on a small number of cases can indicate to a manager that the system is capable of making expert-level decisions. An ES should continue to be evaluated after deployment.

X. ES APPLICATION AREAS

Expert systems can be classified in several ways. One way is by the general problem areas they address. For example, diagnosis can be defined as "inferring system malfunctions from observations." Diagnosis is a generic activity performed in medicine, organizational studies, computer operations, and so on. The generic categories of ES are listed in Table II. Some ES belong to two or more of these categories.

XI. ES BENEFITS

There are thousands of ES in use in almost every industry and every functional area. Many ES have a profound impact, shrinking the time for tasks from days to hours, minutes, or seconds, and that nonquantifiable benefits include improved customer satisfaction, improved quality of products and services, and accurate and consistent decision making. For many firms, ES have become indispensable tools for effective management. There are many examples of ES and their benefits on vendors' web sites. Below we list some of the major *potential* ES benefits. Most of these are due to the fact that ES work faster than people, and their results are consistent.

- Increased output and productivity
- Decreased decision-making time
- Increased process and product quality
- Reduced downtime
- Capture of scarce expertise
- Flexibility
- Easier equipment operation
- Elimination of the need for expensive equipment
- Operation in hazardous environments
- Accessibility to knowledge and help desks
- Ability to work with incomplete or uncertain information
- Provide training
- Enhancement of problem solving and decision making
- Improved decision-making processes
- Improved decision quality
- Ability to solve complex problems
- Knowledge transfer to remote locations

XII. ES PROBLEMS AND LIMITATIONS

Some problems that have slowed down the commercial spread of ES are:

Table II Generic Categories of Expert System Application Areas

Category	Problem addressed
Interpretation	Inferring situation descriptions from observations
Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observations
Design	Configuring objects under constraints
Planning	Developing plans to achieve goals
Monitoring	Comparing observations to plans, flagging exceptions
Debugging	Prescribing remedies for malfunctions
Repair	Executing a plan to administer a prescribed remedy
Instruction	Diagnosing, debugging, and correcting student performance
Control	Interpreting, predicting, repairing, and monitoring system behaviors

- Knowledge is not always readily available.
- Expertise can be hard to extract from humans.
- The approach of each expert to situation assessment may be different, yet correct.
- It is sometimes hard to abstract good situational assessments under time pressure.
- Users of ES have natural cognitive limits.
- ES work well only in a narrow domain of knowledge.
- Most experts have no independent means to check their conclusions.
- The vocabulary that experts use is often limited and not understood by others.
- Help is often required from knowledge engineers who are rare and expensive.
- Lack of trust by end users may be a barrier to ES use.
- Knowledge transfer is subject to a host of perceptual and judgmental biases.
- ES may not be able to arrive at conclusions.
- ES, like human experts, sometimes produce incorrect recommendations.

In 1995, Gill studied the longevity of commercial ES. He discovered that only about one-third of all commercial ES studied survived during a 5-year period. The short-lived nature of so many systems was generally not attributable to failure to meet technical performance or economic objectives. Instead, managerial issues such as lack of system acceptance by users, inability to retain developers, problems in transitioning from development to maintenance, and shifts in organizational priorities appeared to be the most significant factors resulting in long-term ES disuse. Proper management of ES development and deployment can resolve most of these issues in practice. These limitations indicate that ES today may fall short of intelligent human behavior. However, this may change as technology improves.

XIII. ES SUCCESS AND FAILURE

A. ES Success

Several researchers have investigated the reasons ES succeed and fail in practice. As with many management information systems (MIS), two of the most critical factors are the *champion in upper management*, *user involvement,* and *training*. Management must support the project and users must feel ownership. Many studies have shown that the level of managerial and user involvement directly affects the success level of MIS, and specifically ES. However, these alone are not sufficient to guarantee success. In addition,

- The level of knowledge must be sufficiently high.
- Expertise must be available from at least one cooperative expert.
- The problem to be solved must be mostly qualitative (fuzzy), not purely quantitative (otherwise, use a numerical approach).
- The problem must be sufficiently narrow in scope.
- The ES shell characteristics are important. The shell must be of high quality, and naturally store and manipulate the knowledge.
- The user interface must be friendly for novice users.
- The problem must be important and difficult enough to warrant development of an ES (but it need not be a core function).
- Knowledgeable system developers with good people skills are needed.

- The impact of ES as a source of end-users' job improvement must be considered. The impact should be favorable. End-user attitudes and expectations must be considered.
- Management support must be cultivated.

Managers attempting to introduce ES technology should establish end-user training programs, thus demonstrating its potential as a business tool. As part of the managerial support effort, the organizational environment should favor new technology adoption. Finally:

- Business applications for ES are often justified by their strategic impact in terms of gaining a competitive advantage rather than their cost-effectiveness. The major value of ES stems from capturing and disseminating expert-type skills and knowledge to improve the quality and consistency of business operations.
- The most popular and successful ES are those that deal with well-defined and structured applications, or where no more than several hundred rules are needed, such as those in the production area. ES have been less successful when applications require instincts and experienced judgments, as in the human resource management area, or where thousands of rules and their exceptions exist.

In 1996, Gill conducted a survey of 52 successful ES. He found that ES that persist over time change the nature of the users' tasks and jobs in a manner that motivates the continued use of the ES. These tools offer the user a greater sense of control, they increase work-related variety and decrease work-related drudgery, they enable the user to perform tasks at much higher proficiency levels or to assess their own task performance, etc. Gill cautioned ES developers and their managers to recognize that the design features providing such intrinsic motivation must be built within the technology. As soon as the idea for an ES (or, in fact, and information technology based application) has been conceived, it is time to start assessing its impact on user motivations. And, if the outcome of such assessments is that the motivational impacts will most likely be negative, the viability of the development effort should be reconsidered—ESs whose "motivation for use" is negative just do not last very long.

B. ES Failure

There are a number of reasons that ESs fail. If they are never deployed, then there have usually been some major complications in one or more of the earlier phases of development. Usually, these are based on managerial or economic issues. Rarely are failures due to technological problems. For a deployed system, organizational issues are usually the problem. Often system maintenance or economic issues arise. For example, if the environment changes making the ES recommendations infeasible, or the nature of the task simply changes, then the ES should have been updated to reflect the changes. Most of the reasons that ES are permanently discontinued or abandoned are organizational in nature. According to Tsai in 1994, the top three are integration, resistance to change, and finding experienced knowledge engineers. The key to ES success is to develop a high-quality system, to keep it current, and to involve the users through every step of system development.

XIV. THE FUTURE OF ES

Over time we expect to see the following ES developments:

- The development of better manual and automatic knowledge acquisition methods, including uses for working with multiple experts
- More web-based ES including Java-based ES shells
- Better toolkits and shells
- The use of integrated ES to enhance other information systems
- More widespread use leading to improved decision making, improved products, and customer service, and a sustainable strategic advantage.

XV. SUMMARY

Expert systems, a subfield of AI, enable organizations to capture the scarce expertise resource and make it available along with a reasoning capability to anyone who needs it. Expert system technology has demonstrated its ability to impact on organizations' bottom line by providing expertise to nonexperts. Expert systems can often be found providing intelligent capabilities to other information systems. This powerful technology will continue to have major impact on knowledge deployment.

SEE ALSO THE FOLLOWING ARTICLES

Engineering, Artificial Intelligence in • Hybrid Systems • Industry, Artificial Intelligence in • Intelligent Agents Medicine,

Artificial Intelligence in • Knowledge Acquisition • Knowledge Management • Knowledge Representation • Systems Science

BIBLIOGRAPHY

Allen, B. P. (1994). Case-based reasoning: Business applications. *Communications of the ACM,* Vol. 37, No. 3.

Awad, E. M. (1996). *Building expert systems: Principles, procedures, and applications.* Minneapolis/St. Paul: West Publishing Company.

Dennis, A., and Wixom, B. H. (2000). *Systems analysis and design.* New York: Wiley.

Feigenbaum, E., and McCorduck, P. (1983). *The fifth generation.* Reading, MA: Addison-Wesley.

Gill, T. G. (1995). Early expert systems: Where are they now? *MIS Quarterly,* Vol. 19, No. 1.

Gill, T. G. (1996). Expert systems usage: Task change and intrinsic motivation. *MIS Quarterly,* Vol. 20.

Guimaraes, T., Yoon, Y., and Clevenson, A. (1996). Factors important to expert systems success: A field test. *Information and Management,* Vol. 30, No. 3, 119–130.

Hart, A. (1992). *Knowledge acquisition for expert systems.* New York: McGraw-Hill.

Kolonder, J. (1993). *Case-based reasoning.* Mountain View, CA: Morgan Kaufmann.

Russell, S., and Norvig, P. (1995). *Artificial intelligence: A modern approach.* Prentice Hall, Upper Saddle River, NJ: Prentice Hall.

Tsai, N., Necco, C. R., and Wei, G. (1994). Implementing an expert system: A report on benefits realized (Part 1). *Journal of Systems Management,* Vol. 45, No. 10.

Tsai, N., Necco, C. R., and Wei, G. (1994). An assessment of current expert systems: Are your expectations realistic? (Part 1). *Journal of Systems Management,* Vol. 45, No. 11.

Turban, E., and Aronson, J. E. (2001). *Decision support systems and intelligent systems,* 6th ed. Upper Saddle River, NJ: Prentice Hall.

Wong, B. K. (1996). The role of top management in the development of expert systems. *Journal of Systems Management,* Vol. 46.

Zahedi, F. (1993). *Intelligent systems for business.* Belmont, CA: Wadsworth.

Expert Systems Construction

Victoria Y. Yoon
University of Maryland, Baltimore

Monica Adya
DePaul University

GLOSSARY

case-based representation Such representation schemes encode expertise in the form of solved cases from past experience. Characteristics of the problem domain are used to describe these cases.

contrived techniques Contrived techniques support the acquisition of knowledge for purposes of developing an expert system. These techniques involve the deliberate modification of a familiar task in revealing reasoning strategies that have little to do with the expert's usual *modus operandi*.

document analysis Document analysis is a knowledge acquisition technique for expert systems design. It involves the examination and analysis of text-based documents such as previously published literature, manuals, and memorandums.

expert system An expert system is an information system that models expertise in a well-defined domain in order to emulate expert decision-making process.

frame-based representation Frame-based schemes represent expert knowledge in frames that capture descriptive and behavioral information on objects that are represented in an expert system.

fuzzy logic representation A representation using fuzzy rules and sets is similar in nature to rule-based systems with the difference that the rules include statements with fuzzy variables that are assigned fuzzy values.

knowledge acquisition Knowledge acquisition is the process of acquiring and organizing knowledge from multiple knowledge sources, such as human experts and literature, so that expert decision-making processes can be captured, transformed, and coded in an expert system.

knowledge representation Knowledge representation involves a formal representation of the key decision variables and their relationships acquired for the design of an expert system.

rule-based systems These are expert systems in which knowledge is represented in the form of condition-action statements, particularly as IF . . . THEN rules.

task analysis Task analysis is another knowledge acquisition technique that involves the observation and analysis of experts performing familiar tasks.

validation Validation determines whether the expert system corresponds to the system it is supposed to represent. For instance, validation will examine whether the quality of the knowledge represented in an expert system is similar to the skills and knowledge of the human expert.

verification Verification of an expert system refers to the process of examining the well-defined properties of an expert system against its specifications. For instance, effective verification ensures the accuracy and parsimony of the knowledge coded in the expert system.

EXPERT SYSTEMS were one of the earliest techniques in artificial intelligence to gain prominence in academia and industry. After more than five decades of work on expert systems, many organizations claim to have reaped significant benefits from expert systems in terms of cost savings and improved decision quality. However, several studies report that a majority of these expert systems fall into disuse for various technical

and organizational reasons. This article reports the construction of expert systems and address both technical and organizational issues that must be considered in the design of such systems. This article underscores the need to address organizational, managerial, and individual concerns in addition to the technical issues that are often the main focus of expert systems researchers.

I. INTRODUCTION

An expert system is an advanced information system that models expertise in a well-defined domain in order to emulate expert decision-making processes. By encapsulating the knowledge and experiential learning of experts, an expert system provides a means of capturing and leveraging expertise in a particular field and, in turn, improving productivity and competitiveness. Durkin reports that more than 2500 expert systems are functioning and this figure is only about a fifth of systems that might have actually been developed. In the past five decades, expert systems research and applications to real-world applications have matured significantly and these systems are now being deployed in myriad applications such as forecasting, marketing, and scheduling and in the power industry. Several companies have reported significant gains from the deployment of expert systems.

Despite the pervasiveness of expert systems, conclusions about their success and usability have been mixed. Expert systems have been shown to increase productivity, reduce costs, and improve service delivery. Eom examined 440 expert systems in business and concluded that cost savings are probably the primary motivation for their use. Some major companies, such as Digital Equipment, Du Pont, and IBM, have reportedly generated significant financial returns as well as competitive advantage from using expert systems. The manufacturing industry reports that expert systems have been found to be most effective in process planning, product design, layout/facility design, and maintenance. Similar gains have also been reported in the government sector from the deployment of expert systems. In the Florida Department of Motor Vehicles, the Supervisor Assistance System was used to deal with employee misconduct cases. The supervisors felt that the expert system clarified their decisions regarding disciplinary actions and helped them justify these decisions once the recommendations were sent to the management. Supervisors also felt that the system provided a knowledge base that was much easier to use than reading rules and policies from a handbook.

Unfortunately, expert systems have proven to be difficult to implement and institutionalize. Failure in implementing such systems has been attributed to various factors that range from organizational to design and implementation. Gill reported that only one-third of the 73 expert systems in existence in 1987 were in operation in 1992. Reasons for such disuse included maintenance expenses, misalignment with environment, poor sizing of the system, and solving a noncritical problem. Gill also found that some of the most influential hardware and tool companies such as Intellicorp, Teknowledge, and Symbolics had been forced to reorganize and cut back on their supply of expert system tools. To address this problem of many expert system failures, we provide a set of procedures for developing and implementing expert systems. These procedures span a wide range of technical, managerial, and organizational issues.

We discuss the procedures and guidelines for effective design of expert systems. Our discussion focuses on the technical issues that contribute to the effective design of expert systems as well as on organizational issues that contribute to the acceptance of such systems. To aid our discussion, this article relies on the expert systems development life cycle (ESDLC) proposed by Guimaraes and Yoon. There are several proposals for ESDLCs in the existing literature. However, the currently available ESDLCs either have not been validated in practice or have been tested in a limited context. Some of the life cycles proposed in the literature represent models that focus only on the knowledge acquisition stage of expert systems design. The ESDLC used in this chapter attempts an integration of the important elements already proposed in literature.

II. OVERVIEW OF ESDLC

The ESDLC presented by Guimaraes and Yoon consists of nine phases, as shown in Fig. 1. The design life cycle consists of nine phases:

1. *Problem identification:* Identifying the problems and opportunities where the organization can obtain benefits from an expert system. This phase also involves establishing the general goals of the system.
2. *Feasibility study:* Assessing the feasibility of expert systems development in terms of its technical, economic, and operational feasibility.
3. *Project planning:* Planning for the expert systems project, including identifying the development

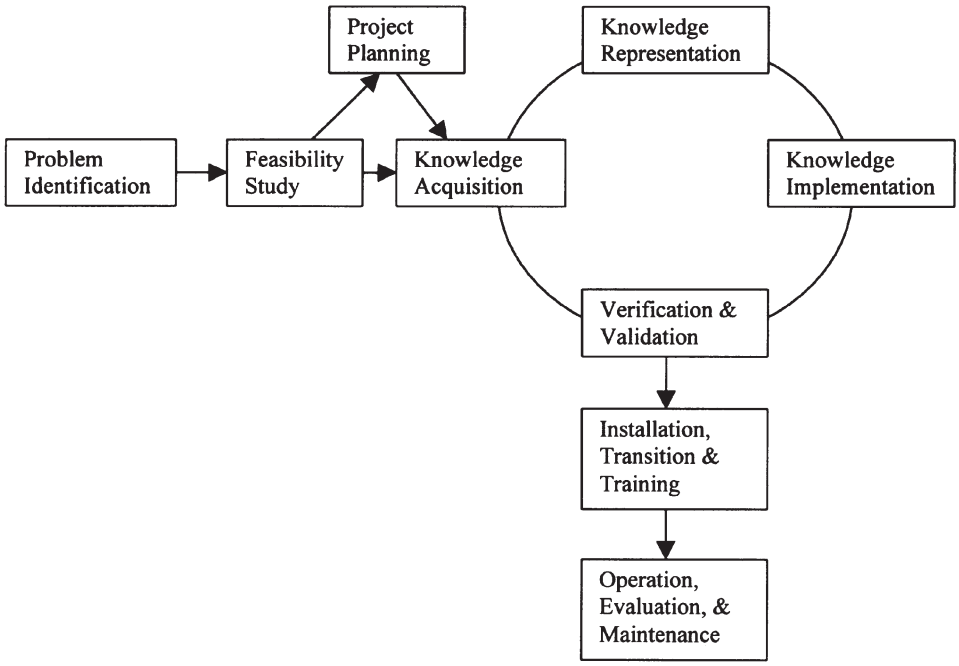


Figure 1 Expert system development life cycle.

team members, the working environment, the project schedule, and the budget.

4. *Knowledge acquisition:* Extracting domain knowledge from domain experts and determining the requirements for the system.
5. *Knowledge representation:* Representing key concepts from the domain, and interrelationships between these concepts, using formal representation methods.
6. *Knowledge implementation:* Coding the formalized knowledge into a working prototype.
7. *Verification and validation:* Verifying and validating a working prototype against the system's requirement, and revising it as necessary according to domain experts' feedback.
8. *Installation/transition/training:* Installing the final prototype in an operating environment, training the users, and developing documentation/user manuals.
9. *Operation/evaluation/maintenance:* Operating the system in the working environment, evaluating its performance and benefits, and maintaining the system.

The prototype expert systems development commences with project approval. Phases 4 through 7 represent an iterative process whereby a prototypical ES is evolved, and the final prototype developed through these iterative phases is installed in an operating environment. The next section presents more detailed descriptions of each phase and discusses existing literature and findings in light of these phases.

III. ESDLC

A. Problem Identification

Several studies have suggested that expert systems should address a task with higher payoff. The first step therefore is to identify the problems and opportunities that drive the need for an expert system in an organization. Successful expert systems are designed to solve core problems in an organization's functioning. For instance, Authorizer's Assistant at American Express aids credit managers in making rapid credit granting decisions. Because credit granting is a core function of American Express, the use of this system has been sustained in the organization for several decades. On the other hand, Honeywell's SYSCON was designed to perform computer system configurations. Because these systems were simple to configure, human experts were already quite adept at the task. Consequently, the system did not introduce any additional quality benefits and was eventually discontinued.

Gill found that 6 systems in his survey of 73 expert systems were ultimately abandoned because the degree of fit between the objectives of the expert system and

those of the organization and the IS department was inadequate. Such evidence underscores the need to clearly identify the objectives of a potential project and align these with the overall business goals. Because the development of an expert system typically entails substantial investment and often involves technologies that are novel to the organization, it is incumbent on managers to ensure that both task and technology are consistent with the organization's objectives and the IS environment in order to obtain the necessary commitment to see the project to completion.

B. Feasibility Study

Determining the feasibility of undertaking the development of expert systems is important in order to obtain management commitment to the project. The feasibility of an expert systems project is determined as for traditional systems. However, economic, technical, and operational feasibility gain more significance because of the high cost of development, the novelty of the technology, and increased possibility of user resistance to the implemented system. In his examination of expert systems, Gill concluded that inadequate cost–benefit justification had a measurable impact on the rates of user penetration and longevity for some systems.

1. Economic Analysis

A system's ability to generate sufficient economic return is perceived to be the most important prerequisite to its development. Economic feasibility typically requires an estimation of the costs of building and maintaining the system, the benefits of and returns from using such a system, and the opportunity cost of the expert system, that is, the losses incurred in continuing with the existing system.

A number of techniques have been proposed for estimating the economic return of expert system applications. These range from formal techniques for cost–benefit analysis to informal discussions of how to assess return on investment. Gill finds three important contributors to such a feasibility analysis that are considered by all techniques:

1. *Start-up costs:* The costs associated with the initial development of the system. This includes both direct costs, such as hardware, software, and programmer time, and indirect costs such as the facilities used and management, user, and expert time.
2. *Ongoing costs:* Pressman estimates that the direct and indirect costs associated with ongoing maintenance and upgrading of an expert system are as high as 70–80% of total costs over a system's lifetime.
3. *Ongoing benefits:* These benefits may take either the form of cost savings (e.g., reduction in personnel) or revenue enhancement (e.g., improved sales) that are experienced as a consequence of using an expert system. A challenge in determining this element of the cost–benefit analysis is getting a good estimate of intangible benefits such as improved customer satisfaction or increased goodwill.

2. Technical Feasibility

Technical feasibility evaluates the technical complexity of the expert system and often involves determining whether the expert system can be implemented with state-of-the-art techniques and tools. In the case of expert systems, an important aspect of technical feasibility is determining the shell in which the system will be developed. The shell used to develop an expert system can be an important determinant to its quality and makes it vital to the system's success. Although the desirable characteristics of an expert system shell will depend on the task and domain requirements, the shell must be flexible enough to build expert reasoning into the system effectively. It must also be easily integrated with existing computer-based systems. Furthermore, a shell providing a user-friendly interface encourages end users to use the system more frequently.

3. Operational Feasibility

This requires an examination of the operational environment and users' acceptance of the system once it has been developed. In particular, operational feasibility evaluates possible resistance from end users that will undermine the potential benefits of the expert system. In general, it is a good strategy to keep the users involved in the design and implementation of the system. End-user involvement has been an important contributor to the future use of expert systems. Expert systems are often applied to tasks that are performed by skilled individuals and, consequently, potential users may be particularly sensitive to such technology as being intrusive. Gill found that 6 of the 73 systems surveyed were abandoned because potential users were concerned about using systems they had not helped develop.

Literature on management involvement clearly indicates that management support is critical for mitigating end-users' negative attitudes toward expert systems and to overcome user resistance to the system. Management support can be useful in getting the users involved in the design phase of the system.

Conducting a thorough feasibility analysis also clearly defines the nature and scope of the project, which further aids in obtaining organizational commitment to the task. Once the organizational approval and commitment have been obtained, the prototype development process can commence. Utilizing expert systems requires the staff to make a conscious change in their behavior, and encouragement by senior management seems to help individuals integrate such change into their work routines.

C. Project Planning

The project planning phase focuses on obtaining management approval and the funds and resources necessary for the expert systems project. The scope and nature of the expert system need to be clearly laid down in order to obtain this commitment. Project planning also involves identifying the participants and their roles in the development and implementation of the system, determining the project schedule, and defining the working environment by choosing the knowledge representation methodology and expert systems shell-based evaluation criteria deemed important by the team.

D. Knowledge Acquisition

An expert system is only as effective as the knowledge that is modeled in it. The acquisition of this knowledge, therefore, is one of the most important and also the most challenging phases of expert systems development. Knowledge acquisition is the process of extracting and organizing knowledge from multiple sources, typically human experts, with the purpose of modeling a decision maker's problem-solving expertise. Indeed, a large volume of research in expert systems has focused on developing effective techniques for knowledge acquisition and on overcoming the "knowledge acquisition bottleneck."

1. The Knowledge Acquisition Bottleneck

Acquiring an expert's knowledge typically involves interactions between the expert and the knowledge engineer (KE) who is responsible for the knowledge acquisition process. The knowledge engineer will either interview the expert or will present a task or decision situation to the expert. This task must be designed such that it taps into their knowledge and skill and reveals their reasoning and judgment.

Several reasons contribute to the knowledge acquisition bottleneck. The KE may have a limited understanding of the problem domain and, consequently, may not be asking the appropriate questions or achieving the depth required to extract the expertise. The expert's responses may be misinterpreted by the KE. More importantly, the challenge lies in extracting this expertise from the experts who are often not aware of the knowledge they use in forming the judgments. They often refer to their most important judgments as "intuitive." Consequently, they may be unable to reveal the knowledge they use.

Effective elicitation of expertise relies not only on adequate training of the KE but also on the nature of KE techniques used for extracting such knowledge from the expert. In general, KE should rely on multiple sources of expertise. The KE can rely on literature, surveys, and task analysis of multiple experts to obtain a good representation of the problem domain.

2. Techniques for Knowledge Acquisition

Knowledge acquisition methods can be divided into two sources: direct and indirect. Direct methods require direct interaction with the experts such as observing the experts at work and asking questions about their reasoning process. Interviews and protocol analyses are other examples of such techniques. Indirect methods focus on obtaining knowledge from texts, published empirical studies, reports, and other printed media. Document analysis and questionnaires fall under this category. The next few sections discuss some of the knowledge acquisition tools available to the KE.

a. Document Analysis

When commencing research in a domain, one must start by reading texts, manuals, and other published manuscripts. For instance, the design of rule-based forecasting was preceded by detailed examination of previously published literature to characterize the domain of time series forecasting. Document analysis must be iterative in nature. The reader must analyze the documents and make extensive notes on the outcome of this analysis. The documents may then be revisited later for clarification and validation.

b. Interviews

Interviews are, by far, the most commonly used approach to knowledge acquisition. In an examination of U.K.-based expert systems, Doukidis and Paul found that of 166 expert systems, 100 relied on interviews as the main source of expertise. Unstructured interviews are usually more exhaustive than structured interviews. However, such interviews may deviate the KE and the expert from the main task, thereby making the interpretation of transcripts challenging. Structured interviews, on the other hand, can reduce the time spent relative to unstructured interviews. Furthermore, if the KE is well trained, structured interviews can be more effective in eliciting expert knowledge. The most common risk of such an interviewing technique is that it is not possible to probe an expert's reasoning process in depth. A combination of structured and unstructured interviews could be more productive since the KE can begin the interviews with a set of predefined questions but may build sufficient flexibility into the interviews to follow the expert's reasoning process to closure.

c. Task Analysis

In the analysis of familiar tasks, one investigates what experts do when they conduct their usual problem-solving or decision-making tasks. Observation is a technique that involves observing the expert at work in familiar environments without interruptions by the knowledge engineer. Although observation is helpful in unraveling a complex problem domain, one of its limitations is its reliance on the KE's ability to deduce the experts' reasoning from his actions. In this regard, protocol analysis is a more explicit and effective technique of knowledge elicitation. In this method, the expert is asked to solve a problem and while performing this task, is asked to verbalize his thoughts. This procedure generates a protocol that can be transcribed and analyzed for propositional content. Collopy and Armstrong conducted protocol analysis on five forecasting experts in order to elicit forecasting expertise. The experts were presented with time series and were asked to think aloud while forecasting the series. Protocols generated from this procedure yielded very explicit rules, which were then coded into a rule-based forecasting system.

Repertory grid analysis aims at gaining insight into the expert's metal model of the problem domain. In an interview, the expert is first asked to identify some objects in the domain of expertise. She is then asked to compare three objects at a time and name one trait at a time that two of the objects possess and the third does not. The expert is then required to identify the opposite of that trait and rate the objects according to the importance of the traits. This process is repeated until all the objects have been compared. The technique is useful in gaining a subjective understanding of the domain but is effective primarily when the number of objects to be compared is small.

d. Contrived Techniques

Hoffman et al. report on literature that supports the effectiveness of deliberate modification of a familiar task in revealing reasoning strategies that have little to do with the expert's usual *modus operandi*. In a technique called "20 questions" proposed by Grover, the expert is provided with little or no information about a particular problem to be solved and must ask the elicitor for information needed to solve the problem. The information that is requested along with the order in which it is requested can provide insights into the expert's problem-solving strategy.

Literature clearly supports the use of multiple techniques for a thorough elicitation of expertise. In general, a two-stage approach must be taken to elicit knowledge. In the first stage, document analysis must be conducted to gain a good conceptual understanding of the domain and possibly yield some general rules for the system. This will involve the examination of empirical and theoretical literature in the domain, memos and intraorganizational communications, notes made by experts, and other such documentation. Unstructured interviews must be conducted not just with the expert but with other users in the organization and, wherever possible, the experts must be observed at work. A thorough examination of the domain at this stage will allow the researcher to develop a detailed conceptual model of the domain. Prior studies have suggested that knowledge that is acquired in the early stages of knowledge acquisition is often used to constrain subsequent knowledge elicitation. In the second stage, structured interviewing, protocol analysis, and contrived techniques can be used to extract explicit rules that can be coded in the expert system. These techniques can also be used to validate, refine, or extend the general rules obtained from the first phase.

E. Knowledge Representation

Knowledge representation involves representing the key concepts and relations between the decision variables in some formal manner, typically within a framework suggested by an expert systems shell. Most representation mechanisms must provide support for

three aspects of knowledge—conceptual representation, relational representation, and uncertainty representation. As such, four schemes are commonly used for knowledge representation.

1. *Rule-based representation:* Such a scheme represents knowledge in the form of IF . . . THEN rules. For instance, a rule can be coded as "IF the credit rating of the applicant is poor, THEN do not grant the loan." The rules are processed through a backward or forward chaining process, or a combination of the two. Rule-based representations allow the inclusion of uncertainty management through the use of confidence factors. Due to their simplicity of representation and ease of use, rule-based representations remain the most popular representation scheme for expert systems.
2. *Frame-based representation:* Frame-based schemes represent the knowledge in frames that capture descriptive and behavioral information on objects that are represented in the expert system. Because frame-based representations share a lot in common with object-oriented programming, they are powerful representation mechanisms and are increasingly becoming popular.
3. *Case-based representation:* Such representation schemes encode expertise in the form of solved cases from past experience. Characteristics of the problem domain are used to describe these cases. When a new case is presented to the expert system, the representation scheme supports a comparison with stored cases and provides a decision that best represents the closest match based on some distance measure. Case-based representations are most effective when the domain is supported by an adequate number of cases.
4. *Fuzzy logic representation:* A representation using fuzzy rules and sets is similar in nature to rule-based systems with the difference that the rules include statements with fuzzy variables that are assigned fuzzy values. For instance, a rule might be stated in fuzzy terms as "IF the credit rating is *very bad*, THEN do not approve loan for the next two years." Fuzzy values are represented mathematically in fuzzy sets. Fuzzy logic is then applied to these rules and sets to process the reasoning. Fuzzy logic is a powerful representation technique and has yielded performance at par with human operators in certain areas such as control systems.

In general, the representation technique selected must be simple and intuitive to the task domain. Collopy et al. recommend that the representation scheme selected must support full disclosure. In other words, the knowledge coded into the expert system must be simple to understand when examined by a person unfamiliar with the task domain.

F. Knowledge Implementation

Implementing expert knowledge requires turning the formalized knowledge into a working computer program or prototype. Prototyping is the most popular implementation technique and has several benefits that are essential for expert system success and usage. The first is that prototyping requires a high degree of interaction between the end users and system developers. Consequently, it ensures user participation to a large degree. The significance of such user involvement in the design and development phases has already been emphasized earlier in this chapter. A second benefit of prototyping is that user interaction improves the user's ability to provide system specifications more clearly in an evolutionary fashion. This is an important factor in successful systems development and implementation. Finally, the rapid prototyping approach to systems development supports ongoing verification and validation of the expert system.

G. Verification and Validation

Verification of an expert system refers to the process of examining the well-defined properties of an expert system against its specifications. Validation determines whether the expert system corresponds to the system it is supposed to represent. Vermesan lays out several components of a complete verification and validation plan. Effective verification and validation must ensure the following:

- The quality of the knowledge in the expert system is comparable to expert knowledge and skills.
- All the knowledge in the knowledge base is referenced and that environment does not try to access nonexistent knowledge.
- There are no contradicting rules in the knowledge base.
- The rules are represented as accurately as possible in accordance with the expert's specifications.
- There is a low percentage of instances when the expert system fails to arrive at a solution.
- The knowledge base is representative of only that problem domain for which the expert system is designed.

Verification and validation of expert systems is critical for ensuring the internal and external validity of the system. A well-validated system is easier to accept than systems for which no validation and testing have been conducted. Unfortunately, very few expert systems meet this criterion. Santhanam and Elam surveyed 430 knowledge-based systems in decision sciences between the 1980 and 1995. They found that only 10 of these systems had been validated.

Several techniques are available to expert systems designers to enable verification and validation of their systems. Of course, a combination of these techniques would be most effective. One set of these approaches is informal in nature and relies primarily on the developer's experience for locating such errors. In this case, the developer will examine and review the rule base and attempt to identify rules and rule sets that are redundant, inconsistent, unused, unsatisfiable, and so on. Informal techniques are inexpensive but are not effective in situations where complex rule dependencies exist.

Formal verification and validation techniques involve the examination of expert system performance on a set of test cases. Collopy and Armstrong validated their rule-based forecasting system on a set of 90 time series. The system was first tested on a set of 36 series. Results on this set were used to refine and correct the rule base. This procedure was repeated with two more independent sets of time series. Though empirical in nature, such validations become very specific to the test cases and results from such comparisons may not be generalizable. Collopy et al. also suggest the use of a Turing test to assess the quality of the expert system. This test determines whether a panel of experts can distinguish the differences in the reasonableness of outputs from an expert system and a real expert.

Several systems such as KRUST, IN-DEPTH, and IMPROVER are now being developed for automating the process of verification and validation of expert systems. However, these are not complete in any way and typically perform only simple checks that are not effective in detecting anomalies in larger, more complex expert systems. Clearly, developers will benefit from using a combination of verification and validation techniques that are suited for their systems.

The traditional view of verification and validation focuses primarily on an in-depth analysis and examination of the rule base. One aspect that is often overlooked is that of system refinement based on utility testing with users. Yoon et al. provide evidence on the positive relationship between the impact of expert systems on the end users' job and the success of such systems. If the expert system does not improve the work environment significantly in terms of either reducing the workload or making the work environment more efficient, later stages of expert system implementation may suffer. Consequently, users must be actively involved in the verification and validation stages to examine such issues. Typically, users should be asked to examine decision quality, response time, and user friendliness of the system.

H. Installation, Transition, and Training

In this phase, the final prototype is converted into the operational environment. This typically involves determining the method of delivery and transition from the old system to the new system. By this phase, the acquisition of inputs and the dissemination of outputs must be clearly defined and in place. Documentation and user manuals should be developed.

Organizations must devote adequate resources at the implementation stage to achieve success when introducing an expert system. In this regard, it is important that managers effectively market the system to the users and provide adequate training. Berry et al. report that in their examination of expert systems in the public sector, they found that the amount of formal education a manager receives appears to have no effect on his level of utilization of the expert system. Instead, the adequacy of training is an important determinant of usage indicating that good training can overcome potential obstacles to implementation. These results were also supported by Guimaraes et al.

Adequate training of users will help to reduce the fears that often accompany the use of expert systems and artificial intelligence technologies, in general. Prereau suggests the use of seminars to disseminate the positive impact of expert systems on end-user functioning. Demonstrations, online tutorials, and formal training courses should be organized and should not only focus on system use and results but also on troubleshooting and extracting the most from the system options.

I. Operation, Evaluation, and Maintenance

The final phase of expert systems construction is the longest due to its ongoing nature. This phase deals with putting the system in operation and continuously evaluating and maintaining the system per user requirements. Evaluation of the system focuses on three aspects: organization, users and managers, and system

performance. Organizational impact of expert system deployment largely focuses on measuring the benefits to the organization primarily in terms of cost reductions or savings. Over a period of time, issues such as operational efficiency and competitive advantage can also be realized.

User and manager attitudes are monitored to identify potential problems with system disuse. As discussed in the previous sections, literature has suggested that user involvement and training and managerial commitment to the project have been shown to be strong motivators for expert system success and use. Managers must constantly revisit user's motivations to ensure long-term system usage. Ideally, an operational expert system should be routinely refined and updated according to changing user and task requirements. However, such maintenance is expensive since construction of expert systems requires substantial knowledge of both task domains and specialized development tools. Retaining individuals with such specific skills is challenging and losing even a single developer can mean reworking a large part of the project.

IV. CONCLUSIONS

Because expert systems have proven to be a viable technology for organizations, a significant amount of literature has focused on suggesting ways to develop them. Due to the expenses involved in developing and maintaining such systems, substantial research efforts have also focused on identifying factors that contribute to the success and usage of such systems. Most of this literature relies on an examination of expert systems once they have been implemented and have been in use for extended periods of time or have failed in sustaining their stronghold as a decision support tool in the organization. Consequently, one can only examine the impact of design decisions on expert system success in retrospect. In our description of the ESDLC in this chapter, we have attempted to tie these two pieces of expert systems research together in an attempt to allow expert systems managers, designers, and users to make informed design decisions early in the process of expert systems construction. Here are some of our recommendations are:

- Expert systems must be applied to core business problems to gain high payoff.
- Although technical and operational feasibility are important factors in gaining managerial support for expert systems development, such a venture must undergo rigorous economic assessment in order to gain managerial support.
- Obtaining managerial commitment early in the design process is important in terms of harnessing organizational resources and funds, motivating users about the effectiveness of the project, and ensuring user involvement in the design of the expert systems.
- Effective management of user motivations and involvement is crucial to system acceptance and usage in the postimplementation phases.
- User training is important for user acceptance.
- Expert system shells must be selected judiciously by giving careful consideration to features such as methods of knowledge representation, costs, and support, etc.
- Developers must select knowledge representation schemes that are suited to the task and are supported by the expert system shell.
- Multiple sources of knowledge are recommended for populating the knowledge base. In another variation to this, multiple experts should be used to support knowledge elicitation.
- Verification and validation must be an ongoing process and multiple approaches to validation must be used.
- Adequate resources and efforts must be put into retaining key development personnel in order to provide long-term support for the expert system.

Gill points out that a significant part of the literature on expert systems focuses on either the technical or the organizational aspects of expert systems use and development. The discussion of expert systems in this article clearly indicates that the two issues are inseparable and that it is important to effectively address both these issues early in the design process to possibly ensure the longevity and success of the expert system.

SEE ALSO THE FOLLOWING ARTICLES

Ethical Issues in Artificial Intelligence • Expert Systems • Hybrid Systems • Knowledge Acquisition • Knowledge Representation • Machine Learning

BIBLIOGRAPHY

Alavi, M. (1984). An assessment of the prototyping approach to information system development. *Communications of the ACM,* Vol. 27, No. 6, 556–563.

Barsanti, J. B. (1990). Expert systems: Critical success factors for their implementation. *Information Executive,* Vol. 3, No. 1, 30–34.

Berry, F. S., Berry, W. D., and Forster, S. K. (1998). The determinants of success in implementing an expert system in state government. *Public Administration Review,* Vol. 58, No. 4, 293–305.

Buchanan, B., *et al.* (1993). Constructing an expert system. *Building expert systems,* F. Hayes-Roth, D. Waterman, and D. Lenat (Eds.). Reading, MA: Addison-Wesley.

Casey, J. (1989). Picking the right expert system application. *AI Expert,* 44–47.

Collopy, F., and Armstrong, J. S. (1992). Rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations. *Management Science,* Vol. 38, 1394.

Collopy, F., Adya, M., and Armstrong, J. S. (2001). Expert systems for forecasting. *Principles of forecasting: A handbook for researchers and practitioners,* J. S. Armstrong (Ed.). Norwell, MA: Kluwer Academic Publishers.

Durkin, J. (1996). Expert system development tools. *The handbook of applied expert systems,* J. Liebowitz (Ed.). Boca Raton, FL: CRC Press.

Eom, S. E. (September–October 1996). A survey of operational expert systems in business. *Interfaces,* Vol. 26, 50–70.

Gill, T. G. (March 1995). Early expert systems: Where are they now? *MIS Quarterly,* 51–81.

Gill, T. G. (September 1996). Expert systems usage: Task change and intrinsic motivation. *MIS Quarterly,* 301–329.

Grover, M. D. (1983). A pragmatic knowledge acquisition methodology. *The psychology of expertise: Cognitive research and empirical AI,* R. R. Hoffman (Ed.). Hillsdale, NJ: Erlbaum.

Guimaraes, T., and Yoon, Y. (Summer 1996). An exploratory assess of the use and benefits of ESDLC in practice. *Information Resources Management Journal,* Vol. 9, No. 3, 15–23.

Guimaraes, T., Yoon, Y., and Clevenson, A. (1996). Factors important to expert systems success: A field test. *Information & Management,* Vol. 30, 119–130.

Hadden, S. G. (1989). Intelligent advisory systems for managing and disseminating information. *Public Administration Review,* Vol. 46, 572–578.

Harmon, P., Maus, R., and Morrissey, W. (1988). *Expert systems: Tools and applications.* New York: Wiley.

Hayes-Roth, F., and Jacobstein, N. (1994). The state of knowledge-based systems. *Communications of the ACM,* Vol. 37, No. 3, 27–39.

Hoffman, R. R., Shadbolt, N. R., Burton, A. M., and Klein, G. (1995). "Eliciting knowledge from experts: A methodological analysis. *Organizational Behavior and Human Decision Processes,* Vol. 62, No. 2, 129–158.

Lee, S. K., and Lee, J. K. (1996). Expert systems for marketing: From decision aid to interactive marketing. *The handbook of applied expert systems,* J. Liebowitz (Ed.). Boca Raton, FL: CRC Press.

Liebowitz, J. (1996). *The handbook of applied expert systems.* Boca Raton, FL: CRC Press.

Liebowitz, J., Krishnamurthy, V., Rodens, I., and Potter, W. (1996). Scheduling. *The handbook of applied expert systems,* J. Liebowitz (Ed.). Boca Raton, FL: CRC Press.

Prereau, D. S. (1990). *Developing and managing expert systems.* Reading, MA: Addison-Wesley.

Pressman, R. S. (1982). *Software engineering,: A practitioner's approach.* New York: McGraw-Hill.

Santhanam, R., and Elam, J. (1998). A survey of knowledge-based system research in decision sciences (1980–1995). *Journal of Operational Research Society,* Vol. 49, 445–457.

Sloane, S. B. (1991). The use of artificial intelligence by the United States Navy: Case study of a failure. *AI Magazine,* Vol. 12, No. 1, 80–92.

Taylor, O., Smith, P., McIntyre, J., and Tait, J. (1996). Power industry. *The handbook of applied expert systems,* J. Liebowitz (Ed.). Boca Raton, FL: CRC Press.

Turban, E. (1992). *Expert systems and applied artificial intelligence,* New York: Macmillian Publishing.

Vamos, T. (1996). Knowledge representation. *The handbook of applied expert systems,* J. Liebowitz (Ed.). Boca Raton, FL: CRC Press.

Vermesan, A. I. (1996). Foundation and application of expert system verification and validation. *The handbook of applied expert systems,* J. Liebowitz (Ed.). Boca Raton, FL: CRC Press.

Waterman, D. A. (1986). *A guide to expert systems.* Reading, MA: Addison-Wesley.

Wong, B. K., Chong, J. K. S., and Park, J. (1994). Utilization and benefits of expert systems in the manufacturing sector. *International Journal of Productions and Operations Management,* Vol. 14, No. 1, 38–49.

Wood, L. E., and Ford, J. M. (1993). Structuring interviews with experts during knowledge elicitation. *Knowledge acquisition and modeling,* K. M. Ford and J. M. Bradshaw (Eds.). New York: Wiley.

Yoon, Y., Guimaraes, T., and O'Neal, Q. (March 1995). Exploring the factors associated with expert systems success. *MIS Quarterly,* 85–106.

Extranets

Deborah Bayles Kalman
University of California, Irvine and Singapore Institute of Management

GLOSSARY

extranet An intranet that allows controlled access by authenticated outside parties.
firewall A set of components that functions as a choke point, restricting access between a protected network (e.g., an intranet) and the Internet.
internet key exchange (IKE) A standard which authenticates devices: the security gateway or client host at each end of an IP security protocol tunnel.
internet protocol (IP) The method by which data is sent from one computer to another on the internet.
intranet Internal closed network based on Internet technology.
IP security protocol (IPSec) A standards-based method of providing privacy, integrity, and authenticity to information transferred across IP networks. IPSec was designed for secure site-to-site (gateway-to-gateway) and remote access (client-to-gateway) tunneling between mutually authenticated devices. IPSec can protect tunneled packets against spoofing, modification, replay, eavesdropping, and other man-in-the-middle attacks, but in order to trust anything received over an IPSec tunnel one must first verify the sender's identity.
layer two tunneling protocol (L2TP) A standard method for tunneling point-to-point protocol (PPP) across the Internet or any intervening IP or non-IP network. Unlike IPSec, L2TP was designed specifically to support traditional remote access.
point-to-point tunneling protocol (PPTP) A networking technology developed by Microsoft that supports multiprotocol virtual private networks. PPTP allows a PPP session to be tunneled through an existing IP connection, regardless of how it was set up. An existing connection can be treated as if it were a telephone line, so a private network can run over a public connection.
tunneling A way of securely transmitting data by encrypting each standard IP packet and then encapsulating it inside a transmission control protocol (TCP)/IP packet.
virtual private network (VPN) Any network built upon a public network and partitioned for use by individual customers. Public frame relay, X.25, and asynchronous transfer mode (ATM) networks are generically referred to as "Layer 2 VPNs." VPNs consisting of networks constructed across several IP backbones are referred to as "IP VPNs."

AN EXTRANET is an intranet that allows controlled access by authenticated outside parties. Typically, an extranet will link the intranets of distributed organizations for the purpose of conducting business. This secure electronic consortium usually consists of an enterprise and its key trading partners, customers, dealers, distributors, suppliers, or contractors.

Extranets were forged from the secure, closed environments of intranets, which are internal closed networks based on Internet technology crossed with the public outreach of the Internet itself. Both extranets and intranets share the requirement of creating high-quality interaction while ensuring security, confidentiality, and controlled access. Extranets, however, take intranets and extend them via "virtual firewalls" to enable collaborative business applications across multiple organizations. Extranets are therefore more

private than the Internet, yet more permeable than an intranet because they allow access to third parties through authentication (see Fig. 1).

There are two basic extranet configurations. The first is to have a direct leased line where an enterprise can have full physical control over the line from intranet to intranet. The second, and most popular, is to set up an extranet through a secure link over the Internet, where the enterprise uses the Internet as a universal private number. If existing intranet and Internet infrastructure is used, an extranet becomes simply a logical overlay, defined only by access privileges and routing tables, rather than a new physical network in its own right. Constructing an extranet then becomes economical and relatively simple.

Because of its popularity and relevance, the second configuration model will be used in this chapter. The open standards of Internet technology have made the creation and adoption of extranets one of the most promising concepts for collaborative businesses today. An ideal extranet scenario calls for seamless deployment across intranet, Internet, and extranet environments. In this way a group of selected linked organizations can collaborate using standard Internet technologies while enjoying the privacy and autonomy of an intranet environment.

I. BUSINESS FORCES SHAPING THE EXTRANET

A. Industry Backdrop

The revenues generated by business-to-business (B2B) sites are now approximately 10 times greater than those generated by business-to-consumer (B2C) sites, and the trend continues unabated. Extranets are the primary enabler of electronic commerce within the B2B sector.

B. IP Protocol

Another reason for the explosive growth of extranets is the almost universal acceptance of the IP protocol as a preferred networking protocol standard. Through

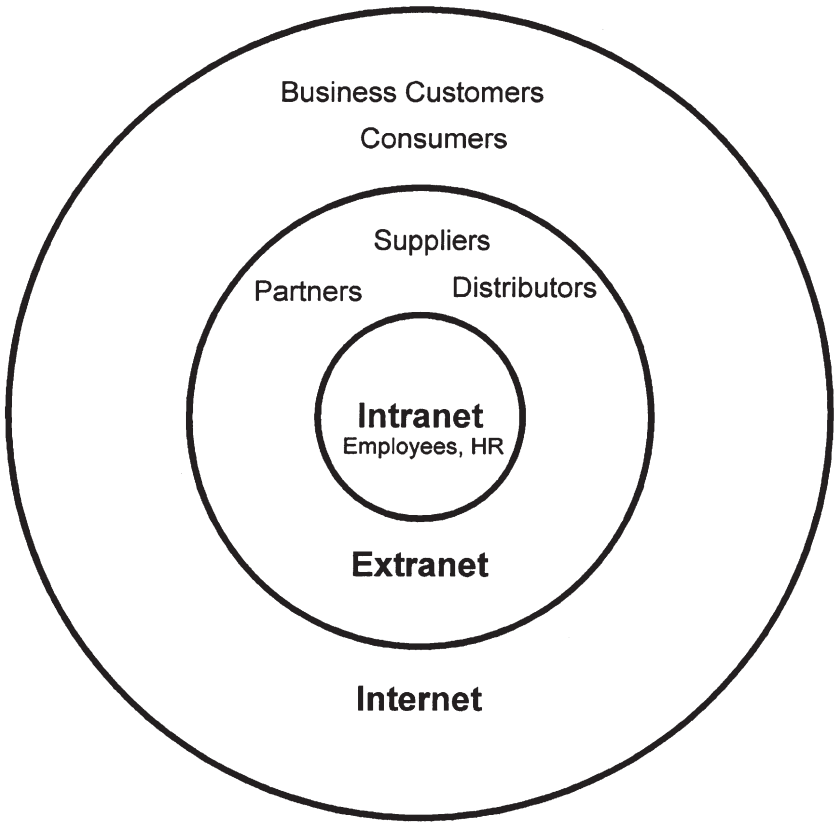


Figure 1 Simple extranet diagram.

widespread adoption of this standard, enterprises can connect intranets together for B2B interoperability relatively easily and inexpensively, whereas in the past those kinds of connections were very difficult and very expensive to build.

C. Electronic Commerce

Conducting electronic commerce via an extranet holds the promise of saving an organization a significant amount of money over paying a value-added network (VAN) provider for an existing electronic data interchange (EDI) application. Because Internet technology is so pervasive and economical, extranets are starting to take over many of the business chores, such as purchase orders and invoices, that have traditionally been handled by EDI on expensive and proprietary VANs.

D. Supply Chain Management

The advent of extranets is revolutionizing supply chain management. Prior to the open architecture of extranet technology, the prospect of linking the product supply chain—manufacturers, suppliers, dealers, offsite contractors, and customers—was virtually impossible. Those that succeeded were expensive proprietary networks that were custom developed to overcome heterogeneous platforms and networks within each organization. Web-based extranets opened up these closed systems by achieving integration across distributed, cross-platform environments.

The automotive industry is a prime area for the use of extranets in supply chain management. One example is Covisint (www.covisint.com), an on-line integrated exchange for participants in the worldwide automotive industry. Covisint is an independent company with joint ownership by Ford, DaimlerChrysler, General Motors, Renault/Nissan, Commerce One, and Oracle. Covisint offers a comprehensive online marketplace for the procurement of automotive parts and supplies and other products and services for members of the automotive industry (i.e., original equipment manufacturers [OEMs], suppliers, partners, dealers, and others).

E. Emerging On-Line Business Trends

The first forays into the Web environment consisted of businesses constructing "brochureware" Web sites that were static and often of marginal interest. This form of communication was based on a one-way, one-to-many model. Although the majority of business Web sites are still in those formative stages, there are several emerging on-line business trends that are shaping the extranet.

1. Interactive Communication

The single greatest trend, and the one that is shaping all others, is the increasing use of other models of interactive communication on the Web. Extranet applications address these main interaction scenarios:

- One-to-many communication. This interaction can take place between teams, department, workgroups, or entire enterprises. A classic example is posting information on a Web page, giving an enterprise immediate printing and mailing cost savings.
- Two-way interactions. The Web and e-mail are ideal for extranet applications such as technical support and on-line customer service. Problems can be submitted by users, and technical support can then respond. Another form of two-way communication is using an extranet to access databases and perform queries for research or to conduct other transactions.
- Many-to-many communication. Probably the most common example of many-to-many communication is newsgroups. An extranet may extend selected access from internal newsgroups to trusted business partners or may create a special forum just for suppliers or customers.

2. Co-opetition

The extranet stretches the new concept of "co-opetition"—the cooperation with competitive organizations for advancing interoperability and total industry advances. The philosophy is one of "all ships rise with the tide." By extending an extranet's reach across multiple organizations, some of which may be directly supplying competitors, an enterprise may realize benefits that far outweigh the risks.

3. Digital Marketplaces and Exchanges

Another business trend is the emergence of on-line digital marketplaces, trading hubs, and exchanges with strong member-centric cultures. These exchanges provide neutral transaction platforms for suppliers

and customers to meet and conduct business, often in the form of reverse auctions. Exchanges usually promise a level of anonymity for all parties so that manufacturers can bid against their own distributors in order to offer potential buyers the lowest prices for their products.

The forecast for digital exchanges is less than optimal, however, for those that remain strictly transaction based. Trading partners will need to embrace collaborative forecasting, planning, and replenishment to form long-term value chain relationships, the by-product of which will be recurring revenues.

II. THE COMPONENTS OF AN END-TO-END EXTRANET SOLUTION

A comprehensive extranet solution encompasses a number of components that interlock to support a distributed cross-organizational network. The extranet solution may be illustrated best within the construct of the classic three-tier client/server architecture. These three tiers include the data and system services tier, the application tier, and the presentation tier. The extranet components that roughly map to each tier are discussed next.

A. Data and System Services Tier

The data and system services tier contains the core technologies that are at the foundation of an extranet solution. Most of these technologies are constantly and rapidly evolving, which means that the foundation of an extranet must straddle the line between solidity and responsiveness to continuous improvement. Some of these technologies include connectivity, extranet servers, databases, security, access control, transaction management, site operations and maintenance, multiplatform interoperability, scalability, and hosting.

1. Connectivity

The nature of an extranet implies connectivity because an extranet must seamlessly connect multiple remote organizations in a closed user group. As previously mentioned, connectivity can be accomplished in two ways: through a direct lease line from intranet to intranet or via a secure link over the Internet. Most organizations utilize the second option by setting up virtual private networks (VPNs). VPNs use advanced encryption and tunneling to permit organizations to establish secure, end-to-end, private network connections over public networks such as the Internet. Implemented correctly, VPNs are scalable, secure, and very cost-effective. A key advantage of VPNs is their compatibility with high-speed broadband technology such as cable and digital subscriber line (DSL). Remote workers can gain high-speed access to their corporate networks over a VPN without resorting to the slow and expensive dial-up services of the past.

2. Extranet Servers

Extranet servers contain the software and systems that power an extranet backbone. They are essential for an effective Web presence, and when properly architected they can scale up to accommodate large numbers of concurrent users. Extranet servers have traditionally been UNIX based due to the robustness and scalability of the operating system; however, the Microsoft's Windows NT/Windows 2000 platform is becoming increasingly popular as Microsoft enhances its offerings.

3. Databases

The most fundamental component of an extranet is the pool of business information that is accessed by the enterprise and its trading partners. As an enterprise and its trading partners get deeper into each other's business processes, there will arise a greater need to provide access to a wider range of databases, documents, and other resources on the respective intranets. These shared resources can then serve as the basis for coordination and collaboration among internal external personnel. For example, an enterprise may choose to provide customers with access to its online purchasing and inventory systems to track the status of their orders or may allow developers real-time access to product specifications and drawings.

An extranet must be able to integrate a variety of databases with Web technology, typically utilizing the Web browser as the front end. The use of extranets as forums for sharing databases can become a powerful means for conducting critical business transactions.

4. Security

Security, at the network and host levels, becomes critical within an extranet. Through the use of firewalls (both physical and virtual), passwords, encryption, and various forms of user authentication, extranets must be able to manage security and accountability.

Interaction and exchange of information throughout the participating organizations must be protected from both the public Internet and the designated extranet members who should not be privy to certain information. For example, suppliers may need to communicate with distributors, but this information should be secured from the view of retailers. The security model must be flexible in its architecture and should be able to provide access controls based on individual, group, organization, transmission type, or other business criteria.

Organizations are increasingly outsourcing the implementation and maintenance of their IP-based extranets/VPNs as security issues become more complex. Service providers must be able to master all of the challenges involved in deploying a remote access VPN. These challenges include client software installation and configuration, dynamically assigned addresses, reliable user authentication, and superb quality of service (QoS) capabilities. AT&T's Managed VPN Tunneling Service, for example, offers services based on standard security and tunneling protocols such as IP security protocol (IPSec), layer two tunneling protocol (L2TP), and layer two forwarding (L2F). Internet key exchange (IKE) is often added to the mix to verify a sender's identity before a transmission is made over an IPSec tunnel.

When outsourcing VPN services, clearly one solution does not fit all organizations. Issues such as remote access requirements, legacy user authentication systems, and overall security and service level requirements must be addressed when planning a secure extranet.

5. Access Control

Access control is at the heart of every extranet. Managing access control, which involves defining user access privileges, managing passwords, user IDs, and authentication schemes, and maintaining user accounts can be a monumental task if the number of users is large. One answer to this challenge is tools that employ lightweight directory access protocol (LDAP) capabilities. With this tool an enterprise can have a global directory that is replicated to all the different services that are running. The directory lists all the directory entries and access control information for people in the enterprise and has a separate section of the directory for people who are at each of the suppliers. The outside parties can literally come into the network and log on to the directory. They authenticate themselves to the enterprise's directory so they appear like internal users, but they have access to only a subset of resources.

6. Transaction Management

Transactions can be defined as each and every request or submission made to the system. An extranet must be able to manage any type of transaction and return the desired result. Transactions can be as simple as transferring a file to as complex as on-line software purchasing, distribution, and licensing. An extranet must have transaction management capabilities that are sophisticated enough to process, track, and manage large volumes of transactions across multiple organizations.

7. Site Operations and Maintenance

UNIX and mainframe systems administrators have long known the importance of maintaining smooth site operations and maintenance. An extranet site is no different: robust utilities that can perform event monitoring and notification, error logging, and system reporting are required. These functions can be performed remotely, and with the advent of Web-based technologies they can be administered through a Web browser on the front end.

8. Multiplatform Interoperability

One of the core requirements of an extranet software solution is that it must be open, portable, and interoperable with different industry standards across multiple platforms. Not only must compatibility with industry standards be achieved at the operating system, hypertext transfer protocol (HTTP) server, and database server levels within an extranet, extranets themselves must also be able to interoperate with one another.

To this end, a group of companies, including Netscape, Oracle, and Sun, has formed an alliance to ensure that their extranets will interoperate. The purpose of the alliance is to ensure that applications and security procedures will interoperate over extranets. The core technologies the alliance is focusing on are JavaScript and Common Object Request Broker Architecture (CORBA) from the object management group and Java. The group also focuses on interoperability of LDAP directory services, signed objects, digital signatures, smart cards, and firewalls, as well as general interoperability of object-based applications.

Absent from the alliance is Microsoft, because the company has its own extranet strategy that is based on

the Windows NT/Windows 2000 platform. Microsoft has built extranet security into the operating system via support for the point-to-point tunneling protocol (PPTP) and secure sockets layer (SSL), thus providing a foundation for VPNs. Microsoft is also working with members of the Internet Purchasing Roundtable to create an open buying on the Internet (OBI) standard and to integrate support for OBI into their commerce products.

9. Scalability

One of the impacts of extranets that has not really been seen before is scalability. Enterprises are now taking data warehousing applications, for example, and giving their suppliers access to those applications. The process of extending those applications to their suppliers might mean that instead of just five or ten internal procurement staff people using that information, suddenly tens of thousands of people are using that application and it may not have been built in such a way to be that scalable. Extranets require such flexibility, extensibility, and scalability in their architecture to accommodate these kinds of changes.

10. Hosting

Hosting refers to where the extranet servers will be housed. An extranet server requires 24 hours per day operation and support, and it must have a continuous high-speed connection to the Internet to be effective. Organizations with 24/7 data center operations may choose to house their extranet servers on-site, but many organizations, regardless of how large they are, choose to co-locate their servers at an Internet service provider's (ISP's) facility. Other organizations choose to house their extranet content on an ISP's servers and save the cost of purchasing and maintaining hardware and software. An ISP can usually furnish the response time and uptime required for business-critical information and extranet performance.

B. Application Tier

Key business activities that are conducted throughout a collaborative extranet environment take place via the application tier. A growing cadre of business applications allow extranet participants to purchase goods and services, manage their businesses, exchange information, and achieve collaborative business objectives. As these business goals and objectives evolve, extranet solutions must be extensible enough to include the addition or modification of applications as required. Applications that are developed and integrated into the extranet must dovetail with the network security plan and not compromise the integrity of existing extranet components. A few of the key extranet applications are described next.

1. Electronic Commerce

The entire sales cycle, from prospecting, qualification, sales close, delivery, and ongoing postsales support, can be performed securely through an extranet. Customer histories and other useful information can be captured simultaneously, greatly enhancing support, maintenance, and the potential for add-on business and upsell to other products.

An extranet can also be set up as an electronic brokerage by offering searchable information about goods and services, reducing the costs of searching for and dealing with customers or suppliers. This self-selection approach transfers more of the selling function to the customer, making transactions timelier and more cost-efficient.

By integrating extranets with existing order entry systems, point of sales terminals, and delivery systems, the costs of managing the buyer and seller relationship are greatly reduced. The ability to facilitate communication and interface processes across the value chain is enhanced, as well as the degree of accuracy at all stages of transactions. Armed with the data and the capabilities of an extranet, a merchant can capitalize on economies of scale, level the production load across many customers, and provide better sales and service at a fraction of the cost of traditional commerce methods.

Contact management systems, e-mail gateways, pager systems, and collaborative groupware applications can be integrated into an extranet, enabling salespeople to proceed through the sales cycle without having to wait for traditional approvals, paperwork, or confirmations. A full-featured extranet can maintain a constant stream of contact throughout the enterprise, so that important business deals are not compromised by missed calls or other frustrations. Customer relationship management, calendaring, and other applications can track a sales prospect through the sales cycle, instantly displaying where a prospect is in each stage of the sales process.

2. Customer Service and Support

The nature of the extranet provides an environment for building stronger customer relationships through col-

laborative, semi-, or fully automated customer service and support. An extranet can provide a self-service model where customers can buy direct, entering their own orders and tracking their own transactions. Even though thousands of customers may be accessing an extranet at any given time, the user experience each of them has is of a one-to-one relationship with the merchant enterprise, which builds increased mindshare and loyalty. Profiles of each customer can be kept in a database and utilized to create dynamic views into the enterprise, customized to the user's level of access and needs. By enlisting the customer as a partner in collaborative product development or in product customization, user training and support are reduced, and customer satisfaction and the potential for add-on sales are greatly increased.

Other customer support functions, such as help desk, e-mail, voice mail, and order status and tracking, can also be greatly facilitated with an extranet solution. A customer can be provided with their own private customized workspace that can be automatically updated with information such as on-line newsletters, tips for product effectiveness, promotional campaigns, customer surveys, and other client-oriented content.

Integration of back-office operations such as call/issue logging, reporting, and analysis to evaluate employee effectiveness can be achieved, giving customers a seamless interface with an enterprise. Customers can also play an active role in the quality of their own support through feedback mechanisms to report problems or questions to a customer support representative. These requests can automatically be forwarded to a corresponding Web-based conference, e-mail, or pager and can then be managed through the extranet to resolution. If a customer requires a phone call or on-site visitation, the customer can fill out a form on-line, which is then automatically forwarded to the appropriate staff member for assignment and follow-up.

3. Version Control and Global Collaboration

A new generation of Web-based version control applications is enabling collaboration via "virtual teams" that come together for the purpose of a project and then disband upon project completion. Version control and configuration management are critical to a well-run extranet. The ability to allow participants to "check out" documents and files, make modifications, and comments on-line and then to check them back in has been expanded to enable concurrent application and site development. A geographically dispersed extranet community can jointly develop projects and not worry about overwriting each other's files or duplicating effort. Since geographical and time constraints are minimized with an extranet, entire virtual corporations have been built using an extranet as a backbone.

Management of distributed teams is especially useful in projects involving multinational commerce, subcontractor relationships, worldwide associations or task forces, specialized development projects, and other activities that require interdepartmental cooperation and/or contribution from a geographically dispersed community.

4. Dynamic Component Assembly

There is no such thing as an off-the-shelf extranet. The requirements of each organization demand that every extranet be individually designed. As with any other software development project, the extranet development life cycle must include extensive requirements definition, analysis, and design and prototyping before construction. Key business goals must be addressed and components must be identified for integration into the solution. With advances in component-based development, extranet applications can include legacy applications that have been wrapped, Java applets, ActiveX controls, and other key building blocks. Dynamic assembly of a diverse range of reusable Web components into cohesive and reliable applications enables the rapid development and deployment of a robust, business-critical extranet solution.

C. Presentation Tier

The presentation tier includes the interface layers between the user and the back-end resources the user wishes to access. Included in the presentation tier of an extranet solution is the capability to customize the interface to each individual user on the fly, according to the user's profile. A group of users can literally be accessing the same data simultaneously, but each user would see a different subset of that data with an entirely different view. In this way individual users and groups can be assured of relevant and useful experiences as they work, collaborate, and communicate within the extranet environment.

The presentation tier can also be extended to include the Web browser front ends to legacy systems and databases and the proliferation of plug-ins that are sent to the client and used when foreign applications are launched. Today's extranet interface layers support a variety of browsers with automatic browser detection and optimization.

III. EXTRANET DEVELOPMENT PLAN TEMPLATE

The following template has been utilized by a number of enterprises to plan their extranets. This checklist is based on a structured software development life cycle approach, as extranet design and implementation should be considered to be a large application development project, rather than a marketing effort.

Extranet Development Plan Template

Part I. The Extranet Opportunity

A. Defining the Role of the Extranet in the Enterprise

Extranet Goals and Objectives

- ❑ What business problem will the extranet solve? Where is the "pain"?
- ❑ The chief executive should understand the reasons for the extranet, the road map, and its milestones and grant full support. How will you achieve this?
- ❑ What is the mission statement of the extranet?
- ❑ Who are the target audiences?
- ❑ What are the characteristics of the vendors, partners, and customers the extranet will reach?
- ❑ How technically adept is each of the audiences?
- ❑ Does management know how an extranet will create business value?
- ❑ Companies should appoint an Internet executive to head all on-line business activities, including fax, electronic data interchange, the corporate intranet, the extranet and electronic commerce. Are you that person? If not, who is?
- ❑ Have you formed a central strategic extranet board within your company so issues can be dealt with from a central corporate policy level?
- ❑ Will the extranet be designed to share data across the enterprise, or will it be limited to supporting one functional area, with a database for that functional area?
- ❑ How will the extranet interact with the corporation's intranet? Different networks may be needed, for example, for high-volume transactions or videoconferencing.
- ❑ Most organizations need reengineering in order to support extranet capabilities. Does your organization understand that probability?
- ❑ Everyone in the company should understand and be excited by the extranet vision. How will that be accomplished?

B. Benefits vs. Costs

Determining ROI

- ❑ Is top management aware of the hard and soft benefits and costs of an extranet?
- ❑ What method will you use to determine the return on investment (ROI) of your extranet?
- ❑ Will the extranet pilot project be able to demonstrate ROI clearly?

Budget for Each Implementation Phase

- ❑ Have you constructed a budget for each phase of extranet implementation?
- ❑ Have you investigated ways to offset costs, such as obtaining vendor sponsorships, coop marketing funds, selling advertising banners, and so forth?

Part II. The Extranet Life Cycle

A. Requirements Definition

Conducting a Needs Assessment

- ❑ Have you conducted a thorough needs assessment in which you have outlined expectations, limitations and demands?
- ❑ Are you comfortable in knowing all the questions to ask? How will you make sure that your questioning is thorough enough to surface all of the requirements?
- ❑ Do you have a set of written questions that will be used to uncover requirements?
- ❑ All extranet users will expect the quality and function of the extranet to be first-rate. The problem arises when a user's expectation is out of sync with the probable outcome. How will you address this potential problem?
- ❑ Part of the requirements analysis should include educating the users about what to expect with the extranet project. How will you accomplish this?
- ❑ Keep in mind that your users' expectations will be guided or misguided by your communication with them. Do you have a clear plan in place for communicating with users on a regular basis?
- ❑ Turnover in your staff and other internal changes should be relayed to users as they relate to the extranet project. Do you have a way to accomplish this without losing morale or support?
- ❑ Do you have a way to let management and users know precisely which players—internal and outsourced—will be handling the project, what changes may occur along the way, and what to expect when they do?
- ❑ Is your extranet development team committed to honest communication throughout the life of the project?
- ❑ Have you evaluated the network and workstation/PC setup throughout your entire company?
- ❑ Will upgrades be made that are necessary to support the extranet?
- ❑ Evaluate your base of expertise. Will you need to send staff to training and/or hire new people?
- ❑ As the extranet evolves, responsibilities on each person's plate are sure to increase and change dramatically. How will you obtain extra resources?
- ❑ Have you developed the acceptance criteria that will

satisfy management and users that the job has been completed?

- ❑ The finished requirements must be *validated* to check that they are a true reflection of the extranet users' needs. Do you have a method for accomplishing this?
- ❑ Is the requirements document comprehensible to nonspecialists?
- ❑ Are users willing to tolerate imperfection as long as the extranet can deliver the functionality users require?
- ❑ Is management in agreement about what quality is and how they would know if the extranet has enough of it?
- ❑ When deadlines approach, do you have an approved method for assessing trade-offs among cost, features, delivery date, and quality?
- ❑ A requirements document should describe exactly *what* you want to do without saying *how* you will achieve it. Does your requirements document do this?

B. Analysis and Design

Extranet Architecture and System Specification

- ❑ Have you taken the initial requirements and expanded them to create the system specification?
- ❑ How will the extranet fit into the environment where it will be operating, and how will it meet the system requirements?
- ❑ How will you verify the specification against the system requirements?
- ❑ Have you broken the system specification down into logical subcomponents?
- ❑ Have you produced a design for each component, showing the data, linkages, and algorithmic structures that can be used to meet the requirements for that component?
- ❑ Can the design be verified against the system specification? Do the subcomponents collectively meet the specification?

Mapping the extranet structure

- ❑ Has an extranet tree structure been developed that visually shows the interrelationship of all of the extranet's components and phases?
- ❑ Does the tree structure diagram clearly map to the system specification?
- ❑ Has the tree structure been presented to and approved by top management and the key user group delegates?

Web site content and interactive features

- ❑ Have you identified all of the existing content for the extranet?
- ❑ Is the content already in electronic form?
- ❑ What new content must be developed?
- ❑ What interactive features are appropriate for each implementation stage?
- ❑ Are the interactive features truly useful or are they gratuitous?

Overall look and feel—The user interface

- ❑ Will the site be designed by a seasoned expert in Web site graphics and design?
- ❑ Is the user interface consistent?
- ❑ Have user interface standards been set?
- ❑ Has the user interface been designed using industry standards?
- ❑ Will the extranet's applications look and feel like other applications developed externally to your organization?
- ❑ Have you explained the rules of how your extranet will work to your users? If it is consistent, then the rules should be simple and few in number.
- ❑ Does the user interface support both novices and experts?
- ❑ Is navigation between screens and on-screen consistent and easy to use?
- ❑ Is color used sparingly in order to speed download time?
- ❑ Put dark text on light backgrounds and light text on dark backgrounds.
- ❑ Are fonts used sparingly and consistently?
- ❑ Are screens simple and uncluttered?
- ❑ Are group boxes and white space used to group logically related items on the screen?

Style guides and templates

- ❑ Is there a clear set of style guidelines and templates for the addition of new content and functionality?
- ❑ Who has final say on approving the look and feel of new content?

C. Prototyping—Selecting a Pilot Project

Why Prototype?

- ❑ Does management fully support the need for the pilot project?
- ❑ Is the pilot project (or prototype) funded?
- ❑ Are there clear agreements to proceed to full implementation of the extranet if the pilot is successful? Are the agreements in writing?

Characteristics of a Good Pilot Project

- ❑ Is the proposed pilot an easily definable project?
- ❑ Does the pilot project solve a real-world business problem?
- ❑ Does it produce the maximum impression with the minimum risk?
- ❑ Have you defined clear criteria for success?
- ❑ Have you identified all of the business groups that will be involved in the buy-off process?
- ❑ Is there a process in place to perform rapid application development (RAD)?

Setting Expectations

- ❑ Are the requirements of your users driving the development of your prototype?
- ❑ What is good about the prototype?
- ❑ What is bad about the prototype?
- ❑ What is missing from the prototype?
- ❑ Do you have criteria for stopping the prototyping process when you find the evaluation process is generating few or no new requirements?
- ❑ Are you working with the people who will use the application when it is done?
- ❑ Have you set a prototyping schedule?
- ❑ Have you evaluated prototyping tools and chosen one?
- ❑ Do you have a plan for enticing the users to work with the prototype?
- ❑ Do you fully understand the underlying business processes behind the prototype?
- ❑ Are you sure you are not investing a lot of time in something that you will probably throw away?
- ❑ Have you documented the purpose and usage of each major component that makes up the prototype?
- ❑ Have you indicated the interfaces of each component and how they interact with one another?

D. Building the Extranet

Security

User access levels and security

- ❑ Have you classified all of your extranet's assets?
- ❑ Do you have a procedure for determining users' access levels?
- ❑ Have you classified the extranet's users and assigned access levels?

Usage policies and procedures

- ❑ Have you developed a clear security policy manual for your company?
- ❑ Has top management co-authored the security policies and procedures, and do they fully support them?
- ❑ Are procedures in place for responding to security incidents?

Firewalls and other security measures

- ❑ Have security tools been identified?
- ❑ Do you have a strong firewall security system in place?
- ❑ Will you be employing encryption technologies?
- ❑ Who will be the "constable" of the extranet?
- ❑ Does the extranet have a demilitarized zone clearly planned or in place?
- ❑ What kind of ongoing security monitoring will be employed?
- ❑ Do all users have anti-virus software on their desktops?
- ❑ What kind of anti-virus measures will be in place for disinfecting files transmitted via e-mail or the Internet?

Version Control

Content development procedures, approval cycles

- ❑ Do you have a clearly defined set of content development, management, and approval procedures?
- ❑ What will be your version control/configuration management system?
- ❑ Have you identified the various virtual teams that will be involved in extranet development?
- ❑ Does your extranet have a private staging area on the Web where content can be reviewed, tested, and modified?
- ❑ Will the extranet be designed so that content can be changed very quickly—overnight in some cases?
- ❑ Turnaround time for content often varies. For example, a press release can move through the departments in less than 1 hour. Marketing materials and technical documents can take several days, especially if the content is new and has never been released in a printed format. What are the procedures for different types of content?
- ❑ Which departments must approve content before it is released? Marketing? IT? Legal?
- ❑ Who is responsible for keeping content current?
- ❑ Who will be tasked with policing the extranet and determining when content is dated?
- ❑ Will business partners and vendors be allowed to contribute content? What are the procedures?
- ❑ Does each business unit run its own set of Web sites, and are they linked to the corporate intranet and/or extranet? If so, how is that managed?

Translation/Localization

Content internationalization and localization

- ❑ Do you have a phased approach to extending your extranet to other countries?
- ❑ How will you localize the content for each country?
- ❑ Will foreign users have input into the content and direction of the extranet?
- ❑ Will your foreign distributors design their own subsites, or will all foreign content be centrally managed?
- ❑ If foreign subsites are developed, how will the content, look and feel, and updates be handled?
- ❑ Will you need to contract outside localization vendors?
- ❑ Will you utilize any machine translation tools?
- ❑ Have provisions been made to compensate for the lack of accuracy by machine translators?
- ❑ How will global e-mail be handled?

Electronic Commerce

Electronic commerce and secure transactions

- ❑ Do you have a separate implementation plan and budget for the electronic commerce portion of your extranet?
- ❑ Does your company have sound business reasons for conducting electronic commerce?
- ❑ Will your electronic commerce model work in tandem with your existing business model?
- ❑ If you conduct commerce on-line, will you be undercutting your existing business partners? How will potential channel conflict be resolved?
- ❑ How much revenue could your company conservatively predict via electronic commerce?
- ❑ How are the security issues going to be addressed?
- ❑ Will your company also distribute software and licensing on-line?
- ❑ How will on-line orders be fulfilled?
- ❑ Are there any export restrictions or value-added tax (VAT) issues to address?
- ❑ Will your company conduct electronic transactions with foreign countries?
- ❑ Has the issue of currency conversion been resolved?
- ❑ Will you be using a third-party clearinghouse to handle the transactions?
- ❑ Does the vendor offer anti-fraud algorithms?
- ❑ Electronic commerce takes much more maintenance than companies anticipate. Do you have a strategy and resources?
- ❑ Forrester Research advocates forming a high-level Internet Commerce Group (ICG). This is not a task force but a full-time staff of about 20 employees coming from both the IT side and marketing groups. Would this model make sense for your company?

Database and Legacy Systems Integration

Analyzing your company's legacy systems and database resources

- ❑ Have you identified all of your company's legacy systems and databases?
- ❑ Have you determined which systems and databases are candidates for integration or front-ending with your extranet?
- ❑ Have you determined how to merge commerce over the Internet with your legacy systems?
- ❑ How will you translate the sales leads or enter orders into the main system? Or, how will you move transactions?
- ❑ Which new extranet users will be allowed to access the legacy systems?
- ❑ Have legacy security issues been addressed to MIS's satisfaction?
- ❑ Are the candidate legacy system interfaces clearly defined?

Bandwidth and Performance Issues

Hardware, software, and bandwidth requirements and costs

- ❑ Have the hardware and software platforms been defined?
- ❑ Is the platform decision truly driven by user requirements, or is the decision a political one, based on strategic alliances?
- ❑ If the decision is based on a strategic business partnership, will the extranet's performance be at risk?
- ❑ Have you planned for different growth scenarios?
- ❑ Has the extranet been designed to minimize download time?
- ❑ Do you have strong policies against spamming?

Server location, hosting, and maintenance

- ❑ Will your company have a dedicated extranet server, or will you host the content on a shared server?
- ❑ Will the server be located in-house, or co-located at an ISP's location?
- ❑ Who will maintain the server?
- ❑ Have you determined your bandwidth requirements?
- ❑ Do you have criteria defined for choosing an ISP?
- ❑ Does your ISP offer any service guarantees?
- ❑ Will caching or mirroring schemes be employed? How?

E. Testing

- ❑ Have you identified a testing method for the extranet?
- ❑ What system will you use for defect counting, tracking, and analysis?
- ❑ Have system-level tests been written?
- ❑ Do they test against the extranet specification?
- ❑ Have unit tests been written for each subcomponent design?
- ❑ How will bugs and change requests be rated and assigned priority?
- ❑ Who will manage the bug list?
- ❑ Have you structured a separate group to test the application components of the extranet? If IS shops do testing, and they are the same people who wrote the code, then there's no objective measurement.
- ❑ By definition, you can not put something under statistical process control if the inputs are always changing. Do you have a release plan that includes "code freezes" at certain stages to enable testing?
- ❑ Do you have a test plan for regression testing?

F. Implementation—Rollout

- ❑ Has a launch plan been developed for the rollout of the extranet?
- ❑ Will there be any associated promotional or media relations activities?
- ❑ What are the criteria for a successful launch?

Implementation phases and priorities

- ❑ Have you broken up the extranet implementation into distinct phases?
- ❑ Has management defined the implementation priorities?
- ❑ Have the components within each phase been prioritized?
- ❑ Have all of the prospective user groups been made aware of the implementation schedule?

Implementation time line and milestones

- ❑ Are there clear success criteria defined for each implementation phase?
- ❑ Are there specific events driving the rollout of each phase?
- ❑ Are the implementation time lines realistic?
- ❑ Are the marketing, sales, and IS departments in agreement on timing?
- ❑ Are there product release deadlines that will be affected by the extranet implementation?
- ❑ Have deliverables been defined for each milestone?
- ❑ Has there been a strong policy put into place to eliminate "feature creep"?

Part III. Monitoring, Measurement, and Maintenance

A. Extranet Statistics and Reporting

- ❑ Have reporting requirements been fully defined by management?
- ❑ Are the reports truly meaningful in a business context?
- ❑ Do any of the extranet partner organizations need the extranet statistics to improve their own businesses?
- ❑ Are there quarterly (or more frequent) meetings scheduled to review the findings with top management?
- ❑ How will the reports be used exactly?
- ❑ Are any of the reports to be kept confidential?
- ❑ How will the results of the reports affect funding and advertising sales?
- ❑ Do extranet users know how the information gathered from the extranet will be used?
- ❑ Have you assured the users that their names or other personal information will not be sold?

B. Staffing/Resources to Maintain and Support Extranet

- ❑ Is there a clear escalation policy in place to resolve problems?
- ❑ Are there clear maintenance procedures in place?
- ❑ Are there guidelines for implementing system upgrades?
- ❑ Have backup procedures been developed?
- ❑ Does an emergency plan exist for blackouts or system failures?
- ❑ Have outside vendors been contracted for maintenance, and are the contracts clearly spelled out?
- ❑ Are there clear guidelines to prevent the premature removal of a primary or backup facility before its replacement is fully operational?
- ❑ Are there any hidden dependencies on old versions of software or hardware components that are no longer available but whose existence is necessary?
- ❑ How will ongoing training be managed?

ACKNOWLEDGMENT

Some material in this article is reprinted by permission from Bayles, D. (1998). *Extranets: Building the Business-to-Business Web.* Upper Saddle River, NJ: Pearson Education, Inc.

SEE ALSO THE FOLLOWING ARTICLES

Firewalls • Internet, Overview • Intranets • Local Area Networks • Mobile and Wireless Networks • Network Database Systems • Network Environments, Managing • Privacy • Security Issues and Measures • Wide Area Networks

BIBLIOGRAPHY

Bayles, D. L. (1998). *Extranets: Building the business-to-business web,* 1st ed. Upper Saddle River, NJ: Prentice-Hall PTR.

McDyson, D. E. (2000). *VPN applications guide: Real solutions for enterprise networks,* 1st ed. New York: Wiley.

Norris, M., and Pretty, S. (2000). *Designing the total area network: Intranets, VPNs and enterprise networks explained,* 1st ed. New York: Wiley.

Phifer, L. (2000). "Tunneling at Layer Two," ISP-Planet, internet.com Corporation.

Szuprowicz, B. O. (2000). *Implementing enterprise portals: Integration strategies for intranet, extranet, and internet resources,* 1st ed. Charleston, SC: Computer Technology Research Corporation.

Wilson, C., and Doak, P. (2000). *Creating and implementing virtual private networks,* 1st ed. Scottsdale, AZ: The Coriolis Group.

Firewalls

Kaushal Chari
University of South Florida

GLOSSARY

bastion host Computers that run proxy software to screen inbound or outbound messages of a protected network.
circuit-level proxy Computer programs that provide security services by screening application layer messages without interpreting application layer protocols.
dual-homed host A special type of bastion host that has two network interfaces for connection to two different networks.
firewalls System consisting of hardware and/or software components that enforce various access control policies and collectively shield organizations from multiple types of security attacks.
internet protocol (IP) The network layer protocol of the Internet that is responsible for routing packets from source to destination hosts.
intrusion detection system (IDS) Security systems that monitor and analyze events for security breaches.
network address translation (NAT) System that shields IP addresses within protected network from being exposed to an external network. This provides security as well as the use of fewer valid IP addresses within an organization.
packet filtering routers Communication devices called routers that use screening rules to filter packets in addition to performing their basic function of routing. They operate at the network and transport layers of the OSI model.
proxy gateways Computer programs that provide security services by screening application layer messages. An *application-specific proxy gateway* uses logic specific to a particular application layer protocol to provide proxy services for that protocol. A *generic proxy gateway* supports proxy services for multiple application layer protocols. In practice, generic proxies are referred to as circuit-level proxies.
stateful packet filtering Screening of packets based on state information.
transmission control protocol (TCP) A transport layer protocol of the Internet that provides reliable connection-oriented service.
user datagram protocol (UDP) A transport layer protocol of the Internet that provides unreliable connectionless service.
virtual private networks Refers to the use of public networks to send confidential data by using encryption and tunneling technologies.

More and more organizations are being connected to the Internet for conducting commerce. With the dramatic growth of the Internet and electronic commerce, network security is becoming a critical issue for many organizations. Most network security solutions deployed in organizations today incorporate firewalls as one of the fundamental components of their overall security plan. Firewalls are systems that protect hardware, software, and data resources within an organization from various types of attacks. Firewall-based security solutions range from simple screening routers that screen IP packets to complex hardware and software systems that screen application layer messages and provide user-level authentication. This article

describes various security threats faced by an organization and then presents different types of firewall security solutions to tackle these threats. A detailed description of firewall technologies is presented along with their risks and benefits to facilitate the understanding of various firewall security solutions.

I. INTRODUCTION

With the rapid growth of the Internet, more and more organizations are interconnecting with external networks to support various activities. These activities range from exchanging messages among various locations of the organization for supporting operations to providing information to potential customers and conducting e-commerce transactions such as buying and selling over the Internet. A whole new class of dot.com companies has appeared during the past few years with the purpose of participating in the e-commerce market. Traditional procurement activities that were once conducted using phones and faxes are now being replaced by supply chain activities centered around electronic markets. The security implications of adopting new business practices that leverage an organization's interconnection to an external network are enormous, since an organization's hardware, software, and data resources are now vulnerable to security attacks from the external network. So, what can organizations do to safeguard hardware, software, and data resources? This article addresses this question and discusses various security solutions based on *firewalls* to secure organizational resources. Firewalls alone do not provide complete security, but rather are used as part of an overall security plan.

Firewalls are systems consisting of hardware and/or software components that enforce various access control policies and collectively shield organizations from multiple types of security attacks. Firewalls are placed at the interface of an organization's protected network and an external network. They are effective in blocking unwarranted traffic coming into the protected network from the external network, while allowing legitimate traffic to pass through. Firewalls are, however, less effective in preventing traffic from leaking out of the protected network. Firewall components such as packet-screening routers operate at the network and transport layers of the OSI model by filtering data packets based on information provided in the headers of network layer protocols (e.g., IP) and transport layer protocols (e.g., TCP and UDP), while components such as application level proxy gateways filter traffic based on user IDs and application messages. The various components of a firewall system handle different types of security attacks.

There are usually three main objectives behind security attacks: intrusion, denial of service, or information theft. Successful intrusions enable attackers to use hardware and software resources within the protected network the same way a legitimate user would use them. Denial-of-service attacks aim to prevent services within the protected network from being offered. For example, an attacker may attempt to crash a competitor's Web server so that the services offered by the competitor to its customers are denied or drastically degraded. An information theft is carried out with the purpose of the attacker gaining such information as credit card numbers, computer accounts, passwords, private information, or trade secrets. In October 2000, Microsoft reported a security breach in which an attacker gained access to top secret source code of Microsoft software. To understand how various attacks can take place, it is imperative to know how Internet protocols work and how these protocols can be exploited to wage a security attack.

This article is organized as follows. Section II presents a brief overview of the TCP/IP suite and services commonly used over the Internet. Section III describes the various types of security attacks in detail. A detailed description of various firewall components is presented in Section IV. This is followed by a description of various firewall architectures in Section V. Section VI presents an overview of intrusion detection systems. This article concludes in Section VII with discussions on notable firewall products and emerging firewall technologies of the future.

II. TCP/IP SUITE AND SERVICES OVERVIEW

The communication protocols used on the Internet consists of protocols at the network layer (e.g., IP, ICMP), transport layer (e.g., TCP, UDP), and application layer (e.g., FTP, SMTP, HTTP). This collection of Internet protocols is commonly referred to as the TCP/IP suite. Figure 1 illustrates the Internet protocol stack.

A. Network Layer

1. Internet Protocol (IP)

IP is a network level protocol that is responsible for transporting data in discrete chunks of bits known as packets from the source machine to the destination machine. IP is an unreliable protocol because it merely

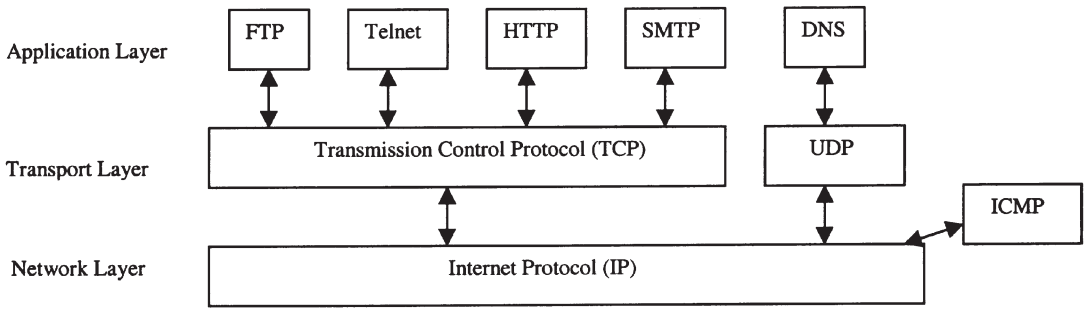


Figure 1 Internet protocol stack.

provides a best effort service and does not guarantee the delivery of packets. IP does not establish any logical connections between source and destination machines, and IP packets belonging to the same message can take different routes to reach a particular destination. IP packets are routed independently of each other and are known as datagrams. All intermediate systems such as routers and end systems such as servers and workstations that are connected to the Internet must implement IP for communication. Typically, an IP address is associated with every network interface of devices connected to the Internet. An IP packet has a header and a data portion. The IP header carries control information for routing packets through the Internet and is analogous to the address label on an envelope, whereas the data portion of an IP packet corresponds to the contents of the envelope.

The current version of the IP is still IPv4, although IPv6 has been proposed. In this article, any references to the IP would imply IPv4, unless stated otherwise. Most fields in the header of IPv4 packets have implications for security that include *Source IP Address, Destination IP Address, Protocol, Fragment Offset,* and *Source Routing Option. Source IP Address* and *Destination IP Address* represent the IP addresses of the source and destination, respectively. *Protocol* identifies the protocol whose data are stored in the data portion of the IP packets. Examples of protocols include Transmission Control Protocol, User Datagram Protocol, and Internet Control Message Protocol. Often, IP packets are fragmented to meet the maximum packet size requirements of intermediate networks on the route from source to destination. The various fragmented pieces of IP packets are then reassembled together at the destination host by using the information in the *Fragment Offset* field. This field indicates the relative position of the fragment data in the original IP packet. IP allows an option whereby the source can specify the route a packet should take to reach the destination. This route is stored under the *Source Routing Option* field.

2. Internet Control Message Protocol (ICMP)

ICMP is used for status reporting over the Internet. ICMP provides error and control messages that are used in a variety of situations. For example, when an IP packet cannot be delivered due to a destination not being reachable a *Destination Unreachable* ICMP message is used. When a route is to be changed a *Redirect* message is used. The ability to reach an IP address is checked using an *Echo Request* message that is used in the Ping utility. To slow down transmission at the source, a *Source Quench* message is generated by the router and sent to the source. ICMP packets occupy the data portion of an IP packet while being transported over the Internet. Although ICMP messages are carried by IP, ICMP is regarded to be at the same level as IP, the network layer level. ICMP header fields that are relevant for enforcing security include the *Type* field, which identifies the type of ICMP message, and the *Code* field, which reports the error code.

B. Transport Layer

1. Transmission Control Protocol (TCP)

TCP is a transport level protocol of the Internet that provides reliable, end-to-end communication between two processes. The requesting process, often known as the client, requests services from the server process. Both client and server processes are accessible on their respective machines by their TCP port numbers assigned to them. Many standard application layer services have *well-known* TCP port numbers assigned by a central authority. For example, a Simple Mail Transfer Protocol server operates at the well-known TCP port 25. TCP carries bytes of data from the higher level process by packaging it into TCP segments. TCP segment data are then packaged by IP into the data portion of IP packets. This is illustrated in Fig. 2.

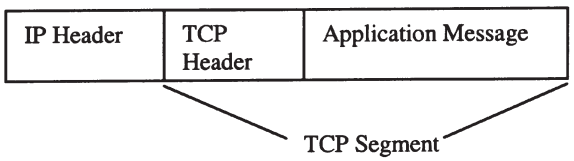


Figure 2 IP packet containing TCP segment as payload.

TCP segment header fields that are relevant for enforcing security include *Source Port, Destination Port, Sequence Number, ACK Flag, SYN Flag,* and *RST Flag. Source Port* and *Destination Port* provide the addresses of the source and destination communicating processes on their respective machines. *Sequence Number* locates the data carried by the TCP segment in the sender's byte stream. *ACK Flag* is used to indicate whether the TCP segment carries an acknowledgment, and the *SYN Flag* is used during the handshaking procedure known as the three-way handshake (Fig. 3) to establish a TCP connection. The client host initiates the handshake by sending a TCP message with the SYN flag set along with a random starting sequence number p of the byte stream in the direction of the server. The server host upon receiving the TCP SYN message sends a TCP message with the SYN and ACK flags set, a random starting sequence number q of the byte stream to the client, and the sequence number $p+1$ of the byte stream expected next from the client. The client, upon receiving the sender's TCP message, then sends a TCP message with the ACK flag set along with sequence number $q+1$of the byte stream expected next from the server. This completes the three-way handshake. The TCP header also includes the *RST Flag* field for resetting a connection.

2. User Datagram Protocol (UDP)

UDP, just like TCP, is also a transport layer protocol of the Internet and is used for transporting data from the source process to the destination process. However, unlike TCP, UDP is unreliable and does not guarantee the delivery of UDP packets. UDP has low overhead and is used by application processes to avoid the large overhead associated with TCP. UDP header fields that are relevant for enforcing security include *UDP Source Port* and *UDP Destination Port.*

C. Application Layer

1. File Transfer Protocol (FTP)

FTP is an application layer protocol that is used for transferring both text and binary files over the Internet. The FTP client process uses a randomly assigned TCP port number x, usually above port number 1023, to establish a control channel to request an FTP connection with an FTP server process listening at the well-known TCP port 21. In the normal mode, the server process uses TCP port 20 to set up a data connection with port number y on the client where $y > 1023$ and $x \neq y$. In the passive mode, the FTP server process uses a port above 1023 for data connection to port y on the client.

2. Simple Mail Transfer Protocol (SMTP)

SMTP is an application layer protocol that is used to transmit electronic mail. The SMTP sender process uses a randomly assigned TCP port above 1023 to send SMTP messages to the SMTP receiver process that is listening at the well-known TCP port 25.

3. HyperText Transfer Protocol (HTTP)

HTTP is an application layer request response protocol that is used to access Web pages over the Internet. The Web client process uses a randomly assigned TCP port above 1023 to send HTTP request messages over TCP to a Web server that is typically listening at the well-known TCP port 80.

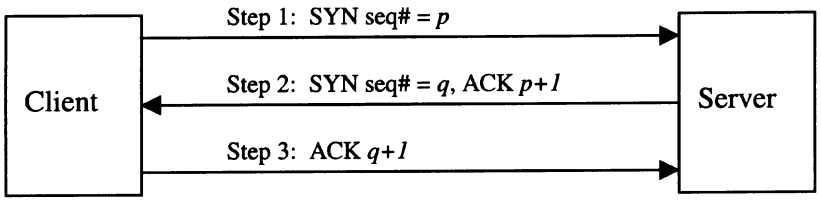


Figure 3 Three-way handshake to establish a TCP connection.

4. Telnet

Telnet is an application layer protocol used to log into a remote computer. The Telnet client process uses a randomly assigned TCP port above 1023 to initiate a Telnet session with a Telnet server process that is listening at the well-known TCP port 23.

5. Domain Name Service (DNS)

DNS is a service available on the Internet to resolve domain names into numeric IP addresses. DNS servers maintain a mapping of domain names to IP addresses. A DNS client issues a DNS query to the local DNS server. If the local DNS server does not have the mapping information to resolve the query, the local DNS server contacts other DNS servers. DNS client processes use UDP to interact with DNS servers that listen at the well-known UDP port 53.

III. NATURE OF SECURITY ATTACKS

As indicated previously, intrusion, denial of service, and information theft are the major goals of security attacks. Most of these attacks exploit vulnerable features of TCP/IP protocols to carry them out. The various kinds of security attacks with different objectives are presented in the next subsection.

A. Security Attack Goals

Various security attacks are summarized in terms of their goals in Table I. While the listing of security attacks in Table I is not exhaustive, Table I does cover many common security attacks. It can be seen from Table I that some security attacks can support multiple goals.

Table I Security Attack Goals

	Goals		
Security attacks	Intrusion	Denial of service	Information theft
Smurf attack		X	
SYN flood attack		X	
Ping of death attack		X	
Land attack		X	
IP spoofing	X	X	X
Tiny fragments attack	X		X
Overlapping fragments attack	X	X	X
Trojan horse attack	X	X	X
Packet sniffing			X
Port scanning	X	X	X
Session hijacking	X		X
Router redirection attack			X

B. Types of Security Attacks

1. Smurf Attack

In the case of a smurf attack, the attacker's objective is the denial of service at the victim host. A utility known as Ping sends ICMP *Echo Request* messages to a target machine to check if the target machine is reachable. The target machine, upon receiving ICMP *Echo Request* messages, typically responds by sending ICMP *Echo Reply* messages to the source. When carrying out a smurf attack, an attacker (host X in Fig. 4) uses a broadcast address for the destination address field of the IP packet carrying the ICMP *Echo Request* and the address of the victim host (host Y in Fig. 4) in the source address field of the IP packet. When the ICMP *Echo Request* messages are sent, they are broadcast to a large number of stations (1 . . . N in Fig. 4). All of these stations then send ICMP *Echo Reply* messages to the victim device, thereby flooding the victim device and perhaps bringing it down.

2. SYN Flood Attack

A SYN flood attack is carried out when a TCP connection is being established. The purpose of a SYN flood attack is to deny service at the victim host. The attacker initiates multiple TCP connections by sending multiple TCP messages with the SYN flag set. The TCP messages are carried by separate IP packets whose source addresses are spoofed by the attacker to that of some unknown host. The receiving host, after allocating precious system resources such as memory, sends back a TCP message with the SYN and ACK flag set for each TCP message received by the source. Under normal operations, the source that initiates a TCP connection request sends a TCP message with the ACK flag set to complete the three-way handshake to begin conversation (see Fig. 3). However, in the case of a SYN flood attack, the source host address is spoofed to an invalid and unreachable address. Therefore, the victim host does not receive any responses

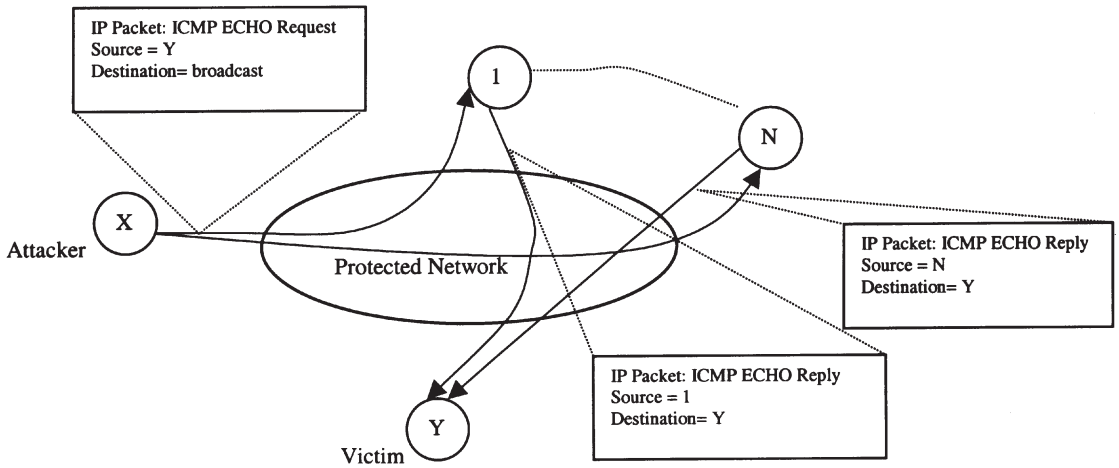


Figure 4 Smurf attack using IP spoofing.

back. Precious resources are blocked at the victim host till various TCP connections time out, typically after 75 seconds. While resources are blocked, new legitimate TCP connection requests are not entertained due to lack of resources. In some instances the victim host also crashes due to lack of resources.

3. Ping of Death Attack

In this attack, the attacker sends oversized ICMP *Echo Request* messages to the victim host. The IP packets carrying ICMP *Echo Request* messages get fragmented as they are routed over the Internet. At the destination host (i.e., the victim), while reassembling IP fragments, the resultant IP packet exceeds its defined maximum size. A poor implementation of IP/ICMP protocol at the victim host would cause a buffer overrun condition, thereby leading to the crash of the victim host.

4. Land Attack

A land attack is also carried out with the purpose of denying service at the victim host. An attacker spoofs the source address of the IP packet to the IP address of the destination host, and the source TCP port to the destination TCP port. This causes the victim (which happens to be both the source and destination host in the IP packet) to crash.

5. IP Spoofing

IP spoofing can be used in different types of attacks. In the case of a smurf attack discussed above, the attacker spoofs the source address of the IP packet to that of the victim host with the objective of denial of service at the victim host. IP spoofing is also done with the objective of intrusion and information theft. The attacker modifies the source address of an IP packet to an address that the firewall is likely to trust, such as the address of a host in the protected network. The firewall would then forward the spoofed IP packet to the protected network. The attacker would use the source routing option to include an IP address in the route that is accessible to the attacker. In this case, the response retraces the route specified in the source routing option and is likely to be intercepted by the attacker.

6. Tiny Fragments Attack

This attack is carried out with the purpose of intrusion or information theft. The IP packets are fragmented into very small pieces such that the TCP header from the attacker requesting TCP connection does not appear in the data portion of the first IP packet (which is empty). This can sometimes circumvent the filtering rules in routers, thereby enabling all packets to pass through.

7. Overlapping Fragments Attack

This attack can lead to denial of service, intrusion, and information theft. The attacker constructs IP fragments in such a way that data from two IP fragments overlap. Many systems that receive overlapped fragments and are not robust enough crash, thereby leading to denial of service. An attacker can add an unacceptable TCP header in the data portion of the second packet that overlaps with the first IP packet

carrying an acceptable TCP header. IP screening routers could let all packets in, based on the acceptable TCP header information in the first packet. Thus the unacceptable TCP header in the second packet sneaks into the protected network and leads to connections that allow intrusion or information theft.

8. Trojan Horse Attack

These attacks are carried out with the purpose of denial of service, information theft, or intrusion. Attackers plant trojan horse programs into the victim system. These programs are capable of sniffing packets and collecting account names and passwords and then passing the information to the attacker. Trojan horses can also execute commands that crash the victim system, thereby leading to the denial of service. Trojan horses are capable of isolating victim hosts when they send ICMP *Destination Unreachable* messages to other hosts and routers even when the victim host is reachable.

9. Packet Sniffing

This is done with the purpose of stealing information. Often trojan horse programs are planted at certain strategic locations. These programs sniff packets containing data of those applications layer protocols that send sensitive information in the clear. Telnet and FTP are good examples where user IDs and passwords are often sent unencrypted.

10. Port Scanning

This is done to gather intelligence on ports that are open at the victim host for the purpose of attacking them. Via port scanning, an attacker is able to identify the victim host's operating system and get more information as to what attacks might be successful at the victim host. An attacker sends a TCP message with the SYN flag set to request connection at a particular port. The victim host can send a TCP message with the SYN and ACK flags set to the attacker, in which case the victim is ready to allow a connection at that port. Alternatively, the victim host can send a TCP message with the RST flag set. In this case, the attacker knows that the port is closed. If the attacker fails to receive a TCP message with the RST flag set, it then assumes that the port is open. A port being open signifies that a particular service is running and reachable. The attacker then has the opportunity to try different attacks on the identified open ports.

11. Session Hijacking

An attacker can hijack a TCP session that is already established with the purpose of intrusion or information theft. This is possible when the hijacker is able to intercept packets sent between two hosts as part of a normal TCP session. The hijacker then has access to the TCP sequence number used and is then able to construct new TCP segments with the right sequence number.

12. Router Redirection Attack

This attack is done with the purpose of information theft. Hosts rely on routing tables to route packets efficiently. The routes in these tables can be updated when a trusted router discovers a better route between two hosts. Typically a router would send an ICMP *Redirect* message to report a better route to the host. The host would then update its routing table and then send packets on the new route. An attacker can pretend to be a trusted router by address spoofing and send ICMP *Redirect* message to the host, directing the host to send packets on a route where the attacker can intercept packets.

IV. FIREWALL COMPONENTS

Organizations are vulnerable to a variety of security attacks as discussed in the previous section. No single device is capable of handling the entire gamut of security attacks. Instead, a system of hardware and software components is used to handle various types of security attacks. This system of components is known as the firewall system. Firewalls enforce various access control policies at different levels. At the network and transport layers, the packet filtering router component in the firewall system is used to screen IP packets based on the information contained in IP, TCP, and UDP headers. At the application level, application level proxy gateways are used to screen application level requests based on user ID and application level commands or message types. A detailed description of various hardware and software components used in firewall systems is presented next.

A. Standard Packet Filtering Routers

Routers route packets that arrive at their inbound ports. The routers used over the Internet are IP routers, because they are designed to read information stored in

the headers of inbound IP packets to route them. Typically, when an IP packet arrives at a router, it waits in an inbound queue. The router then examines the header of the inbound packet to get the destination address of the packet. The router consults an internal routing table to determine how the packet should be routed. Finally, the router places the packet on an outbound queue of the link associated with the route selected. In the case of packet filtering routers, an extra step is added while routing packets. Before sending packets to an outbound queue, a packet filtering router applies a set of packet filtering rules to determine if the packet should be forwarded or be discarded. These rules typically use information such as IP source address, IP destination address, TCP destination port address, and TCP ACK flag.

Examples of various filtering rules are as follows. The example rules below have been expressed in a high-level generic notation. Note that routers from different vendors require rules be specified in their native syntax.

- *Example 1:* To deny an inbound Telnet connection coming from outside into the protected network.

If Direction = Inbound and Source Address

= External and Destination Address

= Internal and Protocol = TCP and Destination Port

= 23 then Deny.

Note that the Telnet server listens at TCP port 23. Hence any inbound IP packet containing TCP payload from an external network address and destined for TCP port 23 on an internal host is filtered.

- *Example 2:* To filter IP packets whose source addresses have been spoofed.

If Direction = Inbound and Source Address

= Internal then Deny.

This rule is associated with the router interface that links the router to the external network. Thus if an attacker spoofs the source address of the packet to an internal address, it can be easily detected since no inbound traffic from the external network arriving at that router interface would have the IP source address of an internal host on the protected network.

- *Example 3:* To permit a Web request to a particular internal host with IP address = X.

If Direction = Inbound and Protocol

= TCP and Destination Address

= X and Destination Port = 80 then Permit.

Often organizations have a single Web server available for public access over the Internet. In the current example, the Web server is located on a host with IP address X and uses the default TCP port 80. Any inbound IP packets carrying TCP segment meant for port 80 on host X are allowed to pass through. It is often prudent to deny all access by default and then add rules that selectively grant access to certain IP addresses for certain application layer services. The ordering of the rules is therefore very important.

B. Stateful Packet Filtering

Certain type of attacks may be beyond the scope of a standard packet filtering router to handle. For example, if TCP connections to the outside network are always required to be initiated from a host within the protected network then any TCP connection initiated from outside should be denied. One way to enforce this in standard packet filtering routers is to use the following rule:

If Direction = Inbound and Source Address

= External and Protocol = TCP and SYN flag

= set and ACK flag = not set then Deny.

Readers will recall that TCP connections are set up using a three-way handshake. The initiator of the TCP connection sends a TCP message with the SYN flag set but not the ACK flag. In response to this, the receiver sends a TCP message with both SYN and ACK flags set. Now, consider the situation where an attacker sends a TCP message with SYN and ACK flags set. A standard packet filtering router will perhaps allow the TCP message to pass through, assuming that the TCP message is in reply to a connection initiated by a host on the protected network. The TCP message is then directed to the destination port on a protected host. Although the TCP protocol at the destination port would drop the TCP message, the entry of such TCP messages would expose ports of hosts on the protected network to outside traffic, thereby leading to port scanning.

The above example illustrates the need for a firewall component that keeps track of state information

such as various active TCP connections initiated from within the protected network so that it can match inbound TCP messages to outbound TCP messages, thereby filtering unwanted traffic. These sophisticated firewall components that use state information to filter packets are known as *stateful* or *dynamic packet filtering* firewalls. Stateful inspection firewalls can maintain state information on many standard protocols. They can create rules on a dynamic basis that last just for a short duration so as to have the ability to open a limited time window for response packets to pass through from a specific external host to the internal host that sent the initial message. For example, UDP is used to carry DNS queries from hosts on the protected network to DNS servers on the external network. A stateful inspection packet filtering firewall will be able to selectively allow only a specific inbound UDP packet from an external network that carries the DNS response back for a DNS query sent from the protected network while blocking the rest of the UDP traffic.

C. Network Address Translation

Network address translation (NAT) is done to shield internal IP addresses as well as TCP/UDP ports within the protected network from being exposed to the outside world. Therefore, by not knowing the address of a protected host, an attacker may not be able to launch a security attack on the host. Network address translation also enables an organization to obtain fewer valid IP addresses. Because addresses assigned to internal hosts are hidden from the outside world, there is a lot more flexibility in assigning addresses to internal hosts.

Figure 5 illustrates how network address translation works. An internal client sends out a request to an external Web server located at 18.181.0.31 that is listening at TCP port 80. The IP packet, carrying a request from the internal client, is intercepted by a network address translation firewall, which then performs network address translation. The firewall changes the source address of the IP packet to its own address, i.e., 131.247.0.1 in Fig. 5, and the TCP source port number is changed to port 4000. The IP packet is then sent to the destination Web server. The response from the Web server is directed to the firewall, which then changes the IP destination address and TCP destination port number to that of the internal client and then transmits the packet. Network address translation firewalls maintain inbound/outbound IP address as well as TCP port address mappings. Inbound packets to a NAT-enabled firewall that do not match one of the current inside-to-outside mappings are discarded, thereby providing additional security.

D. Generic Proxy Gateways

A generic proxy gateway is a process that prevents an external host from establishing a direct application layer connection with an internal host on the protected network and vice versa. The term *generic* implies that the process can handle different application layer protocols. When a generic proxy gateway is used, an external host establishes a TCP connection with a proxy gateway for inbound traffic. The proxy gateway then establishes a separate TCP connection with an internal host. In the case of outbound traffic, an internal host establishes a connection with the proxy gateway and the proxy then establishes a separate TCP connection with an external host. The proxy gateway verifies the legitimacy of the connection while setting up connections. Once the two TCP connections are set up, the proxy gateway transfers data from one end to the other. Generic proxy gateways work with many application layer services. They provide user-level authentication and extensive logging capabilities and are good at filtering out unwanted TCP connections and in shielding internal hosts from various types of attacks including TCP SYN flood attacks. In practice,

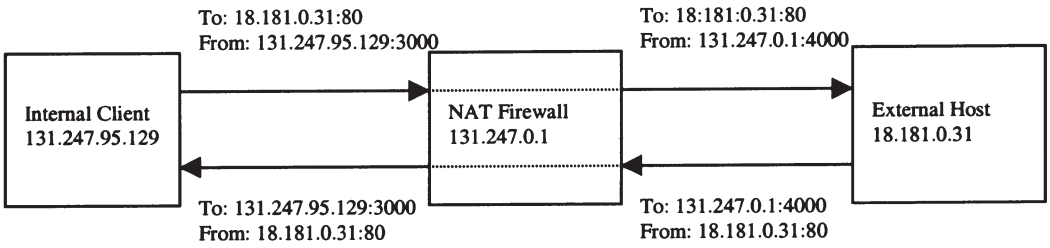


Figure 5 Network address translation.

generic proxy gateways are also referred to as *circuit-level proxies* because they handle application layer messages without interpreting them. Circuit-level proxies are not capable of filtering messages based on the contents of the application layer data. Thus, it is quite possible for application messages containing dangerously formatted data or forbidden commands to slip through circuit-level proxies. Generic proxy gateways shield IP addresses of the internal hosts on the protected network from being exposed to the outside world. This is due to the fact that all IP packets from the external network are directed to the *well-known* IP address of a proxy server host. The responses from internal host to the external hosts are relayed by the proxy server host with the source address of the IP packet set to that of the proxy server host.

E. Application-Specific Proxy Gateways

An application-specific proxy gateway is a process that prevents an external host from establishing a specific type of application layer connection directly with an internal host on the protected network and vice versa. Unlike circuit-level proxies, the logic of an application-specific proxy gateway is dependent on a particular application layer protocol or a service. For example, a separate FTP-specific proxy gateway is required for FTP that is different from the Telnet-specific proxy gateway used for Telnet. In addition to having the capabilities of a generic proxy gateway, an application-specific proxy gateway has the ability to filter application layer messages based on the contents of the application layer messages, user IDs, etc. For example, an application-specific Web proxy can filter HTTP request messages from an internal host that are directed to forbidden Web sites. Application-specific proxy gateways also provide extensive logging capabilities to log connections. Also just like generic proxies, the IP addresses of various internal hosts on the protected network are not exposed to the outside world.

To enable application-specific proxies to intercept messages from internal hosts on the protected network that are directed to external hosts, the following four schemes can be utilized. First, proxy-aware clients could be used, in which case an internal client is configured to send application messages to the proxy server instead of the regular server. This may entail using either modified clients or require setting the location of the proxy server on a standard client. Many standard Web browsers such as Netscape Navigator and Internet Explorer allow users to specify the location of proxy servers. Second, the operating system of the client host could be configured to send any client messages to the proxy server. Third, a router could be used to direct all messages from an internal client to proxy servers. In this case, no modifications are required at the client or the operating system. Fourth, users could be required to send all their external messages to appropriate proxy servers.

F. Bastion Hosts

Bastion hosts are computers that are exposed to the outside world. Bastion hosts support various proxy gateways. To improve their performance, often multiple bastion hosts are deployed, where each host would support few proxy services. For example, separate bastion hosts may be used to run Web proxy and e-mail proxy services. Bastion hosts are typically "hardened" to protect them against security breaches by disabling all unessential services and configuring the operating system of the bastion host to reduce security risks. A special type of bastion host is the *dual-homed host*. In the case of a dual-homed host, two network interface cards connect the host to two different networks. Packets arriving at one interface (i.e., the external interface) of the dual-homed host are not directly sent over the other interface to the protected network. Instead, a dual-homed host may create a new IP packet after verifying the legitimacy of the packet before transmitting the packet on the protected network interface. To prevent automatic forwarding of a packet from one interface to the other, the routing feature in the dual-homed host is usually disabled.

G. Virtual Private Network

Often organizations use the Internet to send data from one organizational location to the other when leasing private lines is not cost effective. The Internet, being a collection of public domain networks, is not under the control of the organization. Thus packets sent over the Internet are susceptible to various security attacks such as packet sniffing, spoofing, and session hijacking. To prevent an attacker from deciphering the contents of a packet captured over the Internet, the packets are encrypted before they are sent. This is what a virtual private network (VPN) does. VPNs use protocols such as L2TP and IPSec to tunnel packets across the Internet. IPSec, developed by the Internet Engineering Task Force (IETF), is a

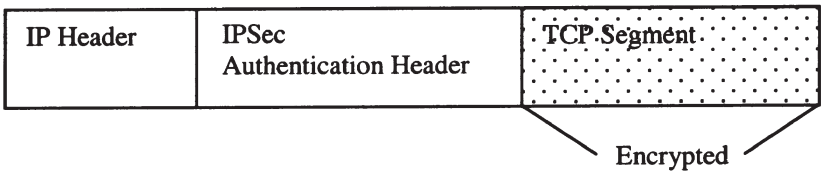


Figure 6 IPSec *Authentication* header and IP packet in the transport mode.

set of protocols that offers authentication as well as privacy at the IP layer level. IPsec supports two headers: an *Authentication* header to carry authentication information (Fig. 6) and an *Encapsulating Security Payload* (ESP) header to support both privacy and authentication (Fig. 7). IPSec operates in two modes: transport and tunnel modes. In the transport mode, the IP packet payload is encrypted while the IP packet header is not as shown in Figs. 6 and 7. In the tunnel mode, the entire IP packet (including the packet header) is encrypted and sent (i.e., tunneled) within another IP packet as shown in Fig. 8. Thus, information in the header of the tunneled IP packet is completely hidden. IPSec supports authentication of the VPN connection using keys and digital signatures and ensures data confidentiality and data integrity. Many firewall products support IPSec.

H. Packet Filtering Routers versus Proxy Gateways

Packet filtering routers operate at the network and transport layers and in addition to performing the basic function of routing, they use screening rules to filter packets. These rules use IP addresses, IP options, TCP/UDP ports, and ICMP message types in making filtering decisions. Packet filtering routers do not open TCP/UDP payloads to filter based on the contents of the application layer messages. Thus, packet filtering routers have higher performance in terms of throughput compared to proxy gateways running on bastion hosts since additional processing is avoided by the packet filtering routers. The flip side to this is that packet filtering routers do not have the ability to support user-level authentication and filtering based on application level messages. Thus, lot of attacks can sneak through a packet filtering router via application layer messages. The filtering rules used in packet filtering routers can be complex, and with little or no capability to support logging, packet filtering routers may not be able to filter all attacks. Application-specific proxy gateways, due to the fact they use application layer messages to make filtering decisions, tend to do more processing than packet filtering routers. Thus proxy gateway performance in terms of throughput is generally lower than packet filtering routers. A proxy gateway has the ability to provide finer-grained control by enforcing user-level authentication and service-specific controls. It has the capability to filter invalid messages for a given protocol. Proxy gateways also have the ability to log more useful and "ready-to-use" information than packet filtering routers. Because proxy gateway logic is often specific to a given application layer protocol or service, often many new applications or "one-of-a-kind" applications do not have their corresponding application-specific proxy gateway products available in the marketplace. Thus application-specific proxies are often custom built.

V. FIREWALL ARCHITECTURES

The previous section presented various components of a firewall system. These components can be put together in different configurations to create a variety of firewall architectures.

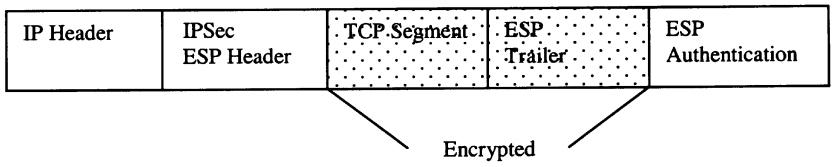


Figure 7 IPSec *Encapsulating Security Payload* and IP packet in the transport mode.

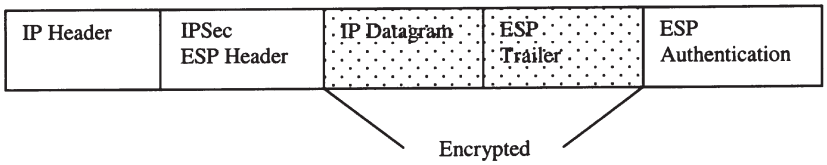


Figure 8 IPSec *Encapsulating Security Payload* and IP packet in the tunnel mode.

A. Screening Router Architectures

This is the simplest architecture in which a router screens inbound and outbound packets. Figure 9 illustrates this architecture. This architecture provides only limited security because there is no application level screening. Therefore attacks concealed in application layer messages can slip through. Also, there is limited logging and no support for user-level authentication. The advantages of this architecture include high performance and low costs.

B. Dual-Homed Architectures

In the dual-homed architecture shown in Fig. 10, the entry to the protected network is controlled by a dual-homed host that has two network interfaces, one linking to the external network and the other to the protected network. To prevent a dual-homed host from directly routing packets, the routing function is disabled. The dual-homed host receives all packets from external hosts and after verifying the legitimacy of the packets (via IP/TCP header information screening, stateful inspection, etc.), it sends out new packets containing the payload of external packets received. The dual-homed host could support firewall software for packet screening (including those based on stateful inspection) as well as proxy gateways. Therefore, application level screening is also possible. One disadvantage with the dual-homed host is that its throughput is not as high as that of many packet-filtering routers.

C. Screened Host Architectures

In the screened host architecture shown in Fig. 11, most inbound external traffic permitted by the screening router is directed to the bastion host only. If a particular application-specific proxy is not supported on the bastion host or when an internal host is connecting to a "safe site," then on a case-by-case basis, response packets from the external network may be allowed to directly go to an internal host. The screening router provides the first level of defense by filtering all packets that are not directed to a legitimate port on the bastion host, with few exceptions involving response packets discussed above. Once the bastion host receives a packet, it unpacks the packet, and the appropriate proxy gateway on the bastion host then ex-

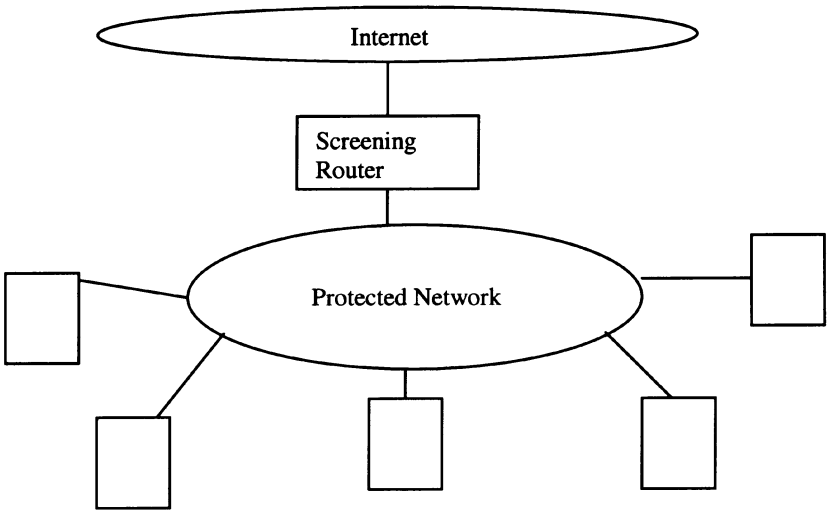


Figure 9 Screening router architecture.

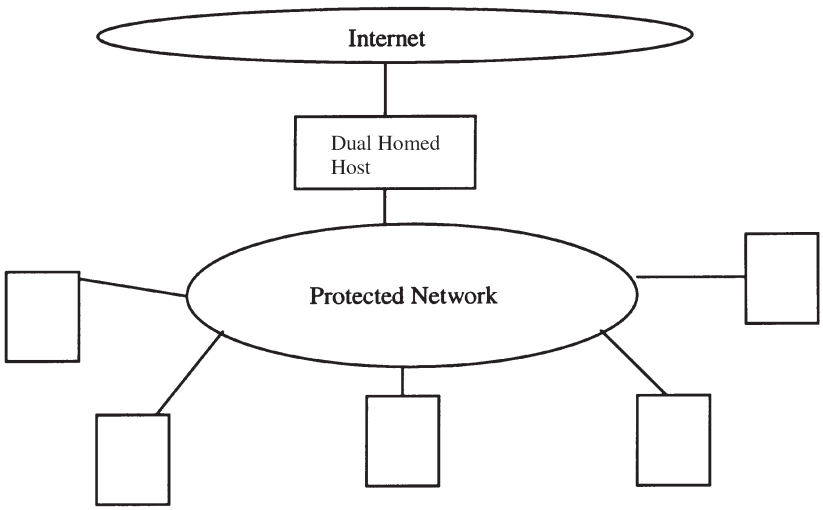


Figure 10 Dual-homed architecture.

amines the application data. Once the application data are verified, the bastion host creates new packets containing application data and sends them to an appropriate internal host on the protected network. Most internal hosts cannot directly send packets to an external host. Instead packets from internal hosts are sent to the bastion host where the proxy gateway screens packets based on user ID and application data. Once the legitimacy of the packets is established, the bastion host sends the packets to the external network. The screening router does not filter packets that are sent by the bastion host to external hosts. The screened host architecture provides more security compared to screening router and dual-homed architectures presented above. The security is enhanced when the screening router traffic is passed through a stateful inspection packet filtering firewall. The main weakness of the screened host architecture is that if the bastion host is compromised, all internal hosts are vulnerable to security attacks.

D. Screened Subnet Architectures

The screen subnet architecture (Fig. 12) is more secure than the screen host architecture. A perimeter

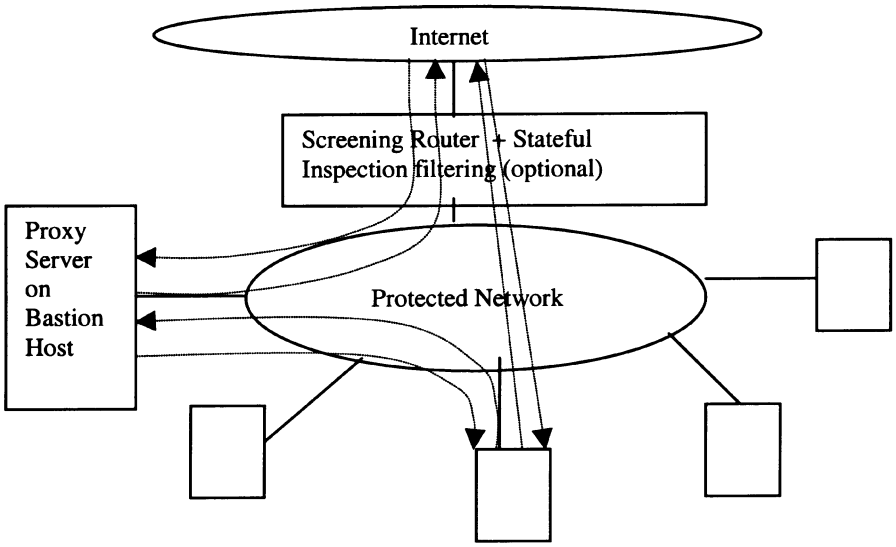


Figure 11 Screened host architecture.

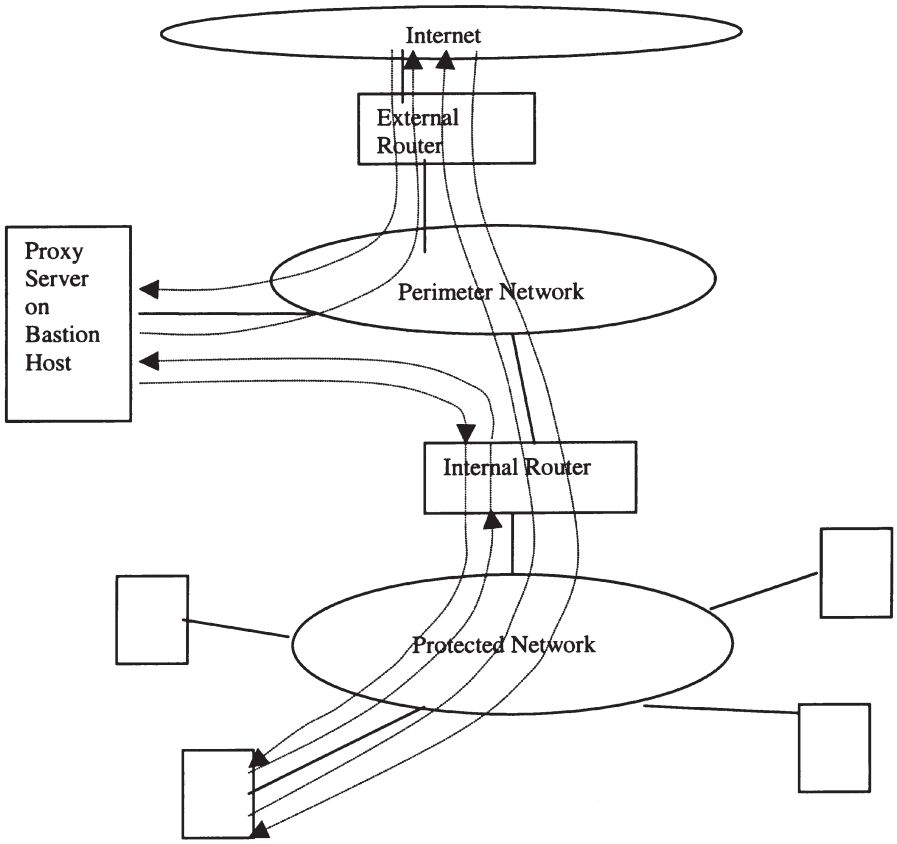


Figure 12 Screened subnet architecture.

network also known as DMZ (demilitarized zone) separates various bastion hosts running proxy servers from the external network and the protected network. Even if a bastion host is compromised, the internal router provides a layer of security to the protected internal hosts through network address translation that hides true addresses as discussed previously. Also, snooping activities are restricted further since the protected network is separate from the perimeter network. Therefore, it has minimum exposure to inbound external traffic. The bastion host can run various proxy gateway services such as SMTP, DNS, FTP, and HTTP. To minimize internal traffic from being exposed on the perimeter network, proxy gateway servers are also installed internally. For example, there could be an internal DNS server on the protected network and an external DNS server on a bastion host on the perimeter network. DNS information, such as names and addresses of various gateways required by external users, can be put on the external DNS server. An internal DNS server placed within the protected network can handle DNS queries for internal clients.

To reduce the load on bastion hosts, only limited services are put on various bastion hosts. Traffic that can be filtered using filtering routers is usually the traffic that is not directed to any bastion host. For example, certain HTTP connections initiated by an internal host to "safe locations" may be allowed between the internal host and the external server without the involvement of a proxy server on a bastion host. The interior router may allow legitimate traffic from a bastion host to the protected network and vice versa, as well as legitimate inbound and outbound traffic resulting from connections initiated by internal hosts to external hosts. The exterior router has filtering rules similar to the interior router with some differences that include rules to protect the hosts on the perimeter network and rules to filter those inbound packets whose source addresses have been spoofed to internal host addresses.

There are many variations of the screened subnet architecture. There could be multiple bastion hosts on a single perimeter network supporting one or more proxy gateway services. This arrangement eases the load on a single bastion host. The perimeter network could be split into two with one dual-homed host con-

necting the two perimeter networks or a series of dual-homed hosts connecting the two perimeter networks in parallel, in which each dual-homed host supports one or more proxy gateways. There could also be multiple perimeter networks. For example, one network could serve to interface with the Internet and others to interface with various business partners over extranets.

VI. INTRUSION DETECTION SYSTEM

Firewall components such as packet filtering routers and proxy gateways have the ability to prevent many security attacks. However, most firewall components cannot prevent an authorized user from launching a security attack from within the protected network, nor can they prevent very sophisticated security attacks. For these reasons, a new class of security systems, commonly known as intrusion detection systems (IDSs), are being deployed. IDSs monitor events and then analyze these events for security breaches. Examples of events include tampering of a systems file by an authorized user, opening of an insecure connection to the external network by an authorized user, and port scanning by an attacker.

IDSs can be classified as host-based or network-based IDS. In the case of host-based IDSs, software for the IDS is installed on the systems being monitored. The host-based IDS software uses log files, audit trails, and systems performance data to detect intrusions. A network-based IDS monitors packets in an organization's network to determine if any intrusion attempts are being made. A network-based IDS could run on multiple target machines or on a dedicated machine that monitors all network traffic.

An IDS consists of a sensor, i.e., a gatherer of information, and an analyzer that analyzes information. Various types of IDS components are available. A system integrity verifier is used to monitor system files for any planted security holes by an attacker. These systems use various mechanisms including cryptographic checksum to determine if a file is tampered. An example of a system integrity verifier is the Tripwire software. A log file monitor is used to monitor log files. These monitors look for certain patterns in the log file that are consistent with an intruder attack. An example of a log file monitor is the Swatch system. Sometimes IDSs use honey pots based on decoys to trap intruders. An example of a honey pot-based product is Mantrap.

Intrusion detection typically involves a high degree of processing to analyze events based on the information collected from a variety of sources. A common approach is to conduct pattern analyses whereby certain strings are searched in the data packets or in log files. For example, if the string "/etc/passwd" is transmitted over the network, then there is a possibility of an attack on a UNIX password file. Anomaly analysis is another approach for detecting security breaches that is based on statistical analysis of data. Anomaly analysis looks for anomalies or unusual behaviors. For example, a Telnet session that is extremely short but uses a lot of CPU time is an anomaly that is a target for deeper analysis.

A network IDS can be placed within as well as outside the protected network. When it is placed outside the protected network, it is capable of gathering various intrusion attack patterns. These patterns can then be incorporated in an IDS within the protected network. There are various types of IDSs based on different architectures. To aid in standardizing the development of IDS as well as in promoting a common terminology, DARPA has supported the development of the Common Intrusion Detection Framework (CIDF). This framework defines various component types used in an IDS.

Many vendors have rolled out IDS products. NetProwler from Symantec (formerly Axent) is a network-based IDS. It allows network administrators to build attack patterns (also known as signatures) without stopping the system. Intruder Alert is an IDS that runs on a host. Other examples of IDS include RealSecure, CyberCop, NFR Network Intrusion Detection, and Secure IDS.

VII. FIREWALL PRODUCTS, EMERGING TECHNOLOGIES, AND THE FUTURE

Many vendors are offering firewall products that range from firewall software to dedicated firewall appliances. The most common types of devices that facilitate security are the IP routers that support packet filtering rules. The firewall software products available typically support one or more features that include stateful inspection packet screening, IPsec support for VPN, network address translation, proxy services, etc. Firewall products include Firewall-1, PIX firewall, firewalls from CyberGuard, and Gauntlet. Both Firewall-1 and PIX support stateful inspection. Firewall products such as Firewall-1, CyberGuard, and Gauntlet, for example, support a content vectoring protocol (CVP). CVP facilitates integration of firewall software with many third-party products to screen viruses, Active-X controls, and Java applets. Many firewall products support the use of a public key infrastructure (PKI) for authentication and LDAP for directory access. SOCKS and FWTK from Trusted Information Systems (now

acquired by Network Associates) are popular proxy packages that support proxy aware clients and applications. FWTK includes proxies for various protocols including Telnet, FTP, and HTTP, whereas SOCKS is a generic proxy software that does not support application-specific logging. Examples of tools for determining security holes include Retina, CyberCop Scanner, Internet Security Scanner, Security Administrator Tool for Analyzing Networks (SATAN), System Analyst Integrated Network Tool (SAINT), Security Auditing Research Assistant (SARA), NESSUS, and NMAP.

A notable trend in firewall technologies is the convergence of packet filtering and proxy services. In the future, many packet filtering routers will go beyond IP/TCP layers to peak into application data. Another notable trend is that more and more packet filtering routers will also be able to support authentication, encryption using PKI, and the logging of high-level information. Similarly, more and more proxy services software will use IP header information to make filtering decisions. Hopefully, faster computers will make application-level filtering go faster.

For authentication, more and more firewalls will use technologies such as biometrics and cryptographic tokens. Biometrics compares an individual's unique physical characteristics such as fingerprints, voice characteristics, or color of eyes, etc., to provide a comprehensive authentication scheme. A biometric-based remote authentication scheme is not totally failure-proof for remote authentications. Cryptographic tokens such as smart cards are emerging as the preferred method for remote authentication.

Firewalls have been widely adopted by many organizations for providing network security. Firewalls have been able to block many common types of attacks. However, new types of attacks have always emerged to beat the security provided by notable firewall systems of the day. Thus firewall technologies have to evolve constantly to meet new security challenges. Firewalls offer only minimal protection against attacks launched by insiders. Thus firewalls would have to be augmented with other systems such as IDSs to reduce security risks. The International Computer Security Association (http://www.icsa.com) has developed criteria for evaluating firewall products. Organizations can benefit from this while developing their security solutions.

ACKNOWLEDGMENTS

The author wishes to thank the two anonymous referees and Jeremy Rasmussen for their valued comments.

SEE ALSO THE FOLLOWING ARTICLES

Computer Viruses • Crime, Use of Computers in • Electronic Data Interchange • Electronic Mail • Encryption • End-User Computing, Managing • Ethical Issues • Internet, Overview • Network Environments, Managing • Privacy • Security Issues and Measures • Software Piracy • Wide Area Networks

BIBLIOGRAPHY

Advanced Research Corporation (2000). Security Auditor's Research Assistant (SARA). Vienna, VA: Advanced Research Corporation. Available at http://www.www-arc.com/sara/index.shtml.

Bakos, Y. (1998). Towards friction-free markets: The emerging role of electronic marketplaces on the internet. *Communications of the ACM,* Vol. 41, No. 8, 35–42.

CERT Coordination Center (1997). CERT® Advisory CA-1993-14 Internet Security Scanner (ISS). Pittsburgh: Software Engineering Institute. Available at http://www.cert.org/advisories/CA-1993-14.html.

Cisco Systems (2001). Cisco secure IDS. San Jose, CA: Cisco. Available at http://www.cisco.com/univercd/cc/td/doc/pcat/nerg.pdf.

Cisco Systems (2001). Cisco secure PIX firewall series. San Jose, CA: Cisco. Available at http://www.cisco.com/univercd/cc/td/doc/pcat/fw.pdf.

Check Point Software Technologies (2000). Check Point FireWall-1 technical overview. Redwood City, CA: Check Point. http://www.checkpoint.com/products/downloads/fw1-4_1tech.pdf.

CyberGuard Corporation (2001). Products & services. Fort Lauderdale, FL: CyberGuard. http://www.cyberguard.com/products/index_ns.asp.

EEye Digital Security. Retina: The network security scanner, white paper. Aliso Viejo, CA: EEye Digital Security, April, 2001. Available at http://www.eeye.com/html/assets/pdf/retina_whitepaper.pdf.

Engler, P. (2000). A survey of the basic functionality of SAINT. Bethesda, MD: The SANS Institute. Available at http://www.sans.org/infosecFAQ/audit/SAINT.htm.

Foley, M. J. (October 27, 2000). Microsoft hack: Was source code altered? *ZDNet News.* Available at http://www.zdnet.com/eweek/stories/general/0,11011,2645871,00.html.

Fyodor (2001). NMAP—The network mapper. Available at http://www.insecure.org/nmap/index.html.

Hansen, S. E., and Atkins, E. T. (1993). Centralized system monitoring with Swatch, technical report. Palo Alto, CA: Distributed Computing Group, Stanford University.

Hurwitz Group (2000). Axent Technologies' NetProwler and Intruder Alert. Framingham, MA: Hurwitz Group. Available at http://www.hurwitz.com and http://www.symantec.com.

Internet Security Systems (2000). A vision for complete protection, white paper. Atlanta, GA: Internet Security Systems. Available at http://www.iss.net.

NESSUS (2000). NESSUS: Introduction. Available at http://www.nessus.org/intro.html.

Network Associates (1999). Next generation intrusion detection in high-speed networks, white paper. Santa Clara, CA:

Network Associates. Available at http://www.pgp.com/products/whitepapers.asp.

Network Associates (2001). Gauntlet Version 6.0. Santa Clara, CA: Network Associates. Available at http://www.pgp.com/products/gauntlet/default.asp.

NFR Security (2001). Overview of NFR network intrusion detection system. Rockville, MD: NFR Security. Available at http://www.nfr.com/products/NID/NID_Technical_Overview.pdf.

Recourse Technologies (2001). Mantrap: A secure deception system, white paper. Redwood City, CA: Recourse Technologies. Available at http://www.recourse.com.

SATAN. Available at http://www.porcupine.org/satan/.

SOCKS (2000). NEC commercial SOCKS products. Available at http://www.socks.nec.com.

Stallings, W. (2000). *Data & computer communications,* 6th ed. Upper Saddle River, NJ: Prentice Hall.

Staniford-Chen, S., Tung, B., and Schnackenberg, D. (1998). The common intrusion detection framework (CIDF). Presented at Information Survivability Workshop, Orlando, FL. Available at http://www.gidos.org/.

Tripwire. Data and network integrity assessment tools: Fundamental protection for business-critical systems, data and applications, white paper. Portland, June, 2001. OR: Tripwire. Available at http://www.tripwire.com.

Trusted Information Systems. TIS Internet firewall toolkit. Available at http://www.tis.com/research/software/.

Zwicky, E. D.,Cooper, S., and Chapman, D. B. (2000). *Building Internet firewalls.* 2nd ed. San Francisco: O'Reilly Publishers.

Flowcharting Techniques

K. B. Lloyd and J. Solak
Indiana University of Pennsylvania

GLOSSARY

CASE tools Computer-aided software engineering refers to a large set of software designed to help analysts, users, and programmers develop and document information systems more efficiently and effectively.

data flowchart One of the earliest flowcharting techniques, predating computers, that depicted the flow and control of data through one or more operations or departments in a company.

data flow diagram One of the most popular structured information system modeling techniques; it uses only four symbols and supports building a process-oriented view of an information system from a general overview through detailed process logic; *use cases* are a similar adaptation in UML.

flowcharting The use of symbols to model the physical and/or logical components of an information system.

object oriented A term that is used to refer to system development planning and/or programming in a context such that related data and code are encapsulated and conceptualized together as a single entity—an object.

program flowchart A flowchart that depicts detailed coding logic that will be implemented via a programming language; a flowchart that can be created via one of several tools, for example, traditional program flowchart, Nassi-Schneidermann chart.

structured techniques A set of information system modeling tools that were first developed primarily in the 1960s and 1970s and were not grouped and labeled as structured techniques until several years after their introduction, for example, data flow diagrams, structure charts.

systems analysis and design The study of the initiation and development of information systems for business.

systems flowchart An information system modeling method used to depict the control and flow of information at a high-level view with an emphasis on physical elements in a system as well as primary processes.

unified Modeling Language (UML) An information system modeling and development tool used in the analysis and design of information systems; a tool that attempts to combine the best aspects of several object-oriented system development methods and other system development tools.

DATA FLOWCHARTING, systems flowcharting, and program flowcharting refer to three categories of modeling techniques used to depict and document one or more parts of an information system. Although there are other related techniques for modeling information systems (e.g., dataflow diagrams, object-oriented diagrams, deployment flowcharts) this discourse focuses on data flowcharts, systems flowcharts, and program flowcharts. These flowcharting

techniques were the first formal system development techniques and began appearing in the late 1950s. The history, application, and relationships among these techniques are discussed. The origins of terminology used to describe these techniques (often a source of confusion in the literature) are also noted and clarified. Other diagramming techniques are discussed only in terms of their relationship to or outgrowth from data flowcharting, systems flowcharting, or program flowcharting.

I. INTRODUCTION

Flowcharting is a term used to describe a family of diagramming techniques used in one or more of the phases of information systems development (systems analysis and design). At the most general level, a flowchart is defined by the ANSI/ISO standards group as "a graphical representation of the definition, analysis, or method of solution of a problem in which symbols are used to represent operations, data, flow, equipment, etc." In chronological order, the three main categories of flowchart methods are data flowcharting, program flowcharting, and systems flowcharting. Current versions of some flowchart tools are placed in a category known as "structured techniques." When not placed in this category they are alternatively viewed as support tools for structured techniques or as business process analysis tools.

Structured techniques refer to graphic modeling tools used for analyzing, designing, or testing an information system. When discussing flowcharting techniques included in or related to structured techniques, the inference is that one is referring to tools developed for computer systems analysts and programmers versus variations adapted for managers to analyze business processes. Debate exists as to which tools qualify as being *structured*. However, there is consensus that, at the minimum, a structured technique must be graphically based and it must employ standards that permit a consistent, traceable, and repeatable application of symbols. In reality, whether or not a structured tool yields a structured result is primarily a function of the individual using the tool. For example, a program flowchart can yield a structured or unstructured result depending on the quality of the design by the human designer. To this end, computer-aided software engineering (CASE) tools, which began appearing in the 1980s, have been responsible for enforcing structure in analysis and design software by preventing human designers from making some choices that would cause a flowchart or other diagram to be inconsistent, incorrect, or difficult to follow.

None of the flowcharting tools were considered to be structured techniques when they appeared in the 1950s and early 1960s. Flowcharting and subsequent techniques were developed to support more expedient and accurate systems development and documentation. One of the earliest classifications of diagramming techniques as being structured was offered by DeMarco in a classic 1978 structured analysis text. At that juncture, no flowcharting method was included in the set. Over time, especially after the advent of CASE tools and the programmed enforcement of construction rules, some flowcharting techniques and later tools have been placed in the "structured" category and others have been labeled as support methods for structured tools. Prevalent structured techniques include dataflow diagrams (DFDs), transform analysis, hierarchical input process output charts, structured Nassi-Schneidermann charts (a type of structured flowchart), program structure charts (a type of structured flowchart), structured English (pseudo-code), decision tables, and decision trees. Numerous other systems development tools are available, many of which are now classified as being structured. The aforementioned represent a small sample of the more popular methods.

As ubiquitous as flowcharting became in information systems development, standards have been incomplete, particularly in the area of implementation and best practice. Formal standards that do exist were developed in collaboration by the International Organization for Standardization (ISO), the American National Standards Institute (ANSI), and the Information Technology Industry Council (ITI). The most recent standard specification, ANSI/ISO 5807-1985, was initially set forth in 1985 and final approval by ANSI was granted in February 1991. For the most part, this standard applies to flowchart symbol names and meanings. Some sections of the standard address symbol positioning and the hierarchical use of certain symbols. However, sparse attention is placed on guidelines for usage and diagram clarity. Common texts and industry records indicate that the ANSI/ISO standard has not been widely followed. Brand naming of flowchart techniques and other extensions of flowcharting have outpaced the ability of standards organizations to review and modify specifications.

To understand the forms and uses of the many types of flowcharting, it is important to consider the context in which flowcharting and related tools became prevalent. Therefore, the following section presents a brief background concerning flowchart devel-

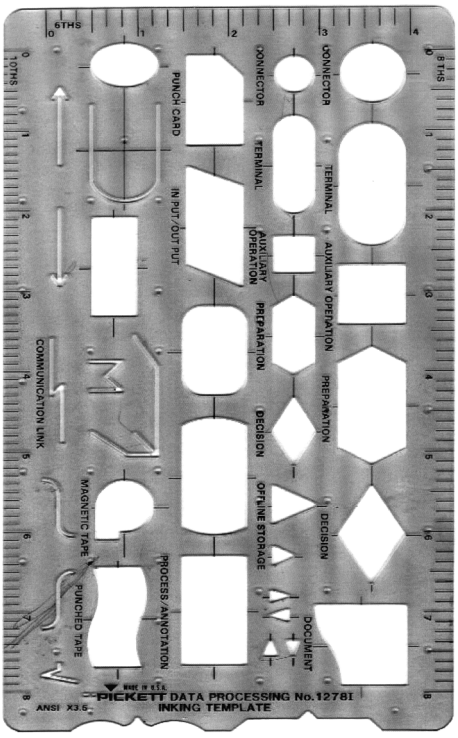


Figure 1 Early template for general flowcharting.

opment. Next, a discussion of the three initial types of flowcharts is presented along with examples. Recent adaptations are presented for each flowcharting method. In the final section, conclusions concerning the future of flowcharting are discussed. Figure 1 depicts an image of an early ANSI-compliant flowcharting template that could be used for program, systems, and data flowcharting. Explanations for the symbols are given in Fig. 2.

II. BACKGROUND OF THE FIRST FLOWCHARTS

Data flowcharts were the first of the three main flowcharting techniques. They existed prior to commercial computers. They were used in noncomputerized environments to record, analyze, and modify the processing and control of data. Data flowcharts were the forerunner of systems flowcharts. Systems flowcharting was a direct extension of data flowcharting with the initial primary difference being the addition of symbols to depict computer equipment, telecommunications, and computer storage devices.

As various versions of systems flowcharts evolved, symbols were added to allow the depiction of decision making and the repetition of activities—something that was not typically shown in data flowcharts. Finally, over a number of years, data flowcharts came to include most of the same symbols that systems flowcharting used. Program flowcharts arrived between data and systems flowcharting. They emerged with the first commercial computers in the early 1950s. Indeed, program flowcharting was the only technique included with the first computer manuals such as the "Programming Univac Systems Instruction Manual 1" for the Univac 1 in 1953.

When commercial information systems development began in the 1950s, systems analysis and design was not a formal discipline and programmers used assembly language (a coding language that used cryptic commands). The primary tool for analysis was a detailed *functional specification*—a large text-based narrative describing the purpose and function of the system. In these pioneering times, the system flowchart and other tools were slow to develop although they would later gain prominence and replace the functional specification. In large development efforts, functional specifications reached thousands of pages in length. It was an unwieldy tool for the analyst/programmer and even more so for the user. Difficulties associated with the functional specification are listed below:

1. The size of the specification document made it difficult to identify sections of the project.
2. Redundant information was widespread in the specifications.
3. The specification document was difficult to change and maintain—it was not flexible for the many updates that occurred during system development.
4. The specification document typically was confusing for users in that they stressed the hardware and software design characteristics of the user's requirements.
5. The specification document did not provide a useful basis for transition from systems analysis to the design and development of the system.

From the functional specification document, analysts and programmers moved more into the design side of systems development. In the beginning, this was done without the support of a systems flowchart. Program flowcharting was introduced with the first machine and assembly programming languages as the primary tool for systems design—a tool for the design of specific program code and storage requirements. The program flowchart was probably the most widely used design technique in the beginning years of programming. The use of program flowcharts, like all

Flowchart symbol	Flowchart symbol name and purpose *These symbols are ANSI compliant, however many symbols in use today are not ANSI compliant	Data flowchart document flowchart	Systems flowchart	Program flowchart
Physical Data Storage and I/O Symbols				
	This symbol depicts a data storage device used for sequential access to data files by a computer—usually magnetic tape.	+	+	
	This symbol depicts a data storage device used for direct or random access to data files (e.g., magnetic disk or optical disk).	+	+	
	This symbol depicts data being stored on a punched card (e.g., magnetic card, paper card, optical card).	+	+	
	This symbol depicts paper tape—usually a punch tape.	+	+	
	This symbol depicts data stored in a printed, human readable form (e.g., printed output, an OCR or MICR document, microfilm).	+	+	
	This symbol depicts data stored in internal memory in a computer.	+	+	
	This symbol depicts the manual entry of data (e.g., keyboard, switches, buttons, light pen).	+	+	
	This symbol depicts any type of video screen being used to display information in human readable form.	+	+	
	This symbol represents data when the medium is unknown.	+	+	+
	This symbol represents data stored in a form suitable for processing when the medium is unknown.	+	+	
Process Symbols				
	This symbol depicts any kind of processing function, for example, executing a defined operation or group of operations resulting in a change in value, form or location of data, or in the determination of which one of several flow directions is to be followed.	+	+	+
	This symbol depicts a named (predefined) process that already exists — a subroutine, utility, or library that consists of program steps that are specified in detail elsewhere.		+	+
	This symbol depicts any process performed by a human.	+	+	
Logic, Flow, and Initialization Symbols				
	This symbol depicts decision logic having a single entry and one or more possible exit paths, but only one of the paths can be selected; logic conditions inside the symbol are evaluated and one path leading out of the decision symbol is selected.		+	+

Figure 2 Symbol comparison chart for flowcharting techniques.

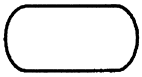
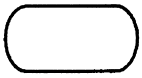
Flowchart symbol	Flowchart symbol name and purpose *These symbols are ANSI compliant, however many symbols in use today are not ANSI compliant	Data flowchart document flowchart	Systems flowchart	Program flowchart
	This symbol, in two parts, depicts the beginning and end of a loop. The top symbol denotes the beginning of a loop structure (repetition logic) and the bottom symbols denotes the end of a looping structure. The conditions for initializing or terminating a value or a process appear in the symbol.		+	+
	This symbol depicts the flow of data and connects symbols in a manner that show control.	+	+	+
	This symbol depicts the flow of data and connects symbols in a manner that shows control and is used instead of the undirected line when a diagram is large and the flow is hence unclear.	+	+	+
	This symbol depicts data transfer via a telecommunication link.	+	+	
	This symbol represents an alternative relationship between two or more symbols. The symbols is also used to surround an annotated area.	+	+	+
	This symbol depicts an exit to, or an entry from another part of the same flowchart, and is used to break a line, and to continue it elsewhere. This symbol is used in pairs—one to indicating a branch to another location and one to identify that location elsewhere—often the branch will have a slightly varied designation such as a circle inside a circle.	+	+	+
	This symbol represents an exit to, or an entry from, the outside environment, for example, start or end of a program flow, external use and origin or destination of data.	+	+	+
	This symbol depicts the initialization of a routine (a named block of program code) or the setting of any value such that subsequent code or activity is changed.	+	+	+

Figure 2 *(continued)*

development tools, was and is company dependent. As with any tool, a program flowchart was only as valuable as the programmer wielding the tool. Inappropriate use and nonuse of program flowcharts resulted in the well-known and much derided practice of unstructured GOTO-oriented programming.

GOTO statements are a method of instructing a program to branch to another section of a program to execute a subprogram. A subprogram is a small block of program instructions designed to perform one specific task, e.g., read in data from a file, search for a record in a file; synonyms for *subprogram* include *module* and *routine*. From a subprogram, control is directed to a lower level subprogram or back to the module that called the subprogram. The leap into programming regularly involved the risky practice of omitting the program flowcharting step due to time constraints or the lack of company standards for program development. It was not so much that the GOTO statement itself was a problem as it was the manner in which programmers applied the GOTO statement.

Specifically, programming was often conducted without a detailed plan, particularly when a program flowchart was not prepared. Even when a program flowchart was prepared, constantly changing user requirements could lead to the burgeoning use of unstructured GOTOs as a means to provide branches to newly added pieces of code. Frequently, when the need for a particular module was realized, the code was written, placed in the program where it would fit, and GOTO statements were inserted to redirect control to and from that code. The resulting code

branched up and down the length of a program creating a disorganized effect commonly known as *spaghetti code*. Moreover, constant time pressures dictated that programmers use few comments in their code. Such code was hard to understand, debug, and maintain.

The ramifications of unstructured GOTO code can be perceived more fully in terms of a book analogy. Imagine a book without an outline or headings. Further, consider a book that begins with an introduction and then directs the reader to a page deep in the discourse to locate the first chapter (and the chapter is not labeled). Next, the reader is directed back to a beginning page located at the end of first chapter, and then forward to the middle of the book for the second chapter, etc. This is quite similar to the unstructured GOTO effect. Clearly, more sophisticated development tools and guidelines were needed.

Early programming was further complicated due to the non-English-like coding language (machine code and assembler) that was used and because operating systems were rudimentary. That is, application programmers had to code instructions at the machine level and they had to issue complex commands for handling memory locations and input/output to storage—all operations that modern application programmers are no longer required to specify because today's operating system handles them. Consider the following line of code written in assembly language for the Univac 1 in 1953:

```
20      B 050.
```

The above line of code erases the contents of two special-purpose memory locations called registers (in this case register A and register X). It also transfers the contents of memory location 50 to both register A and register X. Clearly this type of code is terse and laborious.

The lack of diagramming/modeling techniques contributed to many system development failures. Systems were often late, overbudget, and poorly matched user requirements. This situation was a catalyst for increased development of tools and techniques that could improve system quality by adding clarity and structure to the systems development process. It was in this context that flowcharting techniques and other tools were enhanced or invented to increase the efficiency and effectiveness of the analysis, design, and maintenance of information systems. Directly after program flowcharting was introduced, the data flowchart was modified for computer systems development and referred to as a systems flowchart. Many other diagramming tools followed.

Most of these techniques and tools have become known in retrospect as either *structured techniques* or as tools supporting the structured techniques. That is, there was no specific, controlled effort under which a set of analysis and designed tools was developed and introduced. Instead, many techniques evolved simultaneously in response to systems development problems. It was only after these techniques proliferated and system professionals found them valuable that they became known as structured techniques. DeMarco, Constantine, and Yourdon were leaders in terms of seminal works that explicate these techniques as the basis for structured analysis and design.

Before turning to the details of the specific categories of flowcharting, it is beneficial to view the similarities and differences of the flowcharting symbols for data, systems, and program flowcharts. Figure 2 provides a summary and explanation of ANSI and non-ANSI flowcharting symbols and their overlapping use in data flowcharts, system flowcharts, and program flowcharts.

III. DATA (DOCUMENT) FLOWCHARTING AND SYSTEMS FLOWCHARTING

A. Data Flowcharting—Initial Uses

One of the first tools adopted in computerized business information systems analysis was the data flowchart, also known as a document flowchart. A data flowchart emphasizes tasks, workflows, and changes in data content or structure. Processes are defined, but they are not detailed. The ANSI/ISO 5807-1985 standard clearly delineates that a data flowchart is most similar to a system flowchart. The ANSI specification highlights that the symbols used for these two techniques are now identical except for decision symbols, looping symbols, and a predefined process symbol that are included for system flowcharting. However, in the earliest days of data flowcharting, the computer-related symbols were *not* part of the data flowchart specification. *Note also that a data flowchart is not another name for a data flow diagram (DFD).* DFDs refer to a diagramming method that, while strongly based on data flowcharts and system flowcharts, did not appear until later. Data flowcharts were first employed to model and control the flow of information (typically documents) and to better understand the business processes in the offices of an organization. When data flowcharts were adapted to computerized information systems, the symbols for computer-related input and output devices were added (see Fig. 2).

Traditionally, data flowcharts have not been labeled as a structured technique. More often they have been viewed as a support tool for other structured methods such as DFDs. However, as stated previously, *structured* is a loosely defined term. It has been primarily the lack of supported rules for chart construction, clarity, and usage that has led to a technique being seen as nonstructured. Until recently, another factor responsible for casting a negative light on many development methods was the absence of powerful graphical modeling software such as CASE tools.

Data flowcharts identify data inputs and they trace the flow of inputs through transformations that change them into final outputs or inputs to other processes. Like all flowcharting techniques, data flowcharts have tended to use varying symbols depending on the brand name of the flowcharting template or software being used. Brand name variations have always been a source of confusion—a point frequently noted by leading authors (e.g., Yourdon, DeMarco, Constantine) who cover the structured techniques.

Figure 3 is an example of a data flowchart. This type of flowchart emphasizes not only the physical data and workflows, but also the control of information through a system—in this case a system that handles customer orders. Note the physical emphasis of the flowchart, i.e., documents and people. Following a section of the diagram is illustrative. From the top, we see that a customer credit listing report (a human readable report) and a handwritten order form flow to a human customer credit verification process (note the human process symbol). Next, a paper-based sales order is prepared by a human and one copy is placed in an offline storage file. The triangle symbol with a double line at the top is the symbol used here for offline storage (in this case document storage). This is not an ANSI-compliant symbol but few techniques use only ANSI-compliant symbols. Four additional copies of the sales order are created during the sales order process and they are sent onward for further processing and documentation.

B. Systems Flowcharting—Initial Uses

System, flowcharting is a graphic technique used for representing the functions and physical components involved in one or more programs in an information system. It provides a physical (hardware) oriented overview of input, processing, and storage. It is related to a DFD in functionality. It can be used to understand and discuss a system or as the basis for developing specific program code. Therefore, a system flowchart can be useful in the analysis phase as a communication tool among analysts, programmers, and users. In the design phase, it is useful as a guide to programmers. The system flowchart depicts processing and accentuates control. It uses a subset of symbols from data and program flowcharting (see Fig. 2).

A systems flowchart was most often used to portray an overview of an entire information system. Therefore, a systems flowchart used a separate symbol to represent each type of hardware device. Comparatively, a program flowchart was intended to depict more of the internal coding logic for specific programs and therefore it did not use device symbols.

A small systems flowchart is shown in Fig. 4. It depicts an overview of a section of a simplified library information system. To facilitate comparisons of charting tools, the simplified library information system is used in later sections as the basis for discussing program flowcharts, a structure chart, and a Nassi-Schneidermann diagram. At the beginning of the systems flowchart in Fig. 4, we see that a human processes and enters information regarding a patron request to borrow a book. Information is entered via a keyboard or OCR device. A program determines the validity of the patron's library ID number and then checks to determine if there are any restrictions on the patron's borrowing privileges. Data from the Patron and Book database is used during this process. Information is displayed at a computer screen indicating the status of the patron's ID and borrowing privileges. For valid requests, processing continues and a book is lent to the patron. If the request is invalid, processing stops.

C. Adaptations of Data and Systems Flowcharting

Data flowcharts and systems flowcharts are now nearly identical in terms of symbols employed and their purpose. Differences are minor, e.g., systems flowcharts sometimes tend to depict more information concerning processes. Current adaptations of these flowcharting methods (e.g., quality control process flowcharts, deployment flowcharts) are used by managers. Adaptations of data and systems flowcharts have brought flowcharting full circle. That is, data flowcharting started as a technique for documenting, analyzing, and controlling data flows in a noncomputerized business. The data flowchart was created and used primarily by managers. Systems flowcharting developed as a popular variation of data flowcharting that incorporated computer-related symbols and slightly more emphasis on processes. Systems flowcharting then

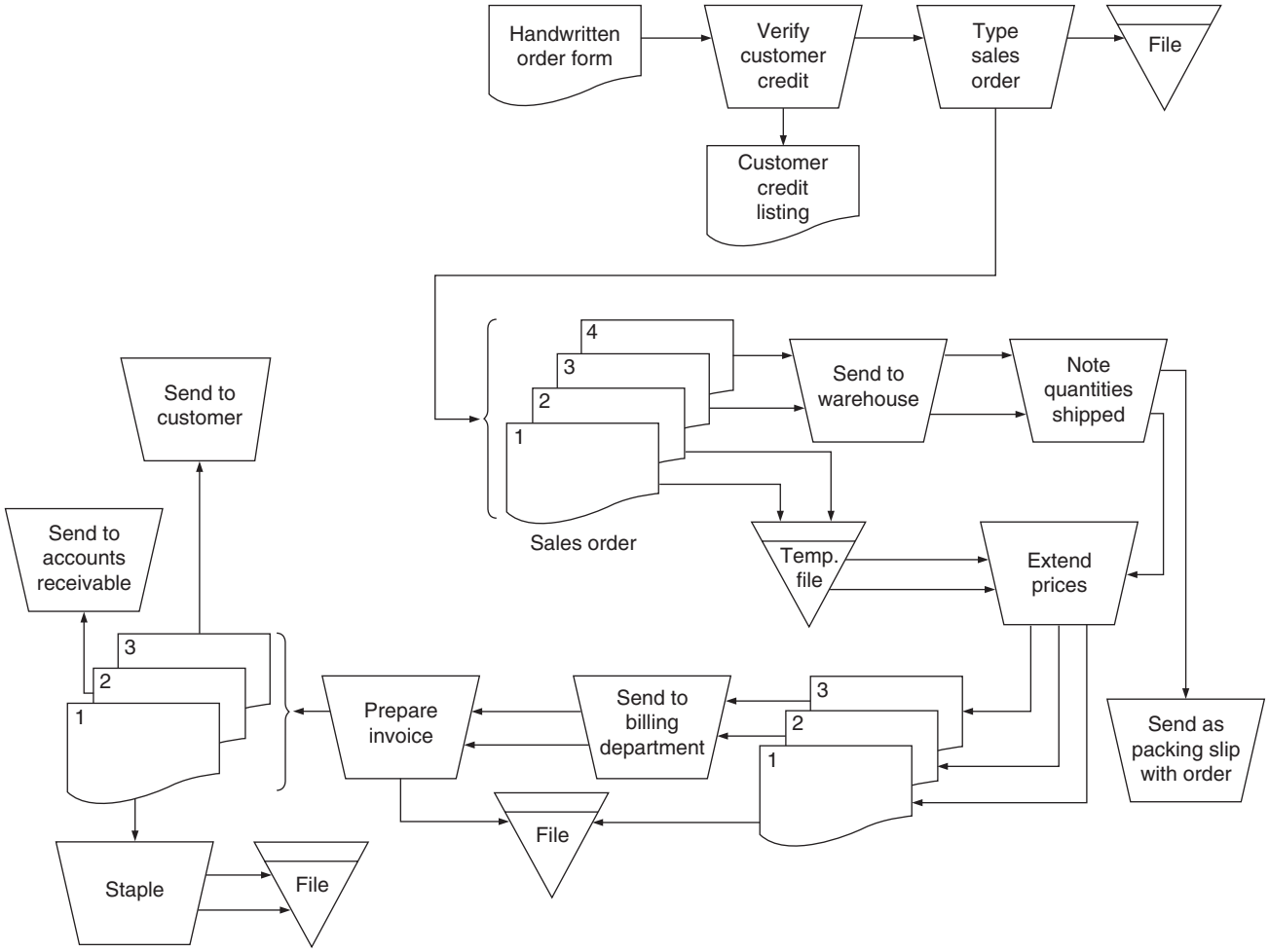


Figure 3 Data flowchart [reprinted with permission from Powers, Adams, and Mills. (1984). *Computer Information Systems Development: Analysis and Design.* New York: Southwestern].

grew in popularity as data flowcharting waned. Systems flowcharting from inception was much more a programmer/analyst tool—a tool, created by the programmer/analyst to document or design information systems and to communicate with managers and other users about an information system. The latest flowcharting techniques in this area have again become tools for managers—tools to assist managers in tasks such as quality analysis and control, process reengineering, and project management.

Modern CASE tools with graphic interfaces are the enablers that have permitted flowcharting to regain relevance as a tool that managers use to model and analyze data flows and business processes. The ability of CASE tools to support the dynamic interactive linking of symbols to pictures, documents, spreadsheets, databases, and other diagrams is a key factor in the reemergence of flowcharting. Such interfaces allow managers and consultants to more quickly and easily create and change process-oriented flowcharts. Changes in one part of a flowchart can be set to automatically update other affected processes. Information contained in a different format (e.g., spreadsheet, database) can be included without having to duplicate and reformat the data. Research is progressing toward the identification and understanding of clarity rules for all types of diagramming tools and other interfaces.

One of the business adaptations of flowcharting is in the area of quality management. A quote from W. Edward Deming (founder of the quality movement) accentuates the importance of the modern flowchart in this area: "Draw a flowchart for whatever you do. Until you do, you do not know what you are doing, you just have a job." The well-known ISO 9000 specification for attaining total quality in the operation of a business has been adapted to flowcharting in both stand-alone software and integrated CASE tools.

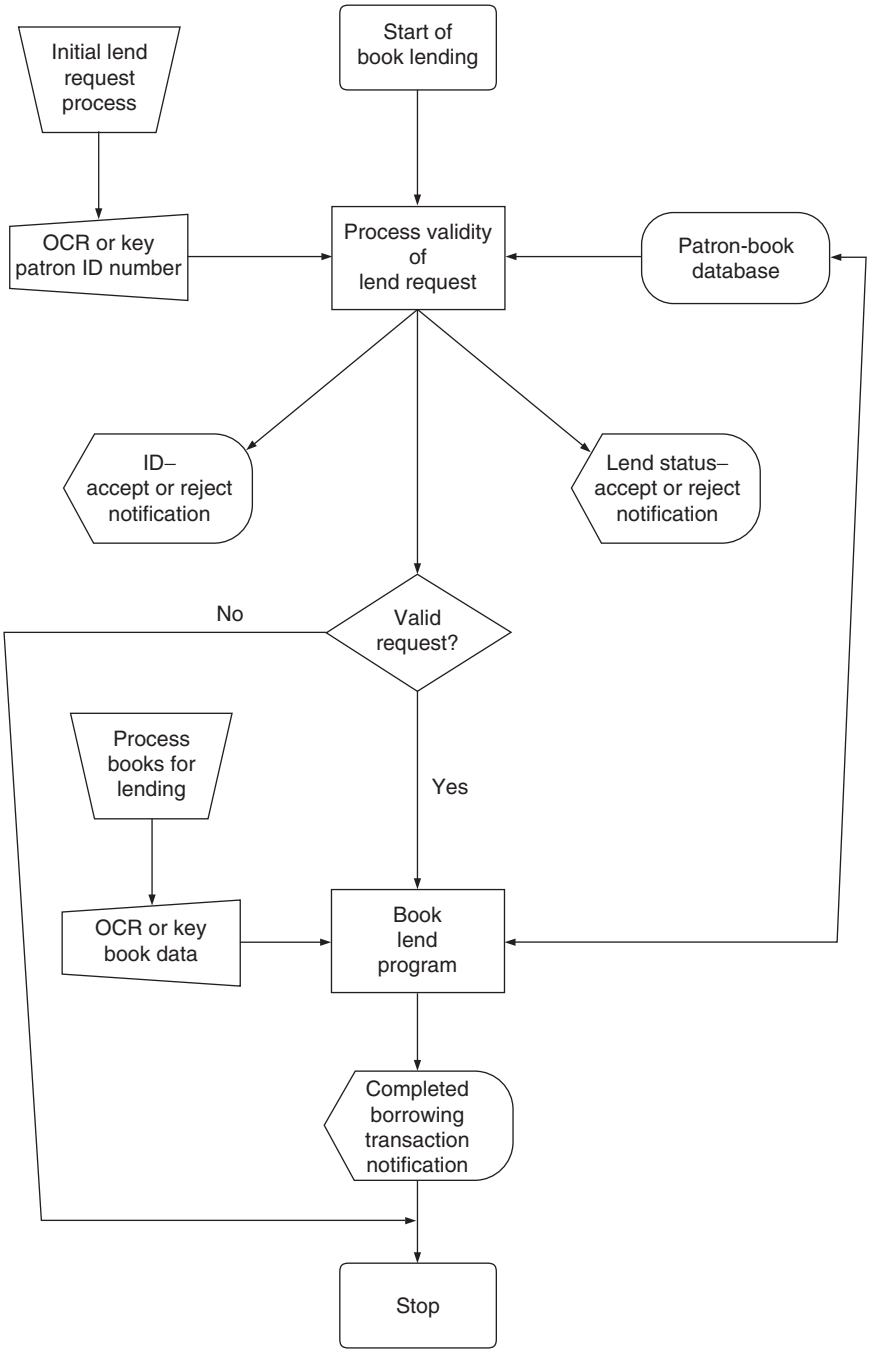


Figure 4 Systems flowchart.

Key to total quality management (TQM) and the ISO 9000 specification is the detailed understanding of business processes. In fact, understanding and detailing business processes is critical to any organizational improvement effort. A quality initiative or any improvement strategy is concerned foremost with processes, e.g., sales processes, purchasing processes, operational processes, and invoicing processes. The goal is to maintain or increase quality and to reduce costs via eliminating redundant or unneeded processes and by reducing cycle times. As discussed below, flowcharting methods designed to assist managers with analyzing business processes became known as process flowcharting tools.

In addition to dynamic interactive linking capabilities, contemporary high-end flowchart-capable software supports the creation of procedures and documentation related to the registration and certification for quality under the ISO standards. Layering of charts, data entry, training, surveys, and process simulation are other common software features for supporting managers and consultants in attaining quality standards. At the center of all such efforts is the process flowchart. The term *process flowchart* has become widely used to refer to any flowchart used to model business processes. Latest applications include the ability to model real-time or time-critical processes in an integrated fashion—linking process modeling and analysis to project management, statistical control, simulation, and cost analysis. Such applications permit a manager to create a process flowchart and then add resources, calendars, and scheduling.

Some applications also support running a simulation on one or more processes. The steps in a process can be run forward and backward in time to show how information enters, changes, and affects each part of a business process. Results from a simulation show where costs and blockages are incurred and hence how resources may be better deployed, or how a process may be carried out more efficiently or cost effectively. Deployment flowcharting is another adaptation of flowcharting to modern business analysis. The 1980s quality movement brought an emphasis on teams and a need to model how a team fits into business processes. Deployment flowcharting (Fig. 5) shows team members across the top of the flowchart. Processes flow in an implied grid-like structure down the chart. Symbols in the flowchart are aligned, layered, and sized in a manner that clearly indicates what teams are involved with each process. Deming created this flowchart variation and Myron Tribus further popularized and named it. Adding the team to the process flowchart increases the amount of information conveyed and helps provide structure.

Figure 5 shows an overview of a product development process. The deployment flowchart shows which team members are involved in each step. Rows of symbols (layers) depict sequence—the "Charge to New Product Development Team" occurs before "Develop Process Template and Write Mission Statement." The width of a symbol is expanded or contracted to indicate which team members (above the symbol) are involved in a process. A dotted line in a symbol indicates that any team member above the dotted line portion of the symbol is not included. Therefore, only the V.P. of Marketing and the Sales department are involved in "Develop Process Template and Write Mission Statement." Colors are used to designate different types of processes, e.g., data collection or development processes are blue, and decision processes are red. The deployment flowchart also shows how information flows between team members.

IV. PROGRAM FLOWCHARTING

A. Initial Uses

Program flowcharting became accepted practice before systems flowcharting. This is an anomaly given that data flowcharts are most similar to systems flowcharts in terms of both the symbols used and their purpose. In today's systems environment, program flowcharting, *in its original form,* is not often used.

Program flowcharts were originally created for the design side of systems development—for designing the logic of a computer program. The ANSI definition of a program flowchart reads: "Program flowcharts represent the sequence of operations in a program." When a program flowchart adhered to the ANSI standard, it borrowed nearly all of its symbols from data flowcharting. The notable symbol differences for program flowcharting include the diamond-shaped decision symbol, the predefined process symbol, and symbols used to denote looping/repetition (see Fig. 2). Absent from the program flowchart are the symbols used to document physical objects (computer-related equipments and human-related actions). Because program flowcharts were designed to model the internal logic of a computer program, the data flowchart symbols for physical objects were not needed. Program flowcharts, like any flowchart, can be created to show an overview or details. Figure 6 shows a simplified overview of the program logic for the lending function in a typical library program. Figure 7 depicts a more detailed program flowchart outlining the steps needed to carry out the Readin Patron ID# Routine shown in Fig. 6.

B. Adaptation of Program Flowcharts

Applied to program flowcharting and systems design, the term *structured* denotes, at the minimum, the use of a structure chart (a structured module-based, program flowchart). Although the term *structure chart* is most commonly used to refer to a module-based program flowchart, it is also used to refer to a grouping of design tools. For example, some discussions categorize the Warnier-Orr diagram, hierarchical input processing output (HIPO), and Nassi-Schneidermann

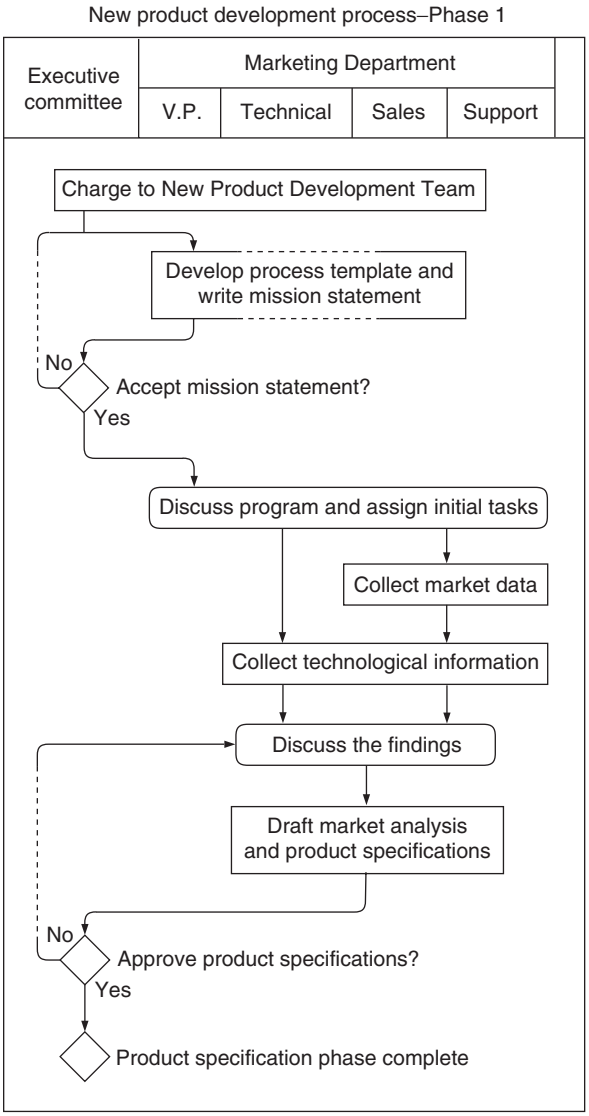


Figure 5 Deployment flowchart. (Reproduced with permission from The Teamflow Company, Bedford, MA, 1999.)

chart as a set of structure charts. It is desirable to refer to such tools as "structured techniques" to avoid confusing the single technique (structure chart) with an entire group of structured tools.

The hierarchy part of the HIPO chart (created by IBM corporation) was the earliest version of a structure chart. IBM referred to the hierarchy part of a HIPO chart as a vertical table of contents (VTOC). A simplified overview level structure chart is shown in Fig. 8. This and other structured adaptations remain in use today.

In a structure chart, emphasis is placed on program diagramming at the module (subroutine) level. Emphasis is also placed on designing a program with modules that exhibit qualities such as "loosely coupled" and "highly cohesive." Loosely coupled modules constitute modules that are less likely to depend on other modules; communication between modules is minimized and restricted to passing data elements and control information. Highly cohesive modules are ones that contain instructions to accomplish *one* specific task. Structure charts often include at least two additional symbols for showing the passing of data between modules—data couples and control flags. Structure charts are developed by breaking down DFDs (remember that *data flow diagrams are not data flowcharts*) typically via one of two processes: transform analysis or transaction analysis. Both are structured processes first

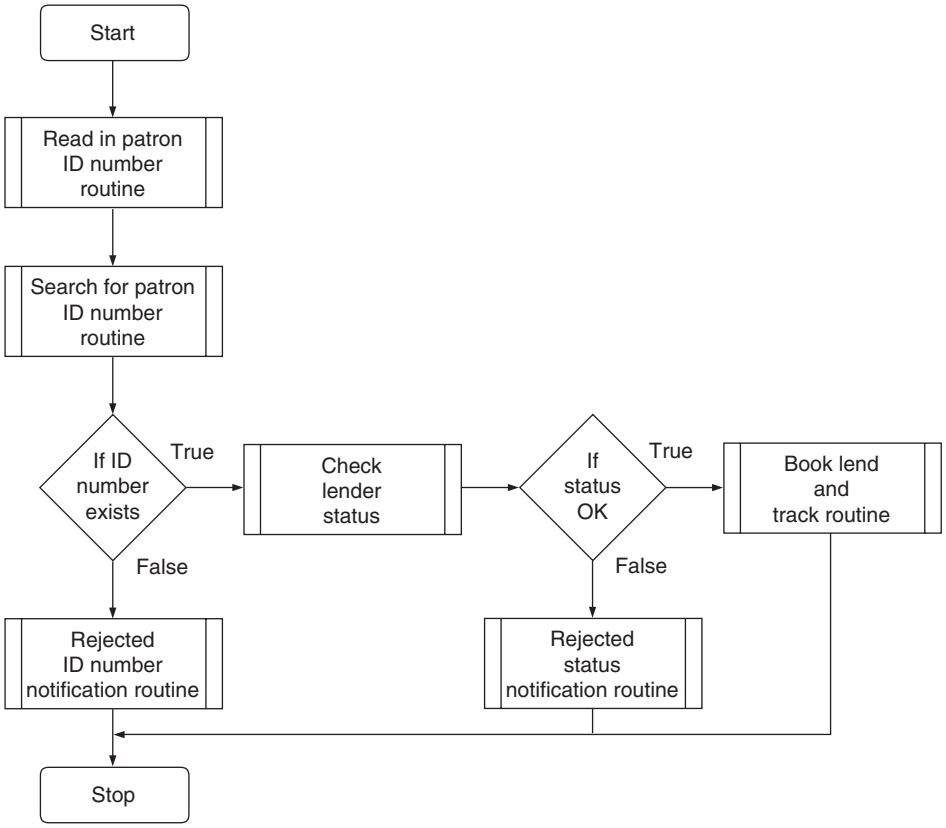


Figure 6 Program flowchart—overview level.

formalized by Yourdon and Constantine. Space limitations preclude a comprehensive discussion of transform analysis, transaction analysis, and the detailed rules for structure chart creation. The *Modern Systems Analysis and Design* text by Hoffer, George, and Valacich provides a thorough and recent discussion of transform analysis, transaction analysis, and structure charts.

A basic structure chart (Fig. 8) shows the names of modules and the order in which modules are called.

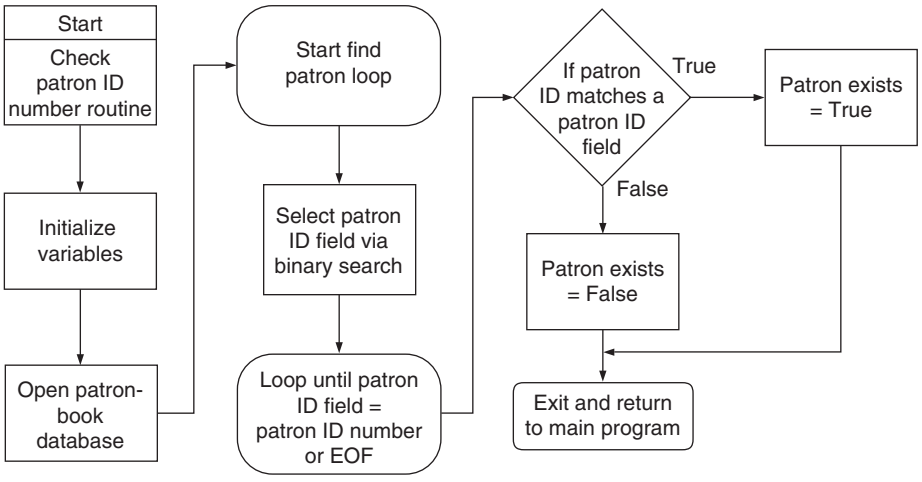


Figure 7 Program flowchart—detail level.

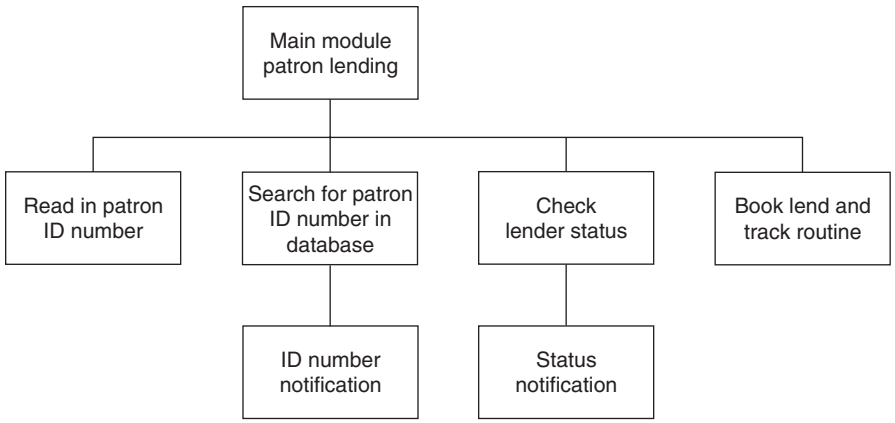


Figure 8 Structure chart.

Modules on the same horizontal level are usually related in terms of function and detail (much like an organization chart for a company). A comparison of this structure chart to the program flowchart in Fig. 6 shows that the structure chart appears more organized, but has fewer details about processes and decision making.

The Nassi-Schneidermann diagram is another tool considered to be part of the flowchart family. It is a structured method for depicting the logic shown in the program flowchart. The basic symbols are as follows. A process or assignment is shown in rectangles. Conditional statements are shown using one or more triangles. Loops are shown using an L-shaped symbol. Figure 9 shows a Nassi-Schneidermann diagram that depicts the same logic as the program flowchart shown in Fig. 7. The Nassi-Schneidermann diagram is visually more organized, compact, self-contained and hence more structured. This structured version of a program flowchart remains in use today.

CASE tools and stand-alone applications have resurrected the usefulness of structured forms of program flowcharting such as the Nassi-Schneidermann diagram. Applications that transform flowcharts to program code, and vice versa, have been developed and remain in use today. Particularly useful are programs that translate cryptic hard-to-follow code such as assembler into well-organized, easy-to-read structure charts and program flowcharts. CASE tools also exist that assist a developer in transforming a DFD into a set of structure charts.

CASE tool adaptations of program flowcharting also remain useful for algorithm design and for visualizing and determining program complexity. Different measures of logical, structural, and psychological complexity can be calculated and understood with the use of program flowcharts. In terms of logical and structural complexity, researchers and analysts can study the number and types of paths, branches, and modules in a program. In terms of psychological complexity, the logic in psychologically difficult programming concepts such as recursion can be visualized, measured, and understood with the aid of a program flowchart.

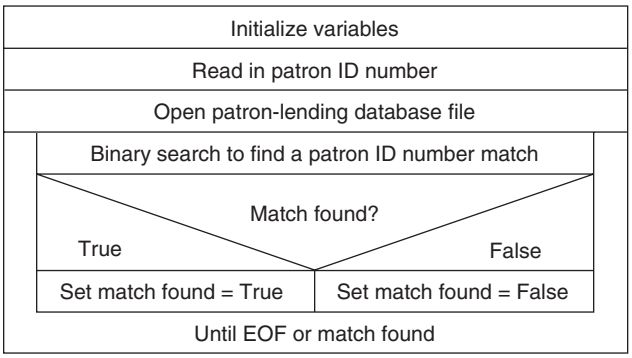


Figure 9 Nassi-Schneidermann diagram.

V. THE FUTURE OF FLOWCHARTING

It seems that flowcharting, in some forms, is here to stay. Adaptations of the three original types of flowcharting (data, program, and systems) are being used today. Variations of data and systems flowcharting have reverted to their origin and have again become process flowcharting tools used by business managers. Program flowcharts have evolved into more structured tools such as the structure chart and the Nassi-Schneidermann diagram. They have also become useful for the analysis and visualization of the complexity of programming code. All such adaptations of flowcharting became

useful due to the development of CASE tools—interactive, integrated, graphical tools for developing, changing, structuring, linking, and visualizing information.

An important related development has been object-oriented (OO) approaches to systems analysis and design. The growing OO paradigm has been suggested by proponents as a movement that will replace all types of structured methods and flowcharting. Yet, the future of flowcharting, especially any type of process flowcharting, has remained strong. Managers and business analysts think in terms of processes and therefore need a modeling method that fits the way they think. Indeed this reality spurred the resurgence of process flowcharting.

The Unified Modeling Language (UML) is a recent OO specification that strives to be as useful to IS professionals in analysis and design as it is for business professionals involved in strategy and process analysis. In short, UML is an integration of entity–relationship (ER) diagramming with prior OO diagramming methods and conventions. It appears unlikely that UML, or any current approach, will succeed with an all-in-one diagramming standard for both business managers and computer system professionals. The key difficulty is that any such tool must be able to fit the process-oriented thinking of managers while at the same time meet the needs of systems designers and programmers for capturing data structure and logic requirements related to computerized implementation. Object-oriented specifications have been found useful for the latter while process-oriented modeling continues to be useful for the former. That process-oriented modeling represents a desired manner of conceptualizing, especially at the overview level, is seen in the persistence of process-oriented methods such as process flowcharting and in the invention of "new" techniques such as the "use case" and state diagrams that are frequently offered as support tools for OO-centered approaches to systems development. The "use case" and the state diagram depict events (processes) and present information in a manner similar to the DFD. Hence, the near future will continue to find process flowcharts and event diagrams along side OO diagrams and a select group of structured tools.

SEE ALSO THE FOLLOWING ARTICLES

Computer Assisted Systems Engineering (CASE) • Database Systems • Data Flow Diagrams • Object-Oriented Programming • Structured Design Methodologies • Systems Analysis • Systems Design

BIBLIOGRAPHY

DeMarco, T. (1978). *Structured analysis and system specification: Tools and techniques.* Upper Saddle River, NJ: Prentice Hall.

Hoffer, J. A., George, J. F., and Valacich, J. S. (1998). *Modern systems analysis and design.* Upper Saddle River, NJ: Prentice Hall.

Kruchten, P. (2000). *The rational unified process, an introduction.* The Addison-Wesley Object Technology Series. Reading, MA: Addison-Wesley.

Lipnack, J., and Stamps, J. (1997). *Virtual teams.* New York: John Wiley & Sons.

Lloyd, K. B., and Jankowski D. J. (1999). A cognitive information processing and information theory approach to diagram clarity: A synthesis and experimental investigation. *Journal of Systems and Software,* Vol. 45, No. 3, 203–214.

McConnell, S. C. (1993). *Code complete: A practical handbook of software construction.* Redmond, WA: Microsoft Press.

Szyperski, C. (1998). *Component software: Beyond object-oriented programming.* Reading, MA: Addison-Wesley.

Whitten, J. L., Bentley, L. D., and Dittman, K. C. (July 1997). McGraw-Hill Higher Education.

Yourdon, E. (1991). *Modern structured analysis.* Norwood, MA: Yourdon Press Computing Series.

Yourdon E., and Constantine, L. (1979). *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design.* Prentice Hall.

Forensics

Donna Wielbo and Kevin L. Lothridge
National Forensic Science Technology Center,
University of Florida

GLOSSARY

class characteristics The properties of an item of evidence that place it in a set with multiple members.
forensic science The application of science as it pertains to the law.
individual characteristics The properties of an item of evidence that can be attributed to a common source with a high degree of certainty.
latent print A fingerprint made when oils or perspiration is deposited onto a surface. Latent prints can't be seen without the application of specialized techniques.
physical evidence Any object that can establish that a crime has been committed, or can provide a link between a crime and its victim, or a crime and the perpetrator.
polymerase chain reaction (PCR) A technique used for replicating or copying a portion of a strand of DNA outside the living cell, which has the capability of producing millions of copies of that DNA strand.
recombinant DNA A DNA molecule formed by joining DNA segments from different sources.
restriction enzyme Any of a large number of nucleases that can cleave a DNA molecule at any site where short sequences of specific nucleotides occur. These enzymes are used extensively in recombinant DNA technology.
restriction fragment length polymorphisms (RFLP) Different lengths of DNA base pairs fragments that are a result of cutting a DNA molecule with restriction enzymes (endonucleases).
ridge characteristics Ridge endings, bifurcations, enclosures, and other ridge details that must match in two fingerprints in order to establish their common origin.
short tandem repeat analysis (STR) Analysis for regions of a DNA molecule that contain short segments that consist of three to seven repeating base pairs.

I. INTRODUCTION

Information systems are generally implemented as a means of facilitating the operation of a business or organization. The information systems currently available to law enforcement agencies and forensic science laboratories play an implicit role in the day-to-day functioning of their operations and the efficient sharing of information between agencies and investigational laboratories to help in the fight against crime. In a forensic capacity, such systems can encompass, and provide, strategic and intelligence information, decision-support information, management information, and data handling. However, the successful design, implementation, and exploitation of such systems is imperative in ensuring the efficiency and the effectiveness of applied forensic science. Information systems supporting forensic investigations need to have considerable human input, and combined with up-to-date technology function to increase the overall effectiveness of forensic investigations.

Forensic science is the application of science as it pertains to the law. The very nature of crime laboratory

analyses generates vast quantities of potentially important and useful data derived from the examination of physical evidence. Physical evidence may include blood and other body fluids such as semen and saliva, which may be used in the partial identification of a suspect or victim. Hair, fibers, soil, paint chips, bullet fragments, and glass particles are types of physical evidence that may link a perpetrator to a crime or crime scene, or be used to help establish the course of events that occurred at a crime scene. Fingerprints or DNA evidence can be used to unequivocally identify a perpetrator, and paint flakes can be used to identify a car involved in a hit-and-run accident. The field of forensic science continues to expand with the incorporation and application of less traditional scientific disciplines. Originating from the practices of applied basic sciences such as chemistry, biology, and physics, the field now includes such disciplines as toxicology, serology, pathology, psychiatry, psychology, linguistics, and odontology, all of which are capable of producing vast amounts of useful data worthy of their own forensic databases. The rapid development of scientific technology now enables the analyses of more complicated materials, with greater sensitivity and selectivity, producing more detailed and complex data than ever before.

With recent developments in information technology, several United States federal, state, and local government law enforcement agencies have dedicated resources to expand and refine information handling systems dedicated to crime analysis and intelligence. Through the development of today's Internet system, inter- and intra-laboratory and agency information networks, and collaborative crime fighting and data sharing initiatives, nationally and internationally, it is now possible to access a wealth of crime-related information. This information is available to all realms of law enforcement and crime laboratories, augmenting the accurate and efficient identification of serial crimes, recidivists, or possible projected crime scene scenarios in the fight against crime.

To achieve this, system developers have been faced with issues of cost-effective, suitable, and secure data storage for enormous amounts of data and case information, easy data retrieval, delivery, and handling for authorized users, shared distributor user interfaces, and collaborative work patterns.

This article introduces many of the information systems supported by national and international law enforcement organizations and agencies that are made available to crime laboratories as tools and resources for the support of investigations, as well as the abundance of databases and libraries available to the practicing forensic scientist or law enforcement agent via the Internet.

II. FEDERALLY MAINTAINED FORENSIC DATABASES

A. Integrated Automated Fingerprint Identification System (IAFIS)

Personal identification is a crucial component of most forensic investigations, from placing a person at a scene, to identifying corpses. The earliest and most significant personal identification system was developed at the end of the 19th century by Alphonse Bertillon. Bertillon's system was anthropometric in nature and involved a series of body and facial measurements designed to establish a process of identification for habitual criminals. This system became widely known as Bertillionage. Nowadays, the Bertillionage system is considered inadequate to identify one person as being distinct from all others. Although the concepts are theoretically possible, the Bertillionage system is extremely susceptible to inconsistent body measurements, reducing the discriminating power of the system. Despite the limitations of Bertillionage, Bertillon is still heralded for his original work that represents one of the first examples of individualization in criminal investigation, and his significant contributions to the development of forensic science.

It was the realization of the value of fingerprinting in personal identification and individualization that led to the inception and growth of the science of fingerprints that was to eventually replace anthropometric systems. William Herschel initiated fingerprinting for personal identification purposes in 1877, when he proposed that fingerprints could be useful in the identification of criminals. It was Henry Faulds who recognized the value of latent prints from a crime scene and proceeded to use fingerprints to identify a criminal in 1880. Through the work of Edward Henry, a fingerprint classification scheme was devised a few years later. Sets of fingerprints were cataloged for identification purposes, resulting in the development of the fingerprint classification system that replaced anthropometry in Europe and North America.

Fingerprints are an individual characteristic; they remain unchanged throughout a person's lifetime, and they have general ridge patterns, or friction ridges, that allow them to be classified and used in personal identification. Basic ridge patterns can be described as loops, whorls, and arches. While 60% of

the population has the loop characteristic in their prints, and 35% have whorls, only 5% have the arch characteristic. A three way process is used to classify fingerprints: the shapes and contours of individual patterns, the finger positions of the pattern types, and relative size, determined by counting the ridges in loops and by tracing the ridges in whorls. The original Henry system, adopted by Scotland Yard in 1901, converted ridge patterns into letters and numbers in the form of a fraction to describe a persons fingerprint profile, however, that system could not accommodate the large numbers of patterns that needed classification. The Federal Bureau of Investigation (FBI) system, used by most agencies in the United States today, is a system based on an applied classification formula. The principles of this system are based on a series of modifications and extensions of the Henry system, recognizing eight different patterns in total: radial loop, ulnar loop, double loop, central pocket loop, plain arch, tented arch, plain whorl, and accidental.

Using the classical methods of examination, the first step in the FBI system, primary classification, provided an insight into full fingerprint classification. This classification system alone can divide all the fingerprints in the world into 1024 groups. A primary fingerprint classification system provides the investigator with a list of candidates, but it cannot identify an individual. Final classification was originally made by detailed visual comparison that was a very time-consuming and laborious task. The Henry and FBI systems are ten finger classifications and are only useful when the investigator has the names of the known suspects, as crimes scenes rarely yield more than partial prints. The FBI revolutionized fingerprint examination through the development and application of the IAFIS. The IAFIS system provides law enforcement agencies the best methods ever to identify suspected offenders and to obtain all possible information about them. The inception of this system ended the labor-intensive, manual system of checking fingerprint records and evidence by reviewing individual cards and latents that had been archived in databases for decades. Traditional identification methods took as long as 10 days to completely check records holding data from 34 million criminal fingerprint files, in an attempt to determine whether a person had an existing criminal record. During the first phase of development the IAFIS system was used to convert all the original fingerprint cards into electronic data, allowing identification procedures to be conducted rapidly and accurately. The IAFIS system is constructed in a way that now enables the FBI to electronically accept fingerprints, from any state, county, or city law enforcement agency, and allows the rapid check of fingerprint files for the identification of suspects and immediate access to their fingerprint records.

B. The Combined DNA Index System (CODIS)

Through the original DNA work of Alex Jeffries and colleagues, and advances in analytical and molecular biology technology, the last 20 years have supported the provision and acceptance of forensic DNA evidence. For the forensic scientist this revelation meant they were now able to link the origin of biological evidence, such as blood, semen, saliva, hair, or tissue to a single individual. The advent of recombinant DNA technology, the discovery of restriction fragment length polymorphisms separable by gel electrophoresis, and the subsequent development of the polymerase chain reaction (PCR), and short tandem repeat analysis for mitochondrial DNA brings us to the current methods applied in DNA profiling today. Current methods require a fraction of the original sample, are faster, and more discriminating. The first applied forensic DNA profile evidence was used to indict Colin Pitchfork in the mid to late 1980s. He was found guilty of murdering two young girls in rural England. While identifying Pitchfork as the perpetrator in the case, the evidence led to the exoneration and release of an earlier suspect. In the United States, the FBI laboratory was the first to use DNA profiling to positively identify a specific individual as the source of a stain. In the past, DNA evidence was traditionally reported as a statistical probability, but now with the implementation of refined techniques and sufficient sample, the FBI can now testify that a questioned biological sample came from a specific person to the exclusion of all others.

Funding provided by the United States federal DNA Identification Act of 1994, supported a five-year program for state and local forensic laboratories, and provided for the creation of a National DNA Index System (NDIS) by the FBI. By gathering and harnessing DNA profile data, the FBI laboratory developed a significant investigative tool, called CODIS. This is a nationally available United States DNA database of DNA profiles from convicted offenders, unsolved crime scenes, and missing persons. The CODIS system enables American state and local law enforcement crime laboratories to electronically compare and exchange DNA profiles, thereby helping to link serial, violent crimes, and the subsequent identification of suspects

by matching DNA from crime scenes with the DNA profiles of convicted sex offenders. By 1998, CODIS had recorded over 400 matches linking serial rape cases, or identifying suspects by matching crime scene evidence to known convicted offenders, facilitating hundreds of violent crime investigations. In addition, the system has significant potential for aiding crime prevention, since the information it provides may be used to identify serial offenders for repeated crimes that might otherwise remain undetected or identified. Since all 50 states have now passed legislation that requires all convicted offenders to provide a biological sample for DNA databasing, the database continues to grow rapidly. By 1999, over 600,000 DNA samples had been collected and more than 250,000 samples had been archived in the database.

C. DRUGFIRE

Crimes that involve firearms make up a large portion of criminal investigations. Firearm evidence may be present in murder, attempted murder, suicide, assault, and rape cases, and is prevalent in drug-related crimes. When firearm evidence is used to its full potential, a number of significant questions can be answered. Such questions include: the nature of the weapon used; whether the weapon was working properly; how far from the target was the weapon fired; the direction of fire; whether a specific weapon fired a specific bullet; and who actually fired the weapon. Firearms identification refers to the study of firearms, and includes topics such as cartridge identification, and bullet and cartridge case identification. Firearms identifications is a separate specialty from forensic ballistics; the two terms are not interchangeable and should not be confused. Although there are literally thousands of different types of firearms, in general they can be broadly classified into two groups: shoulder firearms such as shotguns and rifles, and handguns such as revolvers, pistols, and automatic or semi-automatic weapons. Fundamental to forensic firearms examinations is the fact that no two firearms, even the same make or model, ever produce the same marks on fired bullets and cartridge cases. Each firearm has its own unique surface characteristics resulting from manufacturing processes, use, and wear and tear that can never be reproduced exactly on other firearms. When guns are fired, the firing leaves its own identifiable "fingerprint" in the form of unique microscopic grooves and striations on the bullet and its casings. The integrity of these characteristics change little over time, which allows the identification of firearms that may be recovered months or years after a shooting incident. Cases that involve firearms identification start with a preliminary examination to locate class characteristics. These are intentional or design characteristics common to a specific group or family of items. In the case of firearms, the characteristics that relate to bullets fired from them, includes the caliber of the firearm, and the rifling pattern of the firearm barrel. Cartridges and cartridge cases are examined for class similarities called breech marks, firing pin impressions, extractor marks, and ejector marks. If class characteristics are different the examination usually ends there. If characteristics are similar then the evidence is compared for individual, or unique, characteristics that comprise a match. Individual characteristics are transferred from the firearm to the ammunition it fires, making firearm identification possible. Comparison microscopy is the traditional method used to compare rifling impressions for firearms identification, bullet examination and identification, as well as for comparison of bullet striations in weapon association. If there is no firearm recovered from a shooting crime scene, microscopic examination of the bullets may provide information on general rifling characteristics (GRC).

Significant developments in firearms identification technology have occurred over recent years. Two currently available systems are DRUGFIRE sponsored by the FBI, and IBIS from Forensic Technology Inc. DRUGFIRE is an automated computer technology system developed to assist the FBI and American state and local law enforcement agencies make links between firearms-related evidence. The system was designed to digitize and compare evidence through image analysis. Images of the unique characteristics left on bullets and cartridge cases by the firearms can be compared to each other via specialized computer software programs. These systems allow comparisons to be made between images taken from cases within a single jurisdiction or through networking; the images can be compared to images anywhere in the world. By the end of 1999, DRUGFIRE was operational at 125 sites and the system database contained more than 92,000 evidentiary images from over 167,000 criminal investigations. The success of this information system has resulted in a nationwide, collaborative initiative between the FBI and the Bureau of Alcohol, Tobacco and Firearms (ATF) to make the DRUGFIRE system interoperable with the ATF's own ballistic imaging system CEASEFIRE under the National Integrated Ballistics Information Network (NIBIN) program.

D. CEASEFIRE

CEASEFIRE is a program that was begun by the ATF in 1996, in response to the increasing number of violent firearms incidents. The CEASEFIRE program was developed to provide more efficient investigative methods to enable investigators to trace the links between bullets, shell casings, firearms, and criminals. The program has a mission to reduce illegal gun possession, and subsequently gun violence, by pairing aggressive law enforcement and prosecution efforts to recover illegal handguns, and to target, apprehend, and prosecute offenders who carry firearms. The system combines up-to-date computer systems, and an ever-growing firearms database, while utilizing a national network of experts to help investigate firearms incidents from assault, armed robbery, gang conflicts, drive-by shootings, and murder cases. The CEASEFIRE support program is an invaluable resource to cities that have a history of, or has established, serious organized crime, gang activity, and drug-related incidents, and also provides support to law enforcement agencies that encounter high rates of violent crimes that involve firearms. This is a reliable ballistic identification system that obtains support data from the ATF's, investigative, technical, and forensic services, and can be accessed by authorized users in all law enforcement agencies. Forensic or investigative agencies can use this system to reliably bring cases to closure expeditiously, without having to rely purely on their own investigative resources or on the traditional, more time-consuming methods of firearm investigation. The CEASEFIRE system combines both the investigative skills of the ATF and the technology of a unique ballistic comparison system known as Integrated Ballistic Identification Systems (IBIS). The IBIS system combines the power of two established databases called BRASSCATCHER and BULLETPROOF, through links to the ATF National Tracing Center. This system is a networkable workstation which houses a microscope, illumination control devices, computer hardware, and software, which enables firearms examiners and technicians to make digital images of, and automatically sort and categorize, bullet and shell casing markings. The sorting process can result in fast, high probability ballistic matches, which in turn increases the frequency of investigative leads in shooting incidents. The CEASEFIRE program provides police with the ability to link shooting incidents that might not otherwise be connected using traditional investigative methods, and the ability to match ammunition and shell casings recovered from what may appear to be unrelated incidents. By using this system, firearms trace requests can now be processed within 24–36 hours. This system is also used to collect and evaluate data, which is used to monitor firearms trafficking.

E. National Tracing Center

The National Tracing Center (NTC) of the ATF assists law enforcement agencies by providing systems that trace firearms recovered from crimes worldwide. The system holds more than 100 million firearm records, and is the main repository of all the records of firearms dealers who are no longer in business. Firearms examiners can access the NTC's reference library to identify firearms, firearms manufacturers, and importers. The center also confirms the accuracy of firearms traces and identifies trends and trafficking patterns for investigators. The NTC traces the origin and ownership of recovered firearms used in crimes, and is electronically linked to local law enforcement agencies. The database is built from information obtained from recovered and traced firearms, and can be accessed by authorized users from federal, state, local, and foreign law enforcement agencies. The center is also responsible for collecting, collating, and generating firearms statistics for each state, and providing investigative leads for the law enforcement community by identifying persons involved in suspicious activity, such as unlicensed firearms dealers.

F. National Integrated Ballistic Information Network

The ever-increasing incidence of firearms-associated crimes has resulted in new measures to facilitate the rapid and accurate flow of data and important information between investigating agencies. The FBI and the ATF have recently taken steps to combine their individual databases (DRUGFIRE, which deals with cartridge cases and CEASEFIRE, which deals with bullets), into the National Integrated Ballistic Information Network (NIBIN). DRUGFIRE and a necessary part of the CEASEFIRE system, IBIS, were originally incompatible because the two systems used different lighting techniques in their analyses, resulting in different shadowing effects on the images making comparison difficult. Through assistance from the National Institute of Standards and Technology (NIST), and interagency cooperation, the FBI and

ATF are working to make the two systems compatible through NIBIN. The NIBIN system uses project staff to obtain digital images of bullets and cartridges which are then downloaded to a computer, where they are analyzed and stored in a database for matching against bullets or cartridges fired from the same gun at a later time. Combining both of these established systems, evidence analysis that used to take weeks or months can now take minutes. Bullets recovered from a crime scene in one state can now be compared with evidence from crime scenes across the country to identify suspects and leads by rapidly providing the most likely candidates for matching crime evidence. The database contains images of criminal evidence that is directly connected with gun crimes and images from test firings. The ATF has now implemented a cooperative initiative with gun manufacturers to make digital images of their inventory before it is distributed to retailers. Due to the nature of the data collected for the NIBIN program, investigative leads are provided after a crime has been committed, rather than before. However, through the voluntary efforts of gun manufacturers, matches and serial number traces will become easier. Most law enforcement jurisdictions have the appropriate software and hardware to run NIBIN. The NIBIN database contains information on more than 500,000 of the 220 million firearms in the United States, and it is expected to grow to 800,000 by the year 2002.

G. Arson and Explosives Incidents System

During arson and explosive investigations, the forensic scientist routinely encounters a number of problems not generally present at other crime scenes. The scene itself is often surrounded by devastation and destruction, and the investigator has to collect evidence after further damage has been caused by other visitors who were there first, such as firefighters, supervisory personnel, owners, and onlookers. There is potential loss of evidence from the foot traffic as well as from the processes of fire control and fire extinguishing. The role of an arson examiner is to determine the presence of accelerants introduced to a fire scene and to answer several questions: Where was the origin of the fire? How was the fire started? Was the fire intentional or accidental?

A fire that has been deliberately lit is considered arson. Examinations of debris recovered from scenes can identify gasoline, fuel oils, and specialty solvents. For a source of information relating to arson and explosives incidents the investigator can turn to the Arson and Explosives Incidents System (AEXIS), a flexible database that helps ATF investigations and prosecutions involving arson and explosives activities. The AEXIS database was developed by arson and explosives experts as a means to enable the collection, maintenance, evaluation, and dissemination of information for law enforcement agencies, forensic laboratories, and the community. The AEXIS database can provide agents and forensic investigators with information such as the total number of explosives incidents by target type; the total number of bombing incidents by target type; the total number of explosives incidents by state; the total bombing incident fatalities by target type, including deaths, injuries and property damage; the total number of bombing incidents by motive, as well as information on estimated damage; and the total number of pipe bomb incidents.

H. National Fire Incident Reporting System

Another information system heavily used by the ATF is the National Fire Incident Reporting System (NFIRS). The NFIRS has two objectives: to help state and local governments develop fire reporting and analysis capabilities for their own use, and to obtain data that can be used to more accurately assess and subsequently fight the fire problems at a national level. Within the NFIRS, local fire departments participate by filling out incident and casualty reports as fires occur. The completed incident reports are sent electronically to a state office where the data is validated and combined into a single computerized database. Feedback reports are then generated and routinely sent to participating fire departments. At regular intervals, collated statewide data is sent electronically to the National Fire Data Center at the United States Fire Administration (USFA) and included in the national database. This database is used to answer questions about the nature and causes of injuries, deaths, and property loss resulting from fires, and it represents the world's largest national annual collection of incident information.

I. National Drug Pointer Index

The National Drug Pointer Index (NDPIX), which is physically supported by the National Law Enforcement Telecommunications System (NLETS), first became operational in the United States in October 1997. This drug pointer system was set up to assist fed-

eral, state, and local law enforcement agencies in the investigation of drug trafficking organizations and to preserve resources by preventing duplicate investigations. The development, maintenance, and success of this fast and effective network, which extends to most law enforcement agencies in the United States, requires an ongoing cooperative effort between state, local, and federal law enforcement agencies. Agencies that participate submit active case-targeting information to NDPIX, which responds by sending pointer information from the database back to the agency. The more data factors submitted into the system, the better the chance of identifying possible matches. The NDPIX basically functions like a switchboard, providing a means to rapidly notify agencies of common investigative targets. The actual case information is only shared once the officers or agents make contact with each other via telephone, after being linked by their entries into the NDPIX. The DEA fully participates in the NDPIX and has already entered over 86,000 drug investigative targets into the system as of June 2000. As more law enforcement agencies participate in the NDPIX, the accumulating data should have serious implications in national efforts to bring down drug organizations and the ongoing battle against drug related violence.

J. National Crime Information Center

The National Crime Information Center (NCIC) also established by the FBI, furnishes a computerized database on documented criminal justice information. The database is easily accessed by any authorized criminal justice agency and once an inquiry is made, the system provides rapid disclosure of information in the system obtained from other criminal justice agencies about crimes and criminals. This information can be used to assist the appropriate agencies in catching fugitives, finding missing persons, locating and returning stolen property, and for the protection of the law enforcement agents who have to encounter the persons described in the system. The records in the NCIC are secured from unauthorized access through a number of suitable administrative, physical, and technical safeguards, which include restricting access to investigators on a need-to-know basis, and through the use of locks, alarms, passwords, and data encryption. The data contained in NCIC is provided by the FBI and federal, state, local law enforcement and criminal agencies, authorized foreign agencies and authorized courts.

K. World FACTS

An information database currently under development with forensic capability is the World Forensic Automated Counter Terrorism System (World FACTS). This system is designed to provide forensic laboratories, both in the United States and internationally, information on explosive, firearm, and latent print evidence in terrorist incidents. By utilizing reputable forensic database systems, the FBI laboratory allows foreign national police organizations to access and share information collected in international terrorist incidents.

III. COMMERCIALLY AVAILABLE DATABASES

A. Integrated Ballistics Imaging System

Part of the success of the ATF CEASEFIRE system is its integration with the Integrated Ballistic Identification System (IBIS), (Forensic Technology Inc., Montreal, Canada). IBIS is the first identification system that can correlate and match projectile and shell casing ballistic evidence on one platform. This system is commercially available to all forensic or investigative laboratories, and enables firearms technicians to digitize and store bullet and shell casing images from individual cases. The firearms examiner can use the system to rapidly and accurately enter information, review, and cross-reference fired bullets and expended cartridge case data. The IBIS program can provide numerical rankings or ratings for potential matches from the thousands of pieces of evidence in its database. In a laboratory setting, the firearms examiner evaluates the computer's data assessment, retrieves the potentially matching images for evaluation and comparison on the computer monitor, and then confirms matches using comparison microscopy techniques. Such image analysis can be conducted locally, nationally, or internationally, which opens up tremendous opportunities to link, what were once, unlinkable crimes. This allows firearms examiners to study and compare more items of evidence in less time, and to create leads for law enforcement agents by providing information on crimes committed with the same firearm. The database is dynamic, growing exponentially as more and more evidentiary images are entered. The system uses a bullet-analyzing program (BULLETPROOF), and a cartridge case-analyzing program (BRASSCATCHER) to provide an integrated, easy to use system. The systems can be used to quickly compile, cross-reference, and search databases

for possible evidentiary matches and also have the capabilities for the exchange of electronic evidence with other BULLETPROOF-BRASSCATCHER installations globally. This system works to complement the traditional comparison techniques used in firearms evidence examinations, and can be used to initiate electronic firearms traces through the ATF's NTC. IBISs can be linked via ISDN, Frame Relay, T1, and other commercially available networking systems. This means examiners can routinely check cases with any IBIS on their network, and enable other labs in the network to conduct case peer reviews and confirm matches. MatchPoint is a personal workstation accessory to the IBIS, also provided by Forensic Technology Inc. MatchPoint gives firearms examiners direct access to image match correlation scores and case images that have been entered into IBIS, and consists of a computer and the appropriate software connected over a local area network (LAN) to an IBIS server. The examiner can do the detailed comparison of high-confidence candidates while freeing up IBIS for other technicians or examiners. Rapid Brass Identification (RBI) is a mobile extension of IBIS, which consists of a portable station with a computer and easy-to-assemble microscope for obtaining images of cartridge cases. This system can be used to acquire crime scene breech face and firing pin images, which are then immediately transmitted to IBIS for correlation and analysis over a wide area network (WAN) to an IBIS Data Acquisition Station (DAS), or a DAS Remote (DAS/R). Data transfer can also be executed by connecting directly to a DAS or DAS/R on a LAN. The RBI system can also be used in smaller forensic laboratories and police stations, enabling access to IBIS technology. GunSights is computer reference software produced by Forensics Technology Inc. that provides rapid and accurate identification of firearms for law enforcement agents. The software provides detailed information and high-resolution photographs of thousands of current and historical firearm models. Once installed, basic search features are used to enter the manufacturer, model, or other features of the firearm. Advanced search paradigms are also in place for firearms that have been damaged or altered. By matching the search criteria, GunSights provides a list of matching firearms from its database.

B. Laboratory Information Management Systems

Most forensic laboratories throughout the United States are installing their own Laboratory Information Management Systems (LIMS) for data management, retrieval, and sample tracking. A number of commercially available LIMS have been developed and designed to serve the unique organizational and security needs of forensic laboratories and can be employed to monitor and record the analysis and processing of samples, from the introduction of the sample into the laboratory through to successful completion of the analyses and reporting of the results. The navigation systems are generally easy to use, utilizing point and click technology for navigation and standard interfaces. One of the key issues in forensic science is data security—LIMSs are generally multi-user systems, therefore mechanisms must be in place to ensure that data cannot be tampered with. Most systems have features that allow individual investigators to develop their own personalized folders for their own individual cases, and the capability of making their folders private or public when necessary. Samples can be inventoried, and workflow systems for sample handling and other laboratory processes can be constructed. Within the framework of a laboratory, the LIMS usually enables samples to be tracked from its storage, to the position of a given sample aliquot in an autosampler or microtiter tray. Information can be generated and stored for the creation of aliquots, relevant tests, test results, and associated reports. Most forensic LIMSs have the capability to use electronic bar coding facilitating sample log-in and sample incorporation into the workflow procedure. Data entry is often conducted using a spreadsheet format that allows efficient data mining that makes sample tracking and location of analyses results rapid and easy, as well as providing access to all the associated information for a particular sample or case. This type of information might include the analytical instrument used, its raw data, result specifications, and relevant or important notes. Two important features of most LIMSs include the ability to disseminate specific data to specified destinations within the laboratory and the ability to provide information on the QA/QC functionality of the laboratory.

IV. FIELD SPECIFIC NETWORKS

A. FBI Information Sharing Initiative

In 1999, funds were provided to the FBI to support the Information Sharing Initiative (ISI). These resources funded the upgrade of hardware, software, and networks used to pass information among FBI locations and to outside agencies, including criminal justice and

law enforcement agencies. Through the ISI, the FBI has the capability to electronically share information among field offices working jointly on important cases. Traditionally a large number of FBI technical personnel are deployed to the site of a large case because the computing technology was not available for sharing vast amounts of investigative information between agents in remote or multiple locations. The ISI is under phase-by-phase development over a four-year period with the primary goal of upgrading the infrastructure, and to put in place the applications necessary to manage and analyze the data collected by the FBI, in addition to providing the security required to share information electronically with other law enforcement agencies and government agencies, while maintaining the security of the information. In response to the FBI's need for rapid and easy access to case information, the electronic case file (ECF), has already been developed and implemented. The ECF provides a virtual case file to FBI personnel in all FBI locations worldwide, 24 hours a day. This system makes it possible to distribute information electronically among and across field office locations for the first time ever. Full-text searches can be conducted to find case information that was previously omitted due to the conventional time-consuming manual searches. To date, over 3.5 million documents have been loaded into ECF that are now available for full-text search.

B. Law Enforcement On-Line

Until the advent of the Internet, there was no mechanism in place that provided a means for law enforcement officers to easily communicate among themselves, or access a system designed purely to serve the needs of individual officers or law enforcement managers. In 1995 the FBI established Law Enforcement On-line (LEO). This provides a communications vehicle that links all levels of law enforcement throughout the United States, and supports the broad and rapid distribution of information related to the best technologies and practices in law enforcement. LEO was developed as a user-friendly system, and is supplied free to its users, the law enforcement community. The LEO system provides all law enforcement agents and officers the means to communicate securely with other law enforcement agents throughout the United States, and enables the sharing of information on mutually interesting topics. It is an intranet system exclusively for the law enforcement community, its unique value is its ability to deliver law enforcement sensitive, but unclassified communications services, and distance learning to local, state, and federal law enforcement.

C. Repository of Patterned Injury Data

On a much smaller scale, specialized groups within the field of forensic science are establishing their own databases for local or national use. The University of North Carolina, in collaboration with the Armed Forces Institute of Pathology, and the Office of the North Carolina Chief Medical Examiner, is developing and implementing a digital library to facilitate collaborative practices within forensic medicine. The system is called the Repository of Patterned Injury Data (RPID), and the database includes archived digital forms of important and routinely used collections of medical data, which are stored and presented in a way that allows joint review, discussion, and consultation by several pathologists at one time.

V. FORENSIC SCIENCE AND THE INTERNET

A. On-line Forensic Resources

The evolution of the Internet has provided significant information resources to the forensic scientist. A number of quality sites have been established that provide information on the basic principles of many forensic disciplines and provide everything from general information to published procedures for conducting crime scene investigation, dealing with biological hazards at crime scenes, and technical information and methodologies related to forensic analyses. Many professional organizations now use the Internet to disseminate information on forensic policies, provide electronic journal articles, and advertise professional symposia, meetings, and workshops. These sites also provide links to academic programs, educational seminars, and technology updates, all of which bear relevance to the field or practices of forensic science. Practicing forensic scientists can take on-line courses in subjects relevant to their field as part of their in-house training or to enhance or update their qualifications for courtroom testimony as an expert witness. One of the more practical tools to develop from the Internet are list-serve e-mail networks that enable forensic scientists from around the world to communicate with each other, ask questions regarding analyses, and to discuss topics of mutual interest.

The Internet is also a source of on-line databases available to the forensic scientist and includes the

fields of applied DNA and chemical analyses. For example, the Distribution of Human DNA-PCR polymorphisms database was established in 1996 by a research group in Dusseldorf, Germany. This database coordinates and holds a collection of combined population studies on DNA-PCR polymorphisms that provide an overview of data that has been published to date. When the amount of data collected became unmanageable in script form, the group established a means of providing the data via the Internet and performs database updates every 6 months on collected data and relevant published literature.

B. Database of Nuclear DNA

Another useful DNA database accessible via the Internet is the Database of Nuclear DNA. The database was first proposed in 1995 and was established and implemented by a working group for the Spanish-Portuguese Group of the International Society for Forensic Hemogenetics. The working group, composed of technicians from the Forensic Genetics Laboratory of the Ertzaintza, created a computerized database of allelic frequencies of nuclear DNA markers analyzed by PCR and provided a compilation of allelic frequencies of 52 loci from 30 different populations of Spain, Portugal, and Latin America. The data was then made available via the Internet to members of the GEP-ISFH, as well as the international scientific community.

C. Y-STR Haplotype Reference Database

A very dynamic database is the Y-STR Haplotype Reference Database, created by the continuous submission of Y-STR haplotypes from the International Forensic Y-User Group. Drawing on the utility of identifiable male-specific DNA, the analysis of Y-chromosomal STR systems is an invaluable addition to the standard autosomal loci used in forensic genetics. Markers of individuality of the male-specific part of the Y chromosome can be identified by Y-STR haplotype analysis using a set of short tandem repeat markers that are approved and accepted by the forensic and scientific community. This kind of data is extremely important, especially in cases of rape, sexual assault, and paternity testing. The inception of the database fulfills two objectives: the generation of reliable Y-STR haplotype frequency estimates for both minimal and extended haplotypes for the quantitative assessment of matches in forensic casework, and the assessment of male population stratification, as reflected by Y-STR haplotypes in European and worldwide populations. The International Forensic Y User Group and a number of forensic laboratories worldwide are working collaboratively to collect population data for the large, standardized haplotype database using quality-controlled techniques.

D. Scientific Working Groups

A useful source of information for the forensic scientist is the scientific working groups (SWG) web site. A number of SWGs have been established, one for each main discipline within forensic science. Each group, funded by the FBI, is made up of members from the forensic field of that specialty. The home page for each group provides information related to their forensic application. Information includes guidelines to quality assurance procedures in that particular field, news, public comments, upcoming professional events, symposia and workshops, and information relating to the group members. Scientific working groups are in place for DNA analysis methods (SWGDAM); fingerprinting or friction ridge analysis (SWGFAST); imaging technologies (SWGIT); documents (SWGDOC); firearms and toolmarks (SWGGUN); trace evidence (SWGMAT); digital evidence (SWGDE), and forensic archiving of X-ray spectra (SWGFAX). The European Network of Forensic Science Institutes (ENFSI) has a series of 15 working groups much like the SWG groups in the United States. Of these 15 groups, 5 have active web sites that contain current information. These are the Digital Imaging Working Group, European Document Experts Working Group (EDEWG), DNA Working Group (DNAWG), Fire and Explosion Investigation Working Group, and the Marks Working Group. The ENFSI marks working group has done much work in the area of shoeprints.

E. Chesapeake Area Shoeprint and Tire Track

Chesapeake Area Shoeprint and Tire Track (CAST) is a group of footwear and tire mark examiners who work together to promote education, quality assurance, and casework experiences. It has developed a web site that houses many important sources of information useful to shoe and tire mark examiners. The site includes a tutorial section for footwear.

F. Forensic Science Communications

Forensic Science Communications is an electronic journal provided on-line by the FBI for public use. Established in 1999, and published quarterly, this journal database provides a means of communication between forensic scientists. The e-journal provides current peer reviewed manuscripts that pertain to the applied science and technology of forensics, information on training opportunities, and upcoming meetings, conferences, and workshops related to the field.

SEE ALSO THE FOLLOWING ARTICLES

Crime, Use of Computers in • Electronic Data Interchange • Law Firms • Privacy • Research

BIBLIOGRAPHY

Adams, D. E., and Lothridge, K. L. (July 2000). Scientific working groups. *Forensic Science Communications,* 2:3.

ATF On-line, NIBIN/IBIS web-site, http://www.atf.treas.gov/firearms/nibin_ibis/index.htm.

ATF On-line, Programs, CEASEFIRE web-site, http://www.atf.treas.gov/about/programs/firearms/ceasefire.htm.

BATF, Arson and Explosives Repository web site, http://www.atf.treas.gov/aexis2/index.htm.

CAST—Footwear & Tire Track Impression Evidence Examination resource and contact web site, http://members.aol.com/varfee/mastssite/home.html.

Crime Fighter B.E.A.S.T. web site, Integrated Evidence Tracking Systems for Police Departments and Crime Laboratories. The Porterlee Corporation, http://www.porterlee.com/.

European Network of Forensic Science Institutes web site, http://www.enfsi.org/index.php3.

FBI publications: A report to the american people, chapter 3: Tools and resources to support investigations, http://www.fbi.gov/publications/5-year/1993-98/report6.htm.

Federal Bureau of Investigations IAFIS web site, http://www.fbi.gov/hq/cjisd/iafis/iafisbuilds.htm.

Fisher, B. A. J. (2000). Techniques of crime scene investigation, 6th edition. Boca Raton, FL: CRC Press LLC.

Forensic Technology Inc. web site, http://www.fti-ibis.com/products.html.

GEP-IFSH Database of nuclear DNA, http://www.ertzaintza.net/cgi-bin/ db2www.exe/adn.d2w/INTRODUCCION?IDIOMA=INGLES.

Huckenbeck, W., Kuntze, K., and Scheil, H. G. The distribution of the human DNA-PCR polymorphisms; http://www.uni-duesseldorf.de/WWW/MedFak/Serology/whats.html.

JusticTrax Inc. web-site, http://www.justicetrax.com/contact_us.htm.

LabVantage LIMS web site, http://www.labvantage.com/.

PoliceCenter.com, Law Enforcement Legal Information Source, NIBIN web site, http://policecenter.com/funding/031000nibin.shtml.

Saferstein, R. (1998). Criminalistics: *An introduction to forensic science,* 6th edition, Upper Saddle River, NJ: Prentice Hall.

Stotts, D. *et al.* (1994). A patterned injury digital library for collaborative forensic medicine, http://www.csdl.tamu.edu/DL94/paper/stotts.html.

Thermo LabSystems, LIMS web site, http://www.labsystems.com/products/#LIMS.

U.S. Department of Justice, Drug Enforcement Administration Drug Pointer Index web site, http://www.usdoj.gov/dea/programs/ndpix.htm.

U.S. Department of Justice. Federal Bureau of Investigation, Freedom of Information Act, NCIC, http://foia.fbi.gov/ncic552.htm.

U.S. Fire Administration web site, http://www.usfa.fema.gov/.

Willuweit, S., and Roewer, L. Institute of Legal Medicine, Humboldt University, Berlin, Germany, Y-STR Haplotype Reference Database, http://ystr.charite.de/index_mkl.html.

FORTRAN

Michael Metcalf
Berlin, Germany

GLOSSARY

abstract data type A class of objects defined by a representation-dependent specification with attributes that specify the names and define the abstract meanings of the operations associated with an object of the class.
alias An alternative identification for a named section of computer memory, as represented in a computer program.
compiler A program that translates the text, or source code, of a program into object code.
direct access A method of organizing records in a file such that reference to the individual records is made by an index rather than by any absolute or relative position in the file.
formatted data A representation of data as a string of characters, as opposed to the internal representation of data within a computer memory.
high-level language A programming language that has no necessary relation to the physical computer on which a program will be executed.
low-level language A programming language that specifies the detailed operations to be carried out by the computer on which the program will be executed.
procedure A sequence of instructions forming a unit that may be referenced by name elsewhere in a program.
scope The range within an executable program over which an entity may be accessed.
sequential access A method of organizing records in a file such that access is achieved according to their physical order in the file, beginning with the first record and ending with the last record.
syntax The grammar expressing the manner in which a program is coded.

FORTRAN is a high-level programming language that is widely used in scientific, engineering, mathematical, and other disciplines. The first version was developed by J. Backus *et al.* at IBM in the 1950s and, as the first high-level language, it experienced a rapid spread in use. It was first standardized in 1966 and subsequently in 1978, 1991, and 1997. The language is a procedural, imperative, compiled language with a syntax well suited to a direct representation of mathematical formulas. Individual procedures may be compiled separately or grouped into modules, allowing the convenient construction of very large programs and subroutine libraries. Procedures communicate via global data areas or by argument association. The language contains features for array processing, abstract data types, and, using pointers, dynamic data structures. A major advantage in the use of Fortran is that compilers typically generate very efficient object code, allowing an optimal use of computing resources; it is also comparatively easy to learn, and programs are readily portable. FORTRAN is not an object-oriented programming language, but features to make it so are likely to be available in the next standard.

I. THE DEVELOPMENT OF FORTRAN[1]

A. Origins

In the early days of computing, programming was a tedious activity performed by experts in low-level

[1]See also Metcalf, M. (1992). FORTRAN programming language. In *Encyclopedia of physical science and technology,* Vol. 6, pp. 631–650. New York: Academic Press.

languages that required an intimate knowledge of the hardware being used. Spurred by a perceived economic need to provide a form of "automatic programming" to allow efficient use of manpower and computers, John Backus of IBM proposed, at the end of 1953, to begin development of the FORTRAN programming language (the name being a contraction of FORmula TRANslation). The overriding objective of the development team was to produce a compiler that would produce efficient object code comparable to that of handwritten assembly code.

By comparison, less thought was put into the design of the language itself, except to allow engineers and scientists to write programs without the large overheads of bookkeeping and detailed planning involved in low-level languages. However, this first version, now known as FORTRAN I, already contained early forms of constructs that have survived to the present day: simple and subscripted variables, the assignment statement, a DO-loop, mixed-mode arithmetic, and input/output (I/O) specifications.

Many novel compiling techniques also had to be developed. So it was not until 1957 that the first compiler was released to users of the initial target machine, the IBM 704. First experience showed that, indeed, it increased programmer efficiency and allowed scientists and engineers to program easily themselves. The source form and syntax were a step forward in liberating programmers from the rigid input formats of assembly languages. Fortran was an immediate success.

The ease of learning and the continuing stress on optimization are the two hallmarks of Fortran that have contributed to its continued popularity.

B. From FORTRAN I to FORTRAN 66

Based on the experience with FORTRAN I, it was decided to introduce a new version, FORTRAN II, in 1958. The crucial differences between the two were the introduction of subprograms, with their associated concepts of shared data areas, and separate compilation. FORTRAN II became the basis for the development of compilers by other manufacturers and was described by D. McCracken and E. Organick in their respective textbooks. A more advanced version was developed specifically for the IBM 704—FORTRAN III—but was never released.

In 1961 the IBM users' organization, SHARE, requested from IBM a new version, now called FORTRAN IV, that contained type statements, the logical `IF` statement, the possibility to pass procedure names as arguments, the `DATA` statement, and the `BLOCK DATA` subprogram. Some original features, such as device-dependent I/O statements, were dropped.

FORTRAN IV was released in 1962 and quickly became available on other machines, but often in the form of a dialect. Indeed, the proliferation of these dialects led an American Standards Association (ASA) working group, X3.4.3, to develop a standard definition for the language. In 1966 standards for FORTRAN and Basic FORTRAN were published, based on FORTRAN IV and FORTRAN II, respectively. These were the first programming languages to achieve recognition as national, and subsequently international, standards and are now known as FORTRAN 66.

FORTRAN 66 was made available on almost every computer made at that time and was often pressed into service for a multitude of tasks for which it had never been designed. Thus began a period during which it was very popular with scientists, but, because of limitations such as the lack of character-handling facilities, became increasingly criticized, especially by academic computer scientists (who often preferred the ALGOL language).

C. FORTRAN 77

The permissiveness of the standard, whereby any extensions could be implemented on a given processor so long as it still correctly processed a standard-conforming program, led again to a proliferation of dialects. These dialects typically provided much-needed additional features, such as bit handling, or gave access to hardware-specific features, such as byte-oriented data types. Since the rules of ASA's successor, the American National Standards Institute (ANSI), require that a standard be reaffirmed, withdrawn, or revised after a 5-year period has elapsed, the reformed Fortran committee, now designated X3J3, decided on a revision. This was published by ANSI in 1978 and became known as FORTRAN 77. It was adopted by the International Standards Organization (ISO) shortly afterward. The new standard brought with it many new features, for instance, the `IF...THEN...ELSE` construct, a character data type, and much enhanced I/O. It was largely backward compatible with FORTRAN 66.

The language was rather slow to spread. This was due, in part, to certain conversion problems and also to the decision of one large manufacturer, IBM, not to introduce a new compiler until 1982. It was thus only in the mid-1980s that FORTRAN 77 finally took over from FORTRAN 66 as the most used version. Nevertheless, this standard gave Fortran a new vigor that allowed it to maintain its position as the most widely used scientific applications language of its time, becoming hugely suc-

cessful in this domain. However, it began to yield its position as a teaching language, except for those who would require it in their later careers.

D. The Battle for Fortran 90[2]

After 30 years of existence, Fortran was far from being the only programming language available on most computers. In the course of time new languages had been developed, and, where they were demonstrably more suitable for a particular type of application, they had been adopted in preference to Fortran for that purpose. Fortran's superiority had always been in the area of numerical, scientific, engineering, and technical applications. To bring FORTRAN properly up to date, X3J3, working as a development body for the ISO committee ISO/IEC JTC1/SC22/WG5 (or simply WG5), once again prepared a new standard, now known as FORTRAN 90.

X3J3 itself was a body composed of representatives of computer hardware and software vendors, users, and academia. It reported directly to its parent committee, X3 (computer systems), which was responsible for actually adopting or rejecting the proposed draft standards presented to it. In these decisions, it tried to ensure that the proposals really did represent a consensus of those concerned. X3J3 acted as the development body for WG5, consisting of international experts responsible for recommending that a draft standard become an international standard.

What were the justifications for continuing to revise the definition of the FORTRAN language? As well as standardizing vendor extensions, there was a need to modernize it in response to the developments in language design that had been exploited in other languages, such as APL, ALGOL 68, Pascal, Ada, C, and C++. Here, X3J3 could draw on the obvious benefits of concepts such as data hiding. In the same vein, there was the need to begin to provide an alternative to dangerous storage association; to abolish the rigidity of the outmoded source form; to improve further on the regularity of the language, as well as to increase the safety of programming in the language; and to tighten the conformance requirements. To preserve the vast investment in FORTRAN 77 codes, the whole of FORTRAN 77 was retained as a subset. However, unlike the previous standard, which resulted almost entirely from an effort to standardize *existing practices,* the FORTRAN 90 standard was much more a *development* of the language, introducing features that were new to FORTRAN, although based on experience in other languages. This tactic, in fact, proved to be highly controversial, both within the committee and with the general public, and it was not until 1991, after much vigorous debate and 13 years of work, that FORTRAN 90 was finally published by the ISO.

The main features of FORTRAN 90 were, first and foremost, the array language and abstract data types. The former is built on whole array operations and assignments, array sections, intrinsic procedures for arrays, and dynamic storage. It was designed with optimization in mind. The latter is built on modules and module procedures, derived data types, operator overloading, and generic interfaces, together with pointers. Also important were the new facilities for numerical computation, including a set of numeric inquiry functions, the parameterization of the intrinsic types, new control constructs—`select case` and new forms of `do`—internal and recursive procedures and optional and keyword arguments, improved I/O facilities, and many new intrinsic procedures. Last, but not least, were the new free source form, an improved style of attribute-oriented specifications, the `implicit none` statement, and a mechanism for identifying redundant features for subsequent removal from the language. The requirement on compilers to be able to identify, for example, syntax extensions, and to report why a program had been rejected were also significant. The resulting language was not only a far more powerful tool than its predecessor, but also a safer and more reliable one. Storage association, with its attendant dangers, was not abolished, but was rendered unnecessary. Indeed, experience showed that compilers detected errors far more frequently than before, resulting in a faster development cycle. The array syntax and recursion also allowed quite compact code to be written, a further aid to safe programming.

E. Finishing Touches: FORTRAN 95

Following the publication of FORTRAN 90, two further significant developments concerning the FORTRAN language occurred. The first was the continued operation of the two FORTRAN standards committees, J3 (as X3J3 is now known) and WG5, and the second was the founding of the High Performance FORTRAN Forum (HPFF).

Early on in their deliberations, the standards committees decided on a strategy whereby a minor revision of FORTRAN 90 would be prepared by the

[2]The text in the remainder of this section is based on Metcalf, M., and Reid, J. (1999). *FORTRAN 90/95 explained.* Oxford New York: Oxford Univ. Press. By permission of Oxford University Press.

mid-1990s with a further revision by about the year 2000. The first revision, FORTRAN 95, is the subject of the following section.

The HPFF was set up in an effort to define a set of extensions to Fortran, such that it would be possible to write portable, single-threaded code when using parallel computers for handling problems involving large sets of data that could be represented by regular grids. This version of FORTRAN was to be known as High Performance FORTRAN (HPF), and it was quickly decided, given the array features of FORTRAN 90, that this should be its base language. Thus, the final form of HPF was a superset of FORTRAN 90, with the main extensions being in the form of directives that take the form of FORTRAN 90 comment lines, and so are recognized as directives only by an HPF processor. However, it did become necessary also to add some additional syntax, as not all the desired features could be accommodated in the form of such directives.

The work of J3 and WG5 went on at the same time as that of HPFF, and the bodies liaised closely. It was evident that, in order to avoid the development of divergent dialects of Fortran, it would be desirable to include the new syntax defined by HPFF in FORTRAN 95, and, indeed, the HPF features were the most significant new features that FORTRAN 95 introduced. The other changes consisted mainly of what are known as corrections, clarifications, and interpretations. These came about as it was quickly discovered, as FORTRAN 90 compilers were written and used, that the text of the FORTRAN 90 standard contained a few errors that required correction, some obscure wording that required further textual clarification, and ambiguous statements that required interpretation. (J3 and WG5 processed about 200 requests for interpretation.) All the resulting changes were included in the FORTRAN 95 standard. Apart from the HPF syntax and the corrections, only a small number of other pressing but minor language changes were made.

The details of FORTRAN 95 were finalized in November 1995, and the new ISO standard, replacing FORTRAN 90, was adopted in 1997 as ISO/IEC 1539-1:1997.

II. FORTRAN 95

A. FORTRAN Concepts

For simplicity, only the modern form of the language will be described. Old concepts retained for compatibility with early versions of FORTRAN will not be discussed. One of the hallmarks of FORTRAN is its backward compatibility: features are rarely removed, and, if they are, it is done in a way that allows vendors to continue to support them as extensions.

A complete FORTRAN executable program consists of one main program and zero or more subprograms. The subprograms

- May be stand-alone *external procedures*
- May be contained in another subprogram as *internal procedures*
- may be contained in a module as *module procedures*

(an example appears in Section II.C.5). Main programs, external procedures, and modules are *program units* that can be compiled separately. Subprograms are either *subroutines* or *functions*. These may be referenced by a main program or by one another; the references may be specified to be recursive.

Certain function and subroutine names[3] are defined as part of the FORTRAN language; there are 115 in all. These perform such commonly required tasks as returning the absolute value of a number, `ABS(x)`, or establishing whether a pointer is associated, `ASSOCIATED(ptr)`. A full list of the classes of intrinsic procedures appears in Table I.

A program unit consists of a sequence of statements and optional comment lines (lines beginning with !). The first statement is a header line and the last statement is an `end` statement.

A statement itself is a sequence of syntactic items. Those forming the FORTRAN language will be described below. A statement is written on an initial line and, if necessary, up to 39 continuation lines. A line may have up to 132 characters, and all lines but the last of a statement must be terminated with the character `&`. Any statement may be labeled. Only `FORMAT` statements must be labeled. Statements other than assignment begin with a keyword. Keywords may also appear in other contexts, but none is a reserved word, and whether a sequence of characters represents a keyword or another syntactic item is determined by its context. Blanks are significant within the syntax. The syntactic items are names, numeric labels, keywords, constants, operators, and separators and are composed of the elements of the allowed character set—the upper- and lowercase letters, the digits 0–9, and the special characters such as = + * / ( ) < >, etc.

[3]The main text of this article adopts the arbitrary convention of using uppercase letters for Fortran keywords and names and lowercase letters for other identifiers. Fortran is case insensitive.

Table I Classes of Intrinsic Procedures, with Examples

Class of intrinsic procedure	Examples
Argument presence inquiry function	`present`
Numeric functions	`abs, modulo`
Mathematical functions	`acos, log`
Character functions	`achar, index`
Character inquiry function	`len`
Kind functions	`kind`
Logical function	`logical`
Numeric inquiry functions	`digits, tiny`
Bit inquiry function	`bit_size`
Bit manipulation functions	`btest, ior`
Transfer function	`transfer`
Floating-point manipulation functions	`exponent`
Vector and matrix multiply functions	`dot_product`
Array reduction functions	`all, sum`
Array inquiry functions	`allocated`
Array construction functions	`merge, pack`
Array reshape function	`reshape`
Array manipulation functions	`cshift`
Array location functions	`maxloc`
Pointer association status functions	`associated`
Intrinsic subroutines	`cpu_time`

A name consists of 1–31 alphanumeric characters, including _, the first of which must be alphabetic. It is used to designate the name of a program unit, variable, or other program entity. An example is `mass_of_quark`.

A keyword is a sequence of letters according to the specification of the language. A simplified summary of the keywords that may appear in FORTRAN 95 statements appears in Table II.

There are five types of constants. Three are numeric:

- The signed integer, with a form such as `99`
- The reals such as `12.3` and `1.23e1`
- The complex such as `(1.0, 2.3)`

One is logical, with the values `.true.` or `.false.`; and one is a character, such as `String`. These forms of constants correspond to the five intrinsic data types. In addition, there are forms of binary, octal, and hexadecimal constants associated with the integer data type. Further, the standard defines two kinds of reals, one of a greater precision than the other. Other kinds are permitted, but not required for the nonreal data types. Fortran also allows the specification of abstract data types, with a corresponding form of constant.

FORTRAN defines four classes of operators:

- The arithmetic
- The logical
- The character
- The relational

The arithmetic operators are

- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)
- `/` (division)
- `**` (exponentiation)

These five operators are all binary operators used to form expressions such as `x+y**2`. The subtraction operator is also a unary operator, as in the assignment `y=-x`. Expressions are evaluated from left to right (except for repeated exponentiation), following the order of precedence of the operators (first **, then * or /, then + or −) and the presence of parentheses. The operands of arithmetic expressions may be of mixed types and kinds.

The logical operators in decreasing order of precedence are

- `.not.` (negation)
- `.and.` (intersection)
- `.or.` (union)
- `.eqv.` and `.neqv.` (equivalence and nonequivalence)

Examples are `.not.flag` and `on.or.off`.

There is just one character operator, `//`, denoting the concatenation of two strings. A substring notation may be used to specify part of a character string.

The relational operators are

- `<` (less than)
- `<=` (less than or equal)
- `==` (equal)
- `/=` (not equal)
- `>` (greater than)
- `>=` (greater than or equal)

They are used to form relational expressions involving pairs of operands of numeric or character type, as in `mass < limit`.

All these operators may be used to form expressions involving abstract data types; however, a definition of

Table II Simplified List of FORTRAN 95 Statement Keywords[a]

Non-executable statements	Executable statements
Program units and subprograms `program` `module` `end module` `use` `only` `external` `intrinsic` `subroutine` `recursive, pure, elemental` `function` `recursive, pure, elemental, result` `entry` `contains` `interface` `operator, assignment` `end interface` `module procedure`	Assignment *`variable = expression`* *`pointer => target`* `if` `where` `forall` Program units and subprograms `call` `return` `end` `program, subroutine, function` Dynamic storage allocation `allocate` `deallocate` `nullify` Control constructs `do` `do . . . while` `cycle` `exit` `continue` `end do` `if . . . then` `else if . . . then` `else` `end if` `select case` `case` `case default` `end select` `go to` `stop` `where` `elsewhere` `end where` `forall` `end forall` Input/output `read` `write` `print` `rewind` `end file` `backspace` `open` `close` `inquire` `format`
Type specification `integer` `real` `logical` `complex` `character` `double precision` `type` `implicit none` `implicit`	
Other data specification `parameter` `public` `private` `pointer` `target` `allocatable` `dimension` `intent` `external` `intrinsic` `optional` `save` `type` `end type` `data` `block data` `end block data` `namelist` `equivalence` `common`	

[a]An indented keyword depends on the keyword on the preceding line.

the details of the operation must be provided. In addition, named operators, such as `.multiply.`, may be defined.

In the foregoing text, scalar variables, possessing both a name and a type, have been introduced and used in examples. An ordered sequence of data elements may be specified as a homogeneous array. An array is a variable that has a name, a type, a rank (its number of dimensions), and extents in each dimension (from a lower bound to an upper bound). Whole arrays and subsections of arrays may appear as operands in expressions and in assignments, provided that the rules of conformance (that array operands match in shape) are respected. Single array elements and subsections are specified using integer subscript expressions.

B. Nonexecutable Statements

The statements of FORTRAN, as shown in Table II, can be grouped by function and fall into two classes: the non-executable statements that provide information on, for instance, the attributes of variables, and the executable statements that specify an action, such as an assignment or a subroutine reference. This subsection contains a brief description of a selection of the nonexecutable statements and some of their closely associated executable statements. The descriptions will be by way of examples rather than formal notation, since only a summary can be provided here. Other executable statements are similarly described in Section II.C.

1. Program Units and Subprograms

Each executable program has one main program that begins with an optional `program` statement, as in

```
PROGRAM solve_linear_equations
```

Subprograms must begin with a header line that, for functions, identifies the name, the arguments, and, optionally, the type, the `RESULT` variable, and whether it is any of `RECURSIVE`, `PURE`, or `ELEMENTAL`. A recursive function may reference itself, directly or indirectly; a pure function has no side effects; and an elemental function is defined in terms of scalars but may be referenced for arrays. An example of a function statement is

```
REAL FUNCTION minimum(a, b, func)
```

where `a`, `b`, and `func` are dummy arguments of the function. They are separated by commas, may be variables or dummy procedure names, and provide a means of exchanging data and procedure names between procedures. Functions are referenced as operands of expressions or simply as the right-hand side of an assignment statement, as in

```
f_min = minimum(1.0, 2.0, COSH)
```

A subroutine statement is similar, but it has no type, no result variable, and the arguments are optional:

```
SUBROUTINE finish
```

Subroutines are referenced via a `CALL` statement, as in

```
CALL finish
```

All procedures end with an executable `END` statement, as in

```
END SUBROUTINE finish
```

Data may be shared in FORTRAN not only by argument passing, but also by making them globally accessible in modules. An example is

```
MODULE state
   INTEGER, DIMENSION(100) :: switches
END MODULE state
```

which allows the variable `switches` to be accessed in any program unit that contains the statement

```
USE state
```

An arbitrary number of global variables may be specified in a module. There is a mechanism, via the `ONLY` keyword, that allows just a selection of variables to be accessed. Modules may also contain type definitions; interface blocks, and, following a `CONTAINS` statement, procedure definitions. Within a module (as also for internal procedures), interface checking is automatically provided. Interface blocks provide an alternative mechanism whereby compilers can check the details of a procedure reference without having access to the whole procedure. They also permit, via the `MODULE PROCEDURE` statement, the link to be made between a defined operator for a derived type and the procedure that is actually to perform the operation (as will be seen in Fig. 1). One module may reference another.

2. Type Specifications

The type of a variable, named constant or function, is, by default, determined by the initial letter of its name: integer if it is `i`, `j`, `j`, `l`, `m`, or `n` and real otherwise. However, it is more usual to specify the type explicitly with a type statement:

```
INTEGER :: grid
REAL    :: mass
```

```
MODULE intervals
   TYPE interval
      REAL :: lower, upper
   END TYPE interval
   INTERFACE OPERATOR(+)
      MODULE PROCEDURE add_interval
   END INTERFACE
   INTERFACE ASSIGNMENT(=)
      MODULE PROCEDURE interval_from_real
   END INTERFACE
CONTAINS
   FUNCTION add_interval(a,b)
      TYPE(interval)         :: add_interval
      TYPE(interval), INTENT(IN) :: a, b
      add_interval%lower = a%lower + b%lower
      add_interval%upper = a%upper + b%upper
   END FUNCTION add_interval
   SUBROUTINE interval_from_real(a,b)
      TYPE(interval), INTENT(OUT) :: a
      REAL, INTENT(IN)            :: b
      a%lower = b
      a%upper = b
   END SUBROUTINE interval_from_real
END MODULE intervals
```

Figure 1 A module that defines a type suitable for interval arithmetic, the operation to perform addition between two scalar objects of that type, and the assignment of a real object to an object of type interval.

This form of the type statement, using the double-colon notation, allows other attributes to be specified also. Several entities may appear in a single type statement. These features are described in Section II.B.3.

It is possible to force an explicit type definition by adding to any program unit the statement

```
IMPLICIT NONE
```

As well as being able to specify entities as being of one of the intrinsic types, it is also possible to specify them to be of an abstract data type. In order to be able to do this, the details of the components of the type must first be defined, as in

```
TYPE point
   REAL :: x, y
END TYPE point
```

With this definition of a point available, entities of that type may be specified, as in

```
TYPE(point) :: coord1, coord2
```

An individual component of an entity of a derived type is referenced, as in `coord1%y`. It is possible for a component of a derived type to be of a previously defined derived type.

3. Other Data Specifications

Other attributes may be specified on type statements (or also as separate statements). Thus, a named constant is specified by adding the `PARAMETER` keyword to a type statement, as in

```
COMPLEX, PARAMETER :: i = (0.0, 1.0)
```

Entities in modules are normally freely accessible, or public, via a `USE` statement. This access may be controlled more precisely using the `PUBLIC` and `PRIVATE` attributes and statements. The following sequence sets the default for a module to private and then makes several entities public:

```
PRIVATE
CHARACTER(LEN=99), public :: string
LOGICAL, PUBLIC           :: flag
```

(Here, the character variable `string` is given a fixed length of 99; the default length is 1.)

Pointers in Fortran are not a distinct data type, but an attribute of a variable or function that can be specified as part of the type definition:

```
INTEGER, POINTER :: ptr1
```

A pointer may point at any variable of the same type and rank that is itself specified to be also a pointer or to be a target:

```
INTEGER, TARGET :: dot
```

An array variable is specified by giving it the `DIMENSION` attribute, together with information on its rank (up to seven is possible) and its extent in each dimension. The statement

```
REAL, DIMENSION(-9:9, 100) :: grid
```

specifies a variable to be an array of rank two, with 19×100 elements. It is possible to specify the rank, but not the extents if the array is given the `ALLOCATABLE` or the `POINTER` attribute, allowing the storage to be allocated dynamically during the course of program execution:

```
REAL, ALLOCATABLE, DIMENSION(:, :)
                                  :: grid
```

Variables that are dummy arguments of procedures are normally considered to be defined on entry to the procedure and again on exit. However, a variable may be defined to be for input only, output only, or both by specifying appropriate `INTENT` attributes:

```
INTEGER, INTENT(IN)    :: var1
REAL, INTENT(OUT)      :: var2
COMPLEX, INTENT(INOUT) :: var3
```

Further, a dummy argument does not necessarily have to have a corresponding actual argument if it is given the OPTIONAL attribute:

```
REAL, INTENT(OUT), OPTIONAL :: var2
```

Whether an optional argument is actually present or not can be determined by the intrinsic logical inquiry function PRESENT.

The COMMON, EQUIVALENCE, and BLOCK DATA statements are associated with the static storage model that was the only one available in previous versions of FORTRAN. Now that dynamic storage facilities are available, they are functionally redundant and not described here.

C. FORTRAN Features

1. Expressions and Assignments

Expressions can be classified as numeric, relational, logical, character, or of derived type according to the types of the operands. Examples, for real x and y and logical flag, are

```
SQRT(x**2 + y**2) - 4.8   ! numeric
x <= y                    ! relational
x <= y .and. flag         ! logical
'FORTRAN '//'95'          ! character
```

Such expressions may appear as the right-hand sides of assignment statements when the left-hand sides are of the appropriate type, as in

```
flag = x <= y
```

or as the action statement part of a logical IF statement:

```
IF (x >= 0.0) y = SQRT(x)
```

Expressions and assignments for abstract data types are illustrated in Section II.C.5.

A second form of assignment is the pointer assignment, in which one pointer is set to point at the target of another pointer or at a target object:

```
ptr1 => dot
```

A third form is the where statement which performs array assignment under control of a mask:

```
WHERE (grid >= 0.0) grid = SQRT(grid)
```

following which, the positive values of grid (Section II.B.3) will have been replaced by their roots, other elements of the array remaining unchanged. A more elaborate where construct is also available.

Finally, the FORALL statement (or construct) enables an array assignment to be specified such that the individual element assignments take place independently in any order and, on appropriate hardware, even simultaneously, as in

```
FORALL (i = 1:n) grid(i, i) = x(i)
```

for a numeric array x of sufficient length and an appropriate value of the integer n.

2. Control Constructs

Fortran has five different control constructs, providing flexible mechanisms for program control. Two of these, the WHERE and FORALL, will be further dealt with in the context of the array language in Section II.C.4.

The IF construct is a mechanism for branching depending on a condition. It allows either the execution of a sequence of statements to depend on a condition or the execution of alternative sequences of statements to depend on alternative conditions. As with all constructs, one IF construct may be nested within another. A simple example is

```
IF (i < 0) THEN
   x = 0.0
ELSE IF (k < 0) THEN
   x = 1.0
ELSE
   x = -1.0
END IF
```

in which x acquires a new value depending on the values of i and k.

A similar construct is the CASE construct. However, here only one expression is evaluated for testing, and the evaluated expression may belong to no more than one of a series of predefined sets of values. For ch of type character and other variables of type logical, an example is

```
SELECT CASE (ch)
CASE ('c', 'd', 'r': )
   ch_type = .true.
CASE ('i' : 'n')
   int_type = .true.
CASE DEFAULT
   real_type =.true.
END SELECT
```

in which just one of the logical variables is set true depending on the value of ch.

The DO construct is the means whereby iterations are controlled in FORTRAN. A simple form is

```
DO i = 2, n
   a(i) = a(i-1) + b(i)
END DO
```

which replaces each element of the array a from the second to the *n*th by the foregoing element plus the corresponding element of b. More elaborate control can be achieved through use of the EXIT and CYCLE statements. In

```
DO i = 1, n
   :
   IF (grid(i, i) < 0.0 ) EXIT
   :
END DO
```

(where : denotes some statements), the statement immediately following the construct will be executed should a negative diagonal element of grid be encountered. If, instead, CYCLE was substituted for EXIT, the next iteration of the loop would be taken. Other forms of the DO statement allow an unbounded construct:

```
DO
```

or stepping through the loop with a constant stride:

```
DO i = 1, 100, 3
```

The first of these two forms requires an EXIT statement or other transfer of control to ensure that it terminates. The second will perform only 33 iterations here.

In practice, DO constructs are often deeply nested, but some of their functionality can frequently be obtained through the use of the array language.

3. Modular Programming

The ability of FORTRAN to accommodate the separate compilation of program units is both a strength and a weakness. The strength is derived from the fact that separate compilation allows the incremental construction of very large programs comprising perhaps hundreds or thousands of procedures. These will often be compiled into libraries of object code. The weakness comes from the fact that, when programs consist only of a main program and many external procedures, no interface checking mechanism is automatically available. Thus, if one procedure invokes another with too few or too many arguments, or with arguments whose types or ranks do not correctly match, run-time errors occur that can be very difficult to locate.

Using the modular programming features of FORTRAN, however, enables such errors to be detected and corrected at compile time. At its most extreme, a program is built of a main program that uses one or more modules. Each module contains a group of related procedures. The modules possibly reference one another. Using this program architecture, all the procedure interfaces are explicit and so are known to the compiler. Any mismatch is diagnosed at compile time and a whole class of insidious programming errors is eliminated at an early stage. An example appears in Section II.C.5.

Where large libraries of external procedures already exist, or where it is deemed more appropriate to continue in this style, it is still possible to achieve compile-time interface checking by the use of interface blocks. These interface blocks contain the information required about each procedure interface and they can also be conveniently packaged into one or more modules that are accessed at compile time. Here, the maintenance task is somewhat greater, but a large measure of safety can nevertheless be achieved.

FORTRAN 95 is in this and many other respects a much safer programming language than FORTRAN 77.

4. Array Processing

FORTRAN 95 allows variables and function results to be array valued. Such objects can be manipulated in much the same way as scalar objects. Thus, for the real, conformable (matching in shape) arrays a, b, c, and x,

```
x = (-b + SQRT(b**2 - 4.0*a*c)/(2.0*a)
```

may be written to solve a set of quadratic equations rather than just one. Scalar values within array expressions are "broadcast" across the whole extent of the array variables. In this example, the use of the WHERE or FORALL construct would be necessary to avoid division by zero if this is likely to occur:

```
WHERE (a /= 0.0)
   x = (-b + SQRT(b**2 -
                        4.0*a*c)/(2.0*a)
ELSEWHERE
   x = -HUGE(0.0)
END WHERE
```

The form of the FORALL construct is similar. Any procedure referenced within a FORALL statement or construct must have the PURE attribute to ensure that it has no side effects that could cause the result to depend on the order of execution.

The SQRT function is used here as an elemental function—although defined in terms of scalars it will return an array-valued result for an array-valued argument. Many intrinsic procedures are elemental, and a user-written procedure that is pure may be made elemental by adding the ELEMENTAL keyword to its header line and following certain rules.

An array need not necessarily be specified with a fixed size. If a is an array dummy argument, it may be declared as an assumed-shape array

```
REAL, DIMENSION(:, :) :: a
```

where the actual array bounds are transmitted at run time.

An array that is local to a procedure may be specified as an automatic array whose bounds depend on another argument, as in

```
REAL, DIMENSION(SIZE(a)) :: work
```

to define an array work whose size depends on that of another array a.

Storage may be allocated dynamically to an array at run time. Given the second specification of grid in Section II.B.3,

```
ALLOCATE(grid(50, 100))
```

may be written to give the required space to the array. The space may later be deallocated and then allocated afresh. The ALLOCATE and DEALLOCATE statements are equally useful for arrays that have the POINTER attribute, in particular for dynamic arrays that are components of a derived type (as these may have the POINTER, but not the ALLOCATABLE attribute).

Given an array that has, one way or another, been given bounds, a single (scalar) element may be referenced using a subscript notation as in grid(9, 15). A subsection of the array may be referenced using a triplet notation as in grid(1:10, 10:100:10), which is an array-valued subobject that may, in turn, appear in array expressions and assignments. It is that subsection of grid that consists of its first 10 elements in the first dimension and every 10th element in the second. It is a ten-by-ten, rank-two array.

An array-valued constant is known as an array constructor. It has a variety of forms, with a simple one being

```
grid(1:5, 10) = (/ 1.0, 2.0, 3.0,
                                4.0, 5.0 /)
```

A pointer may be used as a dynamic alias to an array or to an array subobject. If we add the TARGET attribute to the first specification of grid, and define an appropriate pointer array as

```
REAL, DIMENSION(:), POINTER :: window
```

then the pointer assignment

```
window => grid(0:9, 1)
```

makes window a rank-one array of length 10.

The many array functions listed in Table I are an important and integral part of the array language.

5. Abstract Data Types and Data Structures

In Section II.B.2 the concept of derived-data types was introduced. When one of these is defined, it is further possible to define the meanings associated with operations and assignments on objects of that type or between an object of that type and an object of another derived or intrinsic type. The usual mechanism is to specify functions (for operations) or subroutines (for assignments) that perform the necessary task and to place these in a module that can be accessed to gain access to the types, the operations, and the assignments. An example that defines a type suitable for interval arithmetic, the operation to perform addition between two scalar objects of that type, and the assignment of a real object to an object of type interval is shown in Fig. 1.

A snippet of code that makes use of the facilities thus defined would be

```
PROGRAM test
   USE intervals
   REAL             :: a = 1.0
   TYPE(interval)   :: b, c
   b = a      ! defined assignment
   c = a      ! defined assignment
   c = b + c  ! defined operation
   PRINT *, a, b, c
END PROGRAM test
```

(This main program and the module intervals together form a complete, executable program.)

Derived-data types may contain components that have the POINTER attribute. This allows the construction of data structures of arbitrary complexity. If the elements of a sparse vector are to be held as a chain of variables, a suitable data type would be

```
TYPE entry
   REAL                   :: value
   INTEGER                :: index
   TYPE(entry), pointer   :: next =>
                                  NULL()
END TYPE entry
```

A chain can then be specified by

```
TYPE(entry), POINTER :: chain
```

and the first variable can be defined by

```
ALLOCATE(chain)
chain%value = 1.0
chain%index = 10
```

Normally, such a list would be manipulated with the aid of additional pointers that reference, for instance,

its first and current entries and with utility procedures for adding and removing entries, etc. Once again, it would be usual to package the type and the procedures that manipulate the list into a module.

6. Input/Output

FORTRAN defines a comprehensive set of features for performing I/O operations. The basic concepts involved are that data transfer takes place to and from files. A unit of data is a record, and a single data transfer statement may read or write one or more records. Data transfer is device independent.

An external file is said to exist if it is a file to which a program might have access. A file that exists for a running program is said to be connected to that program if it is associated with a unit number known to that program. Connection is achieved either because the file is preconnected by the operating system, frequently the case for terminal I/O, or because an explicit connection has been made by the `OPEN` statement. The `OPEN` statement contains a number of optional parameters that not only allow the connection of a file to a unit, but also allow the status, access method (sequential or direct), formatting style, position, and permitted actions to be specified.

Corresponding to the `OPEN` statement is the `CLOSE` statement that causes a file to become disconnected.

At any time during the execution of a program it is possible to inquire about the status of any unit or file, whether existing or connected, and the various attributes associated with that file, if any. This is carried out with the `INQUIRE` statement. The combination of these three file manipulation statements permits programs to control dynamically their access to files in a portable way.

Simpler levels of file manipulation are provided by the `REWIND`, `BACKSPACE`, and `ENDFILE` statements, each having the function denoted by its name.

The actual data transfer operations are performed by the `READ` statement (for input) and the `PRINT` and `WRITE` statements (for output). These three statements have various forms and are sometimes used in conjunction with the (nonexecutable) `FORMAT` statement. The purpose of the `FORMAT` statement is to provide a description of the transformations that data are to undergo if written in a formatted form. Such format specifications may also appear directly within a data transfer statement. A particular form of specification allows list-directed I/O to be carried out (where the transformations are data driven). No format specification is required for binary or unformatted I/O.

As well as operations on external files, it is possible to perform data transfer to and from internal files, specified as variables of type character.

D. Redundant and Obsolete Features

The procedures under which J3 works require that a period of notice be given before any existing feature is removed from the language. This means, in practice, a minimum of one revision cycle, which for FORTRAN means about 5 years. The need to remove features is evident: if the only action of the committee is to add new features, the language will become grotesquely large, with many overlapping and redundant items. The solution adopted by J3 is to publish as an appendix to the standard a set of two lists showing which items have been removed or are candidates for eventual removal. One list contains the *deleted features,* those that have been removed. The second list contains the *obsolescent features,* those considered to be outmoded and redundant and which are candidates for deletion in a later revision. In practice, because of great pressure to keep language revisions backward compatible with previous ones, the lists are too short to have any real effect on the size of the language. The features in these lists are not described in this article.

E. Subset Languages

At about the time of the publication of FORTRAN 95, another development occurred: the specification of two similar proprietary subset versions of FORTRAN 90, ELF90 and F, that retain modern features while casting aside outmoded ones. These subsets can be regarded as vehicles for the teaching of a safe and reliable style of modern programming.

III. THE STATUS OF FORTRAN

A. Challenges from Other Languages

FORTRAN has always had a slightly old-fashioned image. In the 1960s, the block-structured language ALGOL was regarded as superior to FORTRAN. In the 1970s the more powerful PL/1 was expected to replace FORTRAN. ALGOL's successors, Pascal and Ada, caused FORTRAN proponents some concern in the 1980s. Meanwhile, FORTRAN continued successfully as the workhorse of scientific computing. However, in the late 1980s, two developments began to se-

riously impinge on FORTRAN's predominance in this field: UNIX and object orientation.

UNIX brought with it the highly successful, general-purpose language C. This has been further developed into C++, an object-oriented language. C is widely used for all levels of system programming and has also made inroads into FORTRAN's traditional numerical computing community. C++ has come to dominate many programming applications, especially those requiring sophisticated program interfaces. Another object-oriented language, Java, has also come into widespread use. Whether FORTRAN 95 and its successors will, in the long term, be able to withstand the immense pressure from other languages remains an open question.

B. The International FORTRAN Community

FORTRAN is an international language both in the sense that it is used throughout the world and in the sense that the community of international users has actively participated in the development of recent standards. Furthermore, the Internet and the World Wide Web have facilitated the development of international user communities, for instance the newsgroup `comp.lang.FORTRAN`, the FORTRAN Web site at http://www.fortran.com, and the discussion group at http://www.jiscmail.ac.uk/lists/comp-fortran-90.html. These groups are important in the dissemination of FORTRAN news, such as announcements of new compilers, and as sources of help and advice to users in general. The ACM publishes *Fortran Forum,* a special interest publication on FORTRAN with an international readership and containing articles on FORTRAN language developments and user experience (see http://store.acm.org/acmstore).

C. Current Developments: FORTRAN 2000

The next full language revision, referred to as FORTRAN 2000, was originally planned for 2001. However, the target date for publication is now 2004. Its main features have been chosen: the handling of floating-point exceptions; permitting allocatable arrays as structure components, dummy arguments, and function results;[4] interoperability with C; parameterized data types; object orientation: constructors/destructors, inheritance, and polymorphism; derived type I/O; asynchronous I/O; procedure variables; and various minor enhancements. This activity provides a means of ensuring that FORTRAN remains a powerful and well-honed tool for numerical and scientific applications.

SEE ALSO THE FOLLOWING ARTICLES

C and C++ • COBOL • Java • Object-Oriented Programming • Pascal • Programming Languages Classification • Simulation Languages • Unix Operating System • Visual Basic

BIBLIOGRAPHY

(1984). *Annals of the History of Computers,* Vol. 6, 349–359.

(1996). *Computer Standards & Interfaces,* Vol. 18.

International Standards Organization. (1997). ISO/IEC 1539-1. ISO, Geneva, Switzerland.

Koebel, C., *et al.* (1994). *The high performance FORTRAN handbook.* Cambridge, MA: MIT Press.

Metcalf, M. (1992). FORTRAN programming language. In *Encyclopedia of physical science and technology,* Vol. 6, pp. 631–650. New York: Academic Press.

Metcalf, M., and Reid, J. (1996). *The F programming language.* Oxford/New York: Oxford Univ. Press.

Metcalf, M., and Reid, J. (1999). *FORTRAN 90/95 explained.* Oxford/New York: Oxford Univ. Press.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1996). Numerical recipes in FORTRAN 90: The art of parallel scientific computing. *FORTRAN numerical recipes,* Vol. 2. Cambridge, UK: Cambridge Univ. Press.

[4]These first two features have already been formalized as standard extensions to FORTRAN 95 in ISO/IEC Technical Reports TR 15580 and TR 15581, respectively.

Frame Relay

Ray Hunt
University of Canterbury, Christchurch, New Zealand

GLOSSARY

asynchronous transfer mode (ATM) A dedicated-connection switching technology that organizes digital data into 53-byte cell units and transmits them over a physical medium using digital signal technology. Individually, a cell is processed asynchronously relative to other related cells and is queued before being multiplexed over the transmission path. It is widely used in Telco and ISP backbone networks.

backward error congestion notification/forward error congestion notification (BECN/FECN) BECN/FECN bits are set in the control field of the Frame Relay header and are used as congestion indicators. Use of these bits will assist in preventing data loss due to an excessive data arrival rate.

committed information rate (CIR) The minimum bandwidth which a frame relay network guarantees to provide on a continuous basis. Any application can expect to receive the CIR—and in most situation a great deal more bandwidth.

data link connection identifier (DLCI) A 10 bit addressing mechanism which specifies the virtual circuit which will carry the frame, therefore designating a specific destination port.

data terminal equipment (DTE) In computer data transmission the DTE is the an interface specified by protocols such as RS-232C V.24, X.21, etc. that a computer uses to exchange data with a serial communication device such as a modem.

digital data service (DDS) Also called digital data network, leased line network or private line and is the a fixed and dedicated connection provided by a Telco which is purchased at a specified bandwidth.

discard eligibility (DE) This bit is set in the control field of a Frame Relay header and along with BECN/FECN bits controls whether data can be discarded. Such a decision also related to the application data being carried.

frame check sequence (FCS) A sequence of bits carried with a Frame Relay frame and used to check for errors introduced during transmission. The Frame Relay network will detect errors with the frame's FCS, discarding any frames found to be in error. It is based upon a Cyclic Redundancy Check (CRC) using the V.41 16-bit generator polynomial.

integrated services digital network (ISDN) A set of CCITT/ITU standards for digital transmission over telephone copper wire and other media. ISDN requires adapters at both ends of the transmission and can operate at speeds of multiple 64 Kbps blocks (up to 1.544 or 2.048 Mbps). It is normally offered as a Telco service.

International Telecommunication Union (ITU) The primary international body for fostering cooperative standards for telecommunications equipment and systems. It was formerly known as the CCITT and is located in Geneva, Switzerland.

internet service provider (ISP) A company that provides individuals and other companies access to

the Internet and other related services such as Web site building and virtual hosting. An ISP has the equipment and the telecommunication line access required to have a POP (point-of-presence) on the Internet for the geographic area served.

link access procedure—balanced/D–channel (LAP–B/LAP–D) An ITU-T layer 2 link layer interface protocol standard used as part of X.25. LAP-D is a similar standard specified for use in the D-channel of ISDN networks.

multiple protocol label switching (MPLS) A standards-approved technology for speeding up network traffic flow and making it easier to manage. MPLS involves setting up a specific path for a given sequence of packets, identified by a label placed in each packet, thus saving the time needed for a router to look up the address to the next node to forward the packet to.

packet switching network (PSN). A network in which small units of data (packets) are routed through a network based on the destination address in each packet. Breaking communication down into packets allows the same data path to be shared among many users in the network. Most traffic over the Internet uses packet switching.

permanent virtual circuit/switched virtual circuit (PVC/SVC) A virtual circuit is a circuit or path between points in a network that appears to be a discrete, physical path but is actually a managed pool of circuit resources from which specific circuits are allocated as needed to meet traffic requirements. A PVC is a virtual circuit that is permanently available to the user just as though it were a dedicated or leased line continuously reserved for that user. An SVC is a virtual circuit in which a connection session is set up for a user only for the duration of a connection. PVCs are an important feature of Frame Relay networks.

simple network management protocol (SNMP). The protocol governing network management and the monitoring of network devices and their functions. It is not necessarily limited to TCP/IP networks although this is its primary focus. SNMP exists in three versions and is described formally by the IETF RFCs.

time division multiplexing/statistical time division multiplexing (TDMSTDM) TDM is a scheme in which numerous signals are combined for transmission on a single communications line or channel. Each signal is broken up into many segments, each having very short duration. The allocation for each transmission is can be fixed (Frequency, Time Division Multiplexing) or allocated on a statistical basis (STDM) so that unused allocation can be made available to other users.

FRAME RELAY is a wide area networking architecture designed to provide flexible, high performance interconnection for regional, national, and international networks. Frame relay has an important role to play in the gradual replacement of traditional digital data leased line services as Frame relay frequently offers an improved cost/performance ratio. Also Frame relay is an important network infrastructure used by companies for connection to Internet service providers (ISPs) and as a very viable and cost-effective framework for transport of TCP/IP Internet/Intranet/Extranet services. Frame relay has characteristics which make it an excellent choice for interconnecting distributed local area networks (LANs) which generate intermittent high data rate traffic volumes. The bandwidth requirements for current LAN-based client/server applications such as transaction processing, image and graphics transmission as well as distributed database systems are very different from those used in earlier applications. These modern applications all process large volumes of data, transmit intermittent high-speed bursts, and are intolerant of long delays. To satisfy the requirements of these applications, the bandwidth management system in local-to-local and local-to-wide area networking must offer access to high bandwidth on demand, direct connectivity to all other points in the network, and consumption only of bandwidth actually needed. Frame relay is not only a wide area networking protocol but can be used to provide a range of functions and services including acting as:

- An access interface protocol
- A transport service
- A network processing and interface to routing service
- A network management service

This article discusses a range of background, technical, design, and operational issues by examining:

- What is frame relay?
- Frame relay's protocols
- Frame relay architecture in relation to other wide area telco services
- Congestion control, operation, and performance of frame relay networks

I. INTRODUCTION

A. What Is Frame Relay?

Frame relay is a wide area networking solution designed to provide flexible, high performance interconnection for regional, national, and international networks. It has characteristics which make it an excellent choice for interconnecting distributed LANs. These characteristics can be seen in Fig. 1 and are subsequently explained.

Frame relay is a good choice for handling *bursty* high data rate traffic commonly generated by LAN applications. In the past the predominant technologies used to solve LAN-to-LAN interconnection have been: (1) circuit switching by way of time division multiplexing (TDM) and (2) X.25 packet switching.

Providing frame relay is used in conjunction with an appropriate application environment, it offers a solution that is superior to either of these other two technologies. Frame relay in the wide area network (WAN) combined with Ethernet in the LAN provides an excellent data link architecture to carry traditional TCP/IP traffic. However, it is important to note that there are certain traffic characteristics such as high-volume continuous data, combined voice and conventional data, or low-volume message traffic which would still make the above options applicable in certain circumstances. In general, frame relay is more flexible and cost-effective than TDM, providing better throughput than X.25 packet switching. This results in lower latency and a more scaleable infrastructure for TCP/IP-based applications than for pure router-based interface protocol (IP) networks.

Applications which require high data rates or low delay benefit from frame relay as can be seen in Fig. 2. Frame relay networks support access speeds up to T1 (1.544 Mbps) and *E*1 (2.048 Mbps) in comparison with a maximum of 64 kbps for most X.25 packet-switched networks. Higher data rates of 34 Mbps (E3) and 45 Mbps (T3) have been demonstrated by some vendors and frame relay networks also offer lower delays than those found in conventional packet-based networks.

X.25 was designed to support traditional data networking applications commonly found in the 1980s. This protocol contains features that have been made redundant by intelligent end station higher layer protocols and the low error rate associated with modern digital transmission systems. Its architectural stack is shown in Fig. 3.

In recognizing both the need for higher performance packet switching networks as well as the duplication in the functionality between X.25 networks and the end system higher layer protocols, the designers of frame relay initially created a protocol by simply eliminating the superfluous functionality found within the X.25 protocol. By eliminating this additional functionality and reducing the packet overhead, the designers created a protocol that permitted switches to be built with speed and not error-free transmission as the primary objective. One could note the effect that occurred when the backbone went from analog-based circuits to digital. This resulted in a tremendous reduction in bit errors, which allowed the link-level error checking to be replaced by end-to-end error checking. By removing this overhead, frame relay can be viewed as a simplified X.25 protocol as shown in Fig. 4.

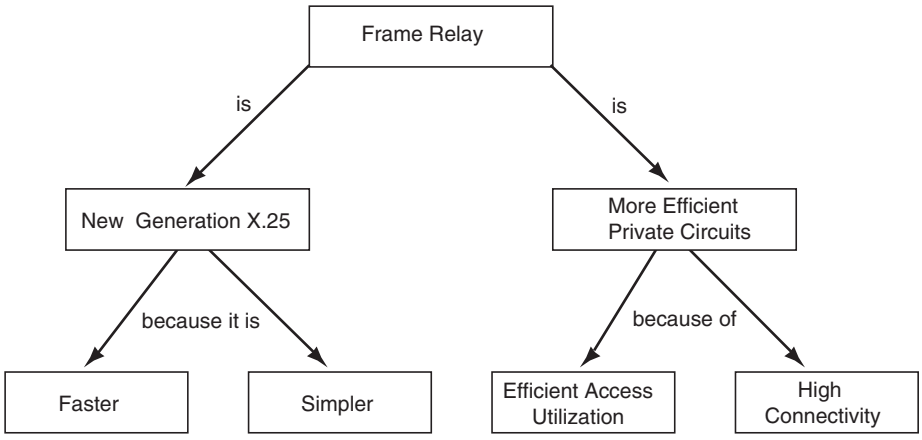


Figure 1 Frame relay characteristics.

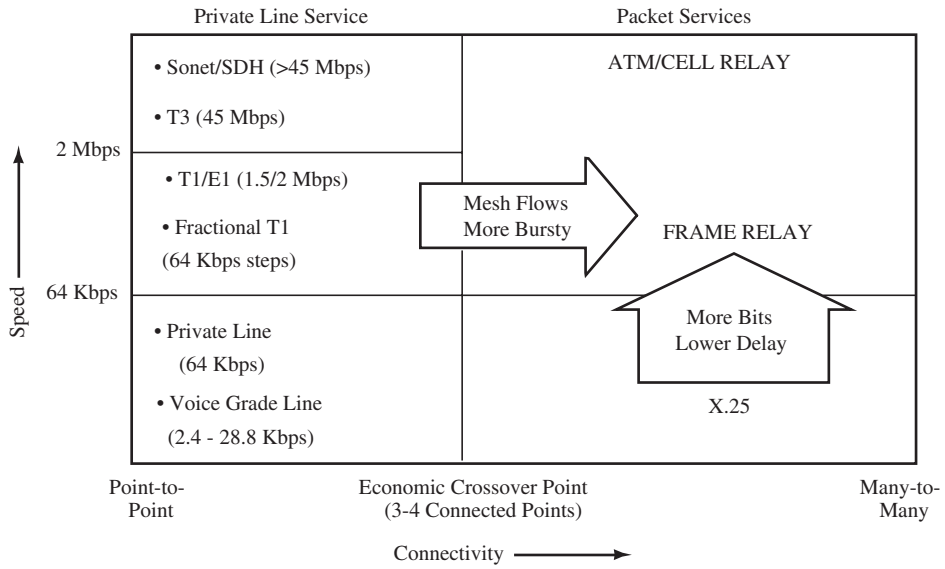


Figure 2 Speed/connectivity relationship for leased line and packet-based services.

The demand for high performance WANs is being driven by two key factors:

1. Lower cost for wide area bandwidth
2. Changing data traffic patterns involving high data rates, bursty traffic, low delay, and high connectivity of data devices from a variety of sources.

In many traditional networks the size of the data burst is no more than one screenfull, being generated by on-line human users. In this situation the primary factor governing network delay is the queuing of the transactions for access to the circuit resulting from the overall volume of the users' traffic.

In modern client/server and distributed network systems, the interacting elements are usually intelligent devices from which relatively large bursts of data are generated. Transactions such as pages of memory or a large number of entries resulting from a database inquiry can generate data bursts of many kilobytes in length. The transmission of a high resolution color screen image can generate a data burst of up to several megabytes.

The characteristics of such data mean that the transport network needs to transfer these bursts with delays of no more than a few milliseconds in order to provide efficient performance for the application.

B. Improved Efficiency over Private Circuits

Frame relay can be viewed as a more cost-effective and efficient private circuit solution. It therefore has a particular region of operating advantage over a number

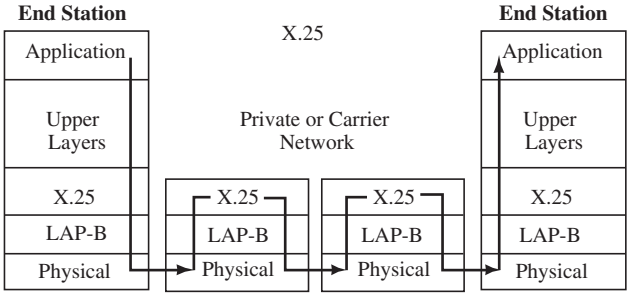


Figure 3 Architectural stack of an X.25 packet switching network.

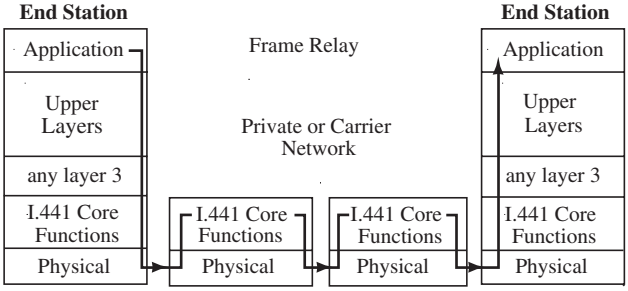


Figure 4 Architectural stack of a frame relay network.

of other technologies which have evolved as a consequence of the changing nature of traffic patterns.

Through the use of statistical time division multiplexing (STDM) and fast switching technology, frame relay allows networks to be configured for the support of low delay, bursty, and high data rate traffic. It therefore provides better utilization of bandwidth for both the network access and backbone trunks.

In conventional TDM, stations are allocated fixed bandwidth slots and once this bandwidth has been allocated it cannot be used by any other station. In an STDM environment—typical of X.25, IP networks, and frame relay—the bandwidth is dynamically shared among multiple stations. When one station has a frame to send, it has access to the entire bandwidth for as long as necessary to transmit this frame. However, if the circuit is in use, the frame must be queued for transmission. For STDM a small amount of overhead must be added to allow for the segregation of frames with various destinations.

Frame relay permits the allocation of the entire bandwidth of the access link to a single user for the duration of the burst. In contrast, TDM limits the allocated bandwidth to a fraction of the overall access circuit speed.

Figure 5a shows an example of a data burst over four node TDM and 5b shows frame relay networks each of which have a 1.536-Mbps access link speed. In the case of the frame relay network, the bandwidth is not subdivided into smaller increments among the connected end stations as is the case with the TDM network. The lower time for access to the network results in an overall improvement in the end-to-end performance of the frame relay network by a factor of three in this example. This disparity in performance increases as the number of locations and connectivity requirements increase.

STDM is the basis of Ethernet and token ring LAN protocols and therefore frame relay networks operate well with these LAN protocols. Further, the number of WAN ports on routers are reduced as can be seen in Fig. 6 where three hardware interfaces are required for access to the TDM network while only one is required for access to the frame relay network. More commonly today, the router can use a single channelized TDM interface (T1/E1) and treat it with three logical channels. This is only possible with internal TDM cards. For a similar reason, more efficient allocation of circuits between frame relay backbone switches is possible.

Some of the key advantages of Frame Relay over circuit switching technologies are:

- Bandwidth savings are unavailable to circuit switches as they are unable to change bandwidth dynamically. Bandwidth is allocated ahead of time at fixed rates and down fixed paths.

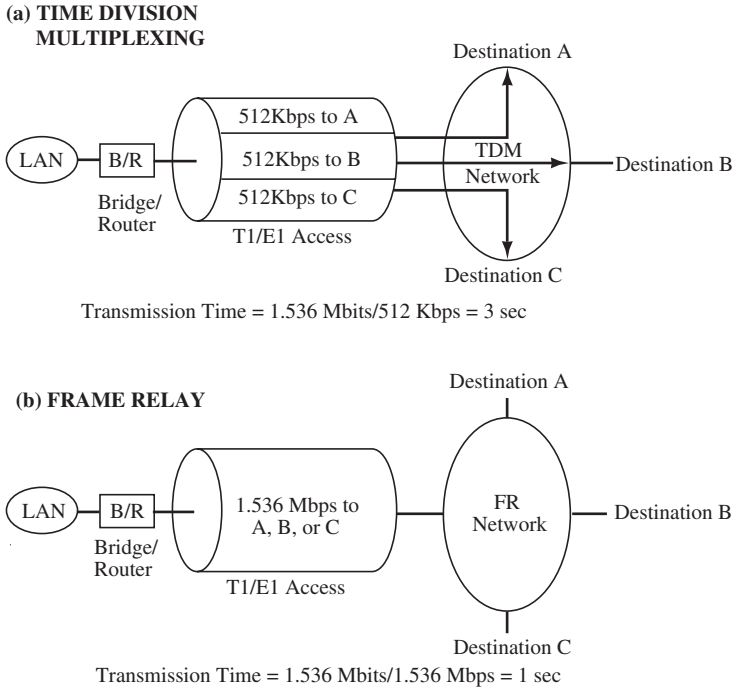


Figure 5 Data burst transmission over TDM and frame relay networks.

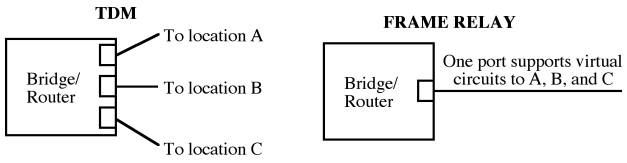


Figure 6 Access ports to TDM and frame relay networks.

- Performance is poor because a circuit switch cannot make high bandwidth available without waste. The resulting lower speeds mean lower throughput. Moreover, since full connectivity is not provided, traffic must be routed through intermediate nodes before reaching its destination thus further increasing the delay.

Frame relay is based upon the equitable sharing of network resources including switches and circuits among the connected users although careful attention must be paid to the setting of the many parameters in order to gain good performance as will be discussed later. From the customer's point of view, where a meshed network is required which must support bursty traffic, a frame relay service is likely to be an attractive option. The reason most customers switch from leased lines to frame relay is the resulting cost savings. A single line can become multiplied within the network backbone which allows service providers to technically oversubscribe the backbone, giving each user access to much higher speed/capacity service when they need it (but charging them much less since most of the time they use much lower capacity).

C. Efficient Access Utilization

Adequate price/performance cannot be guaranteed when the stations using the network generate continuous traffic for substantial periods of time (minutes or even hours). Frame relay networks are therefore not well suited to applications which generate continuous data flows such as are often found with voice and video. In such cases the asynchronous transfer mode (ATM) offers a more satisfactory solution. Frame relay can, however, provide excellent performance when handling bursty data such as often associated with LAN-to-LAN applications. Voice/video solutions can be effective over frame relay, but only when the ratio of these applications to data applications is low. This has therefore limited the voice market to trunking between corporate offices.

Servers on LANs usually communicate as peers. This peer-to-peer communication among servers tends to create traffic patterns that require a mesh topology. The older terminal-to-host architectures were well served by star or tree networking topologies, but the mesh topology of modern networks demands a new level of network performance. Frame relay is well designed to handle these mesh traffic flows commonly found in peer-to-peer architectures by providing logical connections by way of virtual circuits between any pair of end stations in the network.

II. FRAME RELAY, FAST PACKET SWITCHING, CELL RELAY, ATM, IP ROUTING/SWITCHING

A. Architectural Comparison

These terms are frequently used loosely and it is worth taking a moment to clarify their definitions in order to understand the relationship between them.

Frame relay and cell relay are two complementary technologies which form the basis of many of today's modern WANs. Cell relay uses fixed size cells defined at layer 1 of the Open System Interconnection (OSI) model, thus processing and queue management at the network nodes is relatively simple. The cell length is 48 octets (plus 5 octets of header), therefore the overhead is a minimum of 10% which is relatively high with a corresponding reduction in bandwidth. Cells form the basic unit used to carry variable length frames in frame relay networks. Cell relay permits better control of delay within the network and is therefore appropriate for delay-sensitive multimedia applications such as packet voice and video. Cell relay offers improved performance by utilizing a fixed packet length and dispensing with rerouting and retransmission features. Thus more effective use can be made of the high-speed digital links between switching nodes.

Frame relay, on the other hand, has variable length frames—commonly up to 1600 octets (or even 4096 octets in some vendors implementations)—requiring more complex functions at the network nodes but a correspondingly lower overhead of around 1%. It is based upon a layer 2 frame of the OSI model. International Telecomunications Union-Telecommunications Standard Sector (ITU-TSS) and American National Standards Institute (ANSI) worked cooperatively within different frameworks to specify the frame relay interface and they have significant shared representation which has assisted in the development of consistent standards. StrataCom (now part of CISCO Systems, Inc.), Digital (now part of Compaq), CISCO, and Northern Telecom were major players in the early development of frame relay standards.

ATM is sometimes referred to as fast packet switching but is really the *mechanism* by which fast packet

switching is achieved. An ATM network is implemented by the provision of high-speed switches which handle fixed size cells. ATM is a similar concept to frame relay, but is intended for broadband networks operating at much higher speeds than frame relay, which can be considered to be the low end (2 Mbps) and ATM (155 Mbps) the high end of high-speed switching. While ATM supports the simultaneous transmission of voice, video and data, frame relay is orientated to conventional computer data. The variable frame size enables it to cope with bursts of data flows between sites. These two technologies complement each other and provide benefits to users and network providers when frame relay is used as an access protocol to a cell relay (ATM) backbone network.

IP routing is the basis of the Internet but cannot compare with frame relay as it is substantially slower in its operation. In fact frame relay is often used to carry IP traffic consistent with the fact that frame relay operates at layer 2 and IP operates at layer 3 of the ISO architecture. Recently developments in IP switching are challenging frame relay, however, so far most IP switching is LAN based although this is likely to change in the future.

B. Standards

Standards for frame relay are defined in ITU-TSS Recommendations I.122 (Framework for Providing Additional Packet Mode Bearer Services) and I.441 (ISDN User Network Interface Data Link Layer Specification) which form the basis for public implementations. However, with many carriers offering frame relay services, there is a need for standards to interconnect both the public and private arenas. The Frame Relay Forum has taken on the task of applying standards in the interests of users and suppliers alike. Most equipment and service vendors also list the Frame Relay Forum standards, (X.36/X.76) in addition to the I-Series standards mentioned above.

C. Frame Relay in the Spectrum of Telecommunication Interconnection Services

Frame relay is used as an ideal for LAN-to-LAN interconnection service with resulting cost savings over the use of private leased digital data service (DDS) services, particularly as a consequence of its elastic bandwidth. In fact over 80% of frame relay's use is directed at LAN-to-LAN communication with the remainder focusing on host-to-terminal and host-to-host communication.

In some ways frame relay is of more significance to the network provider than the user since 64 kbps to 2 Mbps point-to-point links have been widely available for many years. However, the provision of such bandwidth on a continuous basis is uneconomical for many applications and a more flexible or elastic ("rubber") bandwidth system was needed.

There are many LAN-to-LAN connections that have used DDS point-to-point links most satisfactorily for many years. At the lower end of the spectrum, 64-kbps links have been used by some companies where reasonable

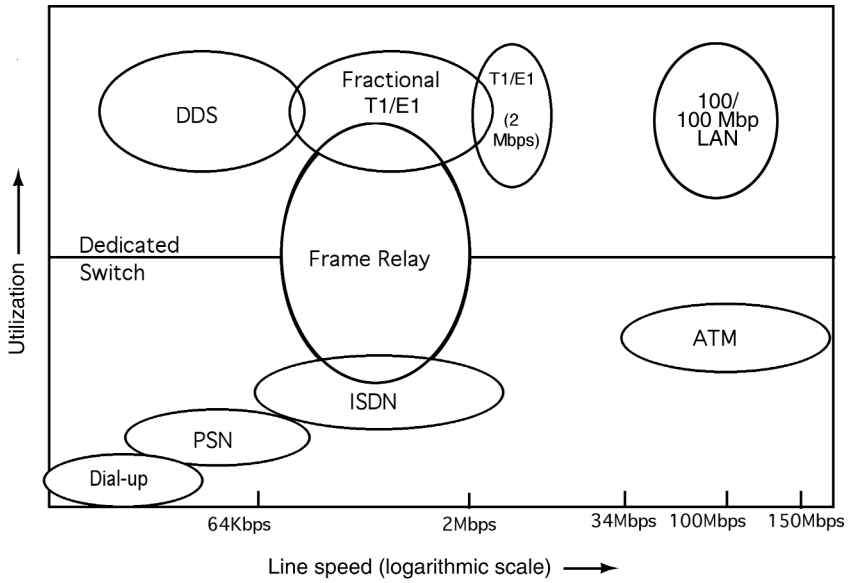


Figure 7 Positioning of telecommunication interconnection services.

data transfer volumes have justified the cost. Further X.25 packet-switching links at a similar speed turned out to be expensive because of the significant connection and volume charges.

Figure 7 shows where frame relay lies in the bandwidth utilization/line speed spectrum covering both switched and dedicated services (note that the line speed axis is a logarithmic one). The table in Fig. 8 provides a comparison of the key technical features of these wide area networking alternatives in comparison with frame relay. In both of these figures it can be seen that overlap with other services does exist and in some situations it may be difficult to choose between some alternatives.

It is generally accepted that DDS (including the various T1/E1 services) is suitable for dedicated interconnection between sites with high volume and/or continuous data flows or where further multiplexing of the services is carried out. Integrated Services Digital Network (ISDN) is appropriate for noncontinuous services where moderate bandwidth is required for short holding times. However ISDN may not offer adequate bandwidth for certain applications. Frame relay is appropriate for major sites and networks requiring bursty traffic flow between interconnected LANs. It is relatively cheap to implement on existing hardware—sometimes it merely requires a software upgrade to the existing X.21 interface on a router.

III. FRAME STRUCTURE

The design of frame relay is based on the assumption that the error rate introduced by the underlying network is very low. Frame relay takes advantage of this fact by not providing error recovery within the structure of the frame. This is done for two reasons:

1. Fiber-optic digital transmission results in very low bit error rates.
2. The combination of modern microprocessor technology and client/server application software makes it appropriate to move intelligence out of the network and into the end stations. Thus in the rare event of an error, recovery can be effected by the end stations.

Frame relay uses a synchronous data transfer protocol at the lower half of the link layer to convey data

	DDS T1/E1	Packet Switched X.25	Frame Relay	ISDN	ATM
Type of Service	Circuit Switched (Leased Circuit)	Packet Switched over SVC & PVC	Frame over PVC (& SVC)	Circuit Switched (B Channel) Packet Switched (D Channel)	Cell Switching
Public/Private	Both	Both	Both	Public	Both
Maximum Speed	1.5/2 Mbps Wideband	64 Kbps Low speed to wideband	2 Mbps (45 Mbps tested) wideband	2 Mbps Low speed to wideband	155 Mbps Broadband
Channel Sharing	Fixed	Statistical	Statistical	Dynamic	Dynamic
Error Control	None	Detection & correction	Detection	D Channel only	None
Transmission Quality	-	Tolerant	Sensitive	Sensitive	Sensitive
Flow and Congestion Control	-	Local and end-to-end	Open & closed loop congestion control	D Channel only	Yes
OSI Layers	1	1-3	1-1.5	1 (B Channel) 1-3 (D Channel)	1-2
Data Unit	Bit	Packet (variable size)	Frame (variable size)	Bit (B Channel) Packet (D Channel)	Cell (fixed size)
Overhead	None	Moderate	Low	Low	High

Figure 8 Comparison of WAN alternatives.

between nodes across the network. This protocol is a subset of the ITU I.441 LAP-D standard.

Frame relay provides transparency to multiple higher level protocols by encapsulating them inside the data field of the frame, thus giving the appearance of an upper layer protocol specific private channel to the user. The frame relay standards designate that frames be formatted as shown in Fig. 9 and the component fields are discussed below.

A. Starting and Ending Flags

Flags are used to delimit the frame in the synchronous bit stream of the physical layer in the same manner as with X.25 (LAP-B) or ISDN (LAP-D) frames. Bit stuffing is used to provide full data transparency. Consecutive frames are separated by a single flag which acts as an ending delimiter for one frame and a starting delimiter for the next frame.

B. Control Field

The control field is two octets long (with optional extension to three or four octets). It contains an address or data link connection identifier (DLCI) as well as control bits for extending the size of the header and for congestion notification.

The DLCI is a 10-bit addressing mechanism and specifies the virtual circuit which will carry the frame, therefore designating a specific destination port. This 10-bit address space is adequate in most situations at present although the extended address (EA) bit permits the support of three and four octets addressing for larger networks in the future. The DLCI normally only has significance within a particular access link. This is called local addressing and allows for the identification of 2^{10} (=1024) destination ports. However, the frame relay standard reserves 32 of these DLCIs for use by the network, as shown in Fig.10.

There are therefore potentially 992 destinations which can be addressed from a single end station. Large networks are difficult to administer when local addressing is used since each access link must maintain a unique list of DLCIs. However, this burden can be improved by the use of global addressing discussed below.

The command/response (CR) bit is not used in frame relay and can therefore be set to any value.

The discard eligibility (DE), forward error congestion notification (FECN), and backward error congestion notification (BECN) bits are used to assist with congestion management and are further discussed under Sections V-VII.

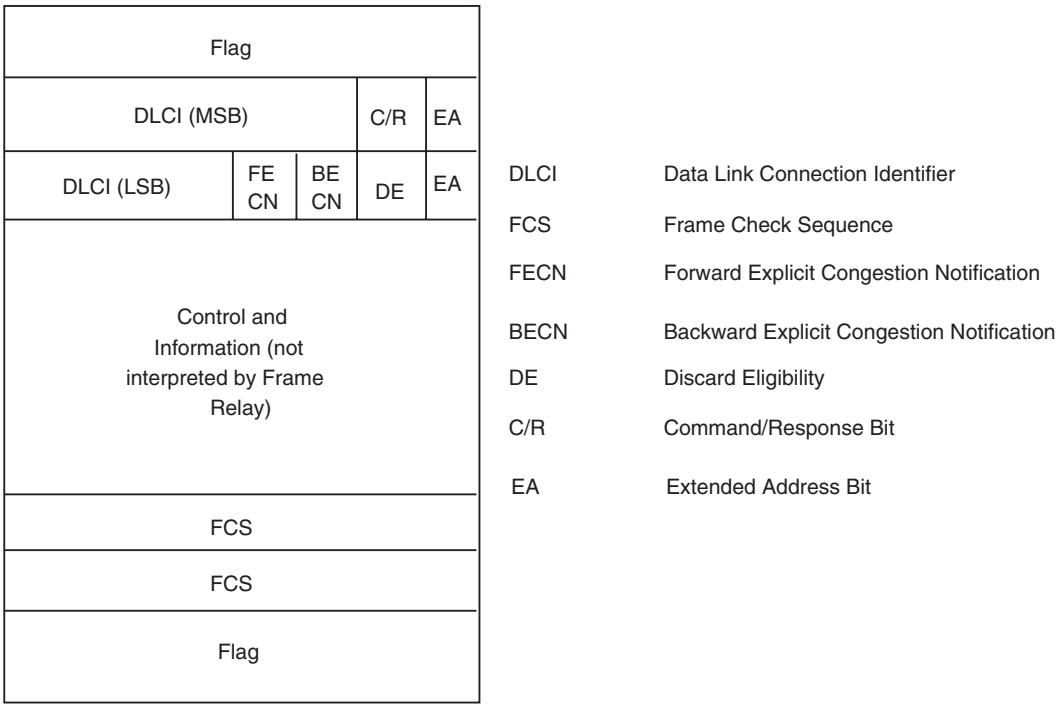


Figure 9 Frame relay frame structure.

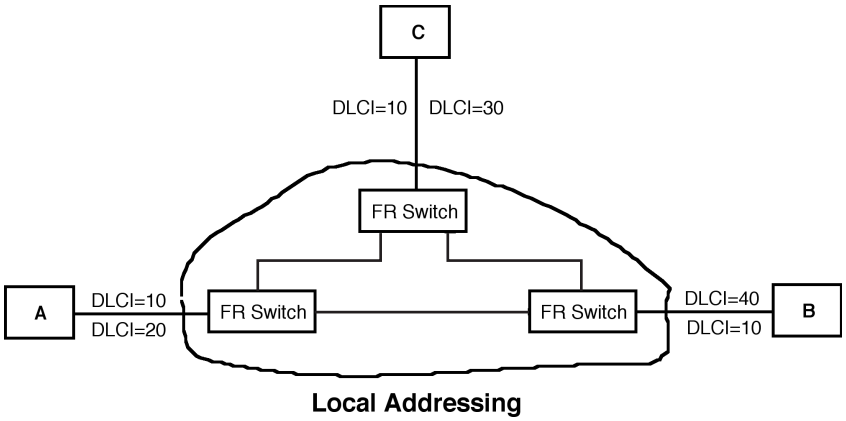


Figure 10 Allocation table for DLCIs.

The EA bit allows the frame relay header to be expanded to three or four octets in order to increase the number of unique DLCIs required for global addressing. In the normal two-octet header, the EA bit in the first octet is set to "0" and in the second octet is set to "1".

C. Local and Global Addressing

In a frame relay network the virtual circuit is identified by the DLCI at the two end stations. For example, in Fig. 11 if station A is logically connected to station B, the virtual circuit is identified as the connection between station A, DLCI 20 and station B, DLCI 10 where the DLCI only has significance at that interface. Further, station A can also be logically connected to station C using DLCI 10 at both interfaces.

As the network increases in size, the number of DLCIs will grow at an exponential rate and DLCI assignment to the virtual circuits becomes a cumbersome affair. Global addressing is an administrative way to improve this where each end station is assigned a unique DLCI. Thus when station A wishes to transmit to station B it will use B's identifier as the DLCI. Similarly station A will receive data from station B over a DLCI which is B's identifier.

D. Multicast

A number of end stations can be designated to be members of a "multicast" group and a specific DLCI is allocated to represent such group addressing. Thus the network takes on the responsibility of delivering multiple copies of the frame to the end stations represented by this DLCI. In the local management in-

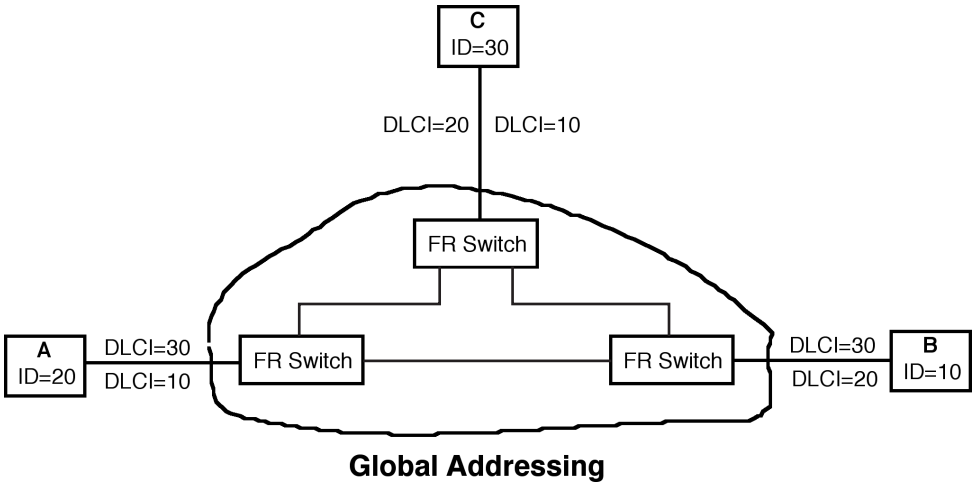


Figure 11 Local and global addressing.

terface (LMI) specification, four DLCIs are reserved to support multicast groups, see Fig. 10.

There are a number of applications which can make use of multicasting. Examples include: distribution of routing table information to routers, distribution of microcode to selected stations, and general document distribution. However, use of multicast addressing can potentially generate high traffic profiles and careful design is necessary in order to ensure satisfactory performance.

E. User Data Field

The user data field is a variable length area used to carry application data along with the upper layer protocol headers. Frame relay frames commonly encapsulate protocols such as TCP/IP as well as proprietary architectural suites. The frame relay standards require devices to support a maximum user data field of 262 octets, although many vendors support frame sizes up to 4096 octets in order to transport LAN application traffic without fragmenting the LAN packets into multiple frames.

F. Frame Check Sequence (FCS)

The frame relay network will detect errors with the frame's FCS, discarding any frames found to be in error. The network will not provide any error correction via retransmission and it is up to the higher layer protocols in the end stations to organize appropriate retransmissions. More specifically the error recovery is moved away from the network architecture and is expected to be provided by the upper layer protocols at the end stations.

The FCS is a cyclic redundancy check (CRC) using the V.41 16-bit generator polynomial. The FCS checks all bits in the frame between the flags. It is implemented in the hardware using a shift register in exactly the same way as with LAP-B and LAP-D frames.

IV. FRAME RELAY FUNCTIONS AND SERVICES

Frame relay is an IP defined by the ITU-TSS and ANSI which specifies network access on a link between an intelligent end station (such as a LAN router) and the network. Frame relay can therefore be viewed as providing:

- An access IP
- A transport service
- Network processing and interface to routing
- Network management services

Each of these will now be briefly described.

A. Frame Relay as an Access IP

The frame relay standard specifies an *access interface* that provides a statistically multiplexed facility and allows multiple logical connections to operate over the physical access link (Fig. 12). These logical connections are maintained via virtual circuits.

The multiplexing facilities of frame relay permit efficient utilization of the access facilities, since each connection can utilize the full bandwidth of the access link when transmitting data. If multiple logical connections are active simultaneously, the access bandwidth is shared via the process of statistical multiplexing. Vendors, routers and other intelligent end-user devices are able to implement procedures based upon priority, or other performance-based criteria, in order to control the connection of the link to the frame relay network. Frame relay's multiplexing capability is provided at the link level of the protocol and is controlled by the DLCI which is contained in the control field of the frame relay header as was shown in Fig. 9.

B. Frame Relay as a Transport Service

Frame relay provides a connection-orientated service that results in a logical connection between the end stations. In the absence of fault conditions, this path is fixed and all frames transmitted follow the same virtual circuit. Frame relay provides a connection-orientated service where all frames are transmitted in sequential order with no loss or duplication. This is different from the connectionless service provided by conventional LANs. Most of these virtual circuits are *permanent* (PVCs) and are fixed at subscription time although *switched* virtual circuits (SVCs) have advantages

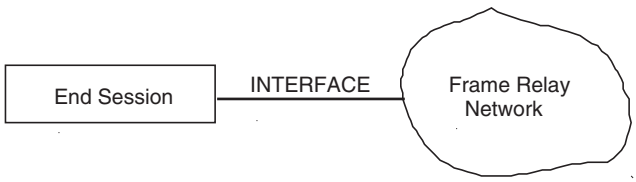


Figure 12 Frame relay as an access IP.

for short-term interconnection requirements. Use of SVCs follows ITU-TSS Recommendation I.450, the signaling protocol used for ISDN networks (Fig. 13).

C. Frame Relay as a Provider of Network Processing and Interface to Routing

Frame relay forms one of the class of technologies entitled fast packet switching which minimizes error-checking and flow control. This means that frame relay can achieve a high throughput than X.25 for any given hardware technology. Another fast packet switching technology is called *cell relay* where cells of the same length are transmitted through the network and even less processing is carried out at the switching nodes. This system forms the basis of ATM networks and multimedia communication (Fig. 14).

Network processing involves establishing and controlling PVCs, routing, allocating adequate bandwidth, and processing frames. These frames are routed across the network using only 1.5 layers of the OSI protocol stack. Only a subset of the full I.441 LAP-D link level protocol is necessary when used in conjunction with error-prone circuits as frames containing errors are simply discarded by the network thus leaving the retransmission to the higher layer protocols in the end stations. This simplifies the functionality of the network by moving a number of functions (equivalent to another 1.5 layers) out of the network and into the end equipment.

Further, current network nodes are very fast and have substantial buffer capacity for internodal trunks that are also of high speed. Therefore link layer flow control and management may not be necessary. By discarding the three most demanding layer 2 functions—error recovery, flow control, and link management—frame relay networks are able to operate at higher speed with a corresponding improvement in performance. In the design of the frame relay protocol, the functions of LAP-D were divided into two sets—core and user selectable functions. Frame relay is only responsible for the former of these two. Specifically, frame relay provides the following functions:

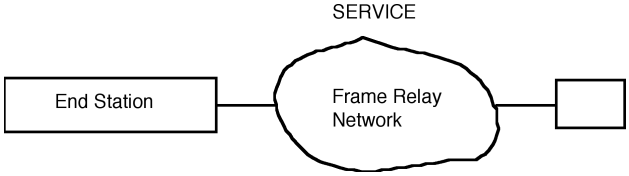


Figure 13 Frame relay as a transport service.

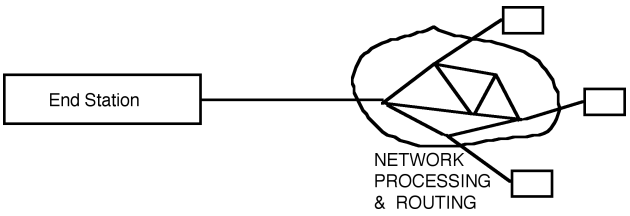


Figure 14 Routing and network processing facilities of frame relay.

- Delimits frames by the use of flags
- Checks for frame integrity (via flags, maximum and minimum frame size)
- Provides data transparency by way of bit stuffing
- Provides multiplexing via DLCI
- Detects errors via the FCS

The user selectable functions include the remaining functions of LAP-D, i.e., error recovery through acknowledgments as well as retransmission and flow control through windowing. A summary of these function can be seen in Fig. 15.

Mechanisms for congestion control were added to the protocol to provide a means for the network to gracefully degrade its performance when the traffic load exceeded the network capacity due to very high traffic bursts or failures within the network.

D. Frame Relay as a Provider of Network Management Services

An important aspect of frame relay is the management of the service which focuses on two areas:

1. **Local management of the interface:** This addresses the messages between the terminal equipment (DTE) and the frame relay network and involves many aspects of the operation of the interface, addition and deletion of DLCIs, LMI extensions, etc. LMI defines a set of messages based upon ITU-TSS Recommendation I.451 message formats in order to facilitate configuration and management of a virtual circuit network. DLCI 1023 is used for the transfer of these messages.
2. **Global management of the service:** This addresses the general health of the network, collects key statistics, and assists with the planning and implementation of the network. Simple Network Management Protocols (SNMP-1, SNMP-2, SNMP-3) are the de facto network management protocol

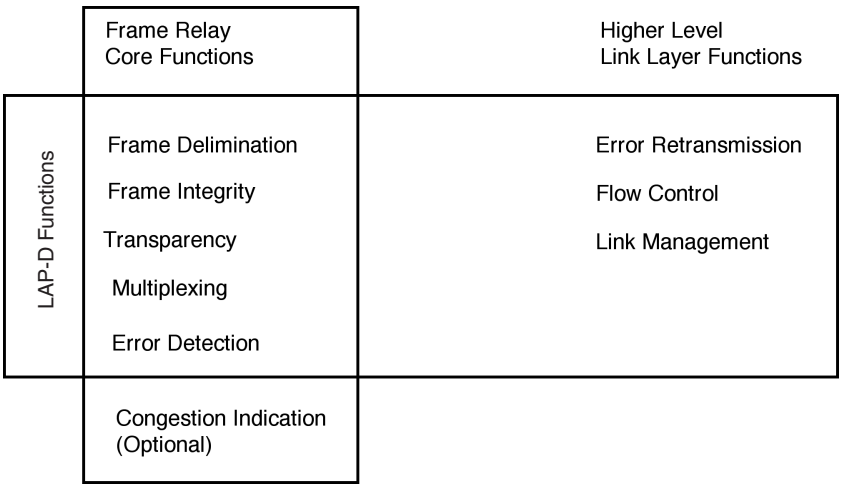


Figure 15 Segregation of LAP-D into core and user selectable functions.

standards. Frame relay networks can be operated as public (shared) networks (usually by the local telco) or as private (dedicated) networks, or some combination (hybrid) of the two.

V. VIRTUAL CIRCUITS AND COMMITTED INFORMATION RATE

The frame relay standard specifies use of either PVCs or SVCs. Virtual circuits in the frame relay context are used for connectivity and addressing and are able to provide the equivalent of private leased circuits but with substantially increased flexibility. All traffic for a particular virtual circuit uses the same path through the network. However, virtual circuits consume no resources when they are not active, but can (in principle) instantaneously carry up to 2 Mbps of traffic. Many PVCs can be established at a single access point between the end station (usually a router) and the network. Each of these PVCs can link different end stations and have different throughput capacity.

The protocols for dynamically setting up SVCs are more complicated than those used for PVCs, although SVCs can be required for international frame relay services. The PVCs in a frame relay network allow any single device to communicate with hundreds of other devices by way of a single access port.

Once frames enter the Frame Relay network they are forwarded along the PVC according to the connection identifier specified by the DLCI in the frame. Data transmitted on a particular PVC is statistically multiplexed onto network trunks along with data from other users. Since this statistical multiplexing process involves variable store-and-forward delays, the throughput for individual transactions will be variable and usually less than the access rate (see Fig. 17).

Nodes in a frame relay network are linked by PVCs each with a committed information rate (CIR) which defines the bandwidth guaranteed to be available to the connection although the maximum amount of bandwidth can be much higher if network capacity is

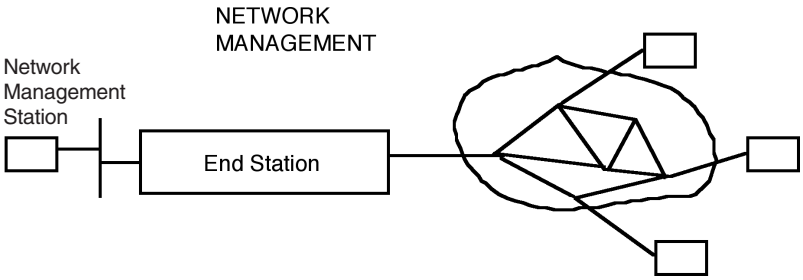


Figure 16 Frame relay and network management.

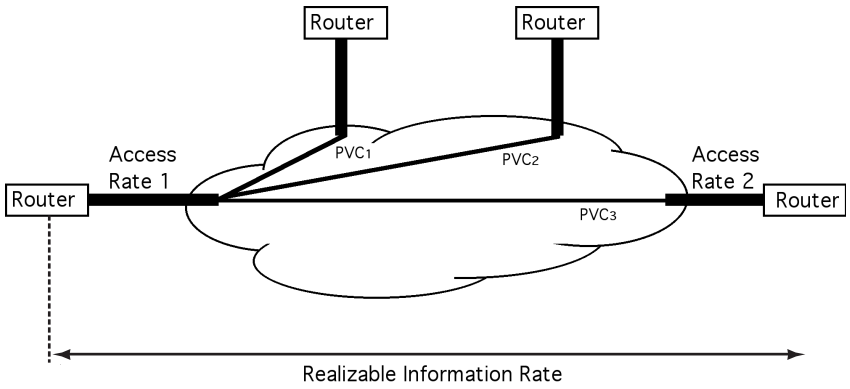


Figure 17 Frame relay interconnection using PVCs.

available. Closely associated with the CIR is the committed burst size (Bc) which specifies the maximum size of a traffic burst which the network commits to deliver during a specified time interval (T) and for a specified virtual circuit, i.e.

$$CIR = Bc/T$$

This does not mean that excess bandwidth is always available as the network can only supply bandwidth up to the access rate of the connection to the network. If the network is congested, the bursts may not make it through the network and data can be discarded. Commonly the CIR can be set to a minimum value of 8 kbps and then increased in blocks of 4 kbps. However, this depends very much upon the equipment supplier and network operator.

CIR can be implemented on a per interface or per PVC basis, and is computed over a small sliding window interval. If the incoming data over the interface (or PVC) exceeds its prespecified CIR, the excess data frames are marked with the discard eligible (DE) bit. Data frames marked with this DE bit are delivered on a best effort basis and are the ones discarded first in the event of network congestion. Although the CIR of the interface cannot exceed the access speed, the total CIR of all the PVCs through an interface can exceed the physical speed of the interface.

The excess burst size (Be) specifies how much data above the committed burst size (Bc), a user can transmit. While CIR deals with the long-term traffic load of an interface (or PVC), excess burst size determines the initial size of a transaction that the network can support. It too is computed over a small sliding window time interval. If the incoming data over an interface (or PVC) exceeds its prespecified excess burst size, then the network will discard the excess data frames. For a PVC, the CIR and excess burst size can be different for the two directions.

For example, Fig. 18 shows that frames 1 and 2 fall within the CIR (or Bc). Frame 3 exceeds the CIR (or Bc) and is counted in the excess burst size (Be) and is therefore marked with the DE bit and *may* be dropped. However, this is highly dependent upon the definition of Tc, which is not standardized. Most equipment uses a Tc of 1 second and there are widespread differences of opinion on when Tc starts—it may be fixed, or floating. A sender may use start of the frame as the reference while the receiver uses end of frame. In reality, the majority of circuits use access line speed as Bc + Be. Finally, frame 4 causes the data rate to exceed the agreed CIR + excess burst size (Bc + Be) and may be discarded by the network thus preventing one user from overloading the network at the expense of other users.

CIR and excess burst size is a form of admission control used to regulate the traffic load at the network interface (or PVC). Selecting the CIR and Be permits a

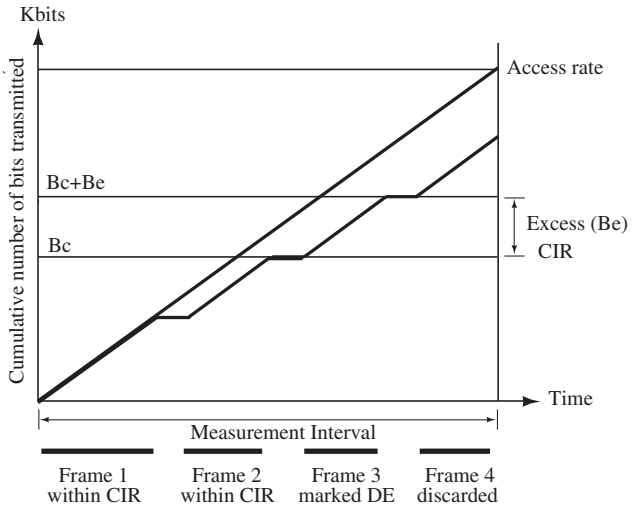


Figure 18 Frame relay capacity management.

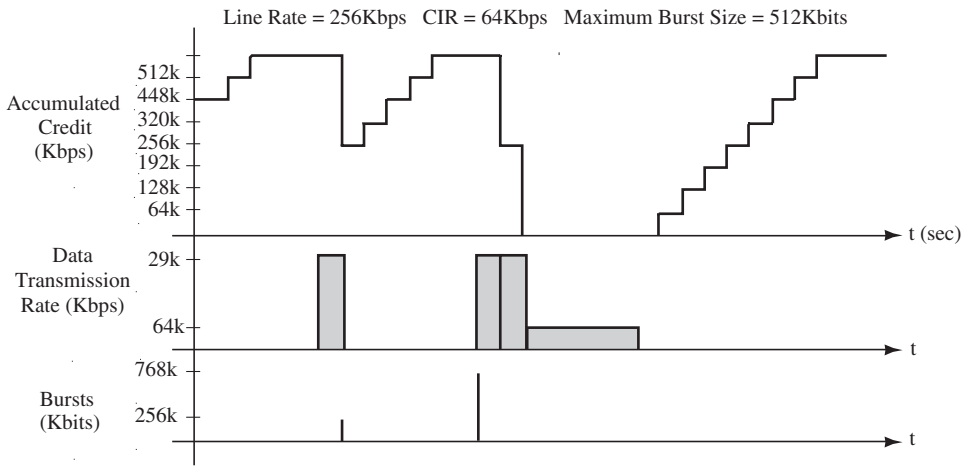


Figure 19 Burst transmission under control of the credit manager algorithm.

user to optimize each PVC for the traffic to be carried. Information in the bursts is transmitted if possible but has a lower probability of getting through. Other admission control mechanisms include sliding windows and the leaky bucket credit manager algorithm.

VI. ADMISSION CONTROL

The function of admission control is to regulate the volume of traffic entering the network in order that data from a single data source will not degrade the network performance as seen by other users. These schemes apply at the network interface and two of the most widely used are the sliding window and leaky bucket credit manager schemes.

Network nodes can still be overloaded even with good admission polices. For example, many end stations might want to communicate with the same distant end station and the combined traffic load to this end station might overload part of the network. The procedures which manage these overload conditions are known as congestion control procedures.

Specification of the CIR is a relatively course measure and it is necessary for the *initial, minimum,* and *maximum* data rates on each PVC to be identified. For example, some PVCs can be configured to provide fixed bandwidth while others can be configured to maximize the cost savings resulting from shared bandwidth.

In the sliding window algorithm admission control scheme, each interface (or PVC) is allocated a certain number of credits. As frames are transmitted the credit allocation is decremented while the receiver returns credits via acknowledgments once the frames have been successfully received. In a well-tuned system the transmitter receives credits back at a rate such that the window is never exhausted.

In the credit manager scheme these credits represent units of data. They are allocated on a time basis to any unused PVC up to a defined maximum. When a frame arrives for transmission, the cells so generated decrement the current credit but are transmitted at the access speed (one credit/cell).

A PVC at the access interface can accumulate credit at a regular rate, up to the limit. When the end station transmits the frames over the PVC, the credit allocation is decremented by the amount of data transmitted. The initial maximum credit limit regulates the initial maximum burst size that the network is willing to accept. When the credits for a PVC are exhausted no further frames will be transmitted until return credit arrives. Under constant load, PVC credits arrive at a rate which permits data to be transmitted at the CIR.

Applications with low utilization and short transaction bursts such as database record processing, terminal/host transactions, electronic mail, and perhaps short file transfers all operate well with this mechanism. Variations and enhancements to these two algorithms have been implemented by various frame relay vendors. An example of the credit manager algorithm in action is shown in Fig. 19.

VII. CONGESTION CONTROL

Congestion control is necessary since there is no mandatory link layer flow control mechanism between the end station and the network as is found in LAP-B and LAP-D. The frame relay standard defines implicit and explicit mechanisms to alleviate network

congestion. With implicit congestion control the intelligent end stations reduce their load into the network once they realize that congestion is occurring. With explicit congestion control, signals are sent from the network to the end stations to inform them of performance problems.

Explicit congestion notification can be achieved by setting the FECN and BECN bits in the control field of the frame. In a congestion situation, the frame relay network may be forced to discard low priority frames, i.e., those with the DE flag set.

Both the FECN and BECN indicators are set by the network when congestion occurs. Figure 20 shows how a network might behave as the traffic increases. Initially the throughput increases in step with the offered load. Beyond a certain point (x) congestion commences and some frames are discarded. Both the FECN and BECN indicators would be set in this mid-congestion region which enables the network to recover from congestion by moving the operating point back into the noncongested region. Forward notification is useful when the receiver can control the transmitter, usually by means of a higher layer windowing mechanism. Backward notification relies upon traffic flowing in the reverse direction and in this way the transmitting device is informed that congestion is occurring. For example, BECN is used when interworking with X.25 traffic. The rate at which X.25 frames are encapsulated is controlled by reducing the X.25 window. Beyond point (y) the network congestion is severe.

The access rate control mechanism has a good deal to do with the size and length of bursts possible above the CIR. Both *Open* and *Closed Loop* Congestion Control Mechanisms are in use and have been implemented by vendors and are described below. There are a variety of congestion algorithms implemented by different vendors as well as many different views on how to set network parameters. The performance comparison of these offerings is difficult in the extreme making the selection of equipment based upon this performance issue very tricky.

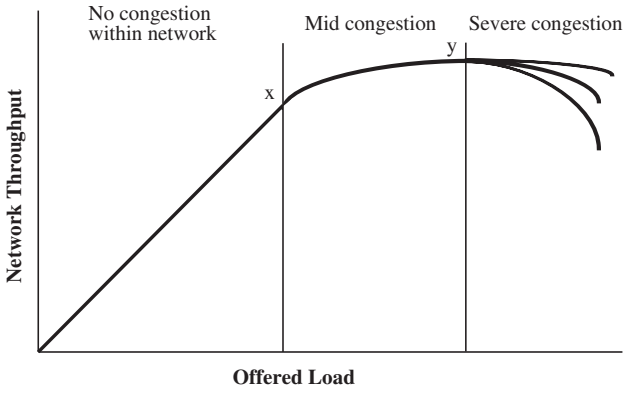


Figure 20 Network throughput versus offered load with various levels of congestion.

A. Open Loop Congestion Control

The simplest (but most inefficient) flow control mechanism is to limit data access to the network by providing explicit window flow control on each PVC with a transmission rate up to the CIR only. Another option is to provide storage capacity for burst data of up to about 5 KB. Although such mechanisms might be adequate for a lightly loaded network, better techniques are required in order to provide fine tuning and greater efficiency under conditions of heavy load. Other open loop congestion control mechanisms such as DE react to congestion by discarding data. This implies that there is no feedback mechanism to indicate congestion and/or no admission control mechanism to prevent congestion. Hence no action is taken until shared truck buffers are full, at which point data must be discarded. Full transmission queues create long delays for all virtual circuits. End-user application performance is reduced by these long queuing delays, and even more so by protocol timeouts caused by discarded frames.

Frames are admitted to the network at the access port speed without buffering. Once the delay experienced by users increases to a critical level, the FECN/BECN bits are set by the network in order to notify end devices that congestion is occurring. Many such devices ignore this notification and, if congestion continues to grow, the DE bit is used in order to decide which frames to discard. The setting of this DE bit is the most common method of handling open loop congestion control. It is the lack of foresight in this open loop system which makes it potentially inefficient as it cannot be assumed that the network will always carry spare capacity.

Although the FECN/BECN bits in the frame are used to signal congestion problems on the network, few access devices or LAN protocols act on this information. Even if they do, the time taken for this congestion information to reach the source end is often outside the time bound caused by the burst of data in the first place. Even this explicit form of congestion notification cannot always be relied upon as a primary congestion avoidance mechanism since reference is required to higher level protocols at end stations. Even assuming that voluntary end-user action is cor-

rectly implemented, in many instances the reaction time is too slow to avoid congestion. However, most protocols operate with a window mechanism and since PVC buffers can often accept up to 64 KB, this turns out to be ample storage in order to accept a full window of data in many situations.

B. Closed Loop Congestion Control

Unlike other congestion control schemes, closed loop control does not rely on the access devices to reduce their transmission rate. This is important because the frame relay flow control signals are presently not implemented in many access devices and in most LAN protocols. Even if implemented, end-to-user device response to explicit congestion notification would typically be too slow to prevent short-term congestion.

This lack of foresight on PVC buffering has the potential to cause congestion where many PVCs share the same access trunk. Once the shared buffer is full, frames will be discarded and all PVCs on the trunk will be affected, even if just one user primarily caused the congestion.

Thus a more satisfactory approach is to adjust the rate at which data is allowed to enter the network at an *individual* PVC's buffer as a result of information derived from the utilization of the interconnecting network trunks. Data received from the user's node at a rate in excess of the allocated rate enters the appropriate PVCs buffer at the network boundary for a limited period of time. Separate buffers for each virtual circuit at the boundary of the network ensure fairness and improve the efficiency of the higher level LAN protocols. The trunk loading parameter information is piggy-backed with normal frames where data is flowing. In the absence of such data, special control frames are transferred thus minimizing any additional load on the network at these critical times.

In closed loop control schemes (such as that implement by CISCO by way of their ForeSight *congestion avoidance algorithm*), the rate at which data is allowed to enter the network on each virtual circuit is adjusted in response to the utilization of the network trunks along that data path at that moment.

By looking across the network to determine the level of bandwidth contention on each data path, the busiest network segment traversed by a virtual circuit will determine the amount of bandwidth allocated to that virtual circuit at the entrance to the network. A virtual circuit which traverses a downstream trunk that is momentarily fully utilized will have its bandwidth allocation at the access node *decreased,* thus avoiding delays and possible data loss at the downstream trunk.

At the same time, another virtual circuit originating at the same node and routed over the same trunk at that node but over less busy downstream trunks may have its bandwidth allocation *increased.* This intelligent bandwidth allocation maximizes the performance of all virtual circuits simultaneously and is an important advantage over open loop congestion control mechanisms (see Fig. 21).

The *initial, minimum,* and *maximum* data rate schemes, described in Section VI, can assist in providing effective closed loop congestion control. For example the *initial* data rate assigned to a PVC is the rate at which data is accepted at the commencement of a burst. The sum of these will normally exceed the access rate since not all PVCs are likely to be active simultaneously. As the utilization of the interconnecting trunks is monitored and found to be low, the initial rate can be incremented until the burst is complete or until any of the interconnecting trunks become fully utilized or until this rate reaches some *maximum* data rate.

Correspondingly, if an interconnecting trunk becomes fully utilized then the rate at which data is accepted onto the PVC is reduced (possibly below the *initial* data rate). In no case is the rate at which data is accepted reduced below the *minimum* rate (which is normally equivalent to the CIR). Different combinations of initial, minimum, and maximum rates can be set for closed loop congestion control in order to suit different end-user application requirements. Thus bandwidth is continuously allocated according to trunk utilization and the capacity of the PVCs. This minimizes the chance that frames will ever have to be marked as discard eligible.

An ideal solution is one which combines an open loop credit allocation system (to control the rate at which frames are accepted at the network boundary) together with a closed loop foresight control system (which monitors the utilization of interconnecting trunks). This data can then be used to control initial, minimum, and maximum bandwidth on each PVC and is an ideal solution to minimize network congestion and to provide adaptive bandwidth allocation.

Further, if properly used, explicit congestion notification by way of the FECN/BECN notification bits will assist in preventing data loss due to an excessive data arrival rate. If data can be buffered at the network boundary the efficiency of upper layer flow control is maximized. Thus data is delayed rather than incurring the problems associated with frame discarding and application time-out delays for the retransmission which inevitably result.

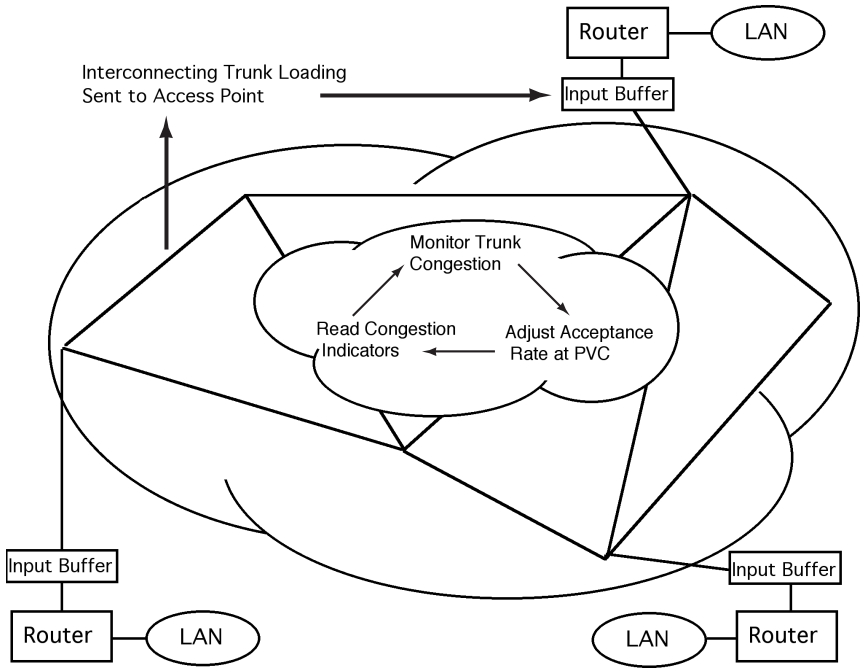


Figure 21 Closed loop congestion control.

VIII. FRAME RELAY PERFORMANCE, OPERATION AND CONFIGURATION

A. Frame Relay Performance and Operational Issues

Early experience with frame relay networks has indicated that properly designed networks do perform well but that there are a number of pitfalls. Careful analysis must be given to the choice of access data rate, the higher level protocols that are to be transported across the network, and the characteristics of the traffic pattern (average and peak transfer rates) as well as the performance characteristics of different vendors' equipment.

Some users indicated that the performance of frame relay over DDS offers very little but that the real advantages lie in improved connectivity, i.e., a "single hole in the wall" with the PVCs providing a replacement for multiple leased DDS circuits thus offering more connectivity per node at a lower cost. Frame relay has the potential to offer savings with this single physical connection philosophy even though customers have to specify the CIR between every pair of nodes on the network. In almost every case the cost of a frame relay configuration will be lower than a partially meshed point-to-point DDS network and always lower than the cost of a fully meshed DDS network.

Although congestion can radically reduce network performance, the answer is to design the network to avoid congestion. However, this is often easier said than done as it is difficult to stop a subscriber from flooding the network with substantial volumes of data if their equipment configuration chooses to ignore congestion control advice from the network.

Although frame relay has congestion management control built in to the protocol by way of the FECN/BECN bits, it appears that a number of frame relay routers on the market explicitly ignore these bits and/or are not intelligent enough to respond to congestion notifications from around the network. The matter is further complicated by the fact that users with routers who do obey the requirements of these congestion bits may in fact be opening up bandwidth for users with routers which do not respond to these control measures. There are, however, two mitigating circumstances. Since most frame relay traffic is TCP/IP based, the upper protocol layers appear to be doing an appropriate job of congestion control at the source. Further, the customer assumes that if they use excessive bandwidth the service provider's ingress switch will drop what cannot be handled. In managed frame relay applications (where the service provider configures the ingress router) BECN notifications are used.

Most frame relay switch manufacturers work on the principle that if extra bandwidth is not available when a substantial burst of traffic is received by the switch, frames are forwarded only up to the CIR with the remainder being buffered. Once the buffer has filled to

a critical point the FECN/BECN bits are set to alert other nodes on the network that this particular PVC is experiencing congestion problems. If the other nodes do not respond by the time that the switch's buffer is full, frames are discarded requiring the user's end-to-end protocols to arrange for retransmission.

The net result is a slow down in the overall performance and the end-to-end protocols which normally implement some form of window-driven protocol do not continue to send frames indefinitely without some form of returning acknowledgment. Hence the dominating factor for specific PVC users becomes *increased delay* and not an excess of *lost data*.

The overall performance objective is to provide maximum throughput on all PVCs while avoiding congestion that can be caused by momentary traffic bursts. The open loop system simply accepts traffic bursts and discards data if sufficient bandwidth is not available while the closed loop system derives continuous feed-back on trunk utilization from across the network and then correspondingly adjusts the rate of data acceptance on each PVC as previously discussed.

B. Network Configuration for Optimum Throughput

In order to determine the access rate and CIR it is essential to analyze traffic volumes and determine to which nodes the traffic is flowing. Getting the CIR correct is a most important issue. If it is *too low* it will cause congestion and poor response times while one that is *too high* will cost the organization financially. There are no magic rules for determining the correct CIR for a particular application. One would normally start with a CIR equivalent to that used for a private point-to-point DDS circuit. For example, if a connection utilizes 25% of a 128-kbps DDS circuit then the CIR could be set to about 32 kbps, although the utilization of a frame relay network is often higher than for a private DDS network. Further, the time period over which this utilization is measured is also a critical factor. Another way to assist in determining the appropriate CIR is to observe the number of DE bits set in a traffic sample. If many frames have this bit set it indicates congestion and the CIR should be set higher while few frames with the DE bit set indicate that the CIR may be too high thus costing the organization unnecessarily.

Although bursts can exceed the allocated traffic capacity, in the long run the network cannot carry significantly more traffic than the negotiated CIR as frame relay networks are vulnerable to overloading, but also the cost of providing excess CIR is high. Another problem occurs when there are too many PVCs per node with low traffic levels per PVC as this results in poor throughput. As the number of PVCs increases, the speed of the access port must be increased accordingly which in turn increases the overall network cost.

Once the CIR is reached, enforcement is up to the network provider who must determine whether additional frames are accepted or rejected and over what time period this might be done. Oversubscription (excess capacity in the network) as discussed earlier is a way of offering some measure of protection for customers who do not have a precise idea about their traffic flows. The benefit of oversubscription is that it reduces access port costs for the user. On the other hand, if the customer under-estimates the CIR, congestion delays are likely to be encountered.

In configuring connections to a frame relay network there are some key issues to keep in mind:

- Determine network traffic requirements, to which locations this traffic flows, and what peak volumes are likely to be encountered (e.g., file sizes).
- Choose access data rates appropriate for the applications. It may be necessary to progressively increment the access rate, e.g. 64 kbps → 128 kbps → 256 kbps → 1 Mbps → 2 Mbps, in single steps.
- Choose a CIR appropriate for the applications. For example, a 9.6-kbps CIR in conjunction with a 128-kbps access rate is unrealistic. In general:

$$\sum_{i=1}^{i=n} \mathrm{CIR}_i <= 0.5\ \textit{Access Rate}$$

  where: CIR_i = committed information rate for the *i*th channel and n = number of PVCs at an individual port operated at the designated *Access Rate*

- It may be necessary to reconfigure PVCs once traffic statistics are available from the traffic management software.
- Ensure that the carrier's network has been designed to address problems such as congestion in a fair manner.
- Full meshing of the network does not necessarily make for an appropriate configuration. It is better to commence with lightly meshed network topologies and work toward more complex meshing only as the delay requirements dictate. However, it should be noted that some routers do not allow transit routing.
- The granularity of the CIR is quite critical if the nodes have low average traffic flows between locations.

- Not all vendors define CIR in the same way. The definition of CIR is determined in part by the standards, in part by the capabilities of the switch, and in part by how much the users know about their traffic profiles.
- Buffering must be allowed for in both the frame relay switch and the DTE (usually a router). The lower the CIR is chosen to be, the more buffering will be required.
- Other factors to be considered include priority queuing in the routers, higher layer applications, and supporting protocols to be operated (for example, this may require adjustments to time-out parameters), what type of congestion control is implemented by the switch manufacturer, and what statistical and management reporting information is available from the network operator.
- Further, equipment conforming to the Frame Relay Forum standard (or a variety of vendor specific implementations of similar functionality) can monitor actual traffic use and frame discard rate (frame delivery ratio) and can (in some cases) also run tests to measure peak performance parameters throughout the network.

IX. CONCLUSIONS

Frame relay has a place in the overall arena of telecommunication interconnection services. It overlaps in part with services such as DDS, packet switching, and some ISDN services as well as offering a good transport infrastructure for many IP networking services. It is also a very important option for LAN-to-LAN interconnection as well as a variety of other application services.

The frame relay interface is designed to be simple, fast and efficient and is optimized for reliable, digital communications circuits. With today's low error rates, it is unnecessary and inefficient to manage acknowledgments and retransmissions at each segment of the network. If a frame is corrupted or lost, it is not retransmitted within the network as acknowledgments and retransmissions are all handled by the end systems.

Frame relay has the potential to save network costs when designed appropriately and where applications in turn are tuned to run on these networks. The fact that there are scenarios where frame relay does not fit well is not unique as similar comments can be made about most networking systems. Frame relay fills a gap in the spectrum and must not be automatically considered to be the solution for LAN or application interconnection in every case.

Higher speed options are increasingly in demand as a result of the enormous growth in the use of the Internet. Frame relay at 34 Mbps (E3) and 45 Mbps (T3) have been demonstrated, although it is not clear yet where the frame relay/ATM boundary will be positioned or whether new switching technologies such as multiple protocol label switching (MPLS) will become predominant. What is more likely is that all these technologies will coexist for many years to come.

SEE ALSO THE FOLLOWING ARTICLES

Integrated Services Digital Network • Internet, Overview • Local Area Networks • Standards and Protocols in Data Communications • Telecommunications Industry • Transmission Control Protocol/Internet Protocol • Wide Area Networks

BIBLIOGRAPHY

Buckwalter, J. T. (1999). *Frame relay: Technology and practice.* Reading, MA: Addison-Wesley.

Foy, S. (2001). *Cisco voice over frame relay, ATM and IP,* (S. McQuerry, and Kelly McGrew, eds.). Indianapolis, IN: Cisco Press.

Goralski, W. J. (1999). *Frame relay for high speed networks.* New York: John Wiley & Sons.

Future of Information Systems

Marlei Pozzebon
École des Hautes Études Commerciales de Montréal and McGill University

GLOSSARY

artificial intelligence (AI) A science and a technology that attempts to mimic human sentience or to emulate certain aspects of human behavior, such as reasoning, inference, learning, and problem solving.

automated teller machine (ATM) A special-purpose transaction terminal used to provide remote banking services.

client/server Platform or network that contains one or more host computers (servers) that provide services to the other interconnected computers (clients) where the responsibilities between the server and the clients are separate: often the client handles data entry whereas the server maintains the database.

collaborative automatic virtual environment (CAVE) Special 3-D, virtual reality room that can display images of people located in other CAVEs all over the world.

data mining Exploring huge repositories of data to uncover patterns, trends, or anomalies.

distributed processing A form of decentralization of information processing made possible by a network of computers dispersed throughout an organization.

extranet A portion of an intranet that is made available to outsiders, often suppliers, consumers, or subscribers.

fuzzy logic systems Computer-based systems that can process incomplete or partly correct data and can solve unstructured problems with incomplete knowledge.

intelligent agent A special-purpose knowledge-based system that serves as a software surrogate to accomplish specific tasks for end users.

internetworked Attribute of a computing environment where Internet, intranet, extranet, and other computer networks are used to support business processes, managerial decision making, and workgroup collaboration among organizational members, customers, suppliers, and other partners.

intranet An internal company network that uses the infrastructure of the Internet and the Web, telecommunications protocols, and browsers.

knowledge worker People whose primary work activities include creating, using, and distributing information. A typical worker in the information age whose performance depends on how well he/she utilizes information and knowledge to create value for the organization.

large-scale integration (LSI) A method of constructing electronic circuits in which thousands of circuits can be placed on a single semiconductor chip.

network computer (NC) A computer that has no hard disk or house applications, but rather has just a browser, memory, keyboard, and a modem to download applications from the Internet.

networking The sharing of computing resources by various computers linked together via a network.

object-oriented language (OO language) Programming language that encapsulates data and operations within objects that are manipulated by programs, increasing the reusability of the code.

open systems Information systems that are not tied to a particular hardware manufacturer or computer system. Different from proprietary systems, they use common standards for hardware, software, applications, and networking, creating a computing environment that allows easy access by end users and their networked computer systems.

optical scanning The generation of digital representations by a device that optically scans characters or images.

personal digital assistant (PDA) Handheld microcomputer device, such as a Palm Pilot, designed for convenient mobile communications and computing.

pointing device Devices that allow the end user to issue commands or make choices by moving a cursor on the display screen.

pull technology Mode of operation on the Internet where users must request data before the data is sent to them. A "web browser" represents a pull technology: the browser must request a web page before it is sent to the user's screen.

push technology Mode of operation on the Internet where data is sent to the users without it being requested. "Electronic mail" represents a push technology when it delivers customized information to the user's screen.

random access memory (RAM) Basic type of semiconductor memory used for temporary storage of data or programs during processing.

smart products Intelligent industrial and consumer products in which embedded microcomputers or microprocessors improve performance and capabilities.

touch device A device that accepts data input by the placement of a finger on or close to the screen.

voice recognition Directed conversion of spoken data into electronic form suitable for entry into a computer system.

telecommuting The use of communications technologies to enable working in a place other than a central location.

wireless technologies Technological devices connected without a physical cable. Considered a transmission medium. Examples are cordless telephone, cellular telephone, wireless LAN, and microwave.

I. INTRODUCTION

Writing about the future of information systems (IS) means writing about rapid change. One might focus on information technology (IT) transforming organizations and the business scenario or on people transforming meanings and uses of information technology. One might also embrace both approaches. Whatever the perspective adopted by the reader, the purpose of this article is to present recent developments and future trends in software, hardware, telecommunications, IS, and knowledge management, major dimensions that enable one to draw a picture of future research and applications in the IS field. Taking into account the pervasiveness of IT in numerous aspects of modern daily life, the present discussion is not only related to challenges faced by IS departments, managers, chief information officers, vendors, suppliers, consumers, and other key business players. It also encompasses the interaction of IT and people in multiple levels of social life—individuals, families, friends, professional associates, organizations, and communities.

Because contemporary evidence suggests that most organizational changes have an important IT-based component, this chapter highlights organization ventures regarding IS trends. These trends move toward an expanding role of IS in business management and an expanding participation of end users and managers in IS-related decisions. In addition, the continuing growth of internetworked IS has provided support for end user, enterprise, and interorganizational computing, communications, and collaboration, including global operations. Advances in telecommunications and computing are increasingly converging and interacting. In fact, one may say that businesses are becoming internetworked enterprises. Some ideas such as technological convergence and internetworking seem to pervade the future scenarios suggested by IS trends. Therefore, this chapter is structured according to five major patterns or trends recognized among the multiple emergent topics that characterize the IS scenario. They are hardware and software developments, the telecommunication revolution, the rise of Internet and web-based applications, digital convergence, and knowledge management (Fig. 1). Improvements on each of these five patterns are so intricately dependent on each other that most emergent IT applications are associated with all of them, and the future of IS has been defined and will be shaped by their intimate relationship and mutual influences.

II. HARDWARE AND SOFTWARE: RECENT DEVELOPMENTS AND FUTURE DIRECTIONS

A. Overall Trends in the Computing Environment

In order to describe today's computing environment and its future trends, it is important to understand

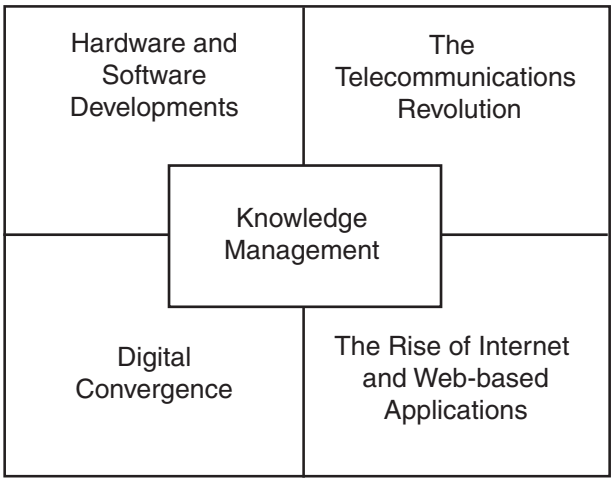


Figure 1 Five convergent trends shaping the future of IS

the evolution of IT over the last few decades (Table I). The 1960s were known as the mainframe era—hardware was centralized and batch processing was dominant. As smaller machines came to market, the possibility of distributing the processing to remote sites became feasible. In the mid-1970s, departmental minicomputers and word processors became available to end users. The 1980s saw the advent of personal computers (PCs) and the consolidation of on-line and distributed systems rather than batch and centralized processing. Networking emerged during the 1990s, when desktops, portable laptop computers, and other devices were now able to work together. The dominant platform became one of client/server, that is, networks of machines with the client on the desktop providing the user interface and the server on the network containing both the data and the applications. Often, the clients are network computers (NCs), devices with high processing and browser power but no hard disk or resident programs. Taking into account such developments and characteristics, hardware and software trends for the 2000s are toward technological convergence, wireless technologies, Internet-based computing, and enterprise-wide networking.

B. Hardware Trends

The major changes in hardware have occurred according to sequences of stages or generations (Table II). The first generation drew upon huge computers using vacuum tubes for processing and memory-circuitry operation. The second generation used transistors and other semi-conductor devices wired to circuit boards in the computer, and magnetic cores for the computer's memory and internal storage. Both first and second generations belong to the mainframe era. The third generation saw the emergence of smaller computers called minicomputers. This was made possible by integrated circuits and tiny chips of silicon with thousands of transistors and other elements of circuitry. The fourth generation, the generation of microcomputers, relies on large-scale integration (LSI) and very large-scale integration (VLSI) technologies, marked by the development of microprocessors (a single chip containing all of the circuits of a CPU) and the replacement of magnetic core memories with memory chips.

Throughout this evolution, the role of mainframes has changed and continues to evolve. They are declining in cost and undergoing remarkable changes in their architecture, becoming more modular and adapted to parallel operation. Taking into account the emergence of client/server architecture, the important role the mainframe can play is as a powerful

Table I The Computing Environment Trends

Period	Computing environment characteristics
1950s and 1960s	Mainframe era and the dominance of batch systems Centralized processing
1970s	Departmental minicomputers and word processors Decentralized processing
1980s	PCs and the consolidation of on-line systems Distributed processing
1990s	Networking and the emergence of the Internet Networked processing
2000s	Enterprise-wide networking and Internet-based computing Internetworked processing

Table II Hardware Trends

Period	Generation of computing	Hardware capabilities: Speed (instructions/second)	Hardware capabilities: Memory (capacity in characters)	Hardware capabilities: Cost (per million instructions)
1950s	First	10^2	10^3	$10
1960s	Second	10^3	10^4	$1.00
1970s	Third	10^6	10^5	$.10
1980s	Fourth	10^7	10^6	$.001
1990s and 2000s	Fifth	10^9	10^9	$.0001

server on networks. The overall trend is toward networked platforms linking powerful workstation computers (multimedia PCs and NCs), network servers (PCs, minicomputers, and mainframes), and portable devices (Personal Digital Assistants or PDAs, palmtops, and multimedia notebooks). In the 21st century, computer developments are moving toward the fifth generation, marked by larger capacities using smaller microelectronic circuits in terms of primary storage and massive capacities using magnetic and optical disks as secondary storage.

Overall, the trend is toward faster chips and lower cost. Speed and memory may become a million or billion times more powerful than in the first generations. In terms of LAN (local area network) and WAN (wide area network) technologies, there is an increasing use of ATMs (automated teller machines), improved backbones, and fiber channels. In addition, wirelesses LANs are new technological phenomena that offer significant flexibility to network configuration and mobility to users across a local scope of coverage. For use across wider geographic spans, a variety of wireless WAN services also have been expanded in several ways.

The major trends in terms of input devices are toward technologies that are more natural and easy to use: the combination of pointing and touching devices, voice and handwriting recognition, optical scanning, and magnetic stripes. In terms of output methods, the trends are toward technologies that communicate more naturally, quickly, and clearly such as voice response and high-resolution video displays.

C. Software Trends

Until the 1960s, IS functions were characterized by transaction processing, record keeping, and traditional accounting applications (Table III). During the 1960s another function emerged: management reports of prespecified information to support decision making. By the 1970s, the concept of decision support systems, providing interactive ad hoc support for the managerial decision-making process, became popularized. These first decades were characterized by in-house programming with eventual use of outside computing services and applications packages. Slowly, programming became centered on modular and structured techniques. With life cycle development methodologies and software engineering, the end users became more involved in the early stages of design and analysis.

The scenario of the 1980s increases in complexity with several new roles for IS. These included end user computing systems and direct computing support for end-user productivity and work, work group collaboration, executive IS providing critical information for top management, knowledge-based expert systems for end users, and strategic information systems providing information to create and maintain strategic products and services. In terms of software development, prototyping became the symbol of the 1980s.

During the 1990s, the rapid growth of the Internet, intranet, extranet, and other interconnected global networks changed the IS capabilities and their role became more strategic. The current trend is toward the structuring of enterprise and global internetworking, where IS is placed as an integral component of business processes, products, and services that help a company gain competitive advantage in the global marketplace.

These recent changes have been supported by two strong software trends: the option to purchase software packages and the consolidation of applications other than transaction processing. Such trends have

Table III Software Trends

Period	Information systems trends	Emergent languages
1950s	Electronic data processing systems, centralized batch systems	Machine languages
1960s	Management information systems; departmental systems	Symbolic languages
1970s	Decision support systems; departmental systems	High-level languages (e.g., COBOL, BASIC, C, C++, FORTRAN)
1980s	End-user computing systems; executive information systems, expert systems, corporate systems	Fourth generation languages (e.g., Oracle 4GE, IBM's ADRS2, APL)
1990s and 2000s	Internetworked information systems, enterprise collaboration systems and Internet-oriented systems	OO languages, web-based languages and technologies (e.g., ADA++, Object Lisp, UML, HTML, XML, Java, JINI)

emerged in the context of progressive open systems as opposed to proprietary applications. Languages such as BASIC and COBOL dominated the proprietary phase. The open phase encompasses PL/SQL, Visual Basic, Power Builder, C++, and an assortment of object-oriented languages. The late 1990s saw the emergence of a new programming trend dominated by Internet-oriented technologies and languages. Their main feature is universality: web browsers have become universal clients, able to be displayed on any client platform (PC, Unix, NC). Likewise, Java language and Jini technologies provide multiplatform operating systems.

Overall, the present scene reveals two main players battling for control of the client/server platform. On the one hand, Microsoft, who defends DCOM architecture (distributed component object model architecture, which defines the components with which their clients interact), Active/X distributed infrastructure, and Visual Basic language. On the other hand, Netscape, Sun, Oracle, and IBM are centered on the Java language, Corba distribution infrastructure, and the IIOP interconnection protocol. In the 2000s, XML (eXtensible mark up language) is becoming a standard that allows web-based applications to exchange structured information in a consistent way. In addition, scripting languages (e.g., ASP and JSP) were created to make interactive web-based applications and are expected to become more popular in the future.

Among the operating systems, the trend is toward systems that are easier and more flexible to install into a network and where software problems can be easily sorted out. The dominance remains with Linux, Windows 2000 and XP, Mac OS 8, and MAC OS X, and alternative open systems. In terms of database, the dominant trend remains oriented to object databases and network databases servers. Such a scenario indicates development in certain directions, but predicting long-term software trends is a risky enterprise.

D. Multimedia Technologies

Multimedia applications are characterized by three features: the combination of content (information) and software (how to control the presentation of the information); the possibility of many kinds of media for presenting information (text, graphic, images, sound, video), and the possibility of presenting information through those forms simultaneously. The result is the integration of two or more types of media, such as text, graphics, sound, voice, video, or animation into a computer-based application, providing the simultaneous presentation of information through that range of forms, where interactivity is a key component.

Currently, the business world is using multimedia in different ways: to support internal processes, to inform customers about products and services, and to improve those products and services. The development of multimedia applications requires ample RAM and disk storage and additional capturing (input devices such as audiocassette players and scanners) and conveying (devices such as speakers and high-resolution monitors) technologies.

Multimedia is a very effective method of communicating information because it enriches presentations, retains the audience's attention, and allows multiple and flexible interaction. The trends predicted for the immediate future are toward the full exploitation of multimedia to the extent that networks will increasingly support the transmission of such information. In addition, the recent emergence of

optical disk technologies called DVD (digital video disk) offers a large capacity for multimedia data storage and high-quality images and sounds. Examples of recent applications of multimedia are electronic books and newspapers, electronic classroom presentation technologies, full-motion videoconferencing, sophisticated imaging, and graphics design tools.

E. 3-D Technologies

Most personal productivity software packages available in 2000 are capable of producing "pseudo" 3-D views of images. They are called pseudo because they are not truly 3-D, but simulated by shadows and colors that are added to create a display that is somewhat realistic. Real 3-D technologies allow the visualization of different angles of the depth of the image, enabling a fuller experience. Commercial solutions developed from 3-D technologies involve high-quality interactive games, intuitive on-line shopping, and more engaging entertainment. The maturity of real 3-D technology depends on the capacity and speed of the development of other technologies such as hard disks, internal memory, CPUs, and monitors. To the extent that such technologies become available and less expensive, the trend suggests that 3-D will become a standard technology in the 2000s.

F. Virtual Reality

Virtual reality is the application of 3-D computer simulation in which the user actively and physically participates. Its key features are the incorporation of 3-D technologies, the simulation of a real-life situation, and the use of special devices that capture physical movements and send physical responses. Among the most recent commercial applications are pilot training, medical learning, and product demonstrations. Researchers working on virtual reality applications argue that some applications can dramatically alter both business and personal life. For instance, the possibility of simulating a great range of real-life situations can lead top managers to view and experience the effects of new organizational structures before proposing them and, similarly, consumers can virtually experience products or services before purchasing them. In addition, the use of collaborative automatic virtual environments (CAVEs), special rooms that allow the image of people to be projected in another location, create the illusion that all are participating in the same activity at the same place.

III. THE TELECOMMUNICATIONS REVOLUTION

Telecommunications are at the core of IT development and are opening new uses for information systems. For this reason, IS trends cannot be dissociated from telecommunication trends. Nonetheless, trends in the telecommunications field are predicted or outlined with great difficulty due to constant and rapid technological changes. Continually, new technologies that apply communications capabilities become available in areas and situations not previously feasible, justifiable, or even possible. Consequences of technological changes are often unanticipated. Social issues (legal, ethical, cultural, and political) emerge with such changes and require more and more consideration and debate. Examples of promising applications of communication technology are telecommuting, medical telediagnosis, automobile navigation by satellite, tracking cargo containers, electronic journalism, and high-speed communications to smart homes.

Advances in digital transmission have allowed the conversion of the public telephone network from analog to digital mode using fiber-optic cables. Satellite circuits have increasingly been used for batch data transmission and television broadcasting. The impact of the use of satellites has been driven by major trends such as higher bandwidth, more powerful satellite transmitters, and lower orbiting satellites. Three important markets, i.e., mobile phones, direct satellite television, and high-speed Internet access are expected to have their communication needs supplied by new satellite systems with lower costs and increased capabilities. Another exemplary illustration of the increasing application of satellites are Global Positioning Systems (GPS). A collection of 24 earth-orbiting satellites continuously transmits radio signals to GPS receivers (small handheld devices). These devices receive radio signals from four satellites at once and determine the user's position, the distance to certain destinations, and the best ways to get there, with the help of road maps. Such technology is now available and the trend is sophisticated in terms of the variety of services, precision, and lower costs.

One of strongest foundations for the telecommunications revolution is microprocessor development. Technological development in the area of telecommunications is closely related to the increasing capabilities of electronic circuit chips. While capacity continues to increase, the price-performance ratio is decreasing at an average rate of 20% per year. Since microprocessors and memory chips are at the core of communication equipment, evolutionary trends in this sector are noteworthy. An illustrative example is

the digital signal processor (DSP), a special type of microprocessor used to manipulate sounds, images, and other signals with precision and high speed. Voice-signal compression and video-signal compression are two examples of applications of DSP that are of particular interest for the communication field. Compression is defined as the technique by which the number of bits required to carry a digitized signal is reduced without losses in characteristics and quality. Voice-signal compression removes redundant bits of information, maintaining the tone and inflection necessary for its recognition. Video-signal compression allows videoconferencing services with good quality, low price, and fast transmission.

Another important improvement includes automatic speech recognition (ASR) techniques. They capture a voice, by a microphone attached to a device, and convert it to text, allowing speech recognition systems. Although until recently the speed of DSPs was not adequate to support the rapid flow of human speech, recent developments are increasing the capability of recognizing continuous speech with different regional and/or individual accents and patterns. ASR devices not only capture spoken words but also distinguish word groupings to form sentences. They are systems that contain a number of IT components such as an input device (microphones), software (that distinguishes words), and a database (that contains a repository of words). Among the future applications promised by the growth of the speech recognition marketplace are the replacement of typing by speaking as the standard technology for home computers; voice-controlled cars, televisions, refrigerators, ovens and other domestic devices; commercial speech-activated attendant systems which include call answer, call center routing, voice-activated dialing and so on.

A. Wireless Communications

The future of wireless communications is promising in several directions. Improvements in display technologies for cellular phones and pagers combined with intelligent and programmable networks will be explored for many types of wireless technologies. The sophistication of wireless technologies has allowed the emergence of smart phones, portable technologies that combine features of digital pagers, cellular phones, laptop computers, and portable printers. Smart phones allow their users to receive and send calls, e-messages, digital pages, and faxes and also to use Internet-access capabilities. The trend is that they will become more powerful and compact. Other examples of the future mobile phone are (1) the visual mobile phone, a mobile unit that transmits color images through a camera mounted in the handset; and (2) intelligent and programmable networks that allow every user of a cellular phone or wireless device to have their user profile stored with relevant information about them (for instance, when any type of message is conveyed to a user, a computer would verify the user profile and translate the message in an appropriate form).

B. Wireless Networks

The emerging wireless technologies have also revolutionized local networks. Wireless LANs are local networks that cover a limited distance without physical cables connecting computers, peripheral devices, and other components. A central point physically connected to a LAN server is established and all wireless communications go through it. Taking into account that setting cables for networks are actually the major expense, the wireless LAN allows increased mobility for future information technology architectures.

Two popular wireless transmission technologies in the LAN area are infrared transmission and spread spectrum transmission. For use across wider geographic spans, a variety of wireless WAN services also have been expanded, such as cellular digital packet data (CDPD), enhanced paging, and two-way messaging, microcellular spread spectrum, etc.

IV. DIGITAL CONVERGENCE

Recent developments in data communications have played such a major role in enabling technological changes that they are becoming more important than computer processing itself. Three major trends are driving the future of data communications and networking: the integration of voice, video, and data (i.e., digital convergence), the pervasiveness of networking, and the emergence of new information services.

The term digital convergence suggests that all data, voice and video, will be digital and integrated in a single device in an interactive way. In fact, the impetus behind this kind of technological development has been the appeal of improving social entertainment applications. Strings of binary digits will be transported by one or more channels and will converge in a selected device (computer or television). Such a device will include features such as phone calls (voice and video), Internet-use, movies on demand, and

information-gathering about mass-market services. Focusing on telephone, computer, television, and cable TV service providers, all providers support the electronic movement of information but use a variety of channels. The idea is to combine some or all of these services in a single device. Such a device could be an interactive television, which would include features of a computer and have Internet capabilities, or could be a multimedia PC with television features. It is clear, however, that television and computers are on convergent paths. It will be some years before the digital convergence scenario is clearer, but its players are already outgrowing their current roles: hardware and software manufacturers, content providers, broadcast and cable television, and long distance and local common carriers.

High-speed connections to the home are one factor that will activate an entirely new set of applications based on digital convergence. Ultimately, broadband communications to the home are derived from the increasing speed of computers, which are on a development path that increases performance by a factor of 10 every 5 years. However, fast computers are not enough. To attain the desired high-speed operation, several ways of connecting homes for high-speed data communications have been explored and will be advanced in the next few years: hybrid fiber-coax, digital subscriber lines (DSL), fiber-optic cables, and wireless technologies based on satellite networks and local multipoint distribution services (LMDS).

The combination of digital convergence with empowered microprocessor chips has accelerated the development of intelligent appliances. The scope is huge, because almost everything that uses electricity uses microprocessors. Intelligent appliances are devices that contain internal systems that control numerous functions and are capable of making decisions. Some household examples are smart vacuum cleaners that automatically adjust settings based on varying densities and weights of dirt, gas ranges that detect when water is about to boil adjusting temperature, washing machines that automatically balance loads to avoid stoppage and so on. These intelligent machines are becoming commonplace and are being equipped with more sophisticated features such as ASR. Behind these new types of appliances is an emerging technology called fuzzy logic, a method of working with information that is incomplete or imprecise. Fuzzy logic is a subfield of artificial intelligence (AI) that allows intelligent home appliances to make decisions. In addition, engineers of household appliances are working on intelligent devices that can interact with the Web. These intelligent networked home appliances will engender many sorts of new services, for instance, web-connected refrigerators that track the next shopping list and send it to an Internet grocer.

The digital convergence is not only related to the entertainment industry and household appliances, but to an improved support to enterprises and their clients with the convergence of wireless telecommunications, Internet, and e-commerce. However, the directions of that evolution are as of yet unpredictable.

V. THE INTERNET AND HIGH-SPEED WEB-BASED APPLICATIONS

One of the major changes in IT is the explosive growth of the Internet and related technologies and applications. Most large organizations have become dependent on web-based technologies, both internally and externally, for their commercial survival. This trend seems to be still evolving. E-business and e-commerce, in particular, will become central to the way large companies do business, and this fact places increasing pressure on IS practitioners to deliver the necessary resources. Among the most important trends in terms of technologies and applications related to Internet, e-commerce, and specific Internet-based applications such as e-card and long distance calls are anticipated. It is expected that a new set of applications will pervade homes and organizations. It is difficult to predict the speed at which most people will have Internet access within the next decade, but tens or hundreds of times faster that the ones commonly used today seems likely. In addition to speed, the form of Internet usage may also change. For instance, unified messaging is a logical progression from e-messages where voice mail, cellular text messages, fax and e-mail will be accessed from only one mailbox, anywhere in the world. Although still in experimental phases, unified messages may be an available service before or by 2005.

A. E-Commerce

Electronic commerce is one area that has increased rapidly, enabled by major developments in IT in recent years. Defined as the entire set of processes that support commercial activities on the Internet, e-commerce is still in its infancy. Although e-commerce has had an important impact on the way organizations work, it is based on Web technologies, which are still immature, and on computer and communications infrastructure, which are still under construction. To continue evolving, e-commerce depends on regulatory and governmental policy developments, infra-

structure consolidation, expansion of the telecommunications sector, and the availability of computing innovations. Among the emergent technologies that promote e-commerce, convergent billing opens the possibility that multiple services and products could be charged to a single bill. On the Internet, convergent billing allows users to make purchases and be charged on their telephone bill.

The terminology for e-commerce is vast: B2B is "business-to-business relationship," B2C is "business-to-consumer relationship," B2E is "business-to-employee relationship," and all other possible combinations (for instance, C2C is "customer-to-customer relationship"). In addition, subareas of e-commerce emerge: m-commerce is "mobile commerce" (a subset of all e-commerce transactions that involves the use of mobile devices like cellular phones and Palm Pilots) and k-commerce is "knowledge commerce" (described in Section VI). All these terms are examples of the richness and unpredictability of the broad topic called e-commerce.

The future directions of e-commerce will depend on areas such as the transformation of the marketplace, the replacement of traditional intermediary functions, the development of new products and markets, and the creation of new and closer relationships between business and consumers. In addition, giving it a potential catalytic effect, the development of e-commerce may accelerate the development of many other sectors already under way, such as electronic banking, one-to-one marketing, and direct booking of a range of services. More and more companies will become virtual organizations, with a mobile workforce and radically different network and support requirements.

Although evaluating the economic impacts (cost and productivity) is key, it is equally important to understand the social implications of e-commerce. Regarding employees, e-commerce is likely to redefine workers' skills and functions and to accelerate phenomena such as de-skilling or re-skilling. Regarding customers, there are many issues surrounding Internet security such as information security and privacy that should be addressed by organizations. Because the foundation of commerce has been built on trust and security, the future of e-commerce depends strongly on the development of methods of effectively providing on-line security.

B. Electronic Cash and Smart Cards

Among the major trends impelled by the explosion of Internet technologies are the development and dissemination of Internet cash transactions. Electronic cash (also called e-cash or digital-cash) is an electronic representation of the available money assigned to someone (a person or a company). Technically, the electronic representations of cash are files consulted and transmitted among electronic players: potential clients, Internet merchants, and electronic banks. The prediction is that in a few years e-cash will be a reality, depending on the definition of standards for payment protocols and governmental regulations. Among the predicted enhanced functions of e-cash is its combination with smart cards and wireless technologies, where e-cash can be downloaded to a mobile terminal and from there onto the smart card.

Similar to a credit card, a smart card is a form of electronic cash that contains a microprocessor chip on which a sum of money can be recorded and updated. Shopping with smart cards requires appropriate reader devices that can read the amount stored on the card, deduct the amount of the purchase, and display the new balance. There are two basic kinds of smart cards: the "intelligent" smart card that contains a CPU that actually has the ability to store and secure information; and the "memory" card, which is primarily an information storage card that contains a stored value which the user can spend in an electronic commercial transaction. It is expected that smart-card technology applications will continue to grow quickly, but such growth depends on two key issues that must be addressed: the adoption of technological standards for use by consumers and merchants everywhere and the planning of a technology migration path, by industry and government, that will allow smart card technology to coexist with established technology investments, such as magnetic stripe technology. Among the trends predicted for the next few years, is the expected use of smart cards in various applications and the development of so-called hybrid cards, cards that contain not only an embedded microprocessor chip or memory module, but also bar coding. In the next few years users will be able to access different hardware systems, such as merchant card readers, ATM machines, and bar code applications, with a single hybrid card.

VI. KNOWLEDGE MANAGEMENT AND KNOWLEDGE WORKERS

Knowledge management involves gathering, organizing, sharing, analyzing, and disseminating knowledge, including learning processes and management information systems, to improve an organization's performance. This represents one of the major sources of opportunities and challenges over the next few years.

For IS theorists and practitioners, this could mean providing instant, flexible access to information databases and data warehouses. However, knowledge management involves more than mere technical skills. A kind of synergy between technological and behavioral capabilities should be considered in order to make knowledge management effective. Such a connection involves the "know-how" accumulated through experience combined with knowing information or knowing where information can be found.

The increasing preoccupation that organizations have with knowledge management indicates the attempt to recognize individual knowledge, to filter and separate the most relevant knowledge, and to organize that knowledge into databases such that other employees can access them. Examples of enabling technologies for knowledge management are intelligent agents, push and pull technologies, information search and retrieval engines, on-line analytical processing, data mining, document management, electronic publishing, and multimedia interoperability. Intranet technologies are also seen as current knowledge management enabling tools because they have been proven useful for an organization looking to catalog, grow, and refine its information repository. In addition to intranets, the improvement of other technologies such as extranets, groupware, videoconferencing, and Web casting, several technology vendors offer solutions that are expected to enable knowledge management. Such solutions are expected to provide the means for storing predefined templates in information databases, which may be used for providing predetermined solutions based on predefined parameters. Among the new trends is the convergence of knowledge management and electronic business or electronic commerce, called k-commerce—the merchandising of recycled knowledge and packaged via electronic networks. Packaging refers to the transformation of tacit knowledge into explicit knowledge, which can be applied to product development or directly traded. While knowledge has always been traded in traditional ways, the Internet changes opportunities for advancing profitability from knowledge resources.

VII. CONCLUDING REMARKS

This chapter has presented recent developments and future trends in software, hardware, telecommunications, IS, and knowledge management. Keeping in mind the rapid pace of such developments and the constant pervasiveness of information technology in aspects of individual and organizational life not yet expected or planned, the risk that the trends drawn here will quickly become obsolete is considerable. However, awareness of these developments and of the suggested trends can be seen as a tool for improving our capability to continue further information systems developments and use.

Perhaps the most important concern regarding the future of IS is to think about the future of its users. Behind each technological trend described in this chapter, there are always human decisions and choices. Therefore, people need to be aware about their decisions regarding the adoption and use of technology. Far from accepting a technological determinist perspective that portrays technology as undeniably shaping human futures or from adopting a dangerous voluntarist position that neglects the influence of technological development, the perspective adopted here is one that highlights the power of ongoing and daily interactions between individuals and technologies. The future of IS lies in critical human knowledgeability.

SEE ALSO THE FOLLOWING ARTICLES

Artificial Intelligence • Developing Nations • Digital Divide, The • Economic Impacts of Information Technology • Electronic Commerce • Extranets • Globalization • Hybrid Systems • Mobile and Wireless Networks • National and Regional Economic Impacts of Silicon Valley • People, Information Systems Impact on • Telecommuting • Virtual Organizations • Virtual Learning Systems

BIBLIOGRAPHY

Alavi, M., and Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly* **25**(1), 107–136.

Ark, W. S., and Selker, T. (1999). A look at human interaction with pervasive computers. *IBM Systems Journal* **38**(4), 504–507.

Coates, J. F. (2000). Manufacturing in the 21st century. *Int. J. Manufacturing Technology and Management* **1**(1), 42–59.

CTR Report (1999). Information technology: Converging strategies and trends for the 21st century. Computer Technology Research Corp., South Carolina.

Davenport, T., and Prusak, L. (2000). Working knowledge: How organizations manage what they know. Boston, MA: Harvard Business School Press.

Fitzgerald, J., and Dennis, A. (1999). Business data communications and networking, 6th Edition. New York: John Wiley & Sons.

Grover, V. and Davenport, T. H. (2001). General perspectives on knowledge management: Fostering a research agenda. *J. Management Information Systems* **18**(1), 5–21.

Haag, S., Cummings, M., Dawkins, J., and Bateman, D. (2001). Management information systems for the information age, 1st Edition. Toronto: McGraw-Hill Ryerson.

Kim, Y. and Kim, Y. (1999). Critical IS issues in the network era. *Information Resources Management Journal* **12**(4), 14–23.

Lucas, H. C. (2000). Information technology for management. 7th Edition. New York: McGraw-Hill.

Martin, E. W., Brown, C. V., DeHayes, D. W., Hoffer, J. A., and Perkins, W. C. (1999). Managing information technology: What managers need to know, 3rd Edition. New Jersey: Prentice Hall.

McNurlin, B. C., and Sprague Jr., R. H. (1998). Information systems management in practice, 4th Edition. New Jersey: Prentice Hall.

O'Brien, J. A. (2000). Introduction to information systems: Essentials for the internetworked enterprise, 9th Edition. New York: McGraw-Hill.

OECD (1999). The economic and social impact of electronic commerce—preliminary findings and research agenda, 1st Edition: Paris: Organisation for Economic Co-operation and Development.

Rowe, S. H. (1999). Telecommunications for managers, 4th Edition. New Jersey: Prentice Hall.

Shaw, M., Blanning, R., Strader, T., and Whinston, A., Eds. (2000). Handbook on Electronic Commerce, 1st Edition. Berlin: Springer-Verlag.

St. John, C. H., Cannon, A. R., and Pouder, R. W. (2001). Change drivers in the new millennium: Implications for manufacturing strategy research **19**(2), 143–160.

Straub, D. W., and Watson, R. T. (2001). Research commentary: Transformational issues in researching IS and net-enabled organizations. *Information Systems Research* **12**(4), 337–345.

Turban, E., McLean, E., and Wetherbe, J. (1999). Information technology for management. Making connections for strategic advantage, 2nd Edition. New York: John Wiley & Sons.

Xephon Report (1999). Information systems: Trends and directions. Xephon Report, 1999 Series, 149 pages. http://www.xephon.com/reports.html.

Xephon Report (2000). E-commerce directions. Xephon Report, 2000 Series, 120 pages. http://www.xephon.com/reports.html.

Game Theory

Theodore L. Turocy *Texas A&M University*
Bernhard von Stengel *London School of Economics*

GLOSSARY

backward induction A technique to solve a game of perfect information. It first considers the moves that are the last in the game, and determines the best move for the player in each case. Then, taking these as given future actions, it proceeds backward in time, again determining the best move for the respective player, until the beginning of the game is reached.

common knowledge A fact is common knowledge if all players know it, and know that they all know it, and so on. The structure of the game is often assumed to be common knowledge among the players.

dominating strategy A strategy dominates another strategy of a player if it always gives a better payoff to that player, regardless of what the other players are doing. It weakly dominates the other strategy if it is always at least as good.

extensive game An extensive game (or extensive form game) describes with a tree how a game is played. It depicts the order in which players make moves, and the information each player has at each decision point.

game A formal description of a strategic situation.

game theory The formal study of decision making where several players must make choices that potentially affect the interests of the other players.

mixed strategy An active randomization, with given probabilities, that determines the player's decision. As a special case, a mixed strategy can be the deterministic choice of one of the given pure strategies.

Nash equilibrium Also called strategic equilibrium, is a list of strategies, one for each player, which has the property that no player can unilaterally change his strategy and get a better payoff.

payoff A number, also called utility, that reflects the desirability of an outcome to a player, for whatever reason. When the outcome is random, payoffs are usually weighted with their probabilities. The expected payoff incorporates the player's attitude toward risk.

perfect information A game has perfect information when at any point in time only one player makes a move, and knows all the actions that have been made until then.

player An agent who makes decisions in a game.

rationality A player is said to be rational if he seeks to play in a manner which maximizes his own payoff. It is often assumed that the rationality of all players is common knowledge.

strategic form A game in strategic form, also called normal form, is a compact representation of a game in which players simultaneously choose their strategies. The resulting payoffs are presented in a table with a cell for each strategy combination.

strategy In a game in strategic form, a strategy is one of the given possible actions of a player. In an extensive game, a strategy is a complete plan of choices, one for each decision point of the player.

zero-sum game A game is said to be zero-sum if for any outcome, the sum of the payoffs to all players is zero. In a two-player zero-sum game, one player's gain is the other player's loss, so their interests are diametrically opposed.

I. WHAT IS GAME THEORY?

Game theory is the formal study of conflict and cooperation. Game theoretic concepts apply whenever the actions of several agents are interdependent. These agents may be individuals, groups, firms, or any combination of these. The concepts of game theory provide a language to formulate, structure, analyze, and understand strategic scenarios.

A. History and Impact of Game Theory

The earliest example of a formal game-theoretic analysis is the study of a duopoly by Antoine Cournot in 1838. The mathematician Emile Borel suggested a formal theory of games in 1921, which was furthered by the mathematician John von Neumann in 1928 in a "theory of parlor games." Game theory was established as a field in its own right after the 1944 publication of the monumental volume *Theory of Games and Economic Behavior* by von Neumann and the economist Oskar Morgenstern. This book provided much of the basic terminology and problem setup that is still in use today.

In 1951, John Nash demonstrated that finite games always have an equilibrium point, at which all players choose actions which are best for them given their opponents' choices. This central concept of noncooperative game theory has been a focal point of analysis since then. In the 1950s and 1960s, game theory was broadened theoretically and applied to problems of war and politics. Since the 1970s, it has driven a revolution in economic theory. Additionally, it has found applications in sociology and psychology, and established links with evolution and biology. Game theory received special attention in 1994 with the awarding of the Nobel prize in economics to Nash, John Harsanyi, and Reinhard Selten.

At the end of the 1990s, a high-profile application of game theory had been the design of auctions. Prominent game theorists have been involved in the design of auctions for allocating rights to the use of bands of the electromagnetic spectrum to the mobile telecommunications industry. Most of these auctions were designed with the goal of allocating these resources more efficiently than traditional governmental practices, and additionally raised billions of dollars in the United States and Europe.

B. Game Theory and Information Systems

The internal consistency and mathematical foundations of game theory make it a prime tool for modeling and designing automated decision-making processes in interactive environments. For example, one might like to have efficient bidding rules for an auction web site, or tamper-proof automated negotiations for purchasing communication bandwidth. Research in these applications of game theory is the topic of recent conference and journal papers (see, for example, Binmore and Vulkan, Applying game theory to automated negotiation, *Netnomics,* Vol. 1, 1999, pages 1–9), but is still in a nascent stage. The automation of strategic choices enhances the need for these choices to be made efficiently, and to be robust against abuse. Game theory addresses these requirements.

As a mathematical tool for the decision maker the strength of game theory is the methodology it provides for structuring and analyzing problems of strategic choice. The process of formally modeling a situation as a game requires the decision maker to enumerate explicitly the players and their strategic options, and to consider their preferences and reactions. The discipline involved in constructing such a model already has the potential for providing the decision maker with a clearer and broader view of the situation. This is a "prescriptive" application of game theory, with the goal of improved strategic decision making. With this perspective in mind, this article explains basic principles of game theory, as an introduction to an interested reader without a background in economics.

II. DEFINITIONS OF GAMES

The object of study in game theory is the *game,* which is a formal model of an interactive situation. It typically involves several *players;* a game with only one player is usually called a *decision problem.* The formal definition lays out the players, their preferences, their information, the strategic actions available to them, and how these influence the outcome.

Games can be described formally at various levels of detail. A *coalitional* (or *cooperative*) game is a high-level description, specifying only what payoffs each potential group, or coalition, can obtain by the cooperation of its members. What is not made explicit is the process by which the coalition forms. As an example, the players may be several parties in parliament. Each party has a different strength, based upon the number of seats occupied by party members. The game describes which coalitions of parties can form a majority, but does not delineate, for example, the negotiation process through which an agreement to vote en bloc is achieved.

Cooperative game theory investigates such coalitional games with respect to the relative amounts of power held by various players, or how a successful coalition

should divide its proceeds. This is most naturally applied to situations arising in political science or international relations, where concepts like power are most important. For example, Nash proposed a solution for the division of gains from agreement in a bargaining problem which depends solely on the relative strengths of the two parties' bargaining position. The amount of power a side has is determined by the usually inefficient outcome that results when negotiations break down. Nash's model fits within the cooperative framework in that it does not delineate a specific timeline of offers and counteroffers, but rather focuses solely on the outcome of the bargaining process.

In contrast, *noncooperative game theory* is concerned with the analysis of strategic choices. The paradigm of noncooperative game theory is that the details of the ordering and timing of players' choices are crucial to determining the outcome of a game. In contrast to Nash's cooperative model, a noncooperative model of bargaining would posit a specific process in which it is prespecified who gets to make an offer at a given time. The term "noncooperative" means this branch of game theory explicitly models the process of players making choices out of their own interest. Cooperation can, and often does, arise in noncooperative models of games, when players find it in their own best interests.

Branches of game theory also differ in their assumptions. A central assumption in many variants of game theory is that the players are *rational*. A rational player is one who always chooses an action which gives the outcome he most prefers, given what he expects his opponents to do. The goal of game-theoretic analysis in these branches, then, is to predict how the game will be played by rational players, or, relatedly, to give advice on how best to play the game against opponents who are rational. This rationality assumption can be relaxed, and the resulting models have been more recently applied to the analysis of observed behavior (see Kagel and Roth, eds., *Handbook of Experimental Economics,* Princeton University Press, 1997). This kind of game theory can be viewed as more "descriptive" than the prescriptive approach taken here.

This article focuses principally on noncooperative game theory with rational players. In addition to providing an important baseline case in economic theory, this case is designed so that it gives good advice to the decision maker, even when—or perhaps especially when—one's opponents also employ it.

A. Strategic and Extensive Form Games

The *strategic form* (also called *normal form*) is the basic type of game studied in noncooperative game theory. A game in strategic form lists each player's strategies, and the outcomes that result from each possible combination of choices. An outcome is represented by a separate *payoff* for each player, which is a number (also called *utility*) that measures how much the player likes the outcome.

The *extensive form,* also called a *game tree,* is more detailed than the strategic form of a game. It is a complete description of how the game is played over time. This includes the order in which players take actions, the information that players have at the time they must take those actions, and the times at which any uncertainty in the situation is resolved. A game in extensive form may be analyzed directly, or can be converted into an equivalent strategic form.

Examples in the following sections will illustrate in detail the interpretation and analysis of games in strategic and extensive form.

III. DOMINANCE

Since all players are assumed to be rational, they make choices which result in the outcome they prefer most, given what their opponents do. In the extreme case, a player may have two strategies A and B so that, given any combination of strategies of the other players, the outcome resulting from A is better than the outcome resulting from B. Then strategy A is said to *dominate* strategy B. A rational player will never choose to play a dominated strategy. In some games, examination of which strategies are dominated results in the conclusion that rational players could only ever choose one of their strategies. The following examples illustrate this idea.

A. Example: Prisoner's Dilemma

The Prisoner's Dilemma is a game in strategic form between two players. Each player has two strategies, called "cooperate" and "defect," which are labeled C and D for player I and c and d for player II, respectively. (For simpler identification, upper case letters are used for strategies of player I and lower case letters for player II.)

Figure 1 shows the resulting payoffs in this game. Player I chooses a row, either C or D, and simultaneously player II chooses one of the columns c or d. The strategy combination (C,c) has payoff 2 for each player, and the combination (D,d) gives each player payoff 1. The combination (C,d) results in payoff 0 for player I and 3 for player II, and when (D,c) is played, player I gets 3 and player II gets 0.

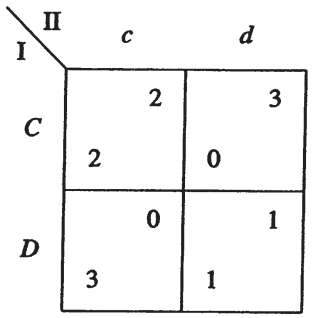


Figure 1 The Prisoner's Dilemma game.

Any two-player game in strategic form can be described by a table like the one in Fig. 1, with rows representing the strategies of player I and columns those of player II. (A player may have more than two strategies.) Each strategy combination defines a payoff pair, like (3, 0) for (*D*,*c*), which is given in the respective table entry. Each cell of the table shows the payoff to player I at the (lower) left, and the payoff to player II at the (right) top. These staggered payoffs, due to Thomas Schelling, also make transparent when, as here, the game is symmetric between the two players. Symmetry means that the game stays the same when the players are exchanged, corresponding to a reflection along the diagonal shown as a dotted line in Fig. 2. Note that in the strategic form, there is no order between player I and II since they act simultaneously (that is, without knowing the other's action), which makes the symmetry possible.

In the Prisoner's Dilemma game, "defect" is a strategy that dominates "cooperate." Strategy *D* of player I

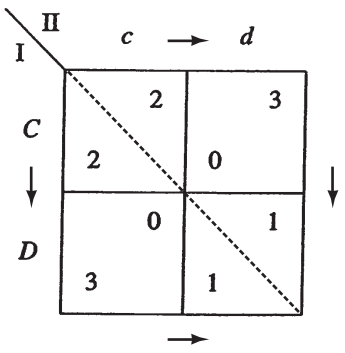


Figure 2 The game of Fig. 1 with annotations, implied by the payoff structure. The dotted line shows the symmetry of the game. The arrows at the left and right point to the preferred strategy of player I when player II plays the left or right column, respectively. Similarly, the arrows at the top and bottom point to the preferred strategy of player II when player I plays top or bottom.

dominates *C* since if player II chooses *c*, then player I's payoff is 3 when choosing *D* and 2 when choosing *C*; if player II chooses *d*, then player I receives 1 for *D* as opposed to 0 for *C*. These preferences of player I are indicated by the downward-pointing arrows in Fig. 2. Hence, *D* is indeed always better and dominates *C*. In the same way, strategy *d* dominates *c* for player II.

No rational player will choose a dominated strategy since the player will always be better off when changing to the strategy that dominates it. The unique outcome in this game, as recommended to utility-maximizing players, is therefore (*D*,*d*) with payoffs (1, 1). Somewhat paradoxically, this is less than the payoff (2, 2) that would be achieved when the players chose (*C*,*c*).

The story behind the name Prisoner's Dilemma is that of two prisoners held suspect of a serious crime. There is no judicial evidence for this crime except if one of the prisoners testifies against the other. If one of them testifies, he will be rewarded with immunity from prosecution (payoff 3), whereas the other will serve a long prison sentence (payoff 0). If both testify, their punishment will be less severe (payoff 1 for each). However, if they both "cooperate" with each other by not testifying at all, they will only be imprisoned briefly, for example, for illegal weapons possession (payoff 2 for each). The "defection" from that mutually beneficial outcome is to testify, which gives a higher payoff no matter what the other prisoner does, with a resulting lower payoff to both. This constitutes their "dilemma."

Prisoner's Dilemma games arise in various contexts where individual "defections" at the expense of others lead to overall less desirable outcomes. Examples include arms races, litigation instead of settlement, environmental pollution, or cut-price marketing, where the resulting outcome is detrimental for the players. Its game-theoretic justification on individual grounds is sometimes taken as a case for treaties and laws, which enforce cooperation.

Game theorists have tried to tackle the obvious "inefficiency" of the outcome of the Prisoner's Dilemma game. For example, the game is fundamentally changed by playing it more than once. In such a *repeated game,* patterns of cooperation can be established as rational behavior when players' fear of punishment in the future outweighs their gain from defecting today.

B. Example: Quality Choice

The next example of a game illustrates how the principle of elimination of dominated strategies may be applied iteratively. Suppose player I is an Internet ser-

vice provider and player II a potential customer. They consider entering into a contract of service provision for a period of time. The provider can, for himself, decide between two levels of quality of service, *High* or *Low*. High-quality service is more costly to provide, and some of the cost is independent of whether the contract is signed or not. The level of service cannot be put verifiably into the contract. High-quality service is more valuable than low-quality service to the customer, in fact so much so that the customer would prefer not to buy the service if she knew that the quality was low. Her choices are to *buy* or *not to buy* the service.

Fig. 3 gives possible payoffs that describe this situation. The customer prefers to buy if player I provides high-quality service, and not to buy otherwise. Regardless of whether the customer chooses to buy or not, the provider always prefers to provide the low-quality service. Therefore, the strategy *Low* dominates the strategy *High* for player I.

Now, since player II believes player I is rational, she realizes that player I always prefers *Low,* and so she anticipates low-quality service as the provider's choice. Then she prefers *not to buy* (giving her a payoff of 1) to *buy* (payoff 0). Therefore, the rationality of both players leads to the conclusion that the provider will implement low-quality service and, as a result, the contract will not be signed.

This game is very similar to the Prisoner's Dilemma in Fig. 1. In fact, it differs only by a single payoff, namely payoff 1 (rather than 3) to player II in the top right cell in the table. This reverses the top arrow from right to left, and makes the preference of player II dependent on the action of player I. (The game is also no longer symmetric.) Player II does not have a dominating strategy. However, player I still does, so that the resulting outcome, seen from the "flow of arrows" in Fig. 3, is still unique. Another way of obtaining this outcome is the successive elimination of dominated strategies: first, *High* is eliminated, and in the resulting smaller game where player I has only the single strategy *Low* available, player II's strategy *buy* is dominated and also removed.

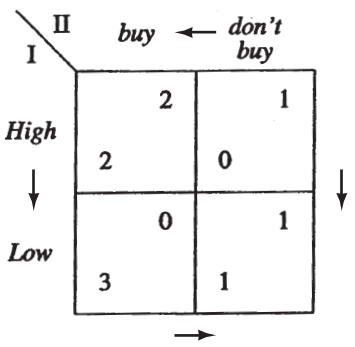


Figure 3 High-low quality game between a service provider (player I) and a customer (player II).

As in the Prisoner's Dilemma, the individually rational outcome is worse for both players than another outcome, namely the strategy combination *(High, buy)* where high-quality service is provided and the customer signs the contract. However, that outcome is not credible, since the provider would be tempted to renege and provide only the low-quality service.

IV. NASH EQUILIBRIUM

In the previous examples, consideration of dominating strategies alone yielded precise advice to the players on how to play the game. In many games, however, there are no dominated strategies, and so these considerations are not enough to rule out any outcomes or to provide more specific advice on how to play the game.

The central concept of *Nash equilibrium* is much more general. A Nash equilibrium recommends a strategy to each player that the player cannot improve upon *unilaterally,* that is, given that the other players follow the recommendation. Since the other players are also rational, it is reasonable for each player to expect his opponents to follow the recommendation as well.

A. Example: Quality Choice Revisited

A game-theoretic analysis can highlight aspects of an interactive situation that could be changed to get a better outcome. In the quality game in Fig. 3, for example, increasing the customer's utility of high-quality service has no effect unless the provider has an incentive to provide that service. So suppose that the game is changed by introducing an opt-out clause into the service contract. That is, the customer can discontinue subscribing to the service if she finds it of low quality.

The resulting game is shown in Fig. 4. Here, low-quality service provision, even when the customer decides to buy, has the same low payoff 1 to the provider as when the customer does not sign the contract in the first place, since the customer will opt out later. However, the customer still prefers not to buy when the service is *Low* in order to spare herself the hassle of entering the contract.

The changed payoff to player I means that the left arrow in Fig. 4 points upward. Note that, compared to

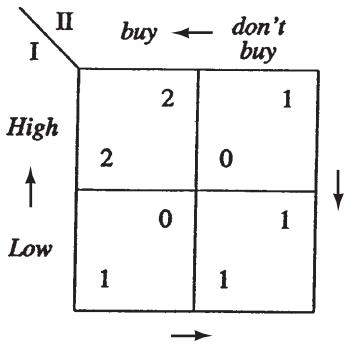


Figure 4 High-low quality game with opt-out clause for the customer. The left arrow shows that player I prefers *High* when player II chooses *buy*.

Fig. 3, only the provider's payoffs are changed. In a sense, the opt-out clause in the contract has the purpose of convincing the customer that the high-quality service provision is in the provider's own interest.

This game has no dominated strategy for either player. The arrows point in different directions. The game has *two* Nash equilibria in which each player chooses his strategy deterministically. One of them is, as before, the strategy combination *(Low, don't buy)*. This is an equilibrium since *Low* is the *best response* (payoff-maximizing strategy) to *don't buy* and vice versa.

The second Nash equilibrium is the strategy combination *(High, buy)*. It is an equilibrium since player I prefers to provide high-quality service when the customer buys, and conversely, player II prefers to buy when the quality is high. This equilibrium has a higher payoff to both players than the former one, and is a more desirable solution.

Both Nash equilibria are legitimate recommendations to the two players of how to play the game. Once the players have settled on strategies that form a Nash equilibrium, neither player has incentive to deviate, so that they will rationally stay with their strategies. This makes the Nash equilibrium a consistent solution concept for games. In contrast, a strategy combination that is *not* a Nash equilibrium is not a credible solution. Such a strategy combination would not be a reliable recommendation on how to play the game, since at least one player would rather ignore the advice and instead play another strategy to make himself better off.

As this example shows, a Nash equilibrium may be not unique. However, the previously discussed solutions to the Prisoner's Dilemma and to the quality choice game in Fig. 3 are unique Nash equilibria. A dominated strategy can never be part of an equilibrium since a player intending to play a dominated strategy could switch to the dominating strategy and be better off. Thus, if elimination of dominated strategies leads to a unique strategy combination, then this is a Nash equilibrium. Larger games may also have unique equilibria that do not result from dominance considerations.

B. Equilibrium Selection

If a game has more than one Nash equilibrium, a theory of strategic interaction should guide players toward the "most reasonable" equilibrium upon which they should focus. Indeed, a large number of papers in game theory have been concerned with "equilibrium refinements" that attempt to derive conditions that make one equilibrium more plausible or convincing than another. For example, it could be argued that an equilibrium that is better for both players, like *(High, buy)* in Fig. 4, should be the one that is played.

However, the abstract theoretical considerations for equilibrium selection are often more sophisticated than the simple game-theoretical models to which they are applied. It may be more illuminating to observe that a game has more than one equilibrium, and that this is a reason that players are sometimes stuck at an inferior outcome.

One and the same game may also have a different interpretation where a previously undesirable equilibrium becomes rather plausible. As an example, consider an alternative scenario for the game in Fig. 4. Unlike the previous situation, it will have a symmetric description of the players, in line with the symmetry of the payoff structure.

Two firms want to invest in communication infrastructure. They intend to communicate frequently with each other using that infrastructure, but they decide independently on what to buy. Each firm can decide between *High* or *Low* bandwidth equipment (this time, the same strategy names will be used for both players). For player II, *High* and *Low* replace *buy* and *don't buy* in Fig. 4. The rest of the game stays as it is.

The (unchanged) payoffs have the following interpretation for player I (which applies in the same way to player II by symmetry): a *Low* bandwidth connection works equally well (payoff 1) regardless of whether the other side has high or low bandwidth. However, switching from *Low* to *High* is preferable only if the other side has high bandwidth (payoff 2), otherwise it incurs unnecessary cost (payoff 0).

As in the quality game, the equilibrium *(Low, Low)* (the bottom right cell) is inferior to the other equi-

librium, although in this interpretation it does not look quite as bad. Moreover, the strategy *Low* has obviously the better *worst-case* payoff, as considered for all possible strategies of the other player, no matter if these strategies are rational choices or not. The strategy *Low* is therefore also called a *max-min* strategy since it maximizes the minimum payoff the player can get in each case. In a sense, investing only in low-bandwidth equipment is a safe choice. Moreover, this strategy is part of an equilibrium, and entirely justified if the player expects the other player to do the same.

C. Evolutionary Games

The bandwidth choice game can be given a different interpretation where it applies to a large *population* of identical players. Equilibrium can then be viewed as the outcome of a *dynamic process* rather than of conscious rational analysis.

Figure 5 shows the bandwidth choice game where each player has the two strategies *High* and *Low*. The positive payoff of 5 for each player for the strategy combination *(High, High)* makes this an even more preferable equilibrium than in the case discussed above.

In the evolutionary interpretation, there is a large population of individuals, each of which can adopt one of the strategies. The game describes the payoffs that result when two of these individuals meet. The dynamics of this game are based on assuming that each strategy is played by a certain *fraction* of individuals. Then, given this distribution of strategies, individuals with better *average payoff* will be more successful than others, so that their proportion in the population increases over time. This, in turn, may affect which strategies are better than others. In many cases, in particular in symmetric games with only two possible strategies, the dynamic process will move to an equilibrium.

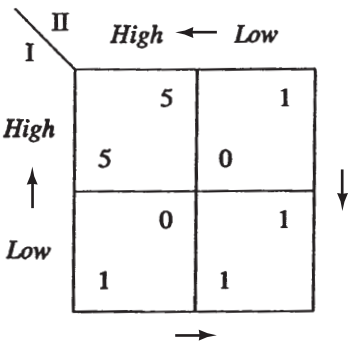


Figure 5 The bandwidth choice game.

In the example of Fig. 5, a certain fraction of users connected to a network will already have *High* or *Low* bandwidth equipment. For example, suppose that one quarter of the users has chosen *High* and three quarters have chosen *Low*. It is useful to assign these as percentages to the columns, which represent the strategies of player II. A new user, as player I, is then to decide between *High* and *Low,* where his payoff depends on the given fractions. Here it will be $1/4 \times 5 + 3/4 \times 0 = 1.25$ when player I chooses *High,* and $1/4 \times 1 + 3/4 \times 1 = 1$ when player I chooses *Low.* Given the average payoff that player I can expect when interacting with other users, player I will be better off by choosing *High,* and so decides on that strategy. Then, player I joins the population as a *High* user. The proportion of individuals of type *High* therefore increases, and over time the advantage of that strategy will become even more pronounced. In addition, users replacing their equipment will make the same calculation, and therefore also switch from *Low* to *High*. Eventually, everyone plays *High* as the only surviving strategy, which corresponds to the equilibrium in the top left cell in Fig. 5.

The long-term outcome where only high-bandwidth equipment is selected depends on there being an initial fraction of high-bandwidth users that is large enough. For example, if only 10% have chosen *High,* then the expected payoff for *High* is $0.1 \times 5 + 0.9 \times 0 = 0.5$ which is less than the expected payoff 1 for *Low* (which is always 1, irrespective of the distribution of users in the population). Then, by the same logic as before, the fraction of *Low* users increases, moving to the bottom right cell of the game as the equilibrium. It is easy to see that the critical fraction of *High* users so that this will take off as the better strategy is 1/5. (When new technology makes high-bandwidth equipment cheaper, this increases the payoff 0 to the *High* user who is meeting *Low,* which changes the game.)

The evolutionary, population-dynamic view of games is useful because it does not require the assumption that all players are sophisticated and think the others are also rational, which is often unrealistic. Instead, the notion of rationality is replaced with the much weaker concept of *reproductive success:* strategies that are successful on average will be used more frequently and thus prevail in the end. This view originated in theoretical biology with Maynard Smith (*Evolution and the Theory of Games,* Cambridge University Press, 1982) and has since significantly increased in scope (see Hofbauer and Sigmund, *Evolutionary Games and Population Dynamics,* Cambridge University Press, 1998).

V. MIXED STRATEGIES

A game in strategic form does not always have a Nash equilibrium in which each player deterministically chooses one of his strategies. However, players may instead randomly select from among these *pure* strategies with certain probabilities. Randomizing one's own choice in this way is called a *mixed* strategy. Nash showed in 1951 that any finite strategic-form game has an equilibrium if mixed strategies are allowed. As before, an equilibrium is defined by a (possibly mixed) strategy for each player where no player can gain *on average* by unilateral deviation. Average (that is, *expected*) payoffs must be considered because the outcome of the game may be random.

A. Example: Compliance Inspections

Suppose a consumer purchases a license for a software package, agreeing to certain restrictions on its use. The consumer has an incentive to violate these rules. The vendor would like to verify that the consumer is abiding by the agreement, but doing so requires inspections which are costly. If the vendor does inspect and catches the consumer cheating, the vendor can demand a large penalty payment for the noncompliance.

Figure 6 shows possible payoffs for such an inspection game. The standard outcome, defining the reference payoff zero to both vendor (player I) and consumer (player II), is that the vendor chooses *Don't inspect* and the consumer chooses to *comply*. Without inspection, the consumer prefers to *cheat* since that gives her payoff 10, with resulting negative payoff −10 to the vendor. The vendor may also decide to *Inspect*. If the consumer complies, inspection leaves her payoff 0 unchanged, while the vendor incurs a cost resulting in a negative payoff −1. If the consumer cheats, however, inspection will result in a heavy penalty (payoff −90 for player II) and still create a certain amount of hassle for player I (payoff −6).

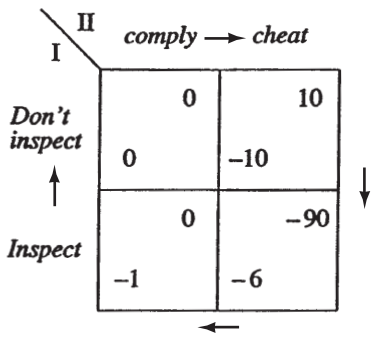


Figure 6 Inspection game between a software vendor (player I) and consumer (player II).

In all cases, player I would strongly prefer if player II complied, but this is outside of player I's control. However, the vendor prefers to inspect if the consumer cheats (since −6 is better than −10), indicated by the downward arrow on the right in Fig. 6. If the vendor always preferred *Don't inspect,* then this would be a dominating strategy and be part of a (unique) equilibrium where the consumer cheats.

The circular arrow structure in Fig. 6 shows that this game has no equilibrium in pure strategies. If any of the players settles on a deterministic choice (like *Don't inspect* by player I), the best reponse of the other player would be unique (here *cheat* by player II), to which the original choice would *not* be a best reponse (player I prefers *Inspect* when the other player chooses *cheat,* against which player II in turn prefers to *comply*). The strategies in a Nash equilibrium must be best responses to each other, so in this game this fails to hold for any pure strategy combination.

B. Mixed Equilibrium

What should the players do in the game of Fig. 6? One possibility is that they prepare for the worst, that is, choose a *max-min* strategy. As explained before, a max-min strategy maximizes the player's worst payoff against all possible choices of the opponent. The max-min strategy for player I is to *Inspect* (where the vendor guarantees himself payoff −6), and for player II it is to *comply* (which guarantees her payoff 0). However, this is not a Nash equilibrium and hence not a stable recommendation to the two players, since player I could switch his strategy and improve his payoff.

A *mixed strategy* of player I in this game is to *Inspect* only with a certain probability. In the context of inspections, randomizing is also a practical approach that reduces costs. Even if an inspection is not certain, a sufficiently high chance of being caught should deter from cheating, at least to some extent.

The following considerations show how to find the probability of inspection that will lead to an equilibrium. If the probability of inspection is very low, for example, 1%, then player II receives (irrespective of that probability) payoff 0 for *comply,* and payoff $0.99 \times 10 + 0.01 \times (-90) = 9$, which is bigger than zero, for *cheat.* Hence, player II will still cheat, just as in the absence of inspection.

If the probability of inspection is much higher, for example, 0.2, then the expected payoff for *cheat* is $0.8 \times 10 + 0.2 \times (-90) = -10$, which is less than

zero, so that player II prefers to *comply*. If the inspection probability is either too low or too high, then player II has a unique best response. As shown above, such a pure strategy cannot be part of an equilibrium.

Hence, the only case where player II herself could possibly randomize between her strategies is if both strategies give her the same payoff, that is, if she is *indifferent*. It is never optimal for a player to assign a positive probability to playing a strategy that is inferior, given what the other players are doing. It is not hard to see that player II is indifferent if and only if player I inspects with probability 0.1, since then the expected payoff for *cheat* is $0.9 \times 10 + 0.1 \times (-90) = 0$, which is then the same as the payoff for *comply*.

With this mixed strategy of player I (*Don't inspect* with probability 0.9 and *Inspect* with probability 0.1), player II is indifferent between her strategies. Hence, she can *mix* them (that is, play them randomly) without losing payoff. The only case where, in turn, the original mixed strategy of player I is a best response is if player I is indifferent. According to the payoffs in Fig. 6, this requires player II to choose *comply* with probability 0.8 and *cheat* with probability 0.2. The expected payoffs to player I are then for *Don't inspect* $0.8 \times 0 + 0.2 \times (-10) = -2$, and for *Inspect* $0.8 \times (-1) + 0.2 \times (-6) = -2$, so that player I is indeed indifferent, and his mixed strategy is a best response to the mixed strategy of player II.

This defines the only Nash equilibrium of the game. It uses mixed strategies and is therefore called a *mixed* equilibrium. The resulting expected payoffs are -2 for player I and 0 for player II.

C. Interpretation of Mixed Strategy Probabilities

The preceding analysis showed that the game in Fig. 6 has a mixed equilibrium, where the players choose their pure strategies according to certain probabilities. These probabilities have several noteworthy features.

The equilibrium probability of 0.1 for *Inspect* makes player II indifferent between *comply* and *cheat*. This is based on the assumption that an *expected payoff* of 0 for *cheat,* namely $0.9 \times 10 + 0.1 \times (-90)$, is the same for player II as when getting the payoff 0 for certain, by choosing to *comply*. If the payoffs were monetary amounts (each payoff unit standing for \$1000, say), one would not necessarily assume such a *risk neutrality* on the part of the consumer. In practice, decision makers are typically *risk averse,* meaning they prefer the safe payoff of 0 to the gamble with an expectation of 0.

In a game-theoretic model with random outcomes (as in a mixed equilibrium), however, the payoff is not necessarily to be interpreted as money. Rather, the player's attitude toward risk is incorporated into the payoff figure as well. To take our example, the consumer faces a certain reward or punishment when cheating, depending on whether she is caught or not. Getting caught may not only involve financial loss but embarrassment and other undesirable consequences. However, there is a certain probability of inspection (that is, of getting caught) where the consumer becomes indifferent between *comply* and *cheat*. If that probability is 1 against 9, then this indifference implies that the cost (negative payoff) for getting caught is 9 times as high as the reward for cheating successfully, as assumed by the payoffs in Fig. 6. If the probability of indifference is 1 against 20, the payoff -90 in Fig. 6 should be changed to -200. The units in which payoffs are measured are arbitrary. Like degrees on a temperature scale, they can be multiplied by a positive number and shifted by adding a constant, without altering the underlying preferences they represent.

In a sense, the payoffs in a game mimic a player's (consistent) willingness to bet when facing certain odds. With respect to the payoffs, which may distort the monetary amounts, players are then risk neutral. Such payoffs are also called *expected-utility* values. Expected-utility functions are also used in one-player games to model decisions under uncertainty.

The risk attitude of a player may not be known in practice. A game-theoretic analysis should be carried out for different choices of the payoff parameters in order to test how much they influence the results. Typically, these parameters represent the "political" features of a game-theoretic model, those most sensitive to subjective judgement, compared to the more "technical" part of a solution. In more involved inspection games, the technical part often concerns the optimal usage of limited inspection resources, whereas the political decision is when to raise an alarm and declare that the inspectee has cheated (see Avenhaus and Canty, *Compliance Quantified,* Cambridge University Press, 1996).

Secondly, mixing seems paradoxical when the player is indifferent in equilibrium. If player II, for example, can equally well *comply* or *cheat,* why should she gamble? In particular, she could *comply* and get payoff 0 for certain, which is simpler and safer. The answer is that precisely because there is no incentive to choose one strategy over the other, a player can mix, and only in that case there can be an equilibrium. If player II would *comply* for certain, then the only optimal choice of player I is *Don't inspect,* making the choice of complying not optimal, so this is not an equilibrium.

The least intuitive aspect of mixed equilibrium is that the probabilities depend on the *opponent's payoffs* and not on the player's own payoffs (as long as the qualitative preference structure, represented by the arrows, remains intact). For example, one would expect that raising the penalty −90 in Fig. 6 for being caught lowers the probability of cheating in equilibrium. In fact, it does not. What does change is the probability of inspection, which is reduced until the consumer is indifferent.

This dependence of mixed equilibrium probabilities on the opponent's payoffs can be explained as terms of population dynamics. In that interpretation, Fig. 6 represents an evolutionary game. Unlike Fig. 5, it is a nonsymmetric interaction between a vendor who chooses *Don't Inspect* and *Inspect* for certain fractions of a large number of interactions. Player II's actions *comply* and *cheat* are each chosen by a certain fraction of consumers involved in these interactions. If these fractions deviate from the equilibrium probabilities, then the strategies that do better will increase. For example, if player I chooses *Inspect* too often (relative to the penalty for a cheater who is caught), the fraction of cheaters will decrease, which in turn makes *Don't Inspect* a better strategy. In this dynamic process, the long-term averages of the fractions approximate the equilibrium probabilities.

VI. EXTENSIVE GAMES WITH PERFECT INFORMATION

Games in strategic form have no temporal component. In a game in strategic form, the players choose their strategies simultaneously, without knowing the choices of the other players. The more detailed model of a *game tree,* also called a game in *extensive form,* formalizes interactions where the players can over time be informed about the actions of others. This section treats games of *perfect information.* In an extensive game with perfect information, every player is at any point aware of the previous choices of all other players. Furthermore, only one player moves at a time, so that there are no simultaneous moves.

A. Example: Quality Choice with Commitment

Figure 7 shows another variant of the quality choice game. This is a game tree with perfect information. Every branching point, or *node,* is associated with a player who makes a move by choosing the next node.

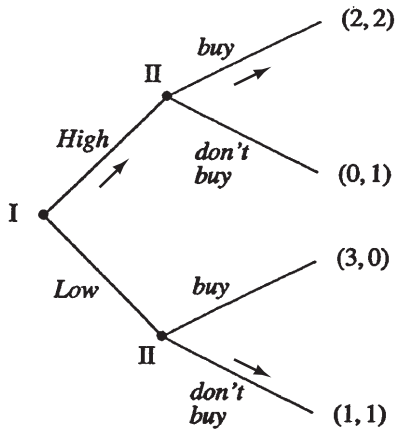


Figure 7 Quality choice game where player I *commits* to *High* or *Low* quality, and player II can react accordingly. The arrows indicate the optimal moves as determined by backward induction.

The connecting lines are labeled with the player's choices. The game starts at the initial node, the *root* of the tree, and ends at a terminal node, which establishes the outcome and determines the players' payoffs. In Fig. 7, the tree grows from left to right; game trees may also be drawn top-down or bottom-up.

The service provider, player I, makes the first move, choosing *High* or *Low* quality of service. Then the customer, player II, is informed about that choice. Player II can then decide separately between *buy* and *don't buy* in each case. The resulting payoffs are the same as in the strategic-form game in Fig. 3. However, the game is different from the one in Fig. 3, since the players now move in sequence rather than simultaneously.

Extensive games with perfect information can be analyzed by *backward induction.* This technique solves the game by first considering the last possible choices in the game. Here, player II moves last. Since she knows the play will end after her move, she can safely select the action which is best for her. If player I has chosen to provide high-quality service, then the customer prefers to *buy,* since her resulting payoff of 2 is larger than 1 when not buying. If the provider has chosen *Low,* then the customer prefers not to purchase. These choices by player II are indicated by arrows in Fig. 7.

Once the last moves have been decided, backward induction proceeds to the players making the next-to-last moves (and then continues in this manner). In Fig. 7, player I makes the next-to-last move, which in this case is the first move in the game. Being rational, he anticipates the subsequent choices by the customer. He therefore realizes that his decision between *High* and *Low* is effectively between the outcomes with pay-

offs (2, 2) or (1, 1) for the two players, respectively. Clearly, he prefers *High,* which results in a payoff of 2 for him, to *Low,* which leads to an outcome with payoff 1. So the unique solution to the game, as determined by backward induction, is that player I offers high-quality service, and player II responds by buying the service.

B. Strategies in Extensive Games

In an extensive game with perfect information, backward induction usually prescribes unique choices at the players' decision nodes. The only exception is if a player is indifferent between two or more moves at a node. Then, any of these best moves, or even randomly selecting from among them, could be chosen by the analyst in the backward induction process. Since the eventual outcome depends on these choices, this may affect a player who moves earlier, since the anticipated payoffs of that player may depend on the subsequent moves of other players. In this case, backward induction does not yield a unique outcome; however, this can only occur when a player is exactly indifferent between two or more outcomes.

The backward induction solution specifies the way the game will be played. Starting from the root of the tree, play proceeds along a path to an outcome. Note that the analysis yields more than the choices along the path. Because backward induction looks at every node in the tree, it specifies for every player a *complete plan* of what to do at every point in the game where the player can make a move, even though that point may never arise in the course of play. Such a plan is called a *strategy* of the player. For example, a strategy of player II in Fig. 7 is "buy if offered high-quality service, don't buy if offered low-quality service." This is player II's strategy obtained by backward induction. Only the first choice in this strategy comes into effect when the game is played according to the backward-induction solution.

With strategies defined as complete move plans, one can obtain the *strategic form* of the extensive game. As in the strategic form games shown before, this tabulates all strategies of the players. In the game tree, any strategy combination results into an outcome of the game, which can be determined by tracing out the path of play arising from the players adopting the strategy combination. The payoffs to the players are then entered into the corresponding cell in the strategic form. Figure 8 shows the strategic form for our example. The second column is player II's backward induction strategy, where "buy if offered high-quality service, don't buy if offered low-quality service" is abbreviated as *H: buy, L: don't.*

I \ II	*H: buy, L: buy*	*H: buy, L: don't*	*H: don't, L: buy*	*H: don't, L: don't*
High	2 2	2 2	1 0	1 0
Low	0 3	1 1	0 3	1 1

Figure 8 Strategic form of the extensive game in Fig. 7.

A game tree can therefore be analyzed in terms of the strategic form. It is not hard to see that backward induction always defines a Nash equilibrium. In Fig. 8, it is the strategy combination *(High; H: buy, L: don't).*

A game that evolves over time is better represented by a game tree than using the strategic form. The tree reflects the temporal aspect, and backward induction is succinct and natural. The strategic form typically contains redundancies. Figure 8, for example, has eight cells, but the game tree in Fig. 7 has only four outcomes. Every outcome appears twice, which happens when two strategies of player II differ only in the move that is not reached after the move of player I. All move combinations of player II must be distinguished as strategies since any two of them may lead to different outcomes, depending on the action of player I.

Not all Nash equilibria in an extensive game arise by backward induction. In Fig. 8, the rightmost bottom cell *(Low; H: don't, L: don't)* is also an equilibrium. Here the customer never buys, and correspondingly *Low* is the best response of the service provider to this anticipated behavior of player II. Although *H: don't* is not an optimal choice (so it disagrees with backward induction), player II never has to make that move, and is therefore not better off by changing her strategy. Hence, this is indeed an equilibrium. It prescribes a suboptimal move in the *subgame* where player II has learned that player I has chosen *High.* Because a Nash equilibrium obtained by backward induction does not have such a deficiency, it is also called *subgame perfect.*

The strategic form of a game tree may reveal Nash equilibria which are not subgame perfect. Then a player plans to behave irrationally in a subgame. He may even profit from this *threat* as long as he does not have to execute it (that is, the subgame stays unreached). Examples are games of market entry deterrence, for example, the so-called "chain store" game.

The analysis of dynamic strategic interaction was pioneered by Selten, for which he earned a share of the 1994 Nobel prize.

C. First-Mover Advantage

A practical application of game-theoretic analysis may be to reveal the potential effects of changing the "rules" of the game. This has been illustrated with three versions of the quality choice game, with the analysis resulting in three different predictions for how the game might be played by rational players. Changing the original quality choice game in Fig. 3 to Fig. 4 yielded an additional, although not unique, Nash equilibrium *(High, buy)*. The change from Fig. 3 to Fig. 7 is more fundamental since there the provider has the power to *commit* himself to high- or low-quality service, and inform the customer of that choice. The backward-induction equilibrium in that game is unique, and the outcome is better for both players than the original equilibrium *(Low, don't buy)*.

Many games in strategic form exhibit what may be called the *first-mover advantage*. A player in a game becomes a first mover or "leader" when he can *commit* to a strategy, that is, choose a strategy irrevocably and inform the other players about it; this is a change of the "rules of the game." The first-mover advantage states that a player who can become a leader is not worse off than in the original game where the players act simultaneously. In other words, if one of the players has the power to commit, he or she should do so.

This statement must be interpreted carefully. For example, if more than one player has the power to commit, then it is not necessarily best to go first. For example, consider changing the game in Fig. 3 so that player II can commit to her strategy, and player I moves second. Then player I will always respond by choosing *Low*, since this is his dominant choice in Fig. 3. Backward induction would then amount to player II not buying and player I offering low service, with the low payoff 1 to both. Then player II is not worse off than in the simultaneous-choice game, as asserted by the first-mover advantage, but does not gain anything either. In contrast, making player I the first mover as in Fig. 7 is beneficial to both.

If the game has antagonistic aspects, like the inspection game in Fig. 6, then mixed strategies may be required to find a Nash equilibrium of the simultaneous-choice game. The first-mover game always has an equilibrium, by backward induction, but having to commit and inform the other player of a pure strategy may be disadvantageous. The correct comparison is to consider commitment to a *randomized choice*, like to a certain inspection probability. In Fig. 6, already the commitment to the pure strategy *Inspect* gives a better payoff to player I than the original mixed equilibrium since player II will respond by complying, but a commitment to a sufficiently high inspection probability (anything above 10%) is even better for player I.

D. Example: Duopoly of Chip Manufacturers

The first-mover advantage is also known as *Stackelberg leadership*, after the economist Heinrich von Stackelberg who formulated this concept for the structure of markets in 1934. The classic application is to the duopoly model by Cournot, which dates back to 1838.

As an example, suppose that the market for a certain type of memory chip is dominated by two producers. The firms can choose to produce a certain quantity of chips, say either high, medium, low, or none at all, denoted by *H*, *M*, *L*, *N* for firm I and *h*, *m*, *l*, *n* for firm II. The market price of the memory chips decreases with increasing total quantity produced by both companies. In particular, if both choose a high quantity of production, the price collapses so that profits drop to zero. The firms know how increased production lowers the chip price and their profits. Figure 9 shows the game in strategic form, where both firms choose their output level simultaneously. The symmetric payoffs are derived from Cournot's model, explained below.

I \ II	*h*	*m*	*l*	*n*
H	0 0	8 12	9 18	0 36
M	12 8	16 16	15 20	0 32
L	18 9	20 15	18 18	0 27
N	36 0	32 0	27 0	0 0

Figure 9 Duopoly game between two chip manufacturers who can decide between high, medium, low, or no production, denoted by *H*, *M*, *L*, *N* for firm I and *h*, *m*, *l*, *n* for firm II. Prices fall with increased production. Payoffs denote profits in millions of dollars.

The game can be solved by dominance considerations. Clearly, no production is dominated by low or medium production, so that row *N* and column *n* in Fig. 9 can be eliminated. Then, high production is dominated by medium production, so that row *H* and column *h* can be omitted. At this point, only medium and low production remain. Then, regardless of whether the opponent produces medium or low, it is always better for each firm to produce medium. Therefore, the Nash equilibrium of the game is (*M,m*), where both firms make a profit of $16 million.

Consider now the commitment version of the game, with a game tree (omitted here) corresponding to Fig. 9 just as Fig. 7 is obtained from Fig. 3. Suppose that firm I is able to publicly announce and commit to a level of production, given by a row in Fig. 9. Then firm II, informed of the choice of firm I, will respond to *H* by *l* (with maximum payoff 9 to firm II), to *M* by *m*, to *L* also by *m*, and to *N* by *h*. This determines the backward-induction strategy of firm II. Among these anticipated responses by firm II, firm I does best by announcing *H*, a high level of production. The backward-induction outcome is thus that firm I makes a profit of $18 million, as opposed to only $16 million in the simultaneous-choice game. When firm II must play the role of the follower, its profits fall from $16 to $9 million.

The first-mover advantage again comes from the ability of firm I to credibly commit itself. After firm I has chosen *H*, and firm II replies with *l*, firm I would like to be able to switch to *M*, improving profits even further from $18 to $20 million. However, once firm I is producing *M*, firm II would change to *m*. This logic demonstrates why, when the firms choose their quantities simultaneously, the strategy combination (*H,l*) is not an equilibrium. The commitment power of firm I, and firm II's appreciation of this fact, is crucial.

The payoffs in Fig. 9 are derived from the following simple model due to Cournot. The high, medium, low, and zero production numbers are 6, 4, 3, and 0 million memory chips, respectively. The profit per chip is $12 - Q$ dollars, where Q is the total quantity (in millions of chips) on the market. The entire production is sold. As an example, the strategy combination (*H,l*) yields $Q = 6 + 3 = 9$, with a profit of $3 per chip. This yields the payoffs of $18 million and $9 million for firms I and II in the (*H,l*) cell in Fig. 9. Another example is firm I acting as a monopolist (firm II choosing *n*), with a high production level *H* of 6 million chips sold at a profit of $6 each.

In this model, a monopolist would produce a quantity of 6 million even if other numbers than 6, 4, 3, or 0 were allowed, which gives the maximum profit of $36 million. The two firms could cooperate and split that amount by producing 3 million chips each, corresponding to the strategy combination (*L,l*) in Fig. 9. The equilibrium quantities, however, are 4 million for each firm, where both firms receive less. The central four cells in Figure 9, with low and medium production in place of "cooperate" and "defect," have the structure of a Prisoner's Dilemma game (Figure 1), which arises here in a natural economic context. The optimal commitment of a first mover is to produce a quantity of 6 million, with the follower choosing 3 million. These numbers, and the equilibrium ("Cournot") quantity of 4 million, apply even when arbitrary quantities are allowed (see Gibbons, 1992).

VII. EXTENSIVE GAMES WITH IMPERFECT INFORMATION

Typically, players do not always have full access to all the information which is relevant to their choices. Extensive games with *imperfect information* model exactly which information is available to the players when they make a move. Modeling and evaluating strategic information precisely is one of the strengths of game theory. John Harsanyi's pioneering work in this area was recognized in the 1994 Nobel awards.

Consider the situation faced by a large software company after a small startup has announced deployment of a key new technology. The large company has a large research and development operation, and it is generally known that they have researchers working on a wide variety of innovations. However, only the large company knows for sure whether or not they have made any progress on a product similar to the startup's new technology. The startup believes that there is a 50% chance that the large company has developed the basis for a strong competing product. For brevity, when the large company has the ability to produce a strong competing product, the company will be referred to as having a "strong" position, as opposed to a "weak" one.

The large company, after the announcement, has two choices. It can counter by announcing that it too will release a competing product. Alternatively, it can choose to cede the market for this product. The large company will certainly condition its choice upon its private knowledge, and may choose to act differently when it has a strong position than when it has a weak one. If the large company has announced a product, the startup is faced with a choice: it can either negotiate a buyout and sell itself to the large company, or it can remain independent and launch its product.

The startup does not have access to the large firm's private information on the status of its research. However, it does observe whether or not the large company announces its own product, and may attempt to infer from that choice the likelihood that the large company has made progress of their own.

When the large company does not have a strong product, the startup would prefer to stay in the market over selling out. When the large company does have a strong product, the opposite is true, and the startup is better off by selling out instead of staying in.

Figure 10 shows an extensive game that models this situation. From the perspective of the startup, whether or not the large company has done research in this area is random. To capture random events such as this formally in game trees, *chance moves* are introduced. At a node labeled as a chance move, the next branch of the tree is taken randomly and nonstrategically by chance, or "nature," according to probabilities which are included in the specification of the game.

The game in Fig. 10 starts with a chance move at the root. With equal probability 0.5, the chance move decides if the large software company, player I, is in a strong position (upward move) or weak position (downward move). When the company is in a weak position, it can choose to *Cede* the market to the startup, with payoffs (0, 16) to the two players (with payoffs given in millions of dollars of profit). It can also *Announce* a competing product, in the hope that the startup company, player II, will *sell out,* with payoffs 12 and 4 to players I and II. However, if player II decides instead to *stay in,* it will even profit from the increased publicity and gain a payoff of 20, with a loss of −4 to the large firm.

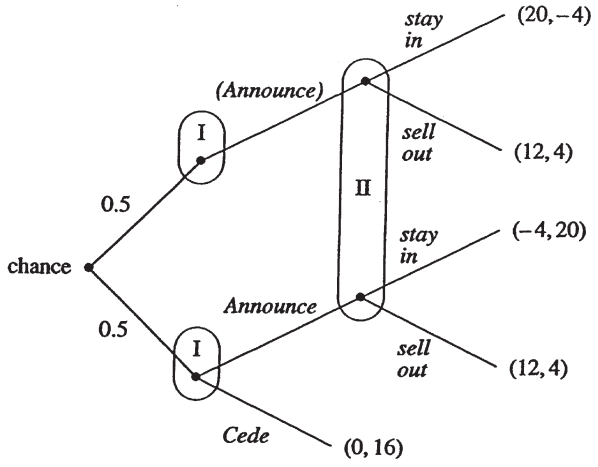


Figure 10 Extensive game with imperfect information between player I, a large software firm, and player II, a startup company. The chance move decides if player I is strong (top node) and does have a competing product, or weak (bottom node) and does not. The ovals indicate information sets. Player II sees only that player I chose to *Announce* a competing product, but does not know if player I is strong or weak.

In contrast, when the large firm is in a strong position, it will not even consider the move of ceding the market to the startup, but will instead just announce its own product. In Figure 10, this is modeled by a single choice of player I at the upper node, which is taken for granted (one could add the extra choice of ceding and subsequently eliminate it as a dominated choice of the large firm). Then the payoffs to the two players are (20,−4) if the startup stays in and (12, 4) if the startup sells out.

In addition to a game tree with perfect information as in Fig. 7, the nodes of the players are enclosed by ovals which are called *information sets.* The interpretation is that a player cannot distinguish among the nodes in an information set, given his knowledge at the time he makes the move. Since his knowledge at all nodes in an information set is the same, he makes the same choice at each node in that set. Here, the startup company, player II, must choose between *stay in* and *sell out.* These are the two choices at player II's information set, which has two nodes according to the different histories of play, which player II cannot distinguish.

Because player II is not informed about its position in the game, backward induction can no longer be applied. It would be better to *sell out* at the top node, and to *stay in* at the bottom node. Consequently, player I's choice when being in the weak position is not clear: if player II stays in, then it is better to *Cede* (since 0 is better than −4), but if player II sells out, then it is better to *Announce.*

The game does not have an equilibrium in pure strategies: the startup would respond to *Cede* by selling out when seeing an announcement, since then this is only observed when player I is strong. But then player I would respond by announcing a product even in the weak position. In turn, the equal chance of facing a strong or weak opponent would induce the startup to stay in, since then the expected payoff of $0.5(-4) + 0.5 \times 20 = 8$ exceeds 4 when selling out.

The equilibrium of the game involves both players randomizing. The mixed strategy probabilities can be determined from the strategic form of the game in Fig. 11. When it is in a weak position, the large firm randomizes with equal probability 1/2 between *Announce* and *Cede* so that the expected payoff to player II is then 7 for both *stay in* and *sell out.*

Since player II is indifferent, randomization is a best response. If the startup chooses to *stay in* with

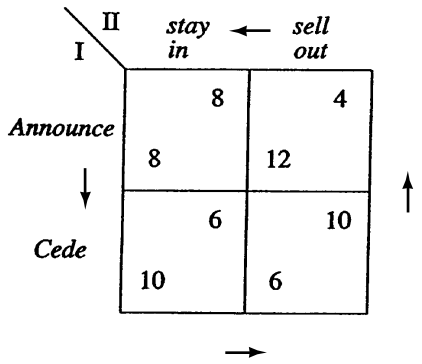


Figure 11 Strategic form of the extensive game in Fig. 10, with expected payoffs resulting from the chance move and the player's choices.

probability 3/4 and to *sell out* with probability 1/4, then player I, in turn, is indifferent, receiving an overall expected payoff of 9 in each case. This can also be seen from the extensive game in Fig. 10: when in a weak position, player I is indifferent between the moves *Announce* and *Cede* where the expected payoff is 0 in each case. With probability 1/2, player I is in the strong position, and stands to gain an expected payoff of 18 when facing the mixed strategy of player II. The overall expected payoff to player I is 9.

VIII. ZERO-SUM GAMES AND COMPUTATION

The extreme case of players with fully opposed interests is embodied in the class of two-player *zero-sum* (or constant-sum) games. Familiar examples range from rock-paper-scissors to many parlor games like chess, go, or checkers.

A classic case of a zero-sum game, which was considered in the early days of game theory by von Neumann, is the game of poker. The extensive game in Fig. 10, and its strategic form in Fig. 11, can be interpreted in terms of poker, where player I is dealt a strong or weak hand which is unknown to player II. It is a *constant-sum* game since for any outcome, the two payoffs add up to 16, so that one player's gain is the other player's loss. When player I chooses to announce despite being in a weak position, he is colloquially said to be "bluffing." This bluff not only induces player II to possibly sell out, but similarly allows for the possibility that player II stays in when player I is strong, increasing the gain to player I.

Mixed strategies are a natural device for constant-sum games with imperfect information. Leaving one's own actions open reduces one's vulnerability against malicious responses. In the poker game of Fig. 10, it is too costly to bluff all the time, and better to randomize instead. The use of active randomization will be familiar to anyone who has played rock-paper-scissors.

Zero-sum games can be used to model strategically the computer science concept of "demonic" nondeterminism. Demonic nondeterminism is based on the assumption that, when an ordering of events is not specified, one must assume that the worst possible sequence will take place. This can be placed into the framework of zero-sum game theory by treating nature (or the environment) as an antagonistic opponent. Optimal randomization by such an opponent describes a worst-case scenario that can serve as a benchmark. A similar use of randomization is known in the theory of algorithms as Rao's theorem, and describes the power of randomized algorithms. An example is the well-known *quicksort algorithm,* which has one of the best observed running times of sorting algorithms in practice, but can have bad worst-cases. With randomization, these can be made extremely unlikely.

Randomized algorithms and zero-sum games are used for analyzing problems in *online computation.* This is, despite its name, not related to the Internet, but describes the situation where an algorithm receives its input one data item at a time, and has to make decisions, for example, in scheduling, without being able to wait until the entirety of the input is known. The analysis of on-line algorithms has revealed insights into hard optimization problems, and seems also relevant to the massive data processing that is to be expected in the future. At present, it constitutes an active research area, although mostly confined to theoretical computer science (see Borodin and El-Yaniv, *Online Computation and Competitive Analysis,* Cambridge University Press, 1998).

IX. BIDDING IN AUCTIONS

The design and analysis of auctions is one of the triumphs of game theory. Auction theory was pioneered by the economist William Vickrey in 1961. Its practical use became apparent in the 1990s, when auctions of radio frequency spectrum for mobile telecommunication raised billions of dollars. Economic theorists advised governments on the design of these auctions, and companies on how to bid (see McMillan, Selling spectrum rights, *Journal of Economic Perspectives* Vol. 8, 1994, pages 145–162). The auctions for spectrum rights are complex. However, many principles for sound bidding can be illustrated by applying game-theoretic ideas to simple examples. This section highlights some of these

examples; see Milgrom, Auctions and bidding: a primer, *Journal of Economic Perspectives* Vol. 3, 1989, pages 3–22 for a broader view of the theory of bidding in auctions.

A. Second-Price Auctions with Private Values

The most familiar type of auction is the familiar *open ascending-bid* auction, which is also called an *English* auction. In this auction format, an object is put up for sale. With the potential buyers present, an auctioneer raises the price for the object as long as two or more bidders are willing to pay that price. The auction stops when there is only one bidder left, who gets the object at the price at which the last remaining opponent drops out.

A complete analysis of the English auction as a game is complicated, as the extensive form of the auction is very large. The observation that the winning bidder in the English auction pays the amount at which the last remaining opponent drops out suggests a simpler auction format, the *second-price* auction, for analysis. In a second-price auction, each potential buyer privately submits, perhaps in a sealed envelope or over a secure computer connection, his bid for the object to the auctioneer. After receiving all the bids, the auctioneer then awards the object to the bidder with the highest bid, and charges him the amount of the second-highest bid. Vickrey's analysis dealt with auctions with these rules.

How should one bid in a second-price auction? Suppose that the object being auctioned is one where the bidders each have a *private value* for the object. That is, each bidder's value derives from his personal tastes for the object, and not from considerations such as potential resale value. Suppose this valuation is expressed in monetary terms, as the maximum amount the bidder would be willing to pay to buy the object. Then the optimal bidding strategy is to submit a bid equal to one's actual value for the object.

Bidding one's private value in a second-price auction is a *weakly dominant* strategy. That is, irrespective of what the other bidders are doing, no other strategy can yield a better outcome. (Recall that a dominant strategy is one that is *always* better than the dominated strategy; weak dominance allows for other strategies that are sometimes equally good.) To see this, suppose first that a bidder bids less than the object was worth to him. Then if he wins the auction, he still pays the second-highest bid, so nothing changes. However, he now risks that the object is sold to someone else at a lower price than his true valuation, which makes the bidder worse off. Similarly, if one bids more than one's value, the only case where this can make a difference is when there is, below the new bid, another bid exceeding the own value. The bidder, if he wins, must then pay that price, which he prefers less than not winning the object. In all other cases, the outcome is the same. Bidding one's true valuation is a simple strategy, and, being weakly dominant, does not require much thought about the actions of others.

While second-price sealed-bid auctions like the one described above are not very common, they provide insight into a Nash equilibrium of the English auction. There is a strategy in the English auction which is analogous to the weakly dominant strategy in the second price auction. In this strategy, a bidder remains active in the auction until the price exceeds the bidder's value, and then drops out. If all bidders adopt this strategy, no bidder can make himself better off by switching to a different one. Therefore, it is a Nash equilibrium when all bidders adopt this strategy.

Most on-line auction web sites employ an auction which has features of both the English and second-price rules. In these auctions, the current price is generally observable to all participants. However, a bidder, instead of frequently checking the auction site for the current price, can instead instruct an agent, usually an automated program provided by the auction site, to stay in until the price surpasses a given amount. If the current bid is by another bidder and below that amount, then the agent only bids up the price enough so that it has the new high bid. Operationally, this is similar to submitting a sealed bid in a second-price auction. Since the use of such agents helps to minimize the time investment needed for bidders, sites providing these agents encourage more bidders to participate, which improves the price sellers can get for their goods.

B. Example: Common Values and the Winner's Curse

A crucial assumption in the previous example of bidding in a second-price auction is that of private values. In practice, this assumption may be a very poor approximation. An object of art may be bought as an investment, and a radio spectrum license is acquired for business reasons, where the value of the license depends on market forces, such as the demand for mobile telephone usage, which have a common impact on all bidders. Typically, auctions have both private and *common value* aspects.

In a purely common value scenario, where the object is worth the same to all bidders, bidders must decide how to take into account uncertainty about that value. In this case, each bidder may have, prior to the auction, received some private information or signals about the value of the object for sale. For example, in the case of radio spectrum licenses, each participating firm may have undertaken its own market research surveys to estimate the retail demand for the use of that bandwidth. Each survey will come back with slightly different results, and, ideally, each bidder would like to have access to all the surveys in formulating its bid. Since the information is proprietary, that is not possible.

Strategic thinking, then, requires the bidders to take into account the additional information obtained by winning the auction. Namely, the sheer fact of winning means that one's own, private information about the worth of the object was probably overly optimistic, perhaps because the market research surveys came back with estimates for bandwidth demand which were too bullish. Even if everybody's estimate about that worth is correct on average, the largest (or smallest) of these estimates is not. In a procurement situation, for example, an experienced bidder should add to his own bid not only a markup for profit, but also for the likely underestimation of the cost that results from the competitive selection process. The principle that winning a common-value auction is "bad news" for the winner concerning the valuation of the object is called the *winner's curse*.

The following final example, whose structure was first proposed by Max Bazerman and William Samuelson, demonstrates the considerations underlying the winner's curse not for an auction, but in a simpler situation where the additional information of "winning" is crucial for the expected utility of the outcome. Consider a potential buyer who is preparing a final, "take it or leave it" offer to buy out a dot com company. Because of potential synergies, both the buyer and the seller know that the assets of the dot com are worth 50% more to the buyer than to the current owner of the firm. If the value of the company were publicly known, the parties could work out a profitable trade, negotiating a price where both would profit from the transaction.

However, the buyer does not know the exact value of the company. She believes that it is equally likely to be any value between $0 and $10 million. The dot com's current owners know exactly the value of retaining the company, because they have complete information on their company's operations. In this case, the expected value of the company to the current owners is $5 million, and the expected value of the company to the prospective buyer is $7.5 million. Moreover, no matter what the value of the company truly is, the company is always worth more to the buyer than it is to the current owner. With this in mind, what offer should the buyer tender to the dot com as her last, best offer, to be accepted or rejected?

To find the equilibrium of this game, note that the current owners of the dot com will accept any offer that is greater than the value of the company to them, and reject any offer that is less. So, if the buyer tenders an offer of $5 million, then the dot com owners will accept if their value is between $0 and $5 million. The buyer, being strategic, then realizes that this implies the value of the company to her is equally likely to be anywhere between $0 and $7.5 million. This means that, if she offers $5 million, the average value of the company, conditioning upon the owners of the dot com accepting the offer, is only $3.75 million—less than the value of the offer. Therefore, the buyer concludes that offering $5 million will lead to an expected loss.

The preceding analysis does not depend on the amount of the offer. The buyer soon realizes that, no matter what offer she makes, when she takes into account the fact that the offer will be accepted only when the value of the dot com turns out to be on the low end. The expected value of the company to the buyer, conditional on her offer being accepted, is always less than her offer. It is this updating of the buyer's beliefs, shifting her estimation of the dot com's value to the low end, which embodies the winner's curse in this example. Having her offer accepted is bad news for the buyer, because she realizes it implies the value of the dot com is low. The equilibrium in this game involves the buyer making an offer of zero, and the offer never being accepted.

This example is particularly extreme, in that no transaction is made even though everyone involved realizes that a transaction would be profitable to both sides. As is generally the case with noncooperative game theory, the equilibrium does depend on the details of the rules of the game, in this case, the assumption that one last, best offer is being made, which will either be accepted or rejected. In general, the winner's curse will not always prohibit mutually profitable transactions from occurring. This example demonstrates the importance of carefully taking into account the information one learns during the course of play of a game. It also shows how a game-theoretic model that incorporates the information and incentives of others helps promote sound decision making.

SEE ALSO THE FOLLOWING ARTICLES

Automata Theory • Cybernetics • Data, Information, and Knowledge • Decision Theory • Future of Information Systems • Goal Programming • Information Theory • Monte Carlo Simulation • Strategic Planning for/of Information Systems • Uncertainty

BIBLIOGRAPHY

Binmore, K. (1991). *Fun and games: A text on game theory.* Lexington, MA: D.C. Health.

Dixit, A. K., and Nalebuff, B. J. (1991). *Thinking strategically: The competitive edge in business, politics, and everyday life.* New York: Norton.

Fudenberg, D., and Tirole, J. (1991). *Game theory.* Cambridge, MA: MIT Press.

Gibbons, R. (1992). *Game theory for applied economists.* Princeton NJ: Princeton University Press.

Myerson, R. B. (1991). *Game theory: Analysis of conflict.* Cambridge, MA: Harvard University Press.

Nasar, S. (1998). *A beautiful mind: A biography of John Forbes Nash, Jr., winner of the nobel prize in economics, 1994.* New York: Simon and Schuster.

Rasmusen, E. (2001). *Games and information: An introduction to game theory,* 3rd ed. Oxford: Blackwell.

Geographic Information Systems

Peter Keenan
University College, Dublin

GLOSSARY

buffers Regions generated around spatial objects.
choropleth maps Thematic maps that display attribute (nonspatial) data values associated with relevant spatial units.
geographic information system (GIS) Also Geographical Information System. Describes a computer system for storing, manipulating, and displaying geographically referenced information.
overlay A method for determining whether different types of geographic features are spatially coincident.
raster data Data that uses a representation of the map represented by pixels in a grid (bitmap).
remote sensing The collection of data from a distance using aircraft or satellites.
spatial data Data about the location and shape of, and relationships between, geographic features.
spatial decision support system (SDSS) A decision support system based on GIS technology.
vector data A complex representation of data built from basic shapes. Three basic objects are used: points, lines, and areas (polygons).
topologically integrated geographic encoding and referencing system (TIGER) A map format widely used for public maps in the United States.

GEOGRAPHIC INFORMATION SYSTEMS (GIS) are an area of information technology (IT) application with a significantly different history from the other information systems discussed in this book. These systems facilitate the display and storage of geographically or spatially related data and allow the integration of this data with nonspatial (attribute) data. Recently, the acronym GIS has also been used as an abbreviation for Geographical Information Science, referring to a body of research on techniques for processing geographic information. A GIS employs these techniques. In this chapter the expression GIS will always refer to a computer system. We will look at the origins of these systems, the type of data used by GIS, and the relationship between GIS and the other systems discussed elsewhere in this book.

I. INTRODUCTION TO GEOGRAPHIC INFORMATION SYSTEMS

A. History of Mapping

Geographic data, also known as spatial data, have always played an important part in human decision making. Many vital activities require geographic data, not the least of which are travel, agriculture, and construction. As people came to have a better understanding of the world around them, a variety of representations of geographic data have been introduced. The best known of these is the map, which shows geographic objects and their spatial relationship. The production of maps has long been a necessary part of human exploration and understanding of the world around us.

Before the computer age, paper maps provided an important information representation that facilitated insight into problems. Military and transport planners found that appropriate maps were needed for their operations. By the mid-19th century, map making had

evolved to a sophisticated level, one example of this was the "Atlas to accompany the second report of the Irish Railway Commissioners," which showed population, traffic flow, geology, and topography all displayed on the same map. This allowed easy understanding of the feasibility of proposed railway routes.

The mid-19th century was also a period of important advances in medicine. At that time there was an ongoing debate in the medical community over the origins and method of dissemination of the cholera disease, which was then widespread. In 1854 John Snow, an English doctor, attempted to get a better understanding of the problem. He used a map of London to plot the locations of the homes of cholera fatalities. He also marked the location of water pumps on the map (Fig. 1). Using this representation, it became clear that those with cholera were clustered around a particular pump. This supported the belief that the disease was spread by contaminated water. Snow asked for that pump to be closed and the epidemic subsided. This example indicates how locations plotted on a map could provide information that was not obvious from an analysis of the nonspatial characteristics of the patients, e.g., ages, diet, etc.

Business use of IT started with its use in payroll and invoice processing in the period after World War II. Early applications employed relatively simple processing that could be automated using the comparatively crude computer technology of the period. The use of computers for geographic applications required the development of more sophisticated technology. A typical payroll application might involve the multiplication of hours worked by the wage rate and the deduction of taxes from the total; less than 10 data values might be needed for this calculation. Spatial processing applications typically require the use of much

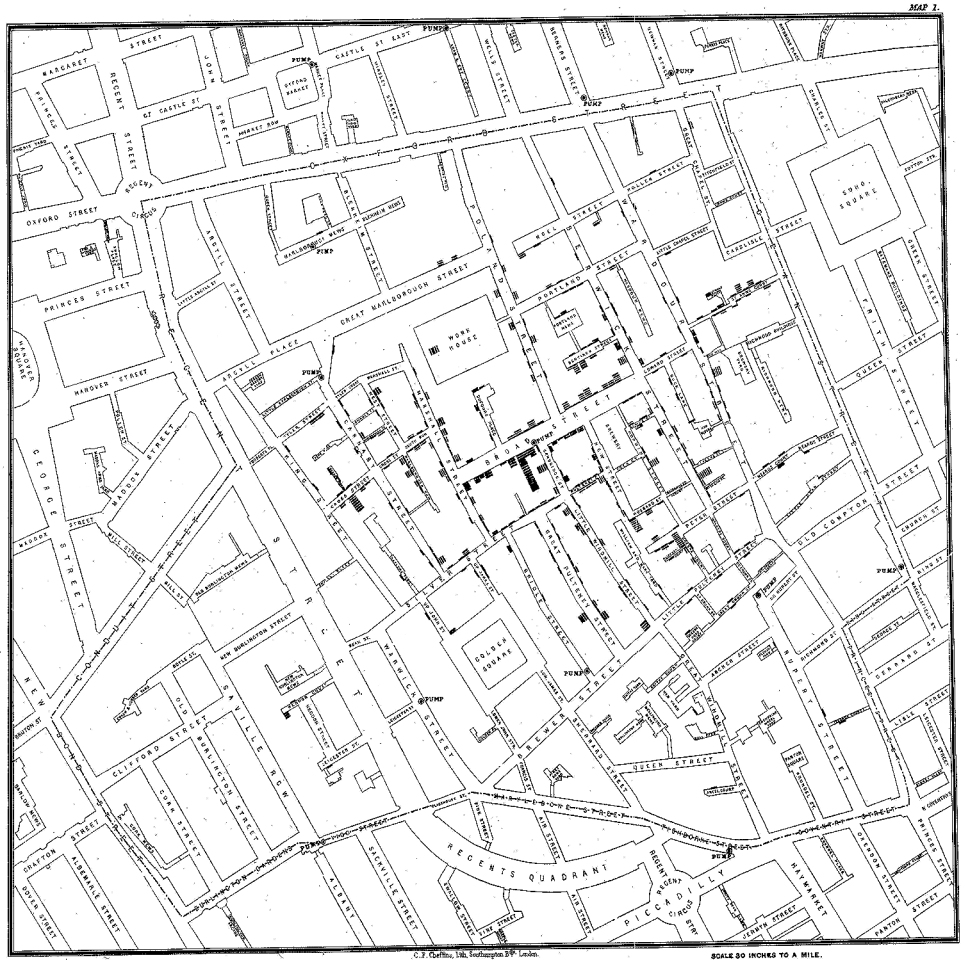


Figure 1 Cholera patients in London, 1854. This map appeared in Dr. John Snow's 1855 book *On the Mode of Communication of Cholera*. Note how most of the dots are centered about the pump on Broad street. [From http://www.ph.ucla.edu/epi/snow/snowmap1a.html.]

larger data sets than traditional data processing. For instance, in the map shown in Fig. 1, there are hundreds of points representing patients in addition to the pumps and the lines marking the streets. In addition to the larger amounts of data involved, geographic processing also requires complex calculations that are much more demanding on computer time than those of routine data processing.

B. Early Use of Computerized Mapping

IT was used for geographic-related activities in the United States in the first decades of computer use, for instance, in the integrated transportation plans in Detroit and Chicago. These plans required maps of traffic flow and volume that integrated transportation information, routes, destinations, origins, and time. In the late 1950s, the Department of Geography at the University of Washington in Seattle conducted research on advanced statistical methods, rudimentary computer programming, and computer cartography. The Canada Geographic Information System (CGIS) is an example of one of the earliest GIS projects. This work started in the mid 1960s and facilitated analysis of the data collected by the Canada Land Inventory (CLI). This data was processed to produce statistics to be used in developing land management plans for large areas of rural Canada. This system introduced many of the concepts utilized by later GIS applications.

The U.S. Bureau of the Census, an important innovator in using new technology, also had a significant role to play in the development of GIS. The bureau recognized the need for a technique to assign census returns to their correct geographical location. Most census data were collected on the basis of street address as the fundamental unit of location. This meant a requirement for address matching to convert street addresses to geographic coordinates. The development of these techniques allowed the data for the 1970 census to be placed into census reporting zones, facilitating aggregation of the results and greatly improving data reporting. Address matching has proved to be an important technology for the use of GIS in urban areas ever since.

The 1960s saw the introduction of many of the basic concepts in GIS, although their widespread implementation awaited further developments in computer technology. This period also saw the foundation of the organizations that were to become ESRI and Intergraph, now two major software companies that supply GIS software. From these foundations the GIS field has expanded greatly, taking advantage of the exponential increase in computing power and the introduction of relevant peripheral devices such as large color screens and printers.

C. Current Geographic Information System Technology

In parallel with the development of computer technology, the functionality of GIS software has greatly increased. Initially, GIS software was run on mainframe computers and then on relatively expensive graphics workstations. However, as computer speeds improved, it has become possible to run GIS software on standard personal computers. The spread of GIS technology has been accompanied by simultaneous growth in the amount of digital data available. The same geographical data sets may be used by many different organizations, as many activities will take place in the same geographic region. Consequently, GIS is somewhat unusual compared to other IT applications in that many users typically outsource both their software and a large part of their data. Any assessment of the scale of the GIS field must take account of the value of the data, as well as hardware and software. A recent report by the International Data Corp. (IDC) suggests that the GIS marketplace in 1999 is worth more than $1 billion U.S. and is growing at 12% per year, this means that market size will double by 2004. The IDC report suggests that growth is fastest for those concentrating on business systems.

II. GEOGRAPHIC INFORMATION SYSTEM DATA

A. Data Collection

Data representation in GIS draws its inspiration from geography and many centuries of cartography, the science of mapping. Maps employ certain conventions to provide a convenient portrayal of the real world. The earth is a sphere, yet maps allow us see a representation of the spatial relationship of continents using a flat page. GIS faces similar problems of representation, as it must also present information on a two-dimensional computer screen and produce printouts of flat paper maps. Information representation in GIS also faces problems resulting from the limitations of computer technology. Issues that arise might include limited storage capacity and round off error associated with machine calculations.

In order to maintain the spatial relationship of objects, GIS databases must record their spatial location

in terms of a projection. Map projections are attempts to portray the curved surface of the earth or a portion of the earth on a flat surface. As a globe cannot be peeled off and laid out flat, flat maps always represent a compromise. A coordinate system (georeferencing system) is needed to associate locations on a map with their real position on the earth's surface. At a global scale we use the latitude and longitude system, which takes into account the curvature of the earth. In practice, a local coordinate system is used for most maps, which simplify or ignore issues arising from the curvature of the earth. A simple coordinate system will provide a reasonably accurate relationship with reality over a limited area. Therefore, smaller countries may have a single system for all maps; larger countries will use different systems for each region. These differences present problems where a GIS must integrate data from two neighboring regions with different systems. Where many local coordinate systems are used, there is closer correspondence with the actual spherical shape of earth. However, there are more boundaries between regions, presenting problems of data integration.

GIS data can be collected from existing paper maps by a process known as digitizing. Data capture from paper maps is a time-consuming and labor-intensive process. One approach is to place the paper map on a special digitizing tablet and to manually trace objects of interest on the map using a device similar to a computer mouse. This is a slow process, but it allows the simultaneous visual identification of the different types of information on the paper map. An alternative approach is to use a scanner that records all of the information on the map. Scanning provides a digital picture of the map, but does not allow easy identification of all of the different types of data on the map. GIS data collected by scanning only becomes useful when it is processed further, for instance, distinguishing those lines which represent roads from those associated with rivers. Further manual processing of the scanned data may be required to achieve this. In many developed countries, mapping agencies have already completed a large proportion of the work required for spatial data collection from existing materials. Future updates to this data can take place on the computer database, so the need for tedious digitizing should reduce in the future.

One important source of information is **remote sensing,** the observation of an object from a distance. The main categories of remote sensing are aerial photography and the use of satellites to observe the earth. Remote sensing can use a variety of electronic sensors in addition to conventional photography. These techniques allow large amounts of data to be collected, but also present problems in the identification of the objects recorded. Satellite data collection allows periodic updating of the data and the identification of seasonal or other changes.

Data collection for GIS has been revolutionized by the use of another satellite-based system, the Global Positioning System (GPS). This allows the easy identification of the location of any point on the earth's surface. This technology is in use by both mapping agencies and GIS users. Individual companies can collect their own spatial data using handheld GPS or equipment installed in vehicles. For example, a delivery company could install GPS in its fleet and easily record the routes taken and the location of customers' premises.

B. Global Positioning Systems

GPS is a satellite-based, radio-navigation system developed and operated by the U.S. Department of Defense. GPS is based on signals from 24 operational satellites in 6 circular orbits 20,200 km (≈12,600 miles) above the earth, spaced in orbit so that at any time a minimum of 6 satellites are in view to users anywhere in the world. The first GPS satellite, the now decommissioned Block I developmental model, was launched in February 1978. The present system reached full operational capability on July 17, 1995.

The satellites continuously broadcast position and time data to users throughout the world. The distance to each satellite is calculated by measuring the time delay for the relevant signal to reach the user's receiver. Measurements collected simultaneously from four satellites are processed to identify the three dimensions of position, velocity, and time. GPS offers absolute positioning on the earth's surface, and a sequence of GPS measurements provides a method of calculating the relative positions of nearby objects. In general, the accuracy of relative measurement is better than the accuracy of absolute positioning. Historically, the GPS signal available to civilians was intentionally degraded and was not as accurate as the encrypted signal used by the U.S. military. However, from May 2000 the full accuracy has been generally available, allowing easy determination of locations to within 1 or 2 meters (3–6 feet).

C. Raster Data

There are two distinct approaches to the representation of data in GIS. The **raster** approach uses a bitmap representation, storing a representation of the map where the entire area of the map is represented by pixels (dots) on a grid. This can be regarded as a form of digital photograph of the map, with the quality dependent on the size of the pixels. Raster representations typically

result from the use of scanning technology to digitize paper maps. The raster format allows an exact duplicate of a paper map to be reproduced and is probably most appropriate when all of the detail on a map is of interest. However, the raster implementation makes it difficult to change the scale of the map. This representation can be very storage intensive, as millions of dots are required to represent a useful mapping region. Compression techniques can reduce the storage required if the map is not very detailed. These compression techniques operate in a similar way to the GIF and JPG bitmap formats used on the World Wide Web.

D. Vector Data

A **vector** representation of a map builds a complex geometric representation from basic shapes, similar to a computer-aided design (CAD) drawing. This approach provides a map representation that is independent of scale. Three basic objects are used: points, lines, and areas (also known as polygons). Points represent a single location and are typically used to identify the existence of objects whose actual dimensions are not of interest in the map in question. A single point on a map of a country may represent a town; this treatment would not be appropriate on a more detailed map where details such as the town boundaries would be displayed.

Lines are used on vector maps to represent linear objects such as roads or rivers. Each line is composed of a number of segments, so that lines appear to be curved. Each point on the line is defined in terms of the coordinate system of the map. In a vector map, a boundary line that forms an enclosed shape represents areas or polygons. Many natural features are represented in this way, e.g., lakes. Areas are also an important form of representation for administrative regions and other political structures (Table I).

E. Data Organization

GIS is used to record information for different types of geographic objects located within the same spatial region. As not all of this detail will be relevant to a given problem, GIS requires effective tools for managing different classes of data and the interactions between them. Within GIS, different classes of data are usually arranged in layers. The basic topographical details, for instance, lakes and rivers, may form one layer. Each type of man-made feature, for instance, roads, canals, or power lines, might constitute subsequent layers. This allows the option of displaying just one of these layers or displaying both where the man-made features are superimposed on the basic topographical map.

Table I Vector Data

Points	Lines	Polygons
Mountain summits	Rivers	Lakes
Towns	Roads	Property enclosures
Houses	Pipelines	Administrative regions

Some types of vector data represent objects that need to be connected. For example, roads generally form a network, as it is very unusual indeed to have a section of road not connected to other roads. Other types of vector data such as pipelines, cables, and railways also need to form connected networks. The accurate computer processing of networks requires high-quality GIS data. Traditional mapmaking techniques tended to emphasize the generation of maps that give a correct visual impression. The human eye may regard two lines that are very close together as being effectively joined, but a computer calculation may not recognize this (Fig. 2). This type of error will prevent the computer from processing the network correctly.

The representation of road networks presents particular problems. Modern limited access highways do not have junctions with all the minor roads around them (Fig. 3), but this may not be properly represented on the map (Fig. 4). Where intersections exist, there may be a complex series of bridges and ramps (Fig. 5).

F. Data Sources

GIS data is untypical of the data used in many of the information systems discussed in this Encyclopedia in that much of the data used is sourced outside the organizations using it. The basic geography of a region is shared by all activities in that region, and a large variety of organizations may make use of the same GIS data. In most countries digital data is available from

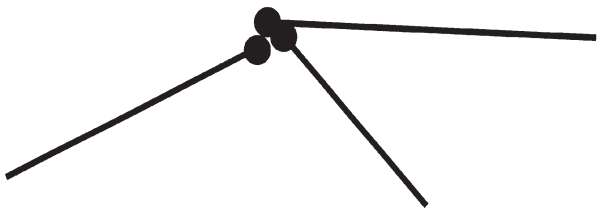


Figure 2 Inaccurate data capture in GIS.

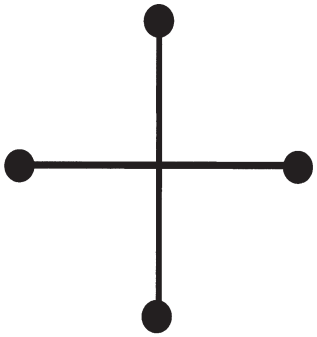


Figure 3 Highway with an overpass.

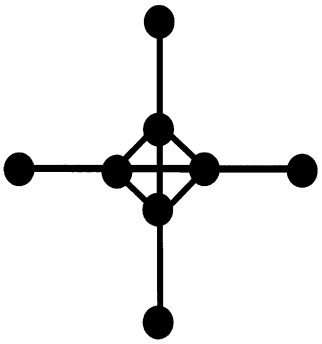


Figure 5 Highway with a complex intersection.

the public mapping agency, for instance, the Ordnance Survey of Great Britain, and also from private sector sources. In the United States, data collected by public agencies is considered to be the property of the public and is essentially available free. The best known of these U.S. data sets is TIGER® (Topologically Integrated Geographic Encoding and Referencing System). These maps provide a digital database of geographic features such as roads, railroads, rivers, lakes, political boundaries, and census statistical boundaries covering the entire United States.

In most European countries, governments seek to recover the cost of collecting the digital data from the users of that data. This means that high charges can be imposed in some countries for the use of digital spatial data and this has slowed the growth of GIS use in some places. In Europe, private sector sources often sell their products as an alternative to the government mapping agencies. In the United States, the basic data is free and the private sector can improve and extend the basic government maps and offer a variety of value-added services.

As the GIS industry has developed, increasing quantities of data have become available for developed countries. In these countries many users find that their needs are well served by off-the-shelf data. As existing databases are more intensively used, the cost of data should be reduced, which in turn will make new applications of GIS economically feasible.

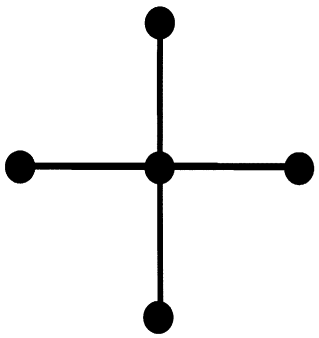


Figure 4 Highway overpass incorrectly represented as a junction.

III. GEOGRAPHIC INFORMATION SYSTEM OPERATIONS

A. Map Graphics

Even relatively simple mapping software can greatly assist in the understanding of geographic data. Just as basic spreadsheet software has a variety of graphic tools for the display of information, GIS software can display spatial information using a variety of map formats. For example, a spreadsheet might contain values for the populations and areas of American states, from these values population density could be calculated. A bar chart could then be produced showing the population density of the various states. The states might be listed in alphabetic order, allowing users to identify the value for the state they were interested in. Such a graph would be extremely cluttered and would not be a very effective way of conveying information.

A superior approach to a traditional chart would be to use a mapping program to represent this information on an actual map of the United States. A **choropleth map** (thematic map) displays attribute data, in this case population, associated with relevant spatial units. This presentation (Fig. 6) allows the user to identify immediately that densely populated states are clustered in the north-east of the lower 48 states of the United States, while the states in the center of country have lower population densities. This allows spatial patterns to be identified, similar to the 19th century map in Fig. 1.

Modern GIS software allows the generation of a variety of chart types, which combine the geographic representation with other types of business graphics (Fig. 7).

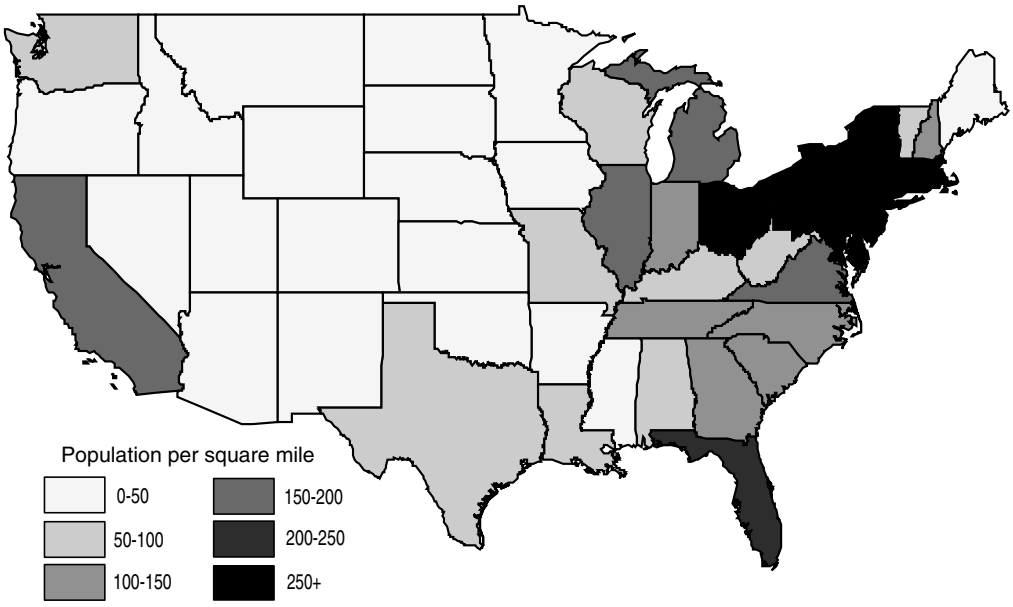


Figure 6 Population density in the lower 48 states of the United States. (Generated from ArcView GIS.)

B. Map Overlay

Information processing can be seen at its most powerful when it allows different sources of data to be synthesized together. A major contribution of database management systems, such as relational database software applications, is their ability to combine information in different data sets. GIS incorporates a spatial database and allows the combination of different spatial data sets. This requires that different types of data be available for the same geographic area and that they be stored in the GIS in a compatible projection. This would allow different types of geographic data to be treated as layers of the same map (Fig. 8).

The different layers may contain polygons (areas), and these may overlap to some extent (Figs. 9 and

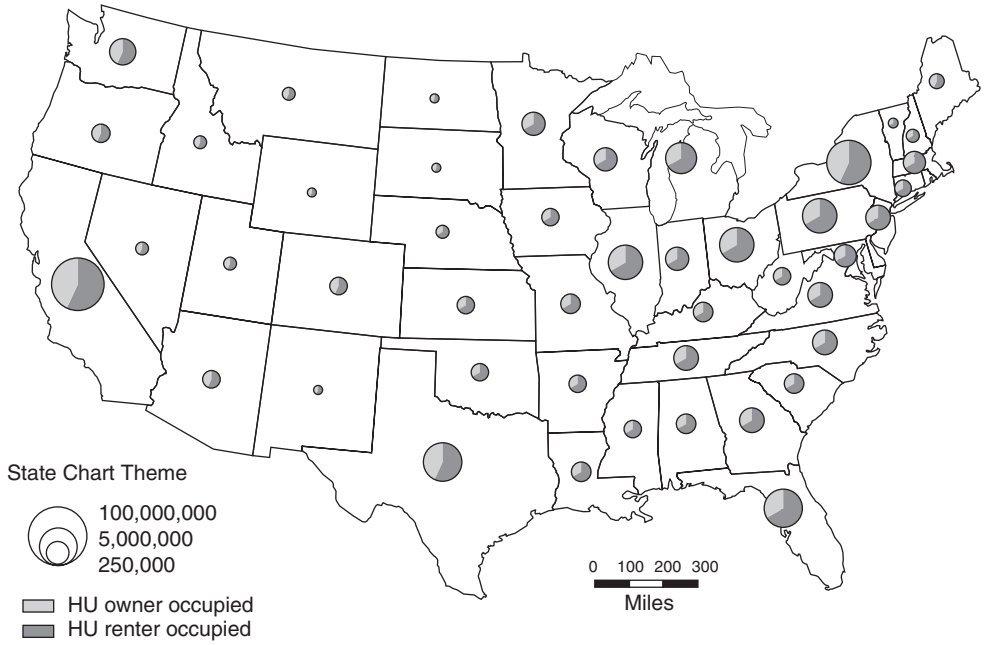


Figure 7 Graph showing the proportions of rented and owner occupied housing in the United States. (Generated using TransCad GIS.)

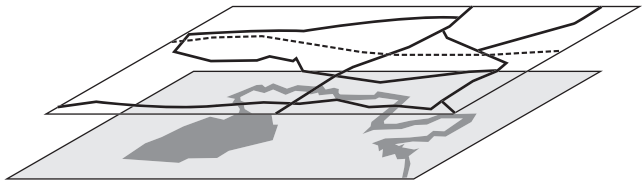


Figure 8 Map layers.

10). For example, one layer might show physical features such as forests, while another layer might show political divisions such as administrative districts and counties. If a person is interested in knowing which administrative regions contained forests (Fig. 10), this can be achieved by using a GIS operation known as **overlay.** Using overlay operations, the GIS software can identify whether spatial features overlap with other spatial objects.

In addition to area overlay, lines or points overlay areas can be identified. For instance, the administrative regions along the path of a river or road could be identified (Fig. 11). This process may not be a straightforward one, as line sections may not end at administrative boundaries.

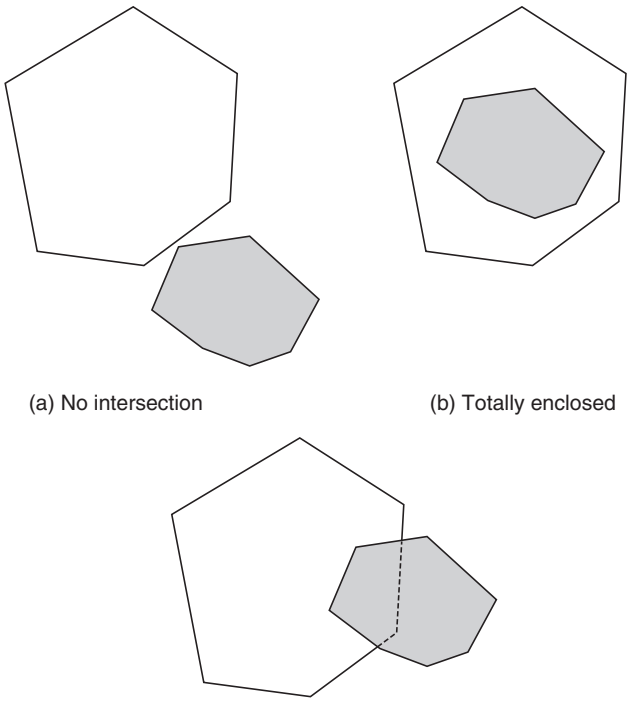


Figure 9 Different types of overlay.

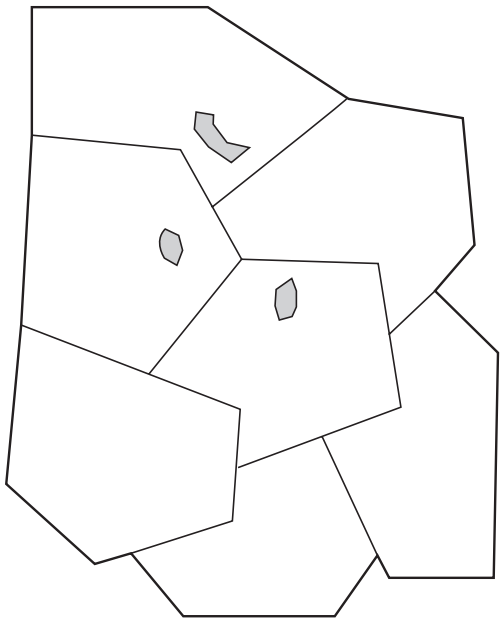


Figure 10 Overlay map showing physical objects and administrative boundaries.

C. Distance Relationships

In dealing with geographic data, it is a common interest to note how near spatial entities are to each other. To allow us to do this, GIS software provides a variety of operations based on distance.

One useful tool is the ability to identify a region within a certain distance from an object on the map; this can be a point, line, or area. The regions generated around spatial objects are known as **buffers** in most GIS software (Fig. 12). Buffers have many practical applications and are widely used in spatial mod-

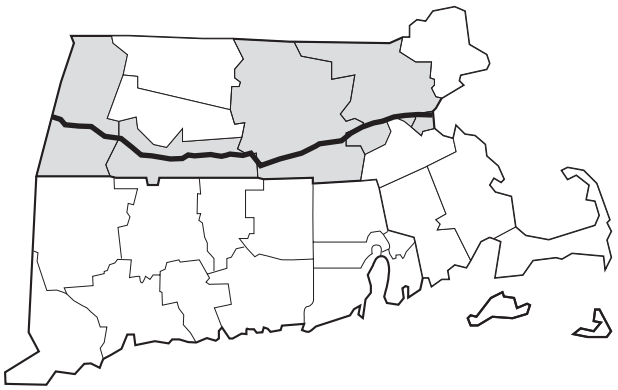


Figure 11 Massachusetts counties along a major highway. (Generated in TransCad GIS.)

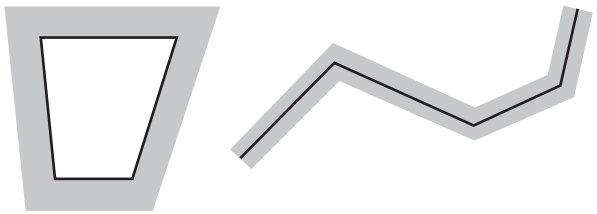


Figure 12 Buffers around spatial objects.

eling. For instance, buffers could be used to identify the regions close to a main road that might be affected by noise. Buffers could also be used to associate delivery locations with routes passing along main roads.

The buffers can be used to generate a new layer on the map and further spatial operations can take place based on this new layer (Fig. 13).

On a road network, distance calculation should involve actual travel paths along the road network, respecting traffic regulations such as one-way streets. Two points that are close in coordinate terms may be quite distant from each other along the road network. Distance calculation on the road network requires a shortest path calculation, and this algorithm is included in many GIS packages. The shortest path may be simply calculated in terms of distance or may be expressed in units of time. In the latter case, a speed must be associated with the network; ideally, each part of the road network can have a different speed reflecting road conditions on that section of road.

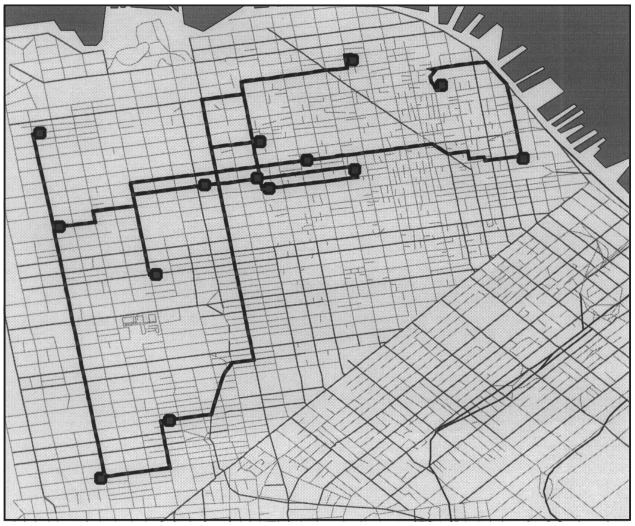


Figure 14 Route on a street network. (Generated from ArcView GIS Network Analyst module.)

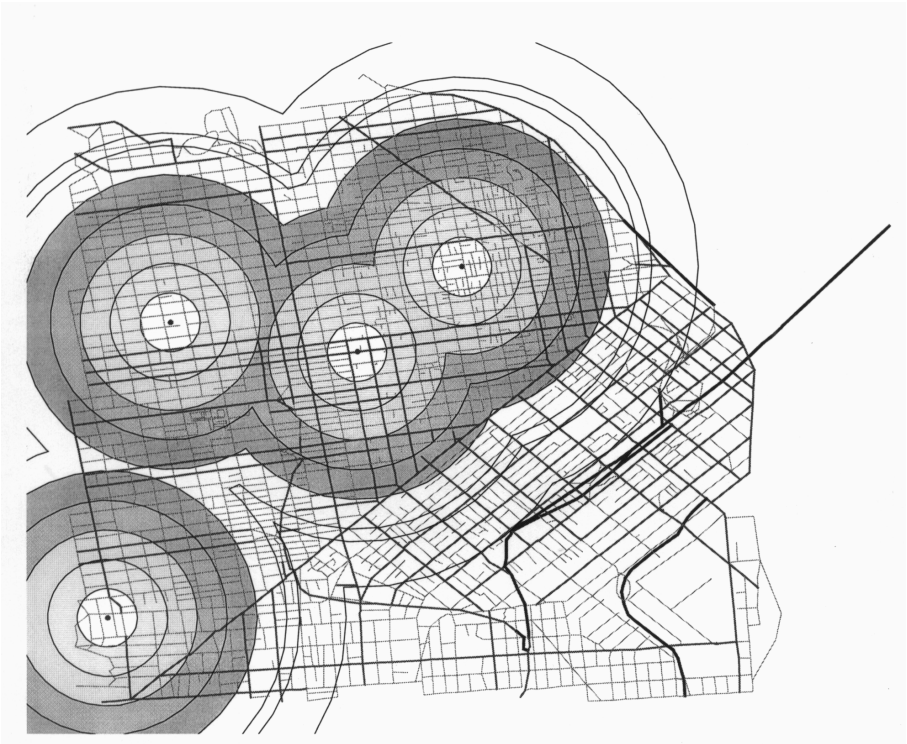


Figure 13 Buffers around points. (Generated from ArcView GIS.)

The identification of a path between two locations can be the starting point for the generation of a complete route visiting a number of locations. With such paths, realistic routes can be generated, as such routes will travel along actual roads and can take account of traffic restrictions recorded in the GIS (Fig. 14).

IV. GEOGRAPHIC INFORMATION SYSTEMS AND INFORMATION SYSTEMS

A. Geographic Information Systems and Decision Support Systems

Given the historically separate development of GIS from the information system field, generally, there is continuing debate on the relationship between GIS and other types of information systems. In particular, many people see a close relationship between GIS and decision support systems (DSS). Geographic data is relevant to many types of problems, including those problem categories where DSS are used. Several DSS applications have used a geographical context for the development of visual interactive techniques and have incorporated a limited GIS-type functionality. Many of the important areas of DSS application overlap with those where GIS can make a contribution. Transport applications have been important in the DSS field and geographic data is widely used here. Marketing applications are also an important area within DSS; GIS can make a major contribution to this type of business application. Within the GIS field there are many references to **spatial decision support systems** (SDSS), although the use of the term DSS is not necessarily based on any great familiarity with the information system field.

Nevertheless, while there is a clear overlap in the applications of GIS and DSS, there is disagreement as to the precise relationship of the two fields. One school of thought is that GIS software can make an important contribution to decision making and therefore GIS is already a form of DSS. This view probably comes most strongly from those fields where the use of geographic data dominates decision making. Another school of thought regards GIS as essentially constituting a spatial database and an interface and procedures to work with that spatial database. This approach suggests that GIS does not have the full model or problem-processing capability for the range of decision support applicable to these problems. This reflects the view that geographic data can have a role to play in a very wide range of activities and that off-the-shelf GIS software cannot have the complete set of models needed for all of these applications. From this perspective, GIS provides a basis for easily building an SDSS, with the integration of additional modeling techniques (Fig. 15).

The use of GIS to build SDSS can make use of new facilities for interaction between software, with techniques such as dynamic data exchange (DDE), object linking (OLE), and open database connectivity (ODBC). These techniques will allow data to pass from the GIS to modeling software that can provide facilities not found in the GIS itself. With the present trend toward an object-oriented future, specialized small applications (or applets) will be available for use as part of a larger package. In this context, the GIS will provide the main interface and database facilities, with applets used for additional modeling or interface requirements. This approach provides a flexibility that allows a wide range of systems to be built that could accommodate a wide range of decisions using both spatial and nonspatial techniques.

B. Geographic Information Systems and Executive Information Systems

Executive information systems (EIS) are used to provide easy access to large volumes of data to address problems of interest to general management in organizations. A further characteristic of EIS is the incorporation of external data sources, many of which are spatially related. These systems emphasize ease of use, and, therefore, the user–system interface is an extremely important component of the system. The importance of spatially related information means that

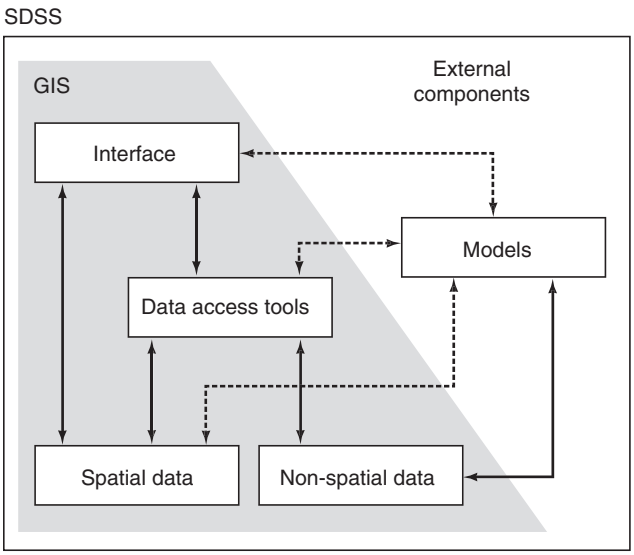


Figure 15 Building an SDSS by integrating models with GIS.

an EIS needs to facilitate spatial query and information retrieval, as well as access to traditional forms of data. A limited map display capability is a recognized feature of EIS-type applications and this presentation is seen as reducing information overload. This trend is made more important by the growing importance of executive support system (ESS) applications that incorporate some additional analysis tools.

Current EIS seek to provide a convenient interface to the large volumes of data made available by transaction processing and routine data collection. If spatial data is to be included in such a system, it must be stored in a form that allows spatial relations to be maintained and must employ systems with an interface that allows spatial queries to be generated. In general, current EIS applications do not exploit the full ability of GIS techniques to facilitate the information needs of top-level management. Future EIS are likely to develop additional capabilities in this direction to build on the large volume of operational spatial data found in many large organizations. For example, many organizations that have large volumes of spatial data, such as utilities or transport companies, are also users of EIS and data warehousing techniques.

C. Geographic Information Systems and the Internet

One of most important recent trends in information systems has been the increasing importance of the Internet and the emergence of electronic commerce. Users increasingly expect to be able to access information from any machine connected to the Internet, and this desire for universal connectivity also includes access to geographic data. However, many users use applications that only require the display of mapping data, rather than fully fledged GIS functionality. If GIS technology is needed, a variety of specialized Internet software tools exist to provide this. Software plug-ins can be used to display maps on Web pages, and appropriate server software exists to support these applications.

In the light of present developments in electronic commerce, new GIS applications will arise. One view might be that the Internet allows equal access to users from all geographic areas, reducing the importance of location. However, location is of importance in the services offered in many electronic commerce applications. Travel is an important electronic commerce application, and GIS data has a clear role to play in Web sites in this field, e.g., tourism sites. Other consumer electronic commerce applications offer goods that must be delivered to the customer. This mode of doing business requires a sophisticated delivery operation, and GIS techniques have an important role to play in the management of this function. In this environment it can be argued that the move to electronic commerce will increase, rather than decrease, the importance of GIS.

D. Geographic Information Systems and Mobile Computing

Mobile computing is an area of IT application that is currently attracting a lot of attention. The rapid growth of mobile telecommunications, especially in Europe and Japan, has meant that information access is no longer restricted to fixed locations. Mobile devices are increasingly becoming data enabled and the advent of technologies such as wireless application protocol (WAP) will allow access to the Internet from any location.

Current mobile technology has a number of important limitations when accessing traditional Internet sites, as mobile devices have small screens and a very limited rate of data transfer. Consequently, information sites that work within these limitations will likely be developed; these sites will be designed specifically for users of WAP or similar mobile technologies. On such sites, GIS has an important role to play in providing customized services for mobile users. There is obvious interest in direction-finding applications that could guide the user to a nearby business or public transport facility. Services such a bus timetables, weather forecasts, and information on traffic congestion are likely to be of interest to the mobile user and GIS technology can contribute to the provision of these services. Future developments will enhance the capabilities of these devices, and mobile users are likely to see the integration of mobile data devices and spatial technologies such as GPS. This will allow the location of the mobile user to be easily identified and will therefore provide the basis for a service customized to that location.

V. CONCLUSIONS

Geographic Information Science has developed many information processing techniques of interest to users of information systems. These techniques have been incorporated in geographic-based systems, known as GIS. GIS developments have exploited changes in technology to provide many useful information representations that have been incorporated into decision-making tools. This development of GIS-based DSS applications parallels previous developments in information systems, where systems have moved toward a greater decision-making emphasis. In the future there will be greater integration of GIS with other forms of information systems, as existing systems are enhanced

to allow the incorporation of spatial data. This process will extend the use of spatial data from those who currently use such data, generally those with training in geographic disciplines. With the incorporation of spatial techniques in a broader range of systems, a much wider user community of GIS will emerge and the GIS technology will reach its full maturity. Against this background a strong case can be made for considering GIS as a new reference discipline for information systems, which would be of potential importance to many of the applications discussed in this book.

SEE ALSO THE FOLLOWING ARTICLES

Decision Support Systems • Executive Information Systems • Geographic Information Systems in Real Estate • Mobile and Wireless Networks • Multimedia

BIBLIOGRAPHY

Burrough, P. A., and McDonnell, R. (1998). *Principles of geographical information systems*. New York: Oxford Univ. Press.

Chrisman, N. (1997). *Exploring geographic information systems*. New York: Wiley.

Clarke, K. C. (1997). *Getting started with geographic information systems*. Upper Saddle River, NJ: Prentice Hall.

Davis, D. E. (1999). *GIS for everyone*. Redlands, CA: Environmental Systems Research Institute, Inc.

Heywood, I., Cornelius, S., *et al.* (1999). *An introduction to geographical information systems*. New York: Addison–Wesley Longman.

Lang, L. (1999). *Transportation GIS*. Redlands, CA: Environmental Systems Research.

Longley, P., Goodchild, M., *et al.* (1999). *Geographical information systems: Principles, techniques, and applications*. New York: Wiley.

McDonnell, R., and Kemp, K. (1995). *International GIS dictionary*. Cambridge, UK: Geoinformation International.

Web Resources

Directions Magazine—http://www.directionsmag.com/

European Umbrella Organisation for Geographic Information—http://www.eurogi.org/

Geoplace (GIS related magazines)—http://www.geoplace.com/

U.S. National Center for Geographic Information and Analysis—http://www.ncgia.org/

Geographic Information Systems in Real Estate

Robert G. Fletcher
California State University, Bakersfield

GLOSSARY

Consolidated Metropolitan Statistical Area (CMSA) Area that has a population of 1 million or more within an urban area and contains Primary Metropolitan Statistical Areas that are equivalent to MSAs but not "free-standing" metropolitan areas.

e-commerce Utilizes the Internet to provide product and service information to potential buyers and sellers.

geographic information system (GIS) An information system that is designed to utilize data referenced by geographic or spatial coordinates.

geographic information systems in business GIS applications to business problems in areas such as accounting, finance, marketing, and real estate.

geographic information systems in government GIS databases utilized by government entities to plan and control spatial activities, and to provide to the private sector for their use.

Metropolitan Statistical Area (MSA) County containing a central city (or twin cities) or an urban area with a population of 50,000 or more with adjacent counties that are socially and economically related to the central-city county.

remote sensing Generally refers to data collection about land, water, or an object by instruments carried aboard satellites or aircraft.

Standard Industrial Classification (SIC) codes Defined by the U.S. Department of Labor utilizing census data to classify industries on the basis of what the companies produce.

I. INTRODUCTION

Geographic information systems (GIS) technology provides an opportunity to improve real estate analysis by linking geography (location) with specific types of data (demographic, consumer profiles, competition, transport networks and travel routes, distance, etc.). There are a number of different property types and people interested in real estate that might benefit from GIS technology. These types of properties include land, industrial, residential, office, retail and shopping centers, and entertainment and recreational properties. Public and not-for-profit land uses such as hospitals, public buildings, and churches also may benefit from GIS analyses. People interested in buying or selling real estate, as well as investors, may also want to use GIS applications.

A variety of types of real estate analyses are possible including but not limited to the following: site location, appraisal applications, selling real estate, financial institution lending practices, compliance with environmental regulations, and national–international applications. Each of these types of analyses is discussed in this article.

Additional uses for GIS in real estate include purchasing real estate, investing in real estate, looking for agglomeration effects in real estate usage patterns, and other applications that are either similar to the above applications or beyond the scope of this article. Because GIS relies heavily on geography, a discussion of common errors in applying geographical analysis is provided.

II. SITE LOCATION ANALYSIS

One of the major uses of GIS databases is for site location analysis. Often, site location analysis is defined as a use looking for a site. For instance, an organization wishes to build a manufacturing plant and is looking for a site for the plant. The organization examines and compares sites in various countries, cities, or rural areas and chooses the best site at least partially based on the results of GIS analysis.

Several products for site analysis are on the market. Examples of GIS software programs for site selection include AnySite, GisPlus, MapInfo, ArcView, SPANS MAP, Atlas GIS, BusinessMap, Scan/US, SpartaSite, and Tactician. Additional software and data vendors are constantly coming into the market. Various sites on the Internet can keep those who are interested abreast of changes in GIS; examples include http://www. directionmag.com, http://www. gisworld.com, http://www.esri.com, and http://www. directionsmag.com.

In any site location analysis, an understanding of who the customers are is a prerequisite to good decision making. GIS data from various sources are available to help in understanding customer characteristics. Syndicated data, collected by research houses or industry consortiums through national surveys, provide information about customer awareness of the real or proposed real estate projects and services, customer preferences regarding the projects or types of projects, customer demographics, lifestyles, product usage, etc. A company or individual can collect primary data and build its own GIS system, although this is a costly and time-consuming process. Finally, a company or individual can utilize its point-of-sale (POS) data to identify and locate people who are actually buying products or services. This information can be modeled into a GIS database to assist in determining the best site locations based on the identification of characteristics most likely to produce positive transactions in terms of building a viable customer base.

Some of the important GIS applications are discussed next.

A. General Location Factors

1. National

GIS databases provide opportunities to locate regional real estate markets for new residential, retail, industrial, and office developments. Major factors that could be included in the databases are demographic, geographic (both locational and environmental), climatic, employment, and types of employment, etc. Quite often the analysis will start with a national breakdown by Metropolitan Statistical Areas (MSAs). Analyses of MSAs are used because of the rich statistical databases available mostly from federal, state, county, and local governments. Alternatively, Consolidated Metropolitan Statistical Areas (CMSAs) can be used to locate prime real estate investment areas throughout the United States; again, rich databases provided by government entities support the analyses of CMSAs. These data can be combined with input/output analyses to identify prime regions with high potential growth. A number of real estate analysts and appraisers prefer this type of model in comparison to econometric modeling because input/output analyses are not influenced by structural change within the economy.

Econometric modeling suffers from the assumption that the structure of the key industries will not change in the future. In many instances, however, regions, states, counties, and local governments are trying to change the structure of the entity's economic or social dimension in order to improve the economic base of the locale or the social structure of the area.

2. Regional

The use of GIS systems is especially helpful at the regional level. GIS databases can identify relevant market area(s) within the region. Geographical factors such as topography can be overlaid with other graphics to indicate prime areas in which to locate new developments. The location of submarkets and facilities is also possible with GIS. For example, the local metropolitan communities within a region can be identified spatially by using Standard Industrial Classification (SIC) codes. These codes are defined by the Department of Labor and provide real estate analysts with a means of classifying local economic activity into basic and service sectors. Thus, a submarket for a specific industry or group of industries can be identified within a region.

Another example would be to locate principal transport facilities and identify major transportation patterns within the region. Major shopping areas and educational facilities can be identified along with community facilities such as recreational areas. The database can be overlaid with other graphics to show the direction of growth within the region. The patterns evolving also can identify the types of growth occurring within a region; i.e. manufacturing in the northern part of a county, services in the central section, residential in the southern part, and retailing/

warehousing in the western section. Finally, location and plans for public utilities, sewers, water lines, etc. can be shown via GIS.

Additional types of regional statistics available by syndicated providers are (1) agriculture, (2) demographics, (3) economic statistics, (4) industry locations, (5) labor force data, (6) transport networks (roads and highways), and (7) unemployment rates. Types of county economic and demographic data include (1) household data by income, (2) personal income by source, (3) per capita income, (4) population by age, race and sex, (5) employment by industry, (6) household income, and (7) employment by industry.

3. Local

Boundary files provide real estate analysts with the opportunity to conduct spatial and demographic analysis locally. One type of boundary file is by zip code. Demographic data by zip code can be overlaid to a map. This information provides population characteristics such as age distribution, number of households, income levels, household size, etc. Another type of boundary file that could provide similar demographic information is to use census enumeration districts. Data at this level could help identify a shopping center's trading area through the use of a gravity model. For example, looking at population and income figures for a specific geographical area may indicate, in general, that the location under study may support a community shopping center. However, adding data about the location of already existing centers may indicate that competitively the population and income base are not large enough to support a new center.

Some types of GIS data include (1) consumer profiles, (2) consumer lifestyles, (3) media behavior, (4) product usage, and (5) consumer demand. These data are usually divided into geographic boundaries such as zip codes or census enumeration districts.

When looking at sites, one must consider future residential developments or planned manufacturing, office, retail, and other planned centers that will be coming on line. City or county jurisdictions may have digitized maps and hardcopy aerial photographs that can provide valuable information for site selection. Some types of information that may be available from city governments are (1) approved development projects, including building permits, (2) current and projected land uses, (3) public transportation locations and traffic counts, (5) hospitals, (6) fire and police, (7) schools, and (8) environmental constraint maps showing endangered species habitat, archaeological sites, noise contours, etc.

Other types of information may be available from local utilities. Most major utilities have GIS databases to help answer questions about a site's water and sewer, electricity, natural gas, and telecommunications services. In certain parts of the country, California, for example, the location and availability of water are vital components for land development. Investors may want to know if sewer and power hookups are available, planned, or installed at the owners' expense before making a decision to locate in an area.

B. Residential Developments

Real estate developers can utilize GIS databases to identify locations for residential development. At the national or regional level, county economic and demographic data are available to provide employment by industry, household data by income, population by age and sex, personal income by source, and per capita income. The information can be used to determine if the location has potential for development and, if so, what kind of development may be warranted, such as first-time housing, buy-up homes, or top of the market estates.

In-migration information may be helpful in addition to data about the existing population base. Does the location being considered attract buyers from other locations or states? If so, what are the characteristics of these buyers? An illustration would include the number of retirees from southern California who were attracted to the Las Vegas area. They could sell their residences in California for a profit, buy property in the Las Vegas area, and still have money left over from the sale to invest in stocks, bonds, travel, or other activities.

Additional information about transport networks (roads and highways), unemployment rates, and geographic boundaries assists developers in pinpointing potential residential sites. Around the Las Vegas area, the availability of water and proper treatment of sewage is also a determinant for the types of development that can occur.

C. Office Buildings

When trying to decide the optimum location of an office building, important factors to consider include (1) transportation patterns, especially distance and congestion; (2) availability of employee conveniences

such as restaurants, banks, and shopping near the proposed site; (3) potential new competitors; (4) new complementary buildings; (5) local absorption rates of office space; (6) fiber optic/cable connections; and (7) vacancies in existing buildings. A GIS system can be developed to provide information about the above by overlaying the relevant data onto a map of the subject area.

The agglomeration effect can be important to the location of office facilities. For instance, lawyers tend to locate around the courts, regulatory agencies, and other lawyers for convenience in carrying out their activities. Health care laboratory services may be looking for sites close to doctors' offices, outpatient facilities, and hospitals. GIS systems can pinpoint the location of other types of offices beneficial to the use looking for a site.

D. Retail Properties

One major application of GIS systems is for the location of shopping centers. A variety of shopping centers exists and require somewhat different data and spatial considerations. For example, a neighborhood shopping center with a supermarket as the anchor tenant generally has a trading area of approximately a 1-mile radius assuming no topographic barriers. Important data to the decision process are demographic information, income levels, lifestyles, competition, major transportation routes, and transportation counts. In contrast, a regional shopping center location analysis can have a trading area of up to 100 miles. Similar information used in neighborhood shopping center analysis would need to be collected and analyzed for this larger trading area. GISs can facilitate this type of hierarchical analysis. Planned centers and growth for the area must be examined along with the existing facilities.

E. Industrial and Special-Purpose Properties

Types of property in this area include manufacturing plants, office-warehouse complexes, motels, hotels, hospitals, etc. Location is an important factor for these types of properties and GIS databases provide a vehicle to ensure that the proper location is found. In general, industrial firms locate near other industrial companies, especially suppliers or near the market for their products. There also may be an agglomeration or clustering effect when a certain industry locates in an area. This agglomeration effect leads suppliers and other related industrial activities to locate in the area.

Another example could be that motels and hotels can be focused around a transportation hub like an airport or can be located close to restaurants, night clubs, movies, shopping centers, tourist recreation areas, etc. GIS databases can provide information about accessibility (roads, railroads, ports, etc.), amenities (utilities, sewage, and power sources), types of firms located in the surrounding area, and distance minimization solutions for alternative site locations.

To further illustrate the potential applications of GISs, the following discussion focuses on important factors that influence the choice of industrial location. For industrial activities that are sensitive to transport costs, industrial locations are generally at the source of the raw materials or near the market. Those industrial activities that lead to weight-losing processes usually locate near raw materials if transport costs are important. Alternatively, industrial processes that are weight gaining most often locate near their primary market(s). GIS databases can provide information about transport routes, location of major markets, distance minimization solutions, etc.

Other types of orientations affect industrial location decisions. Labor-intensive processes generally seek low wage and required skill areas. Power-intensive processes want cheap power locations. Other types of orientations exist for industrial location decisions. In all instances, GIS can provide the relevant information to assist in making good location decisions. For instance, GIS can help avoid environmentally sensitive areas, e.g., wetlands, and overlays of zoning, etc.

F. Distribution Centers

GIS databases play an important role in identifying the prime location for retail outlets (sometimes referred to as "brick-and-mortar" outlets). Historically, the location of retail outlets was dictated by the location of customer markets and manufacturing plants. The advent of e-commerce (electronic commerce) has created inroads in the traditional way to locate retail outlets and has also made the location of distribution centers increasingly important to retail operations. For example, some retailers such as Toys 'R Us include its brick-and-mortar operations in the e-commerce process by allowing Web site customers to avoid shipping charges by picking up their orders and returning merchandise at their local stores. Many giant retailers have problems because no central or

regional distribution centers are in place across the United States. For many of these companies the use of suppliers and company stores for distribution has proven expensive and cumbersome when trying to fill orders made online.

Some e-commerce companies sell only via the Internet and have no physical retail outlets. Giving customers the opportunity to purchase products from a central cyberspace Web site provides convenience to buyers but also introduces an additional concern for businesses selling online. The use of GIS databases is becoming increasingly important to site locations of distribution centers for online types of companies and the traditional brick-and-mortar outlets. Regional distribution centers can be optimally located using point-of-sale (POS) information, demographic data, spatial maps, customer profiles, location of online buyers, and location of present warehouses and transportation modes used. With e-commerce, the entire distribution system may need to be redesigned to ensure order fulfillment (customer satisfaction) capability on the part of the seller.

Another example centers on port facilities where transportation and other related services may be located. For instance, a sea port may need dock facilities for specific types of ships and can benefit from having freight forwarders, customs officials, agents/brokers, financing institutions, warehouses, and connections with trucking, rail, and other transportation, etc., located in the same area. If the sea port is near an airport or connects with rail or truck transportation so that intermodal transportation can be arranged (sea/air transportation, sea/truck transportation, etc.), this can be a great advantage. Singapore's sea and air facilities are an excellent example of this type of agglomeration. GIS can be used to identify the degree and type of agglomeration activities that are located near the site being considered for port development.

III. APPRAISAL APPLICATIONS

Numerous appraisal applications are possible with GIS systems. For example, MSA analysis can be the starting place to define market areas. By double clicking a mouse over a market area, an even more detailed market area map can be shown in many GIS systems. By double clicking again on a submarket area, an even more detailed map can be shown on the computer screen. This map may show the locations of office, shopping centers, industrial properties, etc. by a number on the map. Double clicking on a number provides detailed property records such as historical rents and absorption rates, land and building information, and even a bit-mapped photograph of the building.

Another example is the use of employment data by SIC codes to allow econometric modeling at either the MSA or submarket areas. Alternatively, input/output models can be employed if the analyst believes that structural changes in the local economy have recently occurred or are anticipated in the future.

An additional GIS application involves analyzing a market trade area for an underutilized or proposed shopping center. A gravity model such as Huff's probability formulation (a refinement of Reilly's law of retail gravitation) can be employed to determine the trading area of a shopping center. The gravity model can predict the probability of attracting people from different distances. This information can be multiplied by the dollar expenditure per capita for the type of merchandise sold at the shopping center and then summed to estimate total dollar sales volume. Appraisers can use this information to compute economic rents for the subject property.

Finally, appraisers can use a market area map overlaid with other graphics to indicate the part of the local population that is inaccessible because of topographic or other factors. A variety of new software is becoming available to allow more in-depth analysis by appraisers.

IV. SELLING REAL ESTATE

Real estate brokerage firms and sales agents may benefit from the use of GIS databases when assisting clients in finding properties to purchase. Specific types of information that would assist real estate agents to sell properties are discussed next.

A. Locating Properties for Sale in Specific Geographical Locations

Real estate agents can show property to potential buyers without leaving the real estate office. Buyers specify the price range, the square footage, number of stories, bedrooms, bathrooms, and preferred neighborhood(s). A GIS database takes these characteristics and identifies neighborhoods and properties for sale. Real estate agents take the information from the GIS analysis and show their clients photographs or videos of the properties. The buyers select the residences to visit without leaving the brokerage office.

This can also be done for industrial, office, and retail properties.

B. Provide Select Demographic Data

In some instances, the potential buyers may not know what quality neighborhood to select or which neighborhoods would have the type of people as residents that the buyers prefer as neighbors. Real estate agents ask the buyers to specify selected demographic factors that could help identify potential neighborhoods. Possible demographic data include income levels, age cohorts (young families or retirees), home prices, household size, occupation (professional, white collar, blue collar, etc). A GIS database matching demographic data and neighborhoods assists agents in selecting appropriate areas. Specific properties can then be identified as discussed in the preceding section, and other properties can be eliminated because of neighborhood factors.

C. Comparing Local Residential Prices

Real estate agents benefit from GIS databases as well. Matching home prices with geographical locations, generally neighborhoods, allows sales agents to steer their clients to preferred locations. Additionally, agents can show potential buyers homes with similar price ranges in different neighborhoods.

D. Identifying Important Local Amenities

A GIS database also provides information about important local amenities. Real estate agents are able to show potential clients the locations of schools, shopping centers, churches, restaurants, recreational and environmental amenities, etc. In many cases, the neighborhood amenities can be as important to homebuyers as the homes themselves. Buyers then make choices about the homes they want to visit.

E. Showing Transportation Networks

A GIS database also lets real estate agents show clients the major thoroughfares, freeways, and streets surrounding each property. The importance of this and the above processes is that real estate agents conduct the property searches for the buyers without leaving their real estate offices. Potential real estate buyers choose the residences to visit with considerable savings in search time for the real estate agents and clients because they do not have to do as much driving around looking at numerous homes.

V. FINANCIAL INSTITUTION LENDING PRACTICES

Financial institutions that make real estate loans must be aware of geographic lending patterns. The Community Reinvestment Act (CRA) and the Home Mortgage Disclosure Act require banks to have a thorough geographic understanding of their loan placement. The CRA imposes 12 banking factors on bank real estate loan making. Some of these factors can be handled by utilizing GIS databases. For example, factors D and F compare deposits to total loans in the bank's trade area to determine if discriminatory or other illegal credit practices have occurred. Factor E mandates banks to detail the geographic distribution of its credit applications, extensions, and credit denials. Factor J considers a bank's mortgage loan demand within its service area to the origination of residential mortgages, housing rehabilitation, and home improvement loans. Banks must meet a minimum score on these 12 assessment factors. If banks fall below the minimum, regulatory agencies can prohibit the banks from various activities, including mergers and acquisitions. Consequently, banks have a major interest in the geographic location of their banking activities to ensure regulatory compliance. GIS database systems can directly assist financial institutions to show their compliance.

VI. COMPLIANCE WITH ENVIRONMENTAL REGULATIONS

A. Developers and Investors

Real estate developers and investors need to consider a site's potential environmental hazards. Potential hazards include earthquakes, floods, hurricanes, wildfires, and toxic cleanup. GIS databases can provide, in many instances, historic data about a site's environmental record, including, but not limited to, permit violations and citations. Additional data may be available for surrounding sites as well. Maps are also commercially available indicating weather perils, environmental hazard information, and natural disaster historic data.

B. Banks

Banks also are utilizing internal GIS data to evaluate environmental risk information and assessment needs. The GIS system allows the banks to establish due diligence procedures. The systems provide a means for banks to evaluate the environmental hazards for a particular site and for surrounding properties. For banks, GIS-based environmental tools allow the banks to gain critical insights during the credit process.

C. Regulatory Compliance Information

Most large and some small cities and counties use GIS systems to encode their zoning maps. Quite often these data can be incorporated with other data to provide regulatory compliance information. Some important types of environmental hazards include endangered species habitat, wetlands, scenic easements, noise contours, and archaeological sites.

VII. NATIONAL–INTERNATIONAL REAL ESTATE

GIS databases can be used at the national and international levels to provide valuable information about geographical constraints, socioeconomic characteristics, environmental factors, and other important information at the national and international levels. There are a variety of possible applications. Some examples include large luxury apartment complexes, high-rise office buildings, industrial sites and parks, science parks, warehouse parks, shopping centers, and flagship hotels. Within the United States, prospective international investors could use GIS information to screen out investment opportunities. Additionally, large institutional investors such as real estate investment trusts, pension funds, life insurance companies, mortgage bankers, and international banks can utilize GIS databases to evaluate real estate investments in their portfolios.

Internationally, site location and other types of analyses discussed above are possible in a number of countries with GIS databases. Additionally, GIS applications could be used in tourism research, retail locations, industrial locations, high-tech estates, and international airport siting. However, in certain regions the use of GIS analyses is problematic. For example, many developing countries in Africa lack data to support a GIS system. Poor statistical data or no data at all limit the use of GIS techniques in these types of countries. Other regions around the globe have developing economies with similar data problems. Consequently, GIS analyses are probably confined to developed countries with ample databases and sophisticated mapping capabilities. However, future development of remote sensing technology may assist in overcoming some of these limitations, especially via the use of satellites.

One country with a national GIS system is Singapore. The Urban Redevelopment Agency has mapped Singapore and provides demographic and other relevant data to real estate developers, lenders, etc. Additionally, the Singapore government utilizes this information for urban planning. Because of its small geographic size, Singapore has adopted a national planning process to ensure that there are green belts, efficient transport routes, adequate industrial sites and parks, office building sites, and residential areas serving all developed areas and planned developments.

VIII. COMMON ERRORS IN THE APPLICATION OF GISs

Grant I. Thrall has criticized the use of GIS databases by nongeographers. Thrall's article lists and explains a number of pitfalls facing novice users of GIS systems. Thrall divides these problem into nine areas: (1) geographic scale, (2) discrete versus continuous space; (3) relative versus absolute location measurement, (4) spatial autocorrelation, (5) lack of geographic effects, (6) distance measurements, (7) geocoding, (8) absolute versus proportional measurements, and (8) muddy boots. A brief discussion of each of these is provided based on Thrall's article.

A. Geographic Scale

Real estate analysts creating their own data may obtain invalid measurements if consistent summary measurement procedures are not employed. For example, if the analysis is to be made at the census tract level and census information is to be used in conjunction with new data collected by the real estate analyst, the new data set must have a spatially stratified sample with a statistically appropriate number of observations that is consistent with the census tract data. For example, a single observation cannot be used to represent a census tract since it is not consistent with census statistical sampling procedures.

Another part of the geographic scale problem is geographic position. Essentially, geographic boundaries are established and provide a set of data that are

unique. For example, census tract boundaries are established and demographic data are collected for each tract. If these geographic boundaries (tracts) are changed, a different set of data and conclusions may arise because of the change in boundaries. Real estate analysts need to be aware of this potential problem.

B. Discrete versus Continuous Space

Space can be discrete or continuous depending on what is being measured. In general, altitude is continuous when measured in small increments from location to location, whereas land values may differ substantially because land values are spatially discrete. Surface maps are generally based on spatial interpolation algorithms that use spatially continuous measurements. Because real estate measurements are usually discrete, the real estate analyst must be careful in interpreting the surface information. Thrall recommends that the "surface model be depicted along with a display of the location of the points used in the surface model calculation. The farther the interpolated surface is from an actual observed point, the greater the likelihood that the reported surface is in error."

C. Absolute versus Relative Location

Earth positional coordinates such as latitude and longitude measure absolute spatial location. Relative spatial location utilizes distance between locations and does not have to be assigned earth positional coordinates. In many real estate types of analysis, relative spatial location measurements are used to locate the distance from specific phenomena such as the location of a house from a shopping center. A problem with using relative spatial location measurements is that they provide no information about the direction (east or west, for example) from the site. This information can lead to erroneous conclusions about real estate values since both distance and direction influence property values.

D. Spatial Autocorrelation

Real estate GIS analysis suffers from possible autocorrelation problems just as in time series analysis. In general, properties are more likely to be similar the closer their locations. In most real estate studies, it is recommended that standard statistical procedures be employed to test for spatial autocorrelation and, if found, corrected using standard statistical procedures.

E. Lack of Geographic Effect

Quite often an externality such as noise or pollution may adversely affect local property values. In these instances, one can identify a geographic effect to the impact on property values since the real estate happens to be located in the wrong place. There are, however, instances where an externality may not affect real estate values if a phenomenon covers a large enough geographic area. For example, smog could be considered a negative factor on property values. However, Denver and Los Angeles, with large amounts of smog, have some of the highest property values in the United States. When conducting GIS analyses the real estate analysts need to be certain that there is a geographic effect to changes in property values.

F. Distance Measurements

Two types of distance measurements often used are the "as-the-crow-flies" and Manhattan distance methods. The first method simply measures the straight-line distance between two points disregarding any information about street or roadway patterns. The second method reduces the directional bias of the first method by using a right triangle measurement of the two sides. The Manhattan method accounts for the road patterns and topographical barriers that may bias the "as-the-crow-flies" approach. While potentially superior, the Manhattan distance method may be inappropriate for real estate analysis as travel distance increases. Additionally, distance measurements may be more difficult when dealing with newly built subdivisions where street data are sparse or nonexistent.

G. Absolute versus Proportionate Measurements

When using geographic boundaries such as census tracts the real estate analyst may want to use proportionate data rather than absolute numbers, i.e., per capita income, average income, average land values, etc. The use of absolute values such as total population within a census tract is not advisable because of the lack of uniformity between census tract defini-

tions. These differences could arise because of variations in population, geographic features, or other factors that arise because of the arbitrary assignment of census tract boundaries.

H. Geocoding

Geocoding involves assigning latitude and longitude coordinates by matching address information to the U.S. Census TIGER/line data files. A potential bias may arise when applying this procedure to the suburban periphery or rural areas because address and street information may not exist or is inaccurate. Geocoding provides more accurate information for older neighborhoods, especially those areas near the centers of cities, because street address data are readily available and more accurate.

I. Muddy Boots or "Ground Truthing"

This problem essentially revolves around the possibility that the real estate computational analysis (GIS, regressions, input/output analyses, etc.) may be in error. Simply using a GIS database to analyze a particular area is not necessarily sufficient because the data may not be representative of the geographical area under study. The best method is to verify that the analysis makes sense by visiting the location of the study.

IX. CONCLUSIONS

There are a variety of GIS real estate applications. These applications can focus on finding the best site location for different types of land uses such as residential, office, and industrial. Appraisers can utilize GIS databases to assist in the valuation of properties. Real estate agents may employ GIS data to more efficiently sell real estate properties. Financial institutions can evaluate their lending practices to ensure regulatory compliance. Additionally, financial institutions can employ GIS databases to identify environmental hazards and to evaluate compliance with environmental regulations. Finally, international real estate applications are possible when there are no data limitations.

Major universities often have centers, services, and students that are proficient in providing GIS and remote sensing services at a relatively low cost.

When conducting GIS analyses, real estate analysts must be aware of pitfalls that may bias their study. These pitfalls are correctable if the analysts have a good understanding of geographical principles and procedures.

SEE ALSO THE FOLLOWING ARTICLES

Electronic Commerce • Geographic Information Systems • Sales

BIBLIOGRAPHY

Albers, R. J., and Bible, D. S. (1992). Geographic information systems: Applications for the study of real estate. *Appraisal Journal,* Vol. 60, No. 4, 483–492.

Bible, D. S., and Hsieh, C.-H. (1996). Applications of geographic information systems for the analysis of apartment rents. *Journal of Real Estate Research,* Vol. 12, No. 1, 70–88.

Bidgoli, H. (May–June 1995). Geographic information systems: A new strategic tool for the 90's and beyond. *Journal of Systems Management,* 24–27, 66–67.

Marks, A. P., Stanley, C., and Thrall, G. I. (1994). Criteria and definitions for the evaluation of geographic information systems software for real estate analysis. *Journal of Real Estate Literature,* Vol. 2, No. 2, 227–241.

Murphy, L. (February 1997). Why aren't business schools teaching business geographics? *Business Geographics,* Vol. 5, No. 2, 24–26.

Rodriguez, M., Sirmans, C. F., and Marks, A. P. (1995). Using geographical information systems to improve real estate analysis. *The Journal of Real Estate Research,* Vol. 10, No. 2, 163–173.

Star, J., and Estes, J. (1990). Geographic information systems: An introduction, Upper Saddle River, NJ: Prentice-Hall.

Thrall, G. I. (January 1998). "Common geographic errors of real estate analysis. *Journal of Real Estate Literature,* Vol. 6, No. 1, 45–54.

Thrall, G. I., and Marks, A. P. (1993). Functional requirements of a geographic information system for performing real estate research and analysis. *Journal of Real Estate Literature,* Vol. 1, No. 1, 49–61.

Weber, B. R. (1990). Applications of geographic information systems to real estate market analysis and appraisal. *Appraisal Journal,* Vol. 58, No. 1, 127–132.

Wofford, L. E., and Thrall, G. I. (1997). Real estate problem solving and geographic information systems: A stage model of reasoning. *Journal of Real Estate Literature,* Vol. 5, No. 2, 177–201.

Global Information Systems

Magid Igbaria
Claremont Graduate University

Murugan Anandarajan
Drexel University

Charlie Chien-Hung Chen
Claremont Graduate University

GLOSSARY

culture It is the collective programming of the mind that distinguishes the members of one group or category of people from another. Culture is learned, not inherited. It derives from one's social environment, not from one's genes.

cultural manifestation Cultural differences manifest themselves in symbols, heroes, rituals, and values. Symbols represent the most superficial and value the deepest manifestations of culture, with heroes and rituals in between.

cultural relativism Cultural relativism affirms that one culture has no absolute criteria for judging the activities of another culture as "low" or "noble." However, every culture can and should apply such judgment to its own activities, because its members are actors as well as observers.

globalization The watchword that describes the need for companies and their employees, if they are to prosper, to treat the world as their stage.

heroes Persons, alive or dead, real or imaginary, who possess characteristics which are highly prized in a culture, and who thus serve as models for behaviors.

international business Consists of transactions that are devised and carried out across national borders to satisfy the objectives of individuals and organizations.

internationalization vs globalization Internationalization of business . . . is a concept of an action in which nationality is strongly in people's consciousness. . . . Globalization, by contrast, looks at the whole world as being nationless and borderless. Goods, capital, and people have to be moving freely.

international political economy (IPE) The study of the interplay of economics and politics in the world area. For some, it refers primarily to the study of the political basis of economic actions, the ways in which government politics affect market operations. For others, the principal preoccupation is the economic basis of political action, the ways in which economic forces mold government policies. The two forces are in a sense complementary, for politics and markets are in a constant state of mutual interaction.

multinational enterprise (MNE) Synonymous with the terms multinational company (MNC), global, world, transnational, international, supernational, and superanational corporation. The United Nations defines MNC or MNE as "enterprises which own or control production or service facilities outside the country in which they are based."

nation vs country (nation state) A nation is a "social group" that shares a common identity, history, language, set of values and beliefs, institutions, and a sense of territory. A nation state is synonymous with a "country." A country must be a "sovereign entity" having "rights to determine their own national objectives and to decide how they will achieve them." The nation state needs to combine both elements of culture (nation) and sovereignty. Other terms for the nation state are country and government. The major role of a country is to exercise three major powers—political, economic, and military—to

safeguard its people's best economic interest and security.

rituals Collective activities, technically superfluous in reaching desired ends, but which, within a culture, are considered as socially essential: they are therefore carried out for their own sake.

sociotechnical design of the information system IT is one of five major organizational components that are being influenced by the external environment, social, economic, and political factors. IT has the organizational and societal impacts to other internal components and external factors, and vice versa. When designing IS, all the interrelated factors have to be taken into consideration at the same time.

symbols Words, gestures, pictures, or objects that carry a particular meaning that is only recognized by those who share the culture.

values Among the first things children learn—not consciously, but implicitly. Values were acquired so early in our lives; many values remain unconscious to those who hold them.

World Trade Organization (WTO) In 1995, it replaced the international organization General Agreement on Tariffs and Trade (GATT), which was founded in 1947, to administer international trade and investment accords. The accords have brought major changes to the world trade and investment environment. They will gradually reduce governmental subsidies to industries and will convert nontariff barriers into more transparent barriers. The major contributions of WTO are to improve trade and investment flows around the world.

GLOBAL INFORMATION SYSTEMS (GIS) is the study of the interplay of information systems and globalization trend of the world. While many factors contribute to the globalization trend, GIS professionals, particularly, consider information technology among others as a major driver for globalization and attempt to investigate its relationship with other globalization enablers. This article proposes a generic framework—infrastructural, operational, and organizational components—to help readers understand multiple dimensions of the GIS domain.

I. INTRODUCTION

The discipline of Global Information Systems (GIS) has increased in importance because of the current trend toward globalization. GIS centers on issues relating to information technologies (IT) and other factors which enable globalization. GIS is evolving to cope with the continuous changes in IT as well as globalization enablers. This chapter provides a snapshot of GIS for readers who have had no prior knowledge of it and who are interested in exploring the field further.

The GIS framework proposed in this chapter is divided into three major areas, namely, infrastructural, operational, and organizational. Infrastructural issues are the fundamental ingredients for GIS implementation. These include issues relating to information, computing, telecommunication standards and technologies, as well as Internet standards. Without these infrastructural ingredients, communication across borders will be virtually impossible. The ubiquitous nature of GIS mandates the understanding of operational issues. Operational issues deal with a country's culture, training, management, people, government, legal structure, and organizational factors. The organizational issue is a crucial factor for the successful deployment of GIS at the corporate level. For instance, multinational enterprises (MNEs) need to align their information systems (IS) strategy with their business strategy to deal effectively with organizational issues.

The framework proposed in this article provides an integrated view of GIS with a multilayer analysis. The article concludes with a list of resources, including references and bibliography, for readers interested in further exploring the GIS domain.

II. DEFINING GLOBAL INFORMATION SYSTEMS

Over the last 20 years, the field of GIS has grown and evolved into a major subset within the field of IS. Carmel and Davison report that there are many others terms which are used to describe GIS. These include Global Information Technology, Global Information Technology Management, International Information Systems, and Global Management Information Systems among others.

Understanding the GIS's interdisciplinary nature mandates the prerequisite knowledge of globalization. For instance, the devaluation of China's current Yuan Renminbi in 1994, triggered the devastating economic Asian crisis. The Asian crisis caused the volume of world merchandise exports to drop 6% or more than 300 billion U.S. dollars between the years 1997 and 1998. According to the World Trade Organization (WTO), this was equivalent to one percent of the world Gross Domestic Product (GDP). At the same

time, import and export volumes decreased. This degree of interconnections among the world economies is anticipated to grow continuously. It is important for western MNEs to enter the emerging markets in order to survive in this competitive environment. Despite the existence of many counterglobalization forces, the world GDP is expected to grow to $48 trillion by the year 2010. As AT&T President John Zeglis pointed out, the only two outcomes for all companies in the future is either to go global or go bankrupt.

Researchers suggest that information and knowledge will replace tangible goods as the main source for sustainable competitiveness of the business world. Thus, knowing the way by which information travels across borders, as well as its transfer to knowledge, has become critical for businesses. Every business should be considered first and foremost as an information business. In most industries, information processing activities such as capturing, storing, and processing information represent a large percentage (30%) of their cost structure. Information also defines supplier relationships and is the core competence for some industries, e.g., airlines and retailers.

It is clear that globalization is a unifying theme of many enablers, namely, economics, politics, information, culture, natural environment, and business activities of multinational enterprises and IT, among others. For instance, researchers from the international and political economy (IPE) studies are particularly interested in understanding the relationship between politics and economies and how these contribute to globalization. Anthropologists such as Hofstede, Tomlinson, and Fiddle among others pay particular attention to cultural aspects; while researchers such as Bartlett and Ghoshal among others focus on business activities of the MNEs.

In contrast, GIS researchers emphasize the influence of IT in globalization. Some of the researchers include: Ives and Jarvenpaa; Stephen and Nolan; Hudgins-Bonafield; Huff; Karimi and Konsynski; Laudon and Laudon; Lucas; and Passino and Severance among others. Research in GIS studies typically encompasses varying levels, namely: regions, nations, firms, groups or teams, and individuals.

III. GIS FRAMEWORKS

There are many theoretical frameworks which have examined GIS. Two of the frameworks have been used extensively by researchers. These include the framework of Palvia and colleagues, which takes the economic perspective and suggests examining GIS issues in terms of advanced, less-developed, and underdeveloped countries. The second framework which was proposed by the *Journal of Global Information Management* (JGIM) adopts a general GIS framework to examine all possibilities of GIS issues. However, the scope of these two frameworks is either too broad or too narrow for this article. This article takes a different approach, and examines the infrastructural, operational, and organizational perspectives of GIS. The frameworks provide the reader with an integrative and clearer picture of GIS.

A. The Palvia and Colleagues GIS Framework

This framework divides the world into advanced countries, less-developed countries, and under-developed countries. Key GIS issues, Palvia and colleagues argue, often vary depending upon which category countries belonged. For instance, advanced countries are driven by strategic applications of IS in support for the global operation of MNEs. Countries categorized as less-developed are concerned with the operational efficiency and effectiveness of information systems. The deployment of IT infrastructure is the top priority GIS issue for underdeveloped countries.

According to Palvia and colleagues' framework, many issues overlap for countries moving from one economic level to another. IS productivity paradox is ranked as one of the top ten issues for countries of all three levels. Data security is the top GIS issue for both the advanced and less-developed countries. Many issues have to be discussed repeatedly if this article adopts this framework. More importantly, the rankings of Palvia and colleagues have not been updated since its initial publication. Many new and important issues, particularly the Internet related ones, are not included. Thus adopting the framework of Palvia and colleagues will not reflect the current state of GIS research.

B. *Journal of Global Information Management* Framework

The *Journal of Global Information Management (JGIM)* adopts a general perspective to cover important GIS issues in the past, now, and in the future. Tan believes that GIS issues should include:

- *GIS in business functions:* international marketing, human recourses, research and development, accounting and finance

- *IT in specific regions of the world:* IT in the Asia Pacific, Europe, the Middle East, Africa, Latin and North Americas
- *Management of global IT resources and applications:* IT in government, library, global telecommunications, electronic commerce, data security, IT diffusion and infrastructure, education and IS research

The *JGIM* framework substantiates the GIS construct with a list of rich and contemporary issues. Japan, for instance, cares more about using IT to improve the international competitiveness of its MNEs, while Taiwan is interested in operational issues (e.g., data security and privacy) and Indonesia are interested in infrastructural issues, respectively. Among the electronic commerce issue, user-interface design and productivity measurement are important topics for countries such as the United States and Japan, while technology adoption and deployment of telecommunication infrastructure are more important issues for Nigeria and North Korea. Discussing GIS issues along with the *JGIM* framework can be endless. With the scope of this article, this approach is impractical and cannot provide a constructive and systematic analysis of GIS.

C. GIS Framework Proposed in this Article

In comparison with the GIS frameworks of Palvia and colleagues and *JGIM* (Table I), this framework addresses the GIS discipline from its three generic constituents—infrastructural, operational, and organizational issues (Fig. 1). The GIS infrastructure is the foundation for the future growth of GIS deployment. Even if organizations are built on a robust infrastructure, organizations (e.g., government, firms, and teams) can still fail in executing its GIS projects without considering international operational issues. At the corporate level, major users of GIS, i.e., MNEs, need to constantly align their organizational strategy and structure to respond to the changing global environment. These three GIS constituents together will contribute to the successful implementation of GIS.

1. Infrastructural Issues

IS are typically organized in a manner which supports the activities of an organization. GIS are designed to support the international operations of organizations. When designing a GIS or any other IS, researchers such as Turban, McLean, and Wetherbe suggest that

Table I Comparison of GIS Frameworks

Palvia and colleagues	*JGIM*	Generic framework
Advanced countries	GIS in business functions	**Organizational issues**
Using IS for competitive advantage	IT in specific regions of the world	Strategic alignment of IS and global business strategy
Aligning IS and corporate goal	Management of global IT resources and applications: IT in government, library, global telecommunications, electronic commerce, data security, IT diffusion and infrastructure, education and IS research	
IS strategic planning		
Improving IS productivity		
Data security		
Less-developed countries		**Organizational issues**
IS strategic planning		Culture and technology adoption
The operational issues		Training: supply of IS professionals
Awareness of MIS contribution		People
Quality of input data		Government
Data utilization and security		Laws
Standards in hardware and software		Data and network security
User friendliness of systems		
Improving IS productivity		
Under-developed countries		**Infrastructural issues**
Obsolescence of computing hardware and software		Information, computing, and telecommunication (ICT) technologies and standards
Availability of skilled MIS personnel and development		Internet
Improving IS productivity		

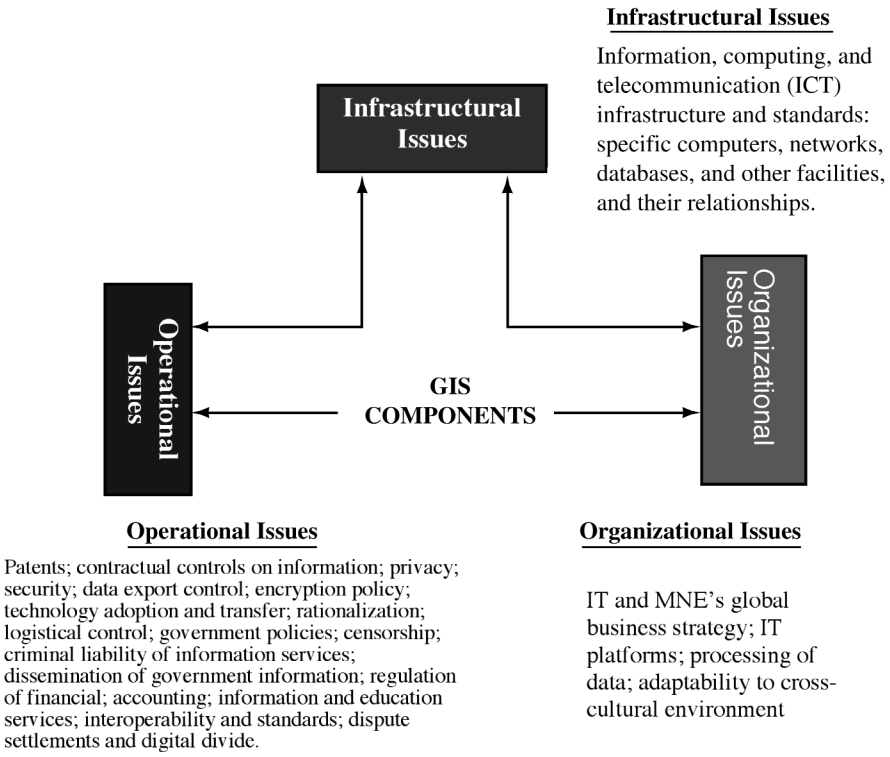


Figure 1 Proposed framework.

IS professionals should first conceptualize the *information architecture*— "the information requirements of the core business of the organization, including the way these requirements are to be met." Once the information architecture is defined, IS professionals need to know its components. Components of information architecture include computers, networks, databases, and other facilities. The relationships between these components are called *information infrastructure*. GIS information architectures and infrastructures are more complex than domestic-based information architectures, because of their international diversity. As is with any other large-scale IS, a well-designed and solid information architecture and infrastructure are the foundation for the growth of a GIS.

2. Operational Issues

GIS is the ultimate example of networked computing. Communication via the Internet, for instance, demands an uninterrupted global electronic connection between receivers and senders. A global communication infrastructure needs to be in place to connect thousands of hardware, software, and telecommunication facilities across borders. When data and information transcend national borders, they have to bypass many points before reaching their destination. Every point of stay is vulnerable to intentional and unintentional entries. GIS's potential vulnerability stresses the importance of security control, auditing, personnel training, procedural management, culture, and national policies. Other than the security factors such as culture and supply of IS professionals, governmental laws can also undermine the effectiveness of GIS implementation.

3. Organizational Issues

Interests in employing IS as competitive weapons for MNEs to achieve international competitiveness can be categorized as an organizational issue in GIS research. MNEs need to align their GIS with other organizational components to cope with or supersede external environment challenges. According to Leavitt and Scott-Morton, organizational components include: (1) organization structure and corporate culture, (2) management and business process, (3) individuals and roles to changes of global economic, and (4) political and social environment. An organization needs

to ensure that GIS are designed as an integral part of the organization along with four other organizational components, and not merely appended to it. This demands that MNEs adopt an interactive planning process between organizational and IS strategies.

The generic framework proposed above is adopted throughout the next three sections. Each section covers important and topical GIS issues.

IV. INFRASTRUCTURAL ISSUES

Prior to the mid-1980s, most governments monopolized their telecommunication industry and offered uncompetitive services. Simon's study found that the convergence of telecommunication industry and the infinite growth of IT capacity have forced many governments to deregulate their telecommunication industry and services. For instance, in 1996 the United States government passed the Telecommunications Act; this was followed by the European Union (EU) which liberalized the telecommunication industry in 1998. In addition, the member countries of the WTO signed the Global Agreement on Basic Telecommunications. Deregulating the telecommunication industry around the world has gradually shaped today's global information infrastructures. The Internet evolution is heavily contingent upon this global infrastructure. Examining telecommunication in general and the Internet's infrastructures in specific can help understand the infrastructural constituents of GIS.

A. Telecommunications

The widespread use of information-processing equipment (e.g., personal computers, workstations, and minicomputers) has made their interconnections crucial for today's communication. Many network options are available in different geographical ranges—local area network (LAN), metropolitan area network (MAN), and wide area network (WAN); wireless area network, as well as the ubiquitous World Wide Web (WWW) or the Internet. Grover and Vaswani's research of 127 telecommunication players in the United States resulted in the categorization of the global telecommunication industry into four key players—communication, distribution, content, and tool providers. Each player offers different elements to allow global communication to take place. Communication providers include local phone, cellular phone, Personal Communication Services (PCS), and other wireless service providers. Distribution providers include broadcast, Internet service provider, cable TV, and long-distance phone service. Tool providers manufacture and supply hardware and software. Hardware providers include routers, modem, Ethernet card, etc., while software providers include operating systems, browsers, etc. Content providers include entertainment firms (e.g., Time Warner), database vendors, and on-line services.

This classification poses important implications for GIS applications in other countries as well. Partnerships among distribution firms can reap the benefits of economies of scale and scope. Content providers often ally with distribution providers in creating a proprietary distribution channel to deliver products or services. The merger of Time Warner and AOL (America Online) is a good example. The alliance between communication and distribution providers empowers them to customize services and products for each individual. Tool providers of hardware and software join hands to speed up their innovation processes.

Driven by such benefits, the telecommunication industry is undergoing the global convergence of communication networks, computing hardware, software, contents, and IT education. Simon believes that this convergence creates opportunities for (1) newer and better telecommunication service and products; (2) fair, but intense global competition of all the players; (3) more affordable innovative products; (4) deregulating the heavily monopolized telecommunication industry of some countries; and (5) global policy and trade battles over which path to take—free markets, regulated markets, or markets in between. Each country needs to have a clear policy directing its telecommunication industry to benefit its people, organizations, country, regions, and the world as a whole. A strong infrastructural constituent is required for the continuous growth of the Internet evolution and the GIS discipline.

B. Internet

The Internet is an example of GIS. According to the International Telecommunication Union (ITU), the Internet became a true mass communication tool for over 50 million people 4 years after it was launched in 1995. In contrast, it took TV, personal computer, radio, and telephone 13, 16, 38, and 74 years, respectively, to achieve the same audience. The Internet has affected every industry. As such, the Internet has become a pertinent issue for the GIS researcher. Thus, it is essential to understand the history, participants, and social and global impacts of the Internet when studying GIS.

As specified in the resolution of the Federal Networking Council (FNC) on October 24, 1995, the "Internet refers to the global information system that:

1. Is logically linked together by a globally unique address space based on the Internet protocol (IP) or its subsequent extensions/follow-ons
2. Is able to support communications using the transmission control protocol/Internet protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols
3. Provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein."

The Internet's history can be traced back to a networking project of the United States Defense Department in the mid-1960s at the apex of the Cold War. The project was named after the department—Advanced Research Projects Agency (ARPANET). This project was to create a defense communication system that was resistant from possible nuclear attack from the former Soviet Union. As the number of different networks increased and crossed the American borders, their incompatibility became a bottleneck for global and open interconnection. In 1982, Bob Kahn and Vint Cerf of the University of California, Los Angeles, developed the *de jure* (open) technical protocol or TCP/IP to solve the interoperability problem. This technological breakthrough was named the "Internet." When the National Science Foundation (NSF) developed the packet-switching network NSFNET successfully in 1985, the Internet gained popularity among the research community.

Corporations began using the Internet for commercial use after the NSF lifted its ban on commercial use in 1992. The Internet's technological sophistication was overcome when the user-friendly graphical user interface (GUI) tools and browser programs of Mosaic allowed nontechnical people to point-and-click the World Wide Web pages. As the Internet become more user friendly, the Internet users reached 40 million people through 10,000,000 hosts, representing more than 150 countries in 1996. According to the Internet Software Consortium (ISC), by January 2001 the world had 109,574,429 Internet host accounts, which were advertised in the Domain Name Service (DNS).

The Internet became a global network of networks that was not owned by any government, but governed by multilateral institutions. These institutions include international organizations, regional and international coordinating bodies, as well as a collaborative partnership of government, companies, and nonprofit organizations. For instance, the American National Standards Institute (ANSI) facilitated the development of standards for the global Internet infrastructure that include data, transmission medium, protocol, and topologies. The Organization for the Advancement of Structured Information Standards (OASIS), a nonprofit and international consortium, commits to the mission of creating "interoperable industry specifications based on public standards such as XML and SGML, as well as others that are related to structured information processing." The most popular web languages, such as SGML, XML, HTML, and CGM are all the innovations of OASIS.

The Internet Corporation for Assigned Names and Numbers (ICANN), a private sector and nonprofit corporation, is in charge of the technical coordination of the Internet activities on a worldwide basis. This includes IP address space allocation, protocol parameter assignment, domain name system management, and root server system management functions. Last year directors of ICANN decided to introduce new top-level domains (TLD) such as .com, .net, .org, .edu, .int, .mil, and .gov. This was because the current TLDs, which were initially issued in 1994, failed to meet the demands of the ever increasing number of Internet users. According to ICANN, TLDs are important in promoting competition within the same category of institutions.

Deploying the global Internet infrastructure was another private initiative to increase the bandwidth of data transmission across different countries. The UUNET, a major international provider of the Internet infrastructure, has already deployed T1, T3, and OC3 across continents. These are the major backbones of today's Internet. Studies indicate that a single fiber optic could deliver 1000 billion bits per section. When the global Internet infrastructure can deliver at least the bandwidth of one fiber optic within and across all countries, the entire world will be able to enjoy multimedia presentation of the Internet as with TV today.

However, the inequality of information and technology communication (ITC) and other complementary assets—education, literacy level, training, security, adoption, and so forth—do indeed exist in different countries. A recent report in the *Economist* indicates that Norway, Singapore, and the United States are the top three countries with approximately 50% Internet penetration rate. In contrast, the penetration rate of the Internet in countries such as India, Egypt, and China is below 5%. It is clear that this "digital divide" is hindering the realization of a total global

community. As the United Nations stated, the challenge is for wealthy nations to "spread technology in a world where half the population does not have a telephone and 4 of every 10 African adults cannot read." The United Nations held the Economic and Social Council (ECOSOC) conference on IT in July 2000 to propose international aids in the deployment of complementary assets that can stop the widening digital divide. Based on its investigation, they found about "90% of the Internet host computers are in high-income countries with only 16% of the world's population." Thus, the lack of complementary assets for IT investment in low-income countries and the digital divide will continue to be an important GIS issue.

The Internet empowers buyers and suppliers to interact electronically and directly. A study by Chircu and Kauffman found that on-line intermediaries are displacing traditional intermediaries to facilitate customers completing the entire transaction process. Siebel System Inc., the largest software provider for customer relationship management solutions, estimates that 80% of sales made on the Internet come from business-to-business (B2B) transactions. Dataquest estimated that on-line B2B transaction volumes would grow from $12 billion in 1998 to $1.25 trillion in 2003. Dataquest's estimated transaction volume does not include financial goods and services that cross national borders. With the speed of the global connectivity and growth of non-proprietary standards, the Internet will be a key player for the continuous growth in both B2B and business-to-commerce (B2C) transactions.

V. OPERATIONAL ISSUES

Robust global telecommunication infrastructures do not guarantee a successful GIS. This could be due to reasons such as, communication misunderstanding due to the cultural factors, lack of IT professionals, lack of IT training, government regulations, and legal and security issues. When deploying IS on a worldwide basis, these operational issues can become particularly important and have to be assessed and addressed in advance. Thus, it is crucial to understand the operational constituent of GIS.

A. Culture

Social anthropologists consider culture as the "collective programming of the mind" to distinguish one group of people from another one. Hofstede states: "Culture is learned, not inherited. It derives from one's social environment, not from one's genes." According to Hofstede, culture should be separated from an individual's personality and human nature. The "ecology fallacy" problem can occur for any study without making any distinction among them. Culture should not be measured in real and absolute value because it cannot be compared.

Culture manifests itself in symbols, heroes, rituals, and values. Symbols represent the most superficial and value the deepest manifestations of culture, with heroes and rituals in between. Legacy about heroes can be written and told while rituals and symbols can be observed. Value, however, is intrinsic and can hardly be observed or told. One way to measure a people's value is to interpret statements about desires. In 1983, Hofstede designed questionnaires to ask IBM employees from 55 countries about how they thought the world ought to be (the desirable) versus what they want for themselves (the desired). Hofstede identified four cultural dimensions to represent values of different cultures—power distance, individualism, masculinity, and uncertainty avoidance. Each cultural dimension is quantified with a specific indicator to illustrate each country's differences.

The Power Distance Indicator (PDI) stands for "the extent to which the less powerful person in a society accepts inequality in power and considers it as normal." Malaysia (PDI = 104), Mexico (81), India (77), Singapore (74), and Brazil (69) are good examples of high power distance culture. In contrast, the countries with lower power distance culture include Austria (11), Israel (13), Ireland (22), Great Britain (35), Australia (36), Canada (39) and the United States (40).

The high Individualism Index (IDV) pertains to "society in which the ties between individuals are loose" while the low IDV or high collectivism refers to "society in which people from birth onward are integrated into strong, cohesive ingroups, which continue throughout a people's lifetime and continues to protect them in exchange for unquestioning loyalty." USA (IDV = 91), Australia (90), Great Britain (89), Canada (80) and South Africa (65) are good examples of high individualism culture. In contrast, the countries with lower individualism or high collectivism culture include Guatemala (6), Taiwan (17), Thailand (20), Mexico (30), Arab countries (38), and Japan (46).

Men who live in the high Masculinity Index (MAS) society are more "assertive, tough, and focused" and women are more "modest and tender." Role differences between genders are not so distinct in the femininity society. Japan (95), Austria (79), Italy (70), Mexico (69), and the United States (62) have higher MAS. Sweden (5), Denmark (16), Thailand (34),

South Korea (39), and Taiwan (45) have lower MAS or a stronger femininity culture.

The extent to which the members of a culture feel threatened by uncertain or unknown situations, and the feelings of nervous stress, predictability, and a need for written rules are good indicators for uncertainty avoidance. Countries with high uncertainty avoidance index (UAI) include Greece (112), Japan (92), France (86), Mexico (82), Israel (81), and Germany (65). The low UAI countries are Singapore (8), Denmark (23), Great Britain (35) and the United States (46).

Straub researched how culture can influence people's attitude toward using IT in achieving individual and organizational productivity gains. He found that people from low IDV countries tend to avoid using IT in supporting individual and organizational productivity. Igbaria and Anandarajan also confirm the evidence by studying computer usage in Nigeria, a country with the low IDV culture (20) in relation to the United States (91). Since low IDV people often come from a culture where indicators of PDI, MAS, and UAI are low, Straub reformulated a computer-mediated support indicator (CMSI) to represent cultural influences to the perceived usefulness of IT and its actual use. Straub's research findings recommend considering cultural factors (e.g., CMSI) when deploying GIS.

All nationalities possess unique cultures that have to be integrated into the design and deployment of GIS. A study by Marcus and Gould on the awareness of cultures in the global web user-interface design showed that Hofstede's cultural considerations can significantly improve the performance and the receptivity of the Web on a worldwide basis. Marcus and Gould suggest that a web design with cultural sensitivity can attract more real and global customers, not visitors to improve revenue of the bottom line. The above discussion suggests that culture influences GIS designs (symbols) and adoption (values). When deploying GIS, cultural impacts have to be carefully assessed and well integrated into GIS.

B. Training

The models of behavior and cultural variability, communication and cultural variability, and impression management strongly suggest that cultural dimensions can influence the characteristics of salespersons and customers, which in turn can affect the sales communication style. The effectiveness of sales training programs is subject to the training of salesperson's sensitivity for different communication styles. In the same token, the managers of HR, IS, and IT departments need to assess the impacts of culture on IT training on a worldwide basis. They need to be aware that IT training programs have to be tailored for trainees with different cultural backgrounds. Bostrom, Sein, and Olfman's IT training strategy framework clearly shows that an effective IT training program has to be customized toward an individual's learning style.

In the most affluent societies, people have many years of schooling. They form mental models or culture to learn possible preferences to a particular learning style. In the large PDI countries, "the educational process is highly personalized" and the communication between the teacher and the student is one way. In the small PDI country, effective learning depends very much on the establishment of two-way communication. They are applicable to the choice of synchronous and asynchronous communication tools for IT training required for people with different PDI cultural backgrounds.

In large IDV countries, two-way communication between teacher and students are highly encouraged. Students expect to be treated individually and impartially in the same classroom, regardless of their backgrounds. In contrast, collectivists prefer the asynchronous communication initiated by the teacher. Learning is more of a one-time process to prepare the youth to participate in the society. The influence of the masculinity can be seen in the classroom behavior. In a masculine culture students often take the initiative to demonstrate their competence and welcome the challenge. Teachers' brilliance and academic reputation could be major determinants for students to learn and improve their learning performance. Students with a higher feminist culture appear less eager to achieve this superficial achievement. The mutual solidarity is the learning goals to feminist culture. Teachers' friendliness and social skills are more important factors to facilitate students through the learning processes. Grades are less important in a feminist society.

Students from the high UAI culture favor more structured learning situations where learning objectives, assignment, deadline, and ways to grade are clearly specified by the teacher, an expert in this knowledge domain (e.g., Germany and Japan). In contrast, open-ended questions with vague learning objectives, broad assignments, and no timetable at all are the preference of students from low UAI culture. Arguing with teachers is considered a benefit of the learning processes.

Cultural differences such as the ones mentioned above can help trainers design better IT training programs with the best-fit learning tools (face-to-face

versus web-based learning) to manipulate the learning behaviors of trainees. For instance, when instructing trainees to design the web pages for customers in both high PDI and low PDI countries, a competent trainer needs to design the course with a different focus. Trainees who design web pages for customers in high PDI countries need to know more about how to structure information, implement explicit and enforced security policy, and design logos that can be associated with authority or expertise. On the other hand, when designing web pages for customers in low PDI countries, trainees need to know more about designing a less structured and hierarchical information flow and user-friendly interface.

C. People

People become part of a community when they are in close proximity to each other, participate in local activities, and share values. A village is a typical example of a traditional community. In the virtual world, physical proximity is no longer necessary. People around the world can choose any activity in any corner of the world because they are participating in global activities locally and share the same value of the community that they believe in. Igbaria defines the virtual society as "a social form where people do not have to live, meet, or work face to face in order to develop or maintain a significant relationship." He identifies five forms of virtual society: telework, computer-supported cooperative work (CSCW), virtual corporations, virtual community, and teledemocracy.

Virtual communities provide people, the time, place flexibilities, the scale of a community, and the speed of forming a community. People no longer need to participate in community activities only when they have holidays and live closer to each other. For instance, the host for the environmentalist community can be in Israel while nationalities of its members can be as many as those of the United Nations. Thus the view of a virtual community is global rather than local.

The size of a virtual community can be as large as it wants to be as long as its activities can attract enough people. Since the location of a virtual community is not a replication of traditional community, local or state governments can barely control the flow of information across their national borders. What makes it even harder for state governments to control the activities of virtual communities is their growth speed. Speculating the disappearance of countries due to virtual communities, some researchers find that the Internet is helping people form virtual communities defined by "common interests" instead of "the accident of physical proximity" at a "spontaneous" speed. In conjunction with the traditional media, virtual communities are growing every minute as long as they can find at least two people sharing the same value.

The virtual community is not just a nonprofit phenomenon. It is effectively applied to the real business to reap some marketing benefits. The study of Hagel and Armstrong reports that virtual communities can bring forth the following benefits to companies:

- Increasing customers' demand for products/services
- Word-of-mouth promotion of products and services
- Stimulate customer feedback to improve products/services
- Generating richer information in customers and markets
- Eliminating separation of advertising and transaction
- Allowing advertising to be seen as helpful, not intrusive

Most organizations are actively using the virtual community to spread their mission, increase customers/members base, and strengthen the interaction among people who believe in the similar value. As a result, each virtual society is creating its own culture, languages, habits, and other sociological factors. This new virtual culture is competing with a nation's traditional culture all the time in the minds of the Internet users. When the Internet users begin spending more time participating in the activities of the virtual society, it is plausible that a new global culture may emerge. This may lead to the demise of national culture.

Many communities or cities are taking advantages of the high bandwidth networking infrastructure. Cohill and Kavanaugh have found that some communities are created to "enhance communication among its citizens, and in particular, to be proactive and a catalyst for empowering citizens to interact with their local government." Another study suggests that a popular virtual community always take into account four factors—design, impacts, critical mass, and access. Government has very limited control of these factors because they are mainly the product of the interactions among web designers, content, and recipients. Although the Chinese government is using gateways to control the inflow of the banned religious and anti-government content, virtual societies allow the information penetration via e-mail, Bulletin Boards system (BBS), and file transfer methods.

D. Government

In this new environment, governments find it much harder to manage the information flow across national borders. Physical borders cannot be applied to the logical boundary created by the Internet. Traditional national entities—land mass, language, culture, history and religions—are gradually vanishing. The Internet is redefining roles of a country.

Internet connection is beneficial for people living in a country where the Internet infrastructures—hardware, software, education, literacy, income, and so forth—are built and widely deployed. Communication via the Internet, however, is not feasible for people in countries like North Africa, poor cities of India and China where the public switched telephone network (PSTN) is not even present. The wealth gap between high-income and low-income countries can be enlarged as fast as the growth speed of Internet hosts. The digital divide is an Internet phenomenon and the gap of information accessibility among ethnicity, income, education, and regions is widening. Countries are facing new challenges of closing the digital divide.

The digital divide seldom concerns national policy makers of a country with the size of a city, such as Singapore or Hong Kong. These countries often find it easier to establish a nationwide telecommunication infrastructure. However, the "digital divide" becomes a grave concern for larger countries, e.g., Mexico, Canada, the United States, Japan, China, Spain, Germany, and South and North Korea. Typically, these countries have wider distribution of incomes and education, as well as disparity of telecommunication infrastructures and computer literacy in different regions.

The *Los Angeles Times* conducted a county-based annual survey on Internet usage. This study found that Orange County, CA, was in many ways ahead of other states in computer and Internet use. The study also found that the Internet divides Hispanic and non-Hispanic population with respect to computer and Internet usage experiences. Potential problems that have resulted from the digital divide include incompetence of Hispanics to locate local IT-related jobs in the short run. In the long run, the degree of economic and social inequality can strongly influence the stability of the United States.

People in higher income brackets use computers and the Internet more frequently than people with lower incomes. This study found that people with incomes higher than $36,000 have about two times the computer usage and three times the Internet usage than people with less income. Another analysis on IT impacts across 36 nations over the years 1985–1993 had similar results. Dewan and Kraemer found that the contribution of IT capital investment to national productivity is much higher in developed nations than developing and less-developed ones. In the long run, countries with free government policy can lead to even higher disparity of computer usage within themselves.

Typically people indirectly authorize their government to act on their best interests because of the perception that a government has better information then they do. However, this assumption becomes obsolete when the Internet can empower citizens to obtain personalized and convenient public service elsewhere. Based on the perfect information, people can choose to live in different places, as well as MNEs can choose different places to operate. Since 1992, Singapore has initiated an e-government project called Singapore ONE. The object of this project was to transform Singapore into a digital island where public service was so convenient, transparent, personalized, and reliable that citizens and MNEs of other countries were attracted to live in Singapore. The Australian and British governments are initiating similar e-government projects, such as Britain's UK On-line.

In addition, e-governments can catalyze economic growth of a country by reducing administrative costs, generating tax revenues, creating supply chain relationship between government and business, enhancing universal education opportunity and eventually avoiding digital divide. When governments streamline their administrative processes, administrative and procurement costs can hugely be saved. If the United States government and EU can save 20% of procurement costs it is estimated that about $110 and $155 billion a year can be saved, respectively. At the same time, citizens can enjoy 24/7 public services from every part of the world. By now, many countries are implementing five main teledemocracy applications according to the Teledemocracy Action News + Network website or TAN-N, a website for the global democracy movement.

1. Voting from the home
2. Scientific deliberative polling
3. Computer-assisted democracy
4. Electronic town meetings
5. Affiliated organizations

However, security can be one of the major opposing forces preventing citizens from participating in e-government projects. Governments maintain confidential information such as social security numbers and criminal and medical records. They need to secure this information from being misused. Another

opposing force is the weakness of existing PSTN. The maximum speed of DSL connectivity services is about 1.5 Mbps. This speed won't be able to handle some image-rich documents with videostream demonstration. This was why Singapore spent $300 million on the Internet infrastructure to attract fiscal and financial sponsorships from business partners. Currently, Singapore has deployed an ATM (655 Mbps) national backbone to support 98% of home Internet access via PSTN and cables according to the Singapore ONE project. Countries that have an equivalent Internet bandwidth include: Norway, United States, Sweden, Canada, Finland, Australia, Denmark, and New Zealand based on the Nua Internet Survey. E-government transformation could be the next move for such governments.

E. Law

What crimes can be categorized as cyber crimes? If employees and hackers sabotage networks for financial gains or personal satisfaction, it can be classified as a cyber crime. The vulnerability of the Internet has caused operations to be terminated, privacy breached, information changed, information of customers stolen, and personal information maliciously used. The Privacy Rights Clearinghouse (www.privacyrights.org), a nonprofit agency based in California, estimates that about 400,000 thefts of identification, costing around $2 billion occur each year. However, there is no universal agreement as to which Internet activities can be classified as computer crimes. The on-line casino, for example, is illegal in the United States, however, gaming casinos on the Internet have legal grounds in Australia and other countries. Since we are all connected to the Internet directly or indirectly, is an American in the United States committing crimes if he gambles on Australian casino websites? Let's consider another extreme example. The Chinese government considers spreading antigovernment information a serious crime. The United States Constitution grants people the right of free speech. What legal grounds can the Chinese government take to punish a Chinese citizen hosting his website in the United States and spreading antigovernment information against the Chinese government from a server located in the United States? Yet another example, New Zealand legalized the Privacy Act in 1993 to protect their citizens' privacy. However, it's difficult to implement this act in an Internet-based setting. For instance, it is considered a crime if someone uses listening devices to intercept private oral communication. This law, however, is restricted to the traditional communication media, and is not applicable to e-mail and facsimile communications. When operating internationally, an MNE needs to be sensitive to the jurisdiction disparity.

Despite many security attempts—protocols, firewalls, encryption, authentication, and access control—Internet users are exposed to cyber crimes consciously and unconsciously. Unfortunately, legal systems of countries vary and are not updated or integrated. Countries try using disparate legal systems to tackle interconnected issues like the Internet-related activities. By and large, these attempts fail.

F. Security

The ubiquitous connectivity of the Internet is made possible by innumerous access points or computers. This poses a serious threat to the resources connected to the Internet communication network if there is a rapid increase in users. On the Internet, every point of access can be intruded upon or damaged in many ways by people intentionally or unintentionally. Such security threats may include human errors, computer abuse or crimes, natural and political disaster, and failures of hardware and software. Of these threats, general and application-based controls can prevent human errors, and hardware and software failure from happening. Yet, the most serious threats to today's global connection are the computer crimes or abuses. There are many forms of computer crimes. Vladimir Zwass identified the ten most common computer crimes or abuse: impersonation, the Trojan horse method, logic bomb, computer viruses, data diddling, the salami technique, superzapping, scavenging, data leakage, and wiretapping.

Of these malicious computer crimes, Kalakota and Whinston report that computer viruses, the Trojan horse method, and the worm can have a devastating effect on the Internet. What makes today's networks so vulnerable are the asymmetric security technologies of different countries. For instance, the security system of RSA is designed with various numbers of bits for different purposes—384 bits for noncommercialization, 512 bits for commercialization, and 1024 bits for military purposes. Since the United States government considers 1024-bit RSA as the technologies of national security, they are not authorized by the International Tariff in Arms Regulation (ITAR) to open to other international communities than within the borders of the United States. Thus, the export control of security technologies has prevented e-commerce from becoming a universal phenomenon.

Many other security technologies, such as PGP (Pretty Good Privacy), IDEA (International Data Encryption Algorithm), DSA (Digital Signature Algorithm), PH HPGs (handheld password generators), and Kerberos have been developed. However, they are not accessible to other countries as are RSA security technologies. Israel, for instance, is extremely advanced in Internet security technologies relative to other countries (e.g., France, and China) where security is not a priority. The operational issue, security, has to be solved before citizens of the world can trust the Internet and universally accept the Internet as a true GIS. That is why organizations that are interested in securing the Internet formed the Internet Fraud Council (IFC) (http://www.internetfraudcouncil.org/). Only through the cooperation of the private, public, and international sectors, Oates believes, can we cope with today's challenges to lessen the impact of computer crimes on the global economy and the public confidence. As explained in *CyberGuard Magazine:* "In the foreseeable future, a secure Internet will be the enabler for a truly global economy. Without the inforsecurity, you will see this Internet vision of global prosperity quickly reduced to that of total anarchy."

VI. ORGANIZATIONAL ISSUES

Infrastructural and operational constituents are key external factors to the success of GIS deployment for MNEs and other international organizations. Ives, Jarvenpaa, and Zwass suggest that strategic alignment between management information systems (MIS) strategy and business strategy is critical for the success of international operation. To gain insight into the GIS execution, this organizational constituent—strategic alignment between MIS and business strategy—needs to be addressed.

A. MNE

MNEs deploy GIS to help service their global customers and to manage worldwide marketing activities for standardized products. The rationalization of global operation for better coordination and logistics control demands accurate, complete, and timely information. The consolidation of different legal requirements and financial markets further stresses the importance of obtaining transparent and integrated information. The success of global operation is tightly contingent upon the quality of information that can be generated from a GIS. Thus, a match between global business strategy and GIS is imperative for the success of the global operation for MNEs.

B. Strategic Alignment between Global Business Strategy and GIS

According to Bartlett and Ghoshal, operational efficiency, local differentiation, and worldwide innovation are the three principal forces shaping the competitive posture of MNEs. A major enabler for international competitiveness is the superior coordination capability of MNEs. The coordination between headquarters, subsidiaries, and suppliers is much more complicated in the international dimension when factors of sociopolitics, economy, language, culture, currency, and IT infrastructure sophistication levels become norms for conducting international business.

Difficulties notwithstanding, Zwass suggests, MNEs can leverage IT in a global fashion to (1) process data fast and accurate, (2) access information instantaneously, (3) coordinate information exchange, (4) span the national boundary, (5) support decision making, (6) formalize organizational practice, (7) differentiate products or services, (8) model international environment, and (9) control production. The GIS is any IS used to support the global operation of MNEs. This view is consistent with Egelhoff's perspective of treating organizations as information-processing systems that have the capacity to fulfill the information-processing requirements facing the organization.

The international marketing research has found that a firm expands internationally by first entering a new market, then followed by local market expansion and global rationalization. Success at each stage heavily relies on the accuracy, timeliness, and relevant information that a GIS can provide. Craig and Douglas, two international marketing experts, believe that a good IS should at least be able to provide three types of information: (1) macroenvironment of a nation; (2) market-specific products and competitive structure; and (3) company sales and performance. The marketing perspective emphasizes the vitality of having a GIS to support international marketing decisions.

Demands for a GIS come from other departments of an MNE as well. From a corporation perspective, Bartlett and Ghoshal identify the different roles of GIS for four types of business strategies (Table II). They argue that MNEs which implement the *transnational* strategy demand that network knowledge be resided in headquarters and subsidiaries. This GIS strategy will enable MNEs to achieve worldwide innovation and

Table II Corporate Strategies and Roles of MIS in the Global Business Environment

Business strategy and structure	Principal characteristics	Decision-making characteristics	MIS role	MIS structure
Multinational (decentralized federation)	Foreign operations regarded as a portfolio of relatively independent businesses	Decision making decentralized to subsidiaries, informal relationships between headquarters and subsidiaries	Financial reporting by subsidiaries to headquarters for control purposes	Decentralized; primarly stand-alone systems and dispersed database
International (coordinated federation)	Foreign operations regarded as appendages to domestic corporation, where core competencies are honed	More vital decisions and knowledge in general developed at headquarters and transferred to subsidiaries	Formal planning and control systems coordinate the entire operation	Largely centralized planning and control systems inplemented on a variety of hardware architectures that ensure links among units
Global (centralized federation)	Foreign operations regarded as pipelines for delivery of goods and services to a unified global market, in search of economies of scale and scope	Decisions made at the center, knowledge developed and retained at the center	Tight central control of subsidiaries through centralized planning, control, and general decision making	Centralized systems and databases
Transnational (integrated network)	Differentiated contributions by all units to integrated worldwide operations	Decision making and knowledge generation distributed among units	Vital coordination role at many levels; knowledge work, group decision making, planning and control	Integrated architecture with distributed systems and databases, supporting management and knowledge work across the organization

From Zwass, V. (1992). *Management Information Systems.* New York: William C. Brown. With permission.

global efficiency, and to quickly sense and respond to local needs for differentiation on the other hand. Kenichi Ohmae's lead-country model also recommends designing a common GIS with local add-on functions for local responsiveness. Zwass further proposed that strategically aligning GIS with global business strategy can help an MNE achieve its international competitiveness. On the contrary, Ives and Jarvenpaa state the misalignment of GIS with global business strategy models could seriously undermine the international competitiveness of MNEs. GIS of this type can be a strategic information system that enables firms to outperform their foreign competitors under today's turbulent environments.

However, not every MNE is ready to adopt the *transnational* strategy. For firms pursuing the *multinational* business strategy, GIS are primarily used to control the financial reporting processes from subsidiaries to headquarters. Foreign subsidiaries have maximum autonomy to respond to diversity and opportunities of their local markets. GIS are being managed independently by each foreign subsidiary. Technology platforms, databases, and applications are chosen and implemented without integrating them across borders. Innovation becomes the focal concerns of the *international* business strategy for MNEs. GIS are centrally planned and controlled at headquarters. The system attempts to exploit information and knowledge generated at headquarters and then diffuses it.

In doing so, GIS needs to coordinate the entire operation of an MNE with formal planning and controlling processes. MNEs attempting to achieve international competitiveness by the economies of scale and scope often adopt the *global* business strategy. GIS are tightly controlled at headquarters to coordinate information of planning, control, and general decision making for subsidiaries. GIS are not affected by the adjustment to local information needs. There is more attention as to whether information and knowledge generated at headquarters can be capitalized on the worldwide basis and distributed to foreign subsidiaries. Bartlett and Ghoshal argue that strategic objectives of operational efficiency, differentiation, and worldwide innovation do not conflict with each other. They can be realized simultaneously with the *transnational* business strategy. GIS can be deployed at different levels to coordinate knowledge generation across subsidiaries and headquarters of different nations.

According to the framework of Bartlett and Ghoshal, an MNE may find itself currently using one particular business strategy and structure. To capitalize on the business strategy and structure, a pair-wise GIS structure has to be carefully designed to provide useful information for the MNE. This concept of strategic alignment is particularly important for the global operation of MNEs because external factors to the organization's design and information-processing requirements of the organization are more complicated than the domestic environment. However, as the international environment shifts, MNEs have to continuously evaluate strategic fitness of their GIS against their business strategy and structure. MNEs have to prepare to make any changes to the GIS if their information systems are no longer supportive for the new information requirements.

VII. CONCLUSION

Advances in IT have become the major driving force for globalization in the New Economy. Leveraging IT to generate and transfer information and knowledge has become strategically important for organizations to gain the sustainability of global competitiveness. Discussing globalization without taking the IT factor into account will not accurately capture the true phenomenon of globalization to date. As a result, the GIS discipline, which emerged in the 1980s, has bridged the knowledge gap between globalization and IT evolution.

Practitioners and academics of GIS are concerned with how IT interacts with other globalization enablers, economy, politics, information, culture, natural environment, and business activities of MNEs. A constructive and systematic framework was proposed to explore GIS discipline through its three generic constituents: infrastructure, operation, and organization. The infrastructural constituent addresses the global telecommunication infrastructures of the Internet. The operational constituent attends to issues of culture, technology adoption, training, supply of IS professionals, people, government, laws, and security. Infrastructural and operational constituents are external factors for the successful deployment of GIS. Successful GIS deployment is also subject to the capability of international organizations, particularly MNEs, to align their business and MIS strategies. Without considering these three constituents at the same time, successful GIS deployment is reduced. Each GIS component is equally important to organizations that want to operate globally.

Such integrative thinking is imperative for the IS professionals who endorse successful GIS. MNEs must learn to manage these three GIS constituents as a whole and then turn them to their advantage when conducting global operations.

SEE ALSO THE FOLLOWING ARTICLES

Developing Nations • Digital Divide, The • Electronic Commerce • Globalization • Globalization and Information Management Strategy • Internet, Overview • Year 2000 (Y2K) Bug Problems

BIBLIOGRAPHY

The major reference list is arranged according to subjects of this chapter. Readers who want to learn more about particular subject of GIS can begin from here.

GIS Overview

Carmel, E., and Davison, R. (2000). What is Global Information Technology? ISWorld.Org. 2000 (On-Line). Available from http://www.american.edu/MOGIT/git/git.htm.

Palvia, P. (1999). *Introduction to Theme: Challenges and Opportunities in Global Information Technology in the New Millennium.* First Annual Global Information Technology Management (GITM) World Conference.

Tan, F. (2001). Journal of Global Information Management: Coverage, Idea Group Publishing. 2001 (On-Line). Available from: http://www.idea-group.com/journals/details.asp?id=99.

Infrastructural Issues

Today's Internet is made possible by the following major organizations.

American National Standard Institute (ANSI): http://www.ansi.org/

Federal Networking Council (FNC): http://www.itrd.gov/

Global network map of UUNET: http://www.uu.net/us/network/maps/

Internet Corporation for Assigned Names and Numbers (ICANN): http://www.icann.org/

Internet Software Consortium (ISC): http://www.isc.org/

Organization for the Advancement of Structured Information Standards (OASIS): http://www.oasis-open.org/

Operational Issues

Anakwe, U., Anandarajan, M., and Igbaria, M. (2000). Management practices across cultures: Role of support in technology usage. *The Journal of International Business Management,* 31(4): 653–666.

Bostrom, R. B., Olfman, L., and Sein, M. K. (1988). The importance of individual differences in end-user training: The case for learning style. *Communication of the ACM,* 133–141.

Hagel, III, J., and Armstrong, A. G. (1997). *Net gain: Expanding markets through virtual communities.* Boston, MA: Harvard Business School Press.

Hofstede, G. (1997). *Cultures and Organizations Software of the Mind.* New York: McGraw-Hill.

Marcus, A., and Gould, E. W. (2000). *Cultural Dimensions and Global Web User-Interface Design: What? So What? Now What?* Proceedings of the 6th Conference on Human Factors and the Web, Austin, Texas.

Straub, D., Keil, W., and Brenner, W. (1997). Testing the technology acceptance model across cultures: A three country study. *Information and Management,* 33: 1–11.

Security and Cybercrime

Oates, B. (2001). Cyber crime: How technology makes it easy and what to do about IT. *Information System Management,* 18(3): 92–96.

Simon, L. D. (2000). *NetPolicy.com: Public agenda for a digital world.* Washington, DC: The Woodrow Wilson Center Press.

Organizational Issues

Applegate, L. M., McFarlan, F. W., and McKenney, J. L. (1999). *Corporate information systems management: Text and cases.* Boston: Irwin McGraw-Hill.

Egelhoff, W. G. (1988). *Organizing the multinational enterprise: An information-processing perspective.* Cambridge, MA: Ballinger Publishing Company.

Ives, B., and Jarvenpaa, S. L. (1991). Applications of global information technology: Key issues for management. *MIS Quarterly,* 15(1): 32–49.

Globalization

Jennifer DeCamp
MITRE Corporation and American University

GLOSSARY

character Defined by the Unicode Consortium as "the smallest component of written language that has semantic value; refers to the abstract meaning and/or shape, rather than a specific shape (see also glyph) . . . the basic unit of encoding."

controlled language A limited set of vocabulary and sentence structures that provide better input to machine translation and thus higher quality output.

encoding A system of computer representations of characters. For instance, in the ASCII or ISO 8859-1 encoding, A = 41. In Unicode or ISO/IEC 10646, A = 0041.

globalization Defined by the Localisation Industry Standards Association (LISA) as "making all the necessary technical, financial, managerial, personnel, marketing, and other enterprise decisions necessary to facilitate localization." Globalization is the full business process that includes internationalization and localization, as well as a broader scope of business issues. (Note: LISA is based in Switzerland and so uses the British English spelling for "localization" in their title but alternates between British and U.S. spellings in documents.) An abbreviation for "globalization" is *g11n,* evolving from *l10n* and *i18n,* discussed below.

glyph font Defined by the Unicode Consortium as "a collection of glyphs used for the visual depiction of character data. A font is often associated with a set of parameters (for example, size posture, weight, and serifness), which when set to particular values, generate a collection of imagable glyphs."

internationalization Defined by LISA as "the process that removes local country requirements and enables use in multiple cultural environments." An abbreviation for "internationalization" is *i18n,* originating from a UNIX programming shorthand for the word (i.e., I + 18 letters + N). Although the terms *"i18n"* and *"L10n"* appear to be declining in use, they still appear in some materials and in names of organizations.

localization Defined by LISA as "the process of modifying products or services to account for differences in distinct markets" An abbreviation for "localization" is *L10n,* originating from a UNIX programming shorthand for the word (i.e., L + 10 letters + N).

machine translation (MT) A computerized system responsible for the production of translations from one natural language into another, with or without human assistance.

post-editing The revision of machine-translated text.

pre-editing The preparation of text prior to machine translation in order to enable better translation.

terminology management system (TMS) A tool for managing terminology, including equivalents of terms in two or more languages.

translation memory (TM) A tool designed to assist in the translation process by recalling same/or similar translation units (TUs) in past translations.

Unicode or ISO/IEC 10646 Defined by the Unicode Consortium as "a fixed-width uniform encoding scheme for written characters and text." Unicode provides unique code points for each character in most of the world's languages.

UTF-8 Unicode Transformation Format, 8-bit form.

I. INTRODUCTION

Globalization is defined by the Localisation Industry Standards Association (LISA) as "making all the necessary technical, financial, managerial, personnel, marketing, and other enterprise decisions necessary to facilitate localization." *Localization* in turn is defined as "the process of modifying products or services to account for differences in distinct markets," such as by providing culturally appropriate currency symbols, date formats, and paper sizes, and usually by providing text input and display in the language of the target market. A related term is *internationalization,* which is defined as "the process that removes local country requirements and enables use in multiple cultural environments" and is a step toward efficient and cost-effective localization. The three terms reflect the historical development of the profession, which started with the conversion of some or all of a piece of software's code and documentation (localization), then implemented better business processes for design and software reuse (internationalization), and then expanded since the late 1990s into providing end-to-end solutions, including business cases, product design, and technical support (globalization) (Fig. 1).

II. GLOBALIZATION PROCESS

Stages of the globalization process include requirements analysis, business case development, product design, terminology management, hardware localization, software localization, documentation translation, sales and technical support, marketing, and continued usability testing.

A. Requirements Analysis

The globalization process generally begins with gathering information on current and anticipated customer requirements. Factors in determining customer requirements may include legal, market, and user requirements.

1. Legal Requirements

Some countries have legal requirements that if a company is going to market products in that country, the software and/or documentation must be in certain languages. For instance, Belgium has a requirement that all software and documentation be provided in French, English, German, and Dutch. Canada has a similar requirement for French and English. Moreover, international organizations such as the North Atlantic Treaty Organization (NATO) and the United Nations (UN) have mandatory requirements for specified sets of languages. These markets are closed to companies that do not meet the localization requirements.

2. Market Requirements

There are often compelling marketing reasons for providing products and documentation specific to certain countries or regions. For instance, the presence of competitive localized products may make it necessary to provide localization for the new product. In addition, rejection of some localization requirements has often led to a country's banning all products from a company.

According to a member of the Microsoft localization team, when Microsoft Corporation shipped its Encarta Encyclopedia to India, the Indian government objected to the map portraying the controversial Kashmir region. With paper encyclopedia, the Indian government would usually provide a stamp on the images they found offensive, thus making the image illegible. However, because Encarta is software, the Indian government banned the import of electronic encyclopedia with the controversial maps of the Kashmir region. They also banned the import of any other Microsoft products until this issue was resolved by Microsoft, substantially reducing the size of the map. In this case, the cost of this localization was balanced against the entire market case for all Microsoft products in India.

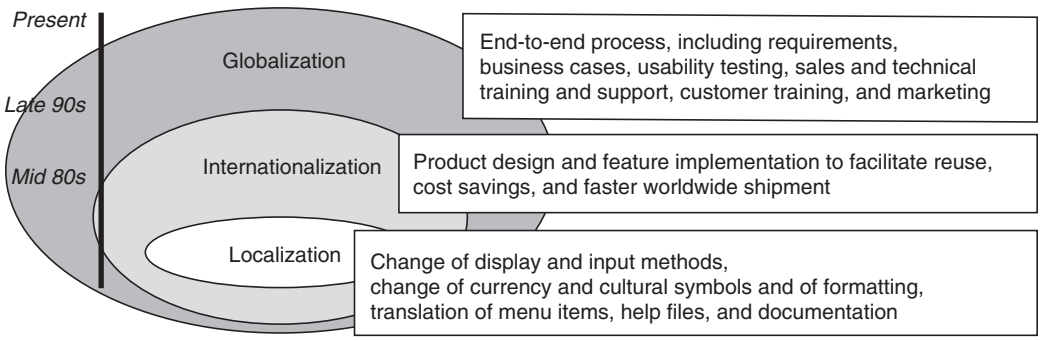


Figure 1 Phases of the globalization process.

3. Other User Requirements

Requirements analysis—an extensive field of study in itself—is often complicated by the complexity of working with foreign cultures and by the potential cost of collecting information from other countries and often in other languages. In some cases, requirements analysis can be conducted by branch offices or sites in the target country or countries. With larger budgets, software designers and developers can arrange to meet with users in the target countries to obtain requirements and to show and obtain feedback on prototypes. For companies or projects with smaller budgets, promising results are being obtained by using remote technologies and feedback and by interviewing people who have recently arrived in the U.S. from the target country (e.g., recent Chinese immigrants in the U.S.). Globalization companies are increasingly providing end-to-end services that include consultation on requirements in foreign markets. For additional information on requirements analysis and usability testing, see *International User Interfaces* (listed in the Bibliography), particularly the article by Jakob Nielson on "International Usability Engineering."

B. Business Case Development

Business cases provide planning and prioritization for funding and also help to obtain high-level management support for localization. In addition, business cases help to ensure adequate funding, a frequent problem with localization, where many costs are not adequately funded but are absorbed by site offices, documentation groups, and other organizations. Business cases should take into account the analysis of primary and secondary markets. They should take into account market limitations, such as tariffs, import restrictions, adherence to copyright agreements, competition, costs, and possibly also comparable case studies, if information on such precedents is available. They should also take into account attitudes of target consumers toward the nationality of the product.

1. Market Analysis

The analysis may include primary and secondary markets.

a. Primary Markets

A primary market may be a country (e.g., Argentina), a region (e.g., Latin America), or a group of countries or regions that speak a similar language (e.g., Spain and Latin America for a Spanish market). Some differences (e.g., differences in keyboard layout, terminology, and currency symbols between French in France and French in Canada) may limit the size of the market for a single localization effort. Information on the size of online markets is provided at http://www.glreach.com/globstats/index.php3. This site lists many of the major world languages, with information on the number of people with Internet access, the percentage of the world's online population, the projected number of people online in that language in 2003, the total population of native speakers of that language, the gross domestic product of that community, and other information pertinent to business cases. Online access may be a good indicator of computer ownership and literacy, figures important for business cases even for software applications that are not necessarily online (e.g., games). *The Ethnologue* at http://www.sil.org/ethnologue/ and the Internet Society at http://www.isoc.org/ are other sources of information of language demographic information that could be helpful in developing a business case.

Note that in developing a business case, it is important to consider not only the sales of localized products, but also the sales of other products leveraged by this localization.

> A 1999 survey by the Localisation Industry Standards Association (LISA) reported $2.5 billion in annual localization expenditures among respondents, which in turn leveraged $50 billion in annual sales in the information technology sector. The survey results are available on http://www.lisa.unige.ch/sig/web_localization/websvey.pdf.

b. Secondary Markets

Business cases may also take into account secondary markets, such as embassies, multinational corporations, governments, libraries, universities, and translation companies, as well as expatriate and immigrant communities. In embassies and in foreign offices of multinational corporations and governments, there is often a need for products that can be serviced in the local country, that can be compatible with those in the home office and other foreign offices, and in some cases (depending on the application), that can produce materials in the languages both of the home country and the host country. Such customers value products where the interface and applications can be switched easily between languages.

Similarly, expatriates and immigrants may value service in the local area and the ability to use a computer application for host country work, but to have the interface and applications primarily operate in a language with which they have a higher comfort level. For instance, Chinese expatriates and immigrants in the United States who do not speak English well may prefer their word processors, browsers, and other applications and also their computer and product interfaces to be in Chinese. Governments, translation companies, libraries, and sometimes multinational corporations often need to be able to receive and process foreign language input and to provide translations and multilanguage text. Often games and other applications are of interest to universities and language learning organizations in other countries as a means of teaching the foreign language.

In some cases, the company may already have a sales and support infrastructure within the country of the secondary market. For example, a U.S. company may have localized its product for Japan; however, the company may already have the sales and support infrastructure in the United States, which could be used to sell not only the U.S. product but also the localized Japanese version. Of course, in this example, training and support materials would need to be provided to the U.S. sales and technical personnel and to the customers.

The 2000 census records 31,844,979 people in the United States 5 years old and over who speak a language other than English at home, of which 13,982,502 do not speak English "very well." According to the census data, there are 17,345,064 people in the United States who speak Spanish at home, of which 8,309,995 do not speak English "very well." There are 4,471,621 people in the United States who speak Asian or Pacific Island languages at home, of which 2,420,355 do not speak English "very well." More detailed information is available at http://www.census.gov.

2. Market Limitations

The business case also needs to take into account limiting factors, including not only standard issues such as competition, but also target market attitudes and adherence to copyright agreements.

a. Attitudes toward Foreign Products

Consumers often prefer to buy products from their own country or to avoid products from particular countries for political or social reasons. Governments may have a similar attitude and may impose import tariffs on foreign products or provide subsidies for local products, thus impairing the ability of the foreign product to compete on the basis of price. As described above, governments and/or international organizations may pass legislation concerning product requirements or provide quotas on monies spent on products from certain countries (e.g., as with NATO). In some cases, it may be advisable or necessary for a business to partner with a local company resident in the target market.

On the other hand, in some areas, consumers may prefer products from a particular country. For instance, the association of the United States with music, computer applications, and fashion is often strongly positive for certain foreign markets. Investigation of consumer attitudes, government regulations, and tariffs is critical in developing a reliable business case.

b. Adherence to Copyright Agreements

As the U.S. Copyright Office points out, "There is no such thing as an 'international copyright' that will automatically protect an author's writings throughout the entire world. Protection against unauthorized use in a particular country depends, basically, on the national laws of that country." The Copyright Office advises first investigating the protections offered to foreign authors (or software developers) within the country where the product will be marketed. The office also points out that protection "may depend on the facts existing at the time of first publication" (i.e., when the software or other localized software is first offered in that country).

Protection may also be provided by international copyright agreements if the country in which the product will be marketed is a party to that agreement and if the company has followed the conditions set forth in those agreements. *Circular 38a: International Copyright Relations of the United States* lists the U.S. copyright relations with specific countries (http://www.loc.gov/copyright/circs/circ38a.pdf). For instance, according to this document, the United States has no international copyright relations and is not a common party to any international software agreements with Afghanistan, so there is no legal protection against any Afghani copyright infringement. On the other hand, the United States and the People's Republic of China both are parties to the Berne Union for the Protection of Literary and Artistic Property (Berne Convention) and the Universal Copyright Convention (UCC), which are the principal international copyright conventions. They are also parties to updates to those revisions, and have two bilateral agreements regarding copyrights. The U.S., thus has some legal recourse against Chinese copyright infringement.

Circular 38 also describes the conditions under which copyrights are protected under U.S. law. An act of October 19, 1976 (Public Law 94-553, 90 Statute 2541) states that published works are subject to protection under specified circumstances, including the following:

1. "On the date of first publication, one or more of the authors is a national or domiciliary of the United States, or is a national, domiciliary, or sovereign authority of a foreign nation that is a party to a copyright treaty to which the United States is also a party, or is a stateless person, wherever that person may be domiciled; or
2. The work is first published in the United States or in a foreign nation that, on the date of first publication, is a party to the Universal Copyright Convention."

Research of the relevant copyright agreements is advisable, particularly employing experts familiar with the copyright laws of the specific countries where the product will be sold.

Within the United States, there is legal recourse, which requires that a product be registered with the Library of Congress, which requires a $30 fee. The U.S. Copyright Offices states: "if registration is made within 3 months after publication of the work or prior to an infringement of the work, statutory damages and attorney's fees will be available to the copyright owner in court actions." If no copyright has been registered, "only an award of actual damages and profits is available to the copyright owner" (see http://www.loc.gov/copyright/circs/circ1.html#cr). In the United States, copyright registration also enables the company to register the product with the U.S. Customs Office, to help prevent importation of products violating that copyright. Registration procedures are also provided at the Web site listed above. Even so, the area of software copyrights is not yet well defined, and there is moreover substantial unauthorized "sharing" of software that diminishes a company's return on investment.

Even with international and national copyright agreements and with attempts to prosecute copyright violations, extensive pirating of software and product ideas still occurs. To protect software, some companies have employed customer identification numbers, passwords, and/or required and enforced product registration to limit use, but with limited success. Some companies offer incentives such as printed documentation, technical support, and upgrades to encourage people to purchase and register the products. Another tactic has been to employ hardware devices in order to run the software. For instance, Sakhr machine translation employs a dongle or physical device that attaches to the computer and is necessary for running the software. Use of the software is thus limited to whoever has the physical device. However, trade-offs need to be calculated between such efforts to limit unauthorized use of the software and the inconvenience such methods provide to authorized users.

> In 1998, the Microsoft product manager for Arabic Office 98 stated that an estimated 95% of the usage of Microsoft Arabic localized products was pirated (i.e., with no revenue to the company).

c. Competition

As with any business case, it is necessary to analyze the competition. Information on localized or native products with similar functionality can often be found on the Internet, although the Web sites may be in the language or languages of the target market. Information may also be available in the annual reports of companies providing similar localized products. It also may be available in patent documentation. The U.S. Department of Commerce can often provide additional help.

3. Costs

On the other side of the ledger, the business case needs to address the costs of providing localization, which may include costs for software design, development, and manufacturing, as well as the packaging, marketing brochures, price sheets, marketing, sales training and support, technical training and support (including help desks and provisions for problem escalation), documentation, and Web site development and updating. For existing localization efforts, it is often difficult to assess the total cost of localization because costs may be spread among and absorbed by multiple organizations within a company.

4. Comparable Cases

Costs and return on investment (ROI) can often be estimated using comparable cases. Through organizations such as LISA and individually in conferences, companies are sharing information about their costs of localization and their ROI.

C. Hardware Localization

Hardware often needs to be localized, particularly with regard to electrical connections (Table I). Voltage differs between countries from 110 (meaning 110–120) to

Table I Sample Differences in Hardware

Country	Voltage	Hertz cycles	Current	Electrical outlets	Telephone jack
United States	110	50	AC	*Ungrounded:* double parallel flat prongs *Grounded:* double parallel flat prongs and rounded third prong	RJ-11
Belgium	220	50	AC	*Ungrounded:* double round prongs *Grounded:* double round prongs	22BELG RJ-11
Brazil	110 220	60	DC	*Ungrounded:* double round prongs; double parallel flat prongs *Grounded:* double round prongs; double parallel flat prongs and rounded third prongs	RJ-11
People's Republic of China	220	50	AC	*Ungrounded:* double round prongs; double diagonal flat prongs *Grounded:* double round prongs; double diagonal flat prongs and flat third prong; double flat prongs parallel with bottom edge of plug and third flat prong perpendicular	RJ-11

220 (meaning 220–240) volts. Hertz cycles (Hz) differ between 50 and 60, and may affect the speed of analog appliances. Fifty hertz generally but not always is correlated with 110 volts, and 60 Hz is correlated with 220 volts. Some countries use more than one standard for voltage and Hertz cycles. Voltage in particular, but also Hertz cycles, used to be a major issue in localization. Now many appliances, including computers, have built-in capabilities for automatically detecting and selecting the proper settings. Transformers can also be easily purchased for transforming the voltage of computers and electronic devices; converters are available for electric appliances such as hair dryers.

An outstanding issue not covered by such detection devices is the use of direct current (DC) instead of alternating current (AC) power in some countries. Information on voltage, Hertz cycles, use of DC, and electrical outlets per country is provided by the American Automobile Association at http://www.walkabouttravelgear.com/wwelect.htm.

Electrical outlets and plugs also vary between and within countries. This issue is often dealt with by providing inexpensive adapters (attachments to plugs that fit the electrical outlet of the target market). However, the use of a nonlocal plug may mark the hardware as being from a particular country (e.g., the two parallel prongs in a plug may mark the system as being from the United States). As discussed in the section on attitudes toward foreign products, such associations with a particular country may help or hinder the sales of a product, depending on attitudes of target market governments and consumers. In some cases, it may be worth providing a separate power cord with the plug appropriate for the particular market.

Modems also vary, although less than voltage, Hertz cycles, and electrical outlets. The RJ-11 telephone jack is most common, and is also often found in conjunction with other styles of jacks. Adapters are available for computer systems. A list of telephone jack styles per country is provided at http://www.walkabouttravelgear.com/modem.htm.

Hardware localization can also be ergonomic. For instance, Japanese have the automobile steering wheel on the left side of the car and drive on the left side of the road. U.S. car companies had little success in selling cars in Japan with the steering wheel on the right side. Hardware localization may concern issues such as the footprint or size of the system, and user preferences concerning factors such as color or fonts.

D. Software Localization

Software localization can range from minimal changes such as providing the Euro currency symbol for applications to be sold in Europe, to providing extensive redesign of the interface. Minimal localization may deal only with locale info (e.g., currency symbols and date formats). More extensive localization may deal with display, input, menus, and documentation. An emerging field involves dealing with graphics, icons, and other computer interface features (Fig. 2).

LISA provides a list of questions to guide localization practitioners through much of this process. The question checklist can be ordered from the LISA Web site at http://www.lisa.unige.ch/home_info.html. In addition, books and/or technical support materials are available for different operating systems. Books

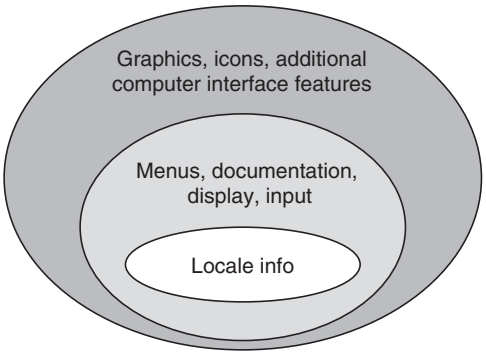


Figure 2 Degrees of localization.

can be obtained from the Multilingual Computing Web site at http://www.multilingual.com.

1. Locale

Locale information includes currency symbols. For instance, most systems sold in Europe were upgraded to handle the Euro symbol. In some cases, this was accomplished by replacing another character in a character encoding or code set (i.e., the mapping of characters in the computer software). In other cases, companies converted to a different encoding, such as Unicode, to be able to include the extra character. Other locale information may include common symbols (such as the telephone symbol in Japanese), number format, measurement, calendars, date formats (including separators such as "/" or "-"), time formats, telephone number formats, page size, and page format. Examples of such differences are shown in Table II. Note that French differs among Canada, France, and Belgium not only in currency symbols but also in the formatting of dates.

2. Display

An additional stage of localization is to provide display capabilities in the language or languages of the target market. For example, for Web pages targeting the Middle East, users need to be able to view Arabic, Persian, and/or Hebrew text. To display a language requires an encoding and a font. With languages with connecting characters and/or different text direction (e.g., left-to-right as in Arabic or top-to-bottom and right-to-left text as in formal Chinese), special rendering software is required. Encodings, fonts, and rendering systems are discussed below.

a. Encodings

An encoding is the computer designation for a character. For instance, in the International Organization for Standardization (ISO) 8859-1 encoding (also known as ASCII), the capital letter A = 41 (Table III). In binary encoding, this number would be 0100 0001. In the Unicode encoding, A = 0041 (or in binary: 0000 0000 0100 0001). For English and many European languages, the Unicode encoding follows ISO 8859-1, except for providing a preceding additional byte.

Characters for English rarely present a problem in displaying text. However, ASCII has only 256 (or 2^8) options with which to designate characters. Therefore, to provide more characters within this 256-bit framework, computer developers have reused the codes by assigning new encodings (also known as *code pages*). Thus the Unicode code point **00CA** designates and results in the display of **Ê** in ISO 8859-1, but **Ę** in ISO 8859-2, and **ъ** in ISO 8859-5. It is not used in CP866, another encoding, and so—depending on the display system—does not appear or appears as a question mark or box (Table IV).

A key issue in display is to ensure that the viewer has the right encoding. For HTML and eXtensible Markup

Table II Sample Differences in Locale Information

Locale information	English in the United States	French in Canada	French in France	French in Belgium
Currency symbol	$	$	F	FB
Decimal marker	.	,	,	,
Measurement	Feet	Meters	Meters	Meters
Time format *(8 minutes and 39 seconds past 6 p.m.)*	6:08:39 p.m.	18:08:39	18:08:39	18:08:39
Short date format *(March 28, 2001)*	3/28/2001	2001-03-28	28/03/2001	28/03/2001
Long date format	March 28, 2001	28 mars, 2001	28 mars 2001	28 mars 2001

Table III Characters Encoded in ASCII and in ISO/IEC 10646

Character	ISO 8859-1 (ASCII)	Binary	Unicode/ISO/IEC 10646	Binary
A	41	0100 0001	0041	0000 0000 0100 0001
B	42	0100 0010	0042	0000 0000 0100 0010

Language (XML), the metadata tag CHARSET (for "character set") is used to designate the appropriate encoding (e.g., CHARSET=KOI-8).

The problem is further complicated by the number of characters available in a code set. When a single byte is assigned for each character, a code set provides code points for 256 characters. Because the Japanese language has more than 6000 characters, and Chinese has more than 10,000, such languages can only be displayed using double byte character sets (DBCS) or multi-byte character sets (MBCS). An implementation of DBCS that has been broadly adopted by industry is Unicode, which is correlated with ISO/International Electrotechnical Commission (IEC) 10646. In the Unicode or ISO/IEC 10646 standards, there is one code point per character and no reuse of code points. Use of this standard often simplifies the display of multilanguage text. There are also different formats of Unicode, including Unicode Transformation Format (UTF) 8, which reduces the encoding from 2 bytes to 1 byte for characters (such as "A" and "B" in Table III) in order to save space and in some cases processing time. Note that for English and many European languages, UTF8 is the same as ISO 8859-1 or ASCII.

An important consideration when using certain encodings (e.g., Japanese Industrial Standard, Big-5, Unicode) is whether the server, operating system, and/or application can support DBCSs. Additional information on text display is available at the Unicode site at http://www.unicode.org/.

b. Fonts

A *font* is the character viewed on the screen or on the printed page. It may be represented using a variety of shapes or glyphs. These characters can also be shown in bold or italic fonts. Thus the character "a" (as a concept) can be portrayed in many ways in different fonts, including those shown in Fig. 3.

Fonts for languages other than English may have marks above or below characters (known as *diacritics*): à â or may have distinctly different characters in a Latin script (e.g., Œ) or in other scripts (e.g., ت, 天). If such fonts are not loaded, text will display as white boxes, black boxes, or other symbols not recognizable as the character.

Size designations of fonts may be misleading, between scripts and even within the same script. For instance, in Fig. 4, the size of each of the characters is shown first in 10 point and then in 18 point. However, even within one point size, the dimensions or actual size of the fonts varies. However, the differences in size affect the layout in a Web page or document and possibly also the page numbering of text. Thus when substituting one language for another, it is necessary to ensure that the fonts are compatible for the graphic layout, regardless of the advertised size. Note that consumers' preferences for certain fonts may vary not only by application (e.g., business letter vs. wedding invitation) but also by country.

Moreover, there are many font sets and font technologies. The differences affect file names in operating systems and applications. Such differences also affect the layout or appearance of the document or Web page. Examples of the different font technologies include the TrueType fonts used in Microsoft Internet Explorer (originated by Apple Computers) and the Bitstream TrueDoc font technology used in Netscape.

There are several means of obtaining the appropriate fonts for the system or application, including providing the font in the operating system and/or application (e.g., in Microsoft Windows) or providing a location where they can be downloaded from the Internet. For primary markets, users generally expect the application or system to already be enabled in the local language or languages.

Table IV The Same Code Point in Different Encoding Systems

Unicode Code Point	Unicode/ISO 10646	ISO 8859-2	ISO 8859-5	KOI-8	Windows 1251	CP866
00CA	Ê	Ę	Ъ	**Й**	к	?

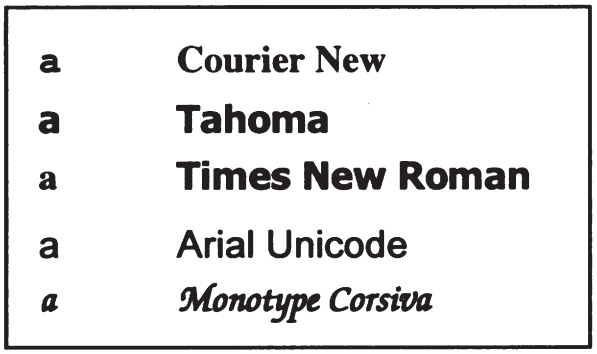


Figure 3 Different portrayals or glyphs of the character "a."

For Internet applications, including Web pages, many more options are available. Fonts and input method editors (IMEs or means of typing a language, as discussed below) can be downloaded from the Microsoft Internet Explorer site and work with both Internet Explorer and Netscape. Web page developers can also use *font preferences, font linking, Dynamic HTML (DHTML),* or *Java* to provide the appropriate fonts. With font preferences, a developer can designate the preferred fonts in priority order to be used in viewing a Web page; however, such preferences may be overridden by the users' system if the users have selected a priority font order in their browsers. Developers can also provide font linking in such a way that when a user views a Web page, the software in the Web page checks to see if the user has the appropriate fonts and, if not, automatically downloads them from a Web site. Dynamic HTML is a JavaScript application. With this approach, when a user views a Web page, the page delivers the appropriate fonts. Java can also be used to provide fonts from a server. However, with Java—unlike the other methods—a user would not be able to view the fonts offline. Java can also be slow when users are working with low-end modems.

c. Rendering Systems

Special word processing software is usually required to correctly display languages that are substantially different from English, for example:

1. Connecting characters, as in Arabic, Persian, Urdu, Hindi, and Hebrew

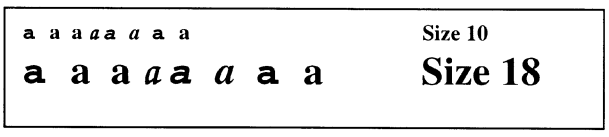


Figure 4 Font glyphs from Fig. 3 in 10 and 18 point type.

2. Different text direction, as in the right-to-left capability required in Arabic, Persian, Urdu, and Hindi, or the right-to-left and top-to-bottom capability in formal Chinese
3. Multiple accents or diacritics, such as in Vietnamese or in fully vowelled Arabic
4. Nonlinear text entry, as in Hindi, where a vowel may be typed after the consonant but appears before the consonant.

Alternatives to providing software with appropriate character rendering systems include providing graphic files or elaborate formatting (e.g., backwards typing of Arabic and/or typing of Arabic with hard line breaks). However, graphic files are cumbersome to download and use, are space consuming, and cannot be electronically searched except by metadata. The second option of elaborate formatting often does not look as culturally appropriate as properly rendered text, and usually loses its special formatting when text is added or is upgraded to a new system. It is also difficult and time consuming to produce. Note that Microsoft Word 2000 and Office XP support the above rendering systems; Java 1.4 supports the above rendering systems except for vertical text.

3. Input Method Editors

For some applications, it is important for the user to be able to enter text. Text entry requires an input method editor (IME). There are many types of IMEs, including the keyboards, handwriting recognition, voice recognition, and special input methods for some Asian languages.

a. Keyboards

The most common IME is a keyboard. Frequently, input is remapped to use different physical keyboards. Often the existing keyboard is mapped to a virtual keyboard (i.e., a picture of a keyboard provided on the screen). There are extensive cultural variations in keyboards. The keyboard most often used in the United States is a QWERTY keyboard, so-called because the five keys on the second row from the top on the left represent these letters. In Europe, the same five keys may be AZERTY or QWARTZ.

There also is considerable variation in the placement of characters on the keyboard space (i.e., the keyboard layout), dating from the variation among typewriter and word processor vendors. Users most often want to use the keyboard layout they know and are reluctant to take the time to learn a new layout. In addition, there may be differing requirements for specialized markets. For instance, users who have learned to

type on a U.S. QWERTY keyboard (including many computer professionals in other countries) often prefer a keyboard where the characters in the language are phonetically mapped to this QWERTY keyboard. Thus the Arab alif (roughly equivalent phonetically to the "a" in English) would be on the "l" key on an Arabic national keyboard but on the "a" key on a U.S. phonetic (sometimes called a "homophonic" or "computer" or "transliterated") keyboard.

Microsoft has tried to adopt national standards where possible. However, in response to customer requests, it has provided phonetic keyboards for some languages, as well as the ability in its latest software (code-named "Whistler") to remap keyboard layouts.

b. Handwriting Recognition

Handwriting recognition is an increasingly popular input method, particularly for Asian languages, such as Chinese, for handheld devices, such as the PalmPilot. It is also supported in Microsoft XP. Sometimes, as with the PalmPilot handwriting input method, user retraining is required. For instance, with the PalmPilot, a "T" is represented as the upper-right corner of a box, without the extension of the top line in what is traditionally known to be the "T" character (Fig. 5).

c. Voice Recognition

Voice recognition is also becoming popular as an input method, particularly for handheld devices. Microsoft XP includes voice dictation. Moreover, ViaVoice has demonstrated voice navigation of the Web. However, while voice recognition is a rapidly emerging technology, it still has substantial limitations. Efficient dictation requires that the user train the system, usually by reading a list of terms to be recognized by the system. Even so, there are often errors, requiring the user to track the written text and select among choices (particularly for homophones or words that sound the same) and/or to make subsequent corrections.

Currently, voice recognition is mainly lexicon based. As a result, the technology is available only in a few languages, and substantial work is required to extend voice recognition to a new language or subject domain. At this time, voice recognition is primarily useful with a limited set of commands, where the system is differentiating between a few hundred choices.

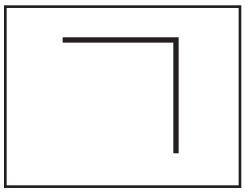


Figure 5 PalmPilot handwriting input for the letter "T."

d. Additional Asian-language IMEs

Voice and handwriting input are popular methods of providing text input for Chinese, Japanese, Korean, and Vietnamese. However, the large number of characters in these languages necessitates IMEs other than the keyboard character entry described above. A common method is phonetic input, which is then converted to the appropriate characters. For instance, in Chinese, users may enter Pinyin, a romanized or Latin-based input, to obtain a selection of characters. The user can then select the appropriate character. Typing compound words or terms in context (such as typing "henhao," meaning "very good," as opposed to "hao" which has the single noncomparative meaning of "good") limits the lookup selection similar to the English compound word "fireplace" limiting the possible meanings of the word "fire."

Phonetic input methods for Japanese include using the phonetic Katakana input method, which converts to a mixture of Katakana and Hiragana Japanese characters when appropriate. For users more comfortable with Latin input, systems usually provide a Romaji input method, which then converts to Katakana, which in turn converts to Kanji and Katakana, and Hiragana.

Other common input methods include the stroke order method, where the keyboard keys are mapped to strokes or graphical pieces of characters. The user then graphically recreates the character. A similar method is the radical input method, where the keys are mapped to common chunks of a character, and the user composes the character from these smaller chunks.

e. Other IMEs

Other input methods include selection of a character from character charts, such as with the Microsoft Office "Insert Symbol" function. Sometimes, a language may be typed off of a different keyboard. For instance, with a single international keyboard, a user can type in multiple languages such as English and French. Many systems also provide a numeric input method. Joysticks, physical keys or buttons, screen buttons, and menu items are additional means for users to provide input, although they are generally not considered IMEs.

4. Menus, Help Files, and Message Files

Menus and screen messages are often localized, where the text is provided in the language appropriate to the users. In some cases—particularly in the secondary markets described above—users may prefer to have menus and messages in a nonlocalized form. For instance, many

computer professionals are familiar with interfaces in English, particularly if they have completed their education in the United States. While they are fluent with the languages of their country, they may not be as familiar with the computer terminology in the languages.

For some languages with different script direction, such as Arabic, the menu items are provided right-to-left across the screen. Thus "File" would be the rightmost item in an Arabic localized system, while it would be the leftmost item in a standard Microsoft U.S. English Office application.

Localization of these items is often more extensive than merely providing translations. For instance, in a system such as Microsoft Office 2000, bolding is designated with a bolded "B." In another language, the letter "B" may be irrelevant (e.g., in German, the term for "bold" is "fett," where an "F" would be a more appropriate symbol). Moreover, many of the shortcuts, which are often based on a letter in the English command (e.g., Control-S for "Save") may be irrelevant in the translation. Extensive examples of these kinds of problems are provided in *Global Interface Design: A Guide to Designing International User Interfaces* by Tony Fernandez.

Particularly effective examples of internationalization are Microsoft Windows 2000 Windows XP. The products were designed to have one product to ship worldwide, with localization content to be added as individual modules. This approach substantially decreased the time to market of the localized markets, making simultaneous product shipments (at least shipments within weeks) possible. The modular approach also simplified administration, since Microsoft needed to handle fewer products. The Microsoft approach had additional benefits of being able to better meet the needs of secondary markets, such as multinational corporations, governments, and universities, by enabling these customers to designate the user interface language separate from the language of the application. In addition, systems administrators in these secondary markets liked being able to change the interface language from the one employed by the user (e.g., Chinese) to one in which they were familiar (e.g., English) in order to install software or troubleshoot problems.

Translation of menus, help files, and message files—as well as translation of other documentation—involves other text issues, including expansion of translated text and use of inconsistent terminology.

a. Expansion of Translated Text

Translated text is often longer than the original text, because of the translator's efforts to adequately explain the meaning of the original. Moreover, terms in European languages are often longer than terms in English, as is shown in Table V from localized instructions for installing a Hewlett-Packard computer docking station. Terms in Chinese may be shorter in horizontal space but require greater vertical space. Designing systems with adequate space for expansion is a basic necessity of internationalization.

b. Consistent Terminology

Use of consistent terminology is a major issue in localization, particularly where different translators are working on the same materials. The appearance of different terms for the same item or action within message files, help files, or documentation can be confusing to the users.

c. Culturally Sensitive Wording

Product names often cause problems across cultures. For instance, MacDonald's needed to change the name of its Big Mac in France, since a "mac" means "pimp" in that culture. Other wording, such as to "abort" a software procedure, can also have offensive connotations, particularly in heavily Catholic cultures.

5. Icons

Appropriate and culturally pleasing graphics often differ between cultures. For instance, the Japanese localized version of Windows 2000 and XP have Office Assistants that include a geisha, who bows and takes notes. This theme is offensive to some women in the United States. As a result, the geisha is not available on U.S. versions of the software. Moreover, icons on U.S. systems are often specific to only a small subculture within the United States. The mailbox used for Macintosh computers, for instance, does not have the same appearance as many other styles of mailboxes in the United States and may be substantially different than the styles in other countries.

Table V Different Text Length in Translations

Language	Text
English	Two PC card slots
German	Zwei PC-Kartsteckplätze
Italian	Dos ranuras de tarjeto de PC
French	Deux logements pour carte PC

6. Applications

The specific applications may have additional localization requirements, particularly in formatting, graphics, content, and approach. If applications involve processing of the language beyond display and input (e.g., spellchecking), it is also helpful to designate the specific language in the metadata of the program or web site.

a. Formatting

Applications in certain languages such as Arabic need to be displayed right to left. Thus, spreadsheets are numbered right to left, and information is entered right to left. Special formatting may be required for other languages as well.

b. Graphics

Appropriate graphics are also important to sales of a product.

A piece of Microsoft clip art showed a Caucasian woman leaning over a Caucasian man, apparently providing directions in an authority or managerial role. Japanese focus groups requested that the artwork be changed to be more culturally appropriate for the localized version, with Japanese figures and with the man in the managerial role. The change was made. However, the focus group still rejected the graphic, since the figure was "obviously a Japanese-American." He was wearing a green shirt. Green was an inappropriate color for a man to wear in business clothing in Japan, so the man must not be Japanese.

Note that significant copyright issues can arise when using existing graphics from the internet or other sources. Under U.S. law, it is no longer necessary to use a copyright symbol or even to have registered materials with the Library of Congress in order to claim copyright ownership. Thus it may not be possible to use existing artwork or other materials without copyright permission from the owners of that material, even if there is no copyright mark.

c. Content

Content is also sometimes localized. For instance, Kohlmeir describes work by the Microsoft Encarta development team to use the name "Islas Malvinas" instead of "Falkland Islands" in the Spanish product, so as not to offend the Argentine market. The team added additional information in areas of special concern (e.g., German reunification for the German edition). They also needed to change or supplement information for the German edition, such as to describe the different sign language conventions in German and the additional wildlife in the German geographic region.

d. Approach

Approach or tone is another item that varies between cultures. U.S. Web sites are sometimes seen by other cultures as overly aggressive or self-promoting, an impression that may have an adverse marketing effect.

When Johns Hopkins established a medical center in Singapore, the company distributed brochures that included terms such as "best patient care available" and "unmatched wealth of advanced treatment." The brochures also described the oncology program director as "internationally known for his expertise in the innovative treatment of breast and lung cancer." While such wording is common in the United States, the Singapore Medical Association in a letter to Johns Hopkins stated, "We view the statements in your brochures to be professionally offensive and culturally insensitive." According to the Singapore medical code of ethics, material should not "disparage, expressly or by implication, the services provided by other practitioners, or claim superiority for the specialist's personal qualities, qualifications, experiences or skills." Research of the culture and review of materials with members of the target audience is critical to providing culturally acceptable Web sites and software.

e. Language

For applications that require more interaction with the language than simple display and input, it is helpful to designate the language in the document or Web site metadata. Values for the metadata tag "LANG" (for "Language") can be found in ISO 639. Two-letter codes are available at http://lcweb.loc.gov/standards/iso639-2/langhome.html. More extensive three-letter codes are available at http://www.w3.org/WAI/ER/IG/ert/iso639.htm. A yet broader set of four-letter codes is now being drafted by ISO.

7. Documentation

Documentation is usually provided in paper and/or electronic form for installation and use of products. For more complex products, different documentation may be provided for systems administrators and technical support, and for users. Efficient translation of documentation involves numerous problems, in-

cluding different paper sizes and the expansion of translated text. Efforts are usually made to separate text from the graphics (or at least to avoid including text as part of the graphic image) to facilitate providing the documentation in multiple languages.

8. Sales and Technical Support

A key aspect of localization is sales and technical support. For localized software products, people in the target market location need to be educated on the products, and appropriate sales and technical support materials and procedures supplied. Such training and support should also be provided to secondary markets. Considerations, particularly with respect to Web marketing and services, include means and hours for obtaining information and support, types of credit cards or other payment accepted, local taxes and tariffs, currency differences, delivery time, and return policies. In some cases, the business may not want to accept soft currencies (i.e., currencies not exchangeable for money from other countries).

Information forms are also substantially different across countries. The U.S. custom of asking for "first name" and "last name" (instead of "family name" and "given name") does not always make sense to Asian cultures that provide the family name first and then the given name. The convention of *state* and *zip code* may also not apply. The term *postal code* is more generic even in English than is the exclusively American *zip code*.

9. Marketing

While Web sites are becoming increasingly popular as marketing tools, paper marketing brochures are still in extensive use. Because of the high cost of such brochures, companies often try to reuse parts of the brochure, such as to use artwork and basic text in English across the United States and Britain. However, there are differences between U.S. and British English not only in spelling and word choice, but also in cultural preferences. In Xerox computer marketing, the European subsidiary requested that the artwork not depict gender and racial diversity (a common practice in the United States), as it clearly marked the image as being of U.S. origin. Of course, specific information on pricing and sales support needs to be added as well, and adjustments need to be made for the difference between the countries in the size of their standard paper.

Web sites are an effective form of marketing and information delivery. However, because Web sites can be viewed internationally, it is important to either plan for global marketing and support or to designate clearly in the site which geographic areas are being serviced. It may also be helpful when targeting specific geographic and cultural markets to provide links from appropriate pages, to advertise on popular portals (e.g., the localized versions of America Online), and to provide special information resources that draw people to one's site. In addition, it is important to keep information current across multiple versions. Further information on Web sites is provided in the next section.

Providing videotapes is another frequent method of marketing. If the videotape is to be provided in multiple languages, it is more efficient to avoid shots of people talking, as such scenes would need to be dubbed or reshot in order to provide localization into other languages. It is also advisable not to record the voice and music on one track, so new voice tracks in other languages can be added without remastering the music track.

III. TECHNOLOGIES

Technologies for localization include Web capabilities, terminology management systems, translation memory systems, and machine translation. These technologies may have overlap and many are specifically designed to interact. Translation memory applications often include terminology management systems; some machine translation systems employ some elements or some of the same functionality as translation memory. Information on specific products is available from the LISA Web site (http://www.lisa.unige.ch/info_localisers.html) and from Multilingual Computing at http://www.multilingual.com.

A. Web Capabilities

The localization issues described above relate to both software applications and to Web sites (see section on fonts for information specific to Web applications). As with menus, help files, and text processing capabilities, the user will need fonts and for some languages (e.g., Arabic) special character rendering software. To input text (e.g., to fill out a form), the user may also need an IME.

Additional information on Web technologies is found in the sections of this article addressing fonts and IMEs. Extensive information on Web site localization can be found under the LISA Special Interest Group (SIG) on this topic: http://www.lisa.unige.ch/sig/web_localization/lisa_gfx.html and from the Web Internationalism and Multilingualism (WINTER) site at http://www.dragoman.org/winter/.

B. Terminology Management Systems

Terminology management systems such as TermStar or MultiTerm are used to manage the specific terms used in localization, together with related support information. Providing consistent terminology within one set of documentation, across upgrades, and across similar products is important for providing materials helpful to the users and for providing a professional image for the company. Such consistency is often difficult just within one language, such as English. Providing it across multiple languages increases the problem, particularly since there may be multiple ways of translating a particular term and since multiple human translators are often working on the documentation.

C. Translation Memory

Translation memory (TM) tools, such as TRADOS Translation Workbench, Transit STAR, or SDLX, provide assistance to the translator by retrieving earlier translations of material (called translation units or TUs), which are usually phrase-l or sentence-level chunks of text. TM tools may search substantial bodies of past text, presenting absolute or partial (i.e., "fuzzy") matches. This retrieval or reuse process may sometimes propagate a bad translation; however, regular editorial review of the materials in the system is recommended to ensure a higher level of quality.

Some tools flag inconsistencies or other problems that the human translator should address. In addition, TM systems may compare versions of text to determine if any changes have been made (e.g., if a developer has added a new line to the original English documentation). Finally, such systems provide means of tracking the complicated process of translation, editing, revision, new translation due to changes in the original text, and new revision.

D. Machine Translation

Machine translation (MT) is the technology of translating one human language into another (e.g., English into French) through full automation or with human assistance. The speed of MT and the process of investing in infrastructure are appealing to businesses. However, MT often provides inaccurate translations, particularly where there is more than one meaning for a word in the original language or where there is more than one interpretation for a term in the target language. Thus "bank" could be a riverbank or a financial institution. The use of domain-specific dictionaries (e.g., agricultural vs. financial dictionaries) is helpful but not sufficient. MT often is not as fluent and readable as human-translated text, and thus may not convey the professional business image that a business would prefer. However, MT combined with review by human translators can be a cost-effective procedure, particularly when a company invests in building company- and technology-specific user dictionaries.

One means of improving the quality of the translations is to use *controlled language* for the input. The controlled language may range from a list of terms acceptable in the translation to grammar checkers that alert the author to types of writing (e.g., incomplete sentences) that may not work well with MT. The writer could then change terms or sentence constructions or flag a term to be added to the MT dictionaries. Some proponents of controlled language claim that it produces more consistent and better written text in the original language, as well as better machine translations. However, some technical writers and programmers object to the additional writing burden, although carefully designed authoring tools and limitation of rules sets can diminish the effort involved in conforming to controlled language parameters.

Translators can also use *pre-editing* tools such as spellcheckers and grammar checkers to provide cleaner text for the machine translation. Careful *post-editing* (review and correction by human translators) is strongly recommended, particularly for materials that are used to convey a professional image of the company or for materials that may have legal implications.

Many MT systems are making increasing use of TM systems. In addition, many MT systems have been particularly effective because—like TM—they capture single translations of terms in their dictionaries (especially user dictionaries) and thus provide standardization.

MT is also attractive in that much of the work of identifying and translating terminology can be conducted before the original documentation, menus, message files, or other text is finalized. Since the original text is rarely finalized until very close to product launch, and since there is substantial marketing pressure to provide the localized versions as soon as possible after the launch of the first product, MT enables the localization team to expedite the localization.

Another application of MT that is being used in localization is the translation of web sites. As with documentation, MT of web sites may expedite the work of human translators, who need to review and edit the work. However, nonreviewed MT may result in miscommunications, a poor business image, and potential legal issues. If unreviewed MT is to be used, it may be more effective to provide a link to machine translation

(e.g., with SYSTRANlinks from http://www.systransoft.com) accompanied by a disclaimer about machine translation. In this way, the business provides a useful tool to the user, but also introduces some buffer against the problems listed above.

IV. TRENDS

The field of localization was estimated by LISA to be a $2.4 billion market in 1998, with an expected 12–15% growth per year. More companies are localizing their products, and more are doing so in an increasing number of languages. The expansion of localization efforts, combined with new technologies, is resulting in increased professionalism, business consolidation, creation of standards, and integration of localization tools with other technologies.

A. Professionalization

In the last decade, an increasing number of professional organizations, conferences, and educational resources have emerged. The Localisation Industry Standards Association, established in 1990, now has more than 180 industry members and holds annual conferences. An education outreach program—the LISA Education Initiative Taskforce (LEIT)—was started in 1998 to provide materials and guidance to the emerging university programs in this area. LISA also conducts surveys on best practices in localization, and has created a primer and guide to help educate businesses on good practices in this field. The International Workshop on the Internationalization of Products and Systems and the Global Management Strategies Conference were both started in 1999 in order to provide forums for discussion of localization issues and sharing of information.

B. Business Consolidation

An item of discussion at recent localization conferences has been the increasing acquisition and consolidation of localization companies. More information is available from the LISA home page.

C. Creation of Standards

Standards are becoming increasingly important with the emergence of eXtensible Markup Language (XML). XML enables communities of interest (such as the globalization industry) to create relevant metadata tags. It also enables the creation of text which can then be used and displayed in multiple ways, with the use of eXtensible Stylesheet Language (XSL). Goals of the globalization community include improving data exchange between terminologies and/or lexicons.

1. Improving Terminology Exchange

Developing standards in order to better share data created in one terminology management or TM systems across other systems has been the focus of several international efforts. This ability to use dictionaries from other systems would significantly enhance the effectiveness of terminology management systems. The ISO Technical Committee (TC) 37 on Terminology has issued ISO 12200, an SGML-based Machine-Readable Terminology Interchange Format (MARTIF), which is currently evolving into an XML-based standard (see http://www.ttt.org/oscar/xlt/ DXLT.html). LISA's OSCAR SIG (Open Standards for Container/Content Allowing Reuse Special Interest Group) has approved a Translation Memory eXchange (TMX) standard (http://www.lisa. unige.ch/tmx/index.html). OSCAR has also adopted the xlt/DXLT standard TermBase eXchange (TBX).

2. Improving Lexicon Exchange

Developing standards designed to facilitate the sharing of lexicons across machine translation systems is aimed at improving the quality of machine translation systems. This effort is being led by a consortium of machine translation companies to develop the Open Lexical Interchange Format (OLIF). Further information on OLIF is available at http://www.otelo.lu/.

3. Improving Terminology and Lexicon Exchange

Ideally, terminologies and lexicons would be shared across terminology management systems, translation managers, and also machine translation. LISA's OSCAR SIG and Standards-Based Access to Lexicographical and Terminological multilingual resources (SALT) are working to harmonize the TMX and OLIF standards with a new XML for Lexicons and Terminologies (XLT). More information on OSCAR can be found at http://www.lisa.unige.ch/tmx/index.html and on SALT at http://www.ttt.org/salt/.

4. Other Standards

ISO TC 37 has produced and is working on standards for terminology, which can be tracked at http://www.iso.ch/ iso.en/stdsdevelopment. In addition, the

Unicode Consortium (see http://www.unicode.org) produces recommendations for standards, which are sent to the World Wide Web Consortium (W3C at http://www.w3.org/International/) and then in some cases ultimately to the ISO and IEC.

D. Improved Technologies

Localization technologies are steadily improving, particularly with the development of standards to enable the sharing of resources. Localization software libraries in Java and in C++ (such as in Sybase's Unilib) have also become available and continue to add functionality.

E. Increased Employment of User Profiles

The ability to designate user profiles is already revolutionizing the localization of systems and applications, particularly browsers and e-mail. For instance, with computer operating systems such as Microsoft Windows 2000 and XP, it is possible to create a user profile that will appear whenever the user logs onto a specific network. With emerging technologies, a user should also be able to designate language choice and cultural elements (clip art, icons, etc.) Moreover, as machine translation improves, it will likely become possible for users to designate the languages in which they would like to receive e-mail and perhaps also to designate such menu items as "Return e-mail in language of original message." Similarly, if a user has a language and geographic area designated in a user profile, it will become increasingly necessary for companies and organizations to provide localized information to browsers and handheld devices connected to the Internet.

F. Increased Integration of Tools

As more tools are developed and/or become localized (e.g., search engines localized to retrieve and summarize Japanese text), demand may increase from consumers to have the tools integrated into applications, Web portals, and other tools.

G. Increased Interest in Multilanguage Market

While localization has generally focused on tailoring software to a particular language and culture, there has been some market recognition of users who have multiple language needs and/or who need to work in languages in which they are not fluent. For instance, with the Windows 2000 and XP operating system, a user can select the language of the user interface and separately select the languages used in word processing. Thus a user may have an English interface, but create a single document with Arabic, Japanese, and English. Moreover, as users face increasing needs to communicate or access information across languages, there is growing interest in tools such as AltaVista's BabelFish, which uses AltaVista's search engine but offers SYSTRAN machine translation to translate the items returned. Chat rooms such as those provided by SlangSoft and SYSTRAN are also incorporating machine translation. The field of globalization may well evolve to serve more of the global needs of users.

SEE ALSO THE FOLLOWING ARTICLES

Developing Nations • Digital Divide, The • Ethical Issues • Global Information Systems • Globalization and Information Management Strategy • Internet, Overview • National and Regional Economic Impacts of Silicon Valley • Year 2000 (Y2K) Bug Problems

BIBLIOGRAPHY

Carmel, E., and Collins, E. (1997). The impact of international copyright management and clearance systems on multimedia markets. *Telematics and Informatics,* Vol. 14, No. 1, pp 97–109.

Day, D., del Galdo, E., and Prabhu, G. (2000). *Designing for global markets 2: Second international workshop on internationalisation of products and systems.* Rochester, NY: Backhouse Press.

del Galdo, E., and Nielsen, J. (Eds.) (1996). *International user interfaces.* New York: John Wiley and Sons.

Esselink, B. (2000). *A practical guide to localization.* Amsterdam: John Benjamins Publishing.

Fernandez, T. (1995). *Global interface design: A guide to designing international user interfaces.* Boston: Academic Press.

Hall, P. A. V., and Hudson, R. (1997). *Software without frontiers.* New York: John Wiley and Sons.

Localisation Industry Standards Association (2000). *The localization industry primer.* Féchy, Switzerland: Localization Industry Standards Association.

Multilingual Computing and Technology Magazine. Sandpoint, ID: Multilingual Computing Inc.

Ott, C. (1999). *Global solutions for multilingual applications: Real-world techniques for developers and designers.* New York: John Wiley and Sons.

Sprung, R. C. (Ed.) (2000). *Translating into success: Cutting-edge strategies for going multilingual in a global age.* Scholarly Monograph Series, Vol. XI. Alexandria, VA: American Translators Association.

Globalization and Information Management Strategy

Jatinder N. D. Gupta
University of Alabama, Huntsville

Sushil K. Sharma
Ball State University

THE INCREASING GLOBALIZATION OF BUSINESS has led firms to seek new, and more appropriate, organizational structures, processes, and cultures. This requires the establishment of appropriate information management strategy to coordinate business processes and to provide coalition mechanisms. The globalization trend requires that companies adapt to changing environments and develop new skills and tools that enable them to work, compete, and cooperate in a global context. Information management strategy in a borderless economy should be based on competitiveness of the company and technological foresight. Much has been written over the last decade on the globalization aspect of organizations. However, little literature is available on how to develop strategies for information management in the global business scenario. This article describes various business drivers for globalization and analyzes the value chain and organizational relationships of international business, strategic management and information system disciplines. It also suggests a framework for information management strategy.

I. INTRODUCTION

Organizations in the 21st century are moving toward global forms of organizations that transcend traditional national boundaries. The key forces driving businesses today are changing from a commodity focus to a customer service focus, improving return on capital, training new knowledge workers, managing diverse global operations, building winning alliances, applying new technologies, managing risk, and planning sustainable development strategy. In such an environment, organizations need to integrate their disparate systems that are housed or located at many different regions. Organizations need an appropriate organizational framework to integrate various technological environments, business processes, and even products and services to coordinate the activities of their sales, manufacturing, and logistics worldwide. Organizations need an effective information management strategy to manage global business activities. It may necessitate changes in organizational structures and information collection policy, ultimately changing the information management strategy.

The article is organized as follows. In Section II, we first describe the business drivers for globalizations. Following this, we discuss the framework for information management strategy and various approaches for information management in Section III. Our discussion of information management includes the possible implementation approaches in a global context. This discussion of information management strategies is followed by the identification of the challenges and obstacles to effective information management in Section III. Finally, Section IV concludes with some closing remarks.

II. BUSINESS DRIVERS FOR GLOBALIZATION

A *business driver* is a force in the environment that influences the direction of business. There are many business drivers that are driving organizations towards

globalization. Changes in competitive environments, changing business models, changing customers, changing technology—advancements in information and communication technologies—and collaborations and alliances are the main business drivers of organizations toward globalization. Each of these is briefly discussed in this section.

A. Changes in Competitive Environments

Traditional business models are no longer sufficient to tackle fierce global competition. In order to survive in global competition, organizations need different business methodologies that can help them to cut costs, improve the quality of products and services, and create value for the businesses. Technology has become critical in creating and sustaining competitive advantages in the global market. The increasing globalization of business has led firms to seek new, and more appropriate, organizational structures, processes, and cultures. This has required the establishment of appropriate information technology platforms to coordinate business processes and to provide coalition mechanisms. The various factors reflecting changes in competitive environments are:

- Globalization of markets
- Continual innovations of products and services
- Buyer oriented markets
- Business through cooperation, alliances, mergers, and acquisitions
- Virtual enterprises—new form of organization
- New business models
- Higher quality expectations
- Demanding customers—customers are more informed and seek more information
- New products and services

B. Global Markets

Organizations have to choose the Internet or Web as a medium to reach customers and conduct business. Technology, the Internet, and revenue models are key to the survival of many organizations in global competition. Today, every organization focusing on global markets has a Web site to offer products and services online. Online business must have a Web site that does more than simply receive online orders or market the company's products and services. The Web site must also be compatible from a technical standpoint and have a viable revenue model. An effectively designed Web site or virtual storefront can provide the fastest way for the company to expand the size of its audience, build a strong corporate brand, and most importantly maximize advertizing revenues.

C. Changing Business Models

The globalization process is changing the way businesses are conducted. The relationship and interaction of various stakeholders such as customers, suppliers, strategic partners, agents, or distributors is entirely changed. The key to survival in the new e-business environment depends upon organizations' ability to adapt to a new, more collaborative, corporate-competition model.

D. Changing Customers' Expectations

The rapidly changing business climate around the world has changed the perspective of the organizations regarding their offerings and customers. Customers are becoming more demanding since customers now have greater awareness of standards of service and have higher expectations when seeking information at times of a significant event. The explosive growth in the use of the Internet has created two main business drivers—increased competition and the need for companies to distinguish their relation with customers, and, just as important, heightened expectations of a more "information savvy" consumer. The Internet revolution has created transparency of transactions and heightened consumer expectations with regard to service. Internet based technologies have empowered customers and today customers want real-time access to data in a format that can be easily uploaded across systems. Many companies are offering opportunities for customers to produce their own products and services by providing an interactive mode. Today, the decisions that used to be made internally are made with and by outsiders—customers or the market as a whole. The easy access that buyers have to competitive information is placing pressure on prices and is encouraging customers to search for substitutes.

The Internet has amplified the shift of power to the customer. Today the customer has timely and immediate access to information regarding just about any subject including products and services. Customers can not only compare attributes of competing products and services, but can make direct contact with the individual and firms with whom they wish to do business.

E. Changing Technology

Advancements in information and communication technologies, the Internet, and other Web technologies have provided many opportunities for the most dramatic economic turnaround in modern times. The Internet is helping to globally expand markets at a low cost. Its facility to communicate information instantaneously across the globe has provided the opportunity for consumers to readily find and compare information about products, services, and prices in a global market. Today, information is transferred so easily across the globe that products and processes can quickly be copied or imitated. The continuous rapid development and innovation have shortened the product life cycle to reinforce competitive pressure and the need to invest more in innovation.

The information technology (IT) revolution has brought many changes in the organizational environment. As the organizations use more technology, they can expand their operations and activities to a global domain. Traditional barriers of international trade are breaking down and new global markets are emerging. Since organizations are competing on global terms, globalization can greatly complicate the task of managing IT in a firm. The IT has demonstrated an ability to change or create the following:

- New business models
- New products and services
- New workflows and procedures
- New relationships with customers and suppliers
- New partnerships and alliances
- Increased span of control
- New form of organizations—"Virtual Organizations"

IT has the potential to create market driven organizations. Flexibility and rapid response are key to success in the new Internet-driven economy for both businesses and customers. The increasing pace of change in market conditions places a premium on building flexible organizations in the business world. Market driven organizations are those organizations which can respond to the new business environment and that have a fluid and flexible internal structure which can respond quickly to changed circumstances. Organizations should focus on their core competences, shedding distracting side businesses and internal production capabilities that could be better performed by outside providers. The priority of market driven organizations should be customers, their preferences, and long-term relationships.

F. Collaborations and Alliances

The nature of the global business environment is changing. Today's companies face a multitude of simultaneous challenges: shorter product life cycles, rising cost pressures, and a growing need to respond to consumers' preferences quickly. Due to increasing competition and the desire to reach global markets, collaborative efforts and alliances have become necessary. Often these challenges place conflicting demands on organizations to offer products and services as a single company. It is a fact that some companies can perform the same business functions for a substantially lower cost. Striking the right alliances with partners that have complementary capabilities, products, and/or markets can be critical for business growth and success. Organizations can integrate their dispersed markets through coordination of their alliance partners. Partnerships can also create tremendous opportunities for the cross selling of products and services through strategic partnerships. To tackle global competition, organizations have to develop international alliances in a win–win situation to create economies of scale.

G. Technology and Change

Globalization, collaboration, and technology are the three most important elements currently driving business. As the competitive environment becomes fiercer, the use of information technology differentiates the leading players from the rest. Thus, it is imperative for the organizations to use new technologies such as the Internet to offer products and services in the global market in a more specific and focused manner. This task cannot be completed alone with its own investment, resources, or technology. It calls for collaboration among technology and business partners. In fact, today, companies can only survive if they incorporate a new management process that encompasses the belief that information systems and business management must act as true partners.

Technology and particularly the Internet or Web technologies are bringing many changes in the organizational environment. Technology is assisting companies

- At their *boundaries,* enhancing environmental information capture (e.g., through databases or public forums) and opportunities for mobile workers (e.g., sending employees into the field to be close to their customers, while maintaining links with internal information systems)

- In their *relations* with partners (e.g., agile corporations), suppliers (e.g., EDI) and targeted customers (e.g., interorganizational information systems)
- In their access to *markets,* e.g., through information malls or electronic markets

III. INFORMATION MANAGEMENT

The goal of information management is to ensure that an organization obtains the maximum real value from its information resource. Information management includes activities like the acquisition, utilization, accessibility, and dissemination of information. The challenge is to integrate information from various levels across processes and organizational boundaries. Information resides at many different levels, in different databases on multiplatforms. The main objective of information management is to satisfy the demand for information and thus deliver value to the business. Demand is expressed in information requirements and the value is expressed in many ways. Business value is derived through the following activities.

1. *Improving existing processes within an organization,* by improving product promotion through mass-customization and one-to-one marketing, offering new direct sales channels for existing products, reducing the cost of some processes (e.g., information distribution), reducing the time to market, improving customer service through automated service and round-the-clock operation, and finally improving the brand image, by offering electronic access to customers.
2. *Transforming the way companies deal with customers,* by accumulating knowledge on their detailed preferences and buying habits, targeting them with specific offers, and in general dealing with them in a more personalized way.
3. *Redefining the products, processes, and business models used today,* by leveraging technology to fundamentally change the ways products are conceived, marketed, delivered, and supported.

The objectives of information management, therefore, are to provide an appropriate set of information tools and technologies, policies, resources, processes, and services to carry out business objectives efficiently. The planning of the information tools and technologies should be such that it improves cost effectiveness, responsiveness, quality, user productivity, and so on. The objectives of information management include the following goals.

- Build and manage a global communications infrastructure to serve the entire enterprise
- Provide an appropriate technology infrastructure for current and future business needs
- Provide a high degree of integration and consistency across the infrastructure to minimize cost and maintain quality
- Support users for improvement in their productivity
- Reduce complexity and nonstandardization to ensure flexibility and responsiveness both at local and global levels

A. Components of Information Management

The components of information management include the technology infrastructure (physical infrastructure and architectures), policies and management processes, and services required for meeting the business needs of the organization.

1. Technology Infrastructure

The technology infrastructure includes hardware and software components to support the applications and information management requirements of the business. The components of technology infrastructure are

Computer hardware
Systems software
Communication and networking systems
Development tools
Application software
Special purpose tools

The technology architecture maps the physical infrastructure with information, processes, and organizational structures. Architectures indicate how the various applications, information stores, and linkages are mapped on the physical model. The types of architectures shown in Table I could be involved in information management.

2. Management Processes and Services

These are the services provided internally or by outsourced suppliers to support technology infrastructures. These include planning and management of the facilities, vendor management, and the technical support for all users.

Table I Architectures and Functions

Architecture	Functions
Information architecture	Identify information requirement for business needs at all levels
Applications architecture	Set applications standards, ensure appropriateness
Data architecture	Coordinate development/establishment of common database management processes
Hardware/operating systems architecture	Specify/monitor the heterogeneous hardware and operating systems
Telecommunication architecture	Telecommunication technologies, integration, and performance

3. Policies

Policies determine how the infrastructure and its support services are managed. These could be security, audit, sourcing, and other contingency plans, etc.

B. Systems and Technologies

The various systems and technologies which need to be managed in an organizational perspective are given in Table II.

1. Application Service Providers (ASPs) and Technology Outsourcing

There is no doubt that technology is becoming the cornerstone of business growth. Many companies are not able to implement state-of-the-art technologies because they lack expertise. Under such circumstances, an increasing number of companies have started using application service providers (ASPs). Application service providers have come to symbolize a new way of delivering information technology services. An ASP is essentially an application that is rented instead of purchased. ASPs provide business solutions for small, medium-sized, and large companies on a rental basis. Rather than spending millions on a new enterprise-wide software implementation, the organizations pay on a per-transaction or per-month basis for someone else to host, run, and manage the software. ASPs have the entire responsibility of managing the infrastructure, hiring manpower, and helping to conduct business online. The costs of supporting any large scale Internet presence across multiple customers, distributors, and suppliers are beyond the capabilities of most IT organizations. With more complex business requirements, IT executives will increasingly look to data hosting centers and application service providers to provide and manage all the resources needed for a major online presence.

2. Web Portals

A Web portal provides an opportunity for connecting businesses through a technology window. Enterprise architects are designing Web services into their system designs. A Web portal is seen as a gateway to communicate with internal as well as external users of products and services. Companies with portal sites have attracted much stock market investor interest because portals provide the company's information both internally and externally and are viewed as able to command large audiences and numbers of advertizing viewers. Portals provide a single point of entry for all the disparate sources of knowledge and information both within and outside an organization, usually through the Internet or a company intranet. Portals are fully integrated with legacy systems, to support collaboration and teams working across diverse communities. Many companies have met these challenges and are developing strategically important portals for competitive advantage. Portals can help companies to interact with their customers, business partners, suppliers, and employees through online tools.

3. Streaming Technologies

Streaming desktop video is one of the recent trends to integrate desktop players (such as Windows Media Player, RealNetworks, and Liquid Audio) and deliver video and audio directly to the browser and company's information systems. These services help to reduce distribution costs and can provide interactive capabilities such as chat, polls, graphics, Q&As, etc.

Table II Systems and Technologies for Information Management

Systems	Technologies
Office automation	DBMS—Oracle, Sybase, DB2, SQL Server, etc.
Online real time systems	Internet, intranet, extranet
ERP, and other legacy systems	ERP software such as SAP, BAAN, Peoplesoft, etc.
Customer relationship management systems	CRM software
Knowledge management systems	Data warehousing, database management technologies, document management systems
Web portals	Web technologies—XML, Java, etc.
Learning and intelligent organization based systems	KM tolls, expert systems, AI based systems
Outsourced software	Applications service providers (ASPs)
E-business	B2B Portals, B2C Portals, and C2C Portals, e-procurement
Visual applications development	Streaming technologies
M-commerce	Wireless technologies
Supply chain automation	ERP and SCM softwares such as SAP, BAAN, Oracle, etc.

4. Mobile Web Devices

Personal digital assitants (PDAs) and digital phones are used to access corporate data and the Internet for business or information purposes. Some companies even offer benefits, enrollment, account management, and corporate directories via hand-held devices. With millions of Internet appliances ranging from phones to hand-held devices and pagers now being used by consumers, the opportunity to completely reinvent tarnished business-to-consumer e-commerce models is great.

5. E-Procurement

Buying on the Internet is a growing trend. Organizations are finding that electronic systems that automate purchasing help them to procure items at the lowest price without the hassle of coordinating purchasing activities within the organization.

6. ERP Systems or Foundation Systems

Most companies today have enterprise resources planning (ERP) systems that are used for transactional processing. ERP systems are divided into functional areas (financials, order fulfillment, manufacturing, distribution, procurement, HR, etc.). Few companies have not implemented ERP systems and many enterprises still have fragmented ERP implementations that are mixtures of ERP and legacy systems.

7. Internet, Intranets, and Extranets

Companies are relying heavily on Internet or networks for conducting their day-to-day business activities. E-mail, groupware, intranet, extranet, and the Internet all have become common platforms to conduct business. The characteristics of new e-merging businesses are those which aggressively link to their customers, suppliers, and business. They do this through the Internet using

(a) websites to handle sales transactions and provide customer service
(b) intranets and enterprise data portals to link employees and give them more access to data
(c) extranets—one of the fastest growing segments of the Internet—which allow information to flow to and from business partners

Through this kind of electronic supply chain, companies can reduce inventory costs by shortening the time it takes customers to order and receive products.

The labels associated with these processes are becoming commonplace: e-procurement, ERP systems, supply chain management, voice technology, call centers, customer relations management, and data and workflow management.

8. E-Business Technologies

E-business is about constantly transforming market spaces and business models by innovating corpora-

tions in time to compete. This is powered by Internet technologies, intellectual capital, and the quest for value. The power of transformation is the essence of e-business investment, but such power has yet to be adequately demonstrated through the existing implementations of e-business models. An e-business model is an approach to conducting electronic business through which a company can sustain itself and generate profitable revenue growth. The business model spells out how a company plans to make money online and how it's competitively positioned in an industry.

9. Wireless Technologies and M-Commerce

The world is moving toward wireless technology. In the last few years, wireless networks for voice communications have enjoyed tremendous expansion and have changed the lifestyles of people. Now people try to gain remote access to Microsoft Exchange or Lotus Notes server while they are mobile. Wireless networking promises both greater work productivity and increased flexibility in our lifestyles. Using wireless technologies, mobile commerce, or m-commerce would enable consumers and business people to shop, bank, close business deals, and even pay for vending machine goods with portable Internet devices. M-commerce implementation is inevitable. Several technologies are being developed to support m-commerce. WAP, General Packet Radio Service (GPRS), and Bluetooth are various forms of wireless technologies that make m-commerce possible.

10. Learning and Intelligent Organizations

The learning organization model is emerging to help firms plan and execute significant organizational change amid rapidly changing business conditions. Learning organizations are generally described as those that continuously acquire, process, and disseminate knowledge about markets, products, technologies, and business processes. This knowledge is often based on experience, experimentation, and information provided by customers, suppliers, competitors, and other sources.

The technology of the future will enable organizations to be cheaper, faster, more flexible, and more competitive. Organizations that are able to align the collective productive and creative energies of their people through teams will be able to maintain and expand their competitive advantage

11. Supply Chain Automation

A company's supply chain encompasses the facilities where raw materials, intermediate products, and finished goods are acquired, transformed, stored, and sold. These facilities are connected by transportation links, along which materials and products flow. Supply chain management is the coordination of material, information, and financial flows between and among all the participating enterprises. Modern supply chains using Internet based technology enable information sharing between various enterprises. With such a system, decision makers along various enterprises in the supply chain have a common base of readily available sales and inventory information. Automating supply chains, therefore, drives costs out of the production process and provides a competitive advantage to a firm.

12. Customer Resource Management (CRM)

Customer resource management is aimed at acquiring new business customers, enhancing profitability of existing customers, and retaining profitable customers for life. Companies are using technologies to profile their most valuable customers and adapt marketing and product development strategies to target those customers. The Internet has enabled companies to reach widely dispersed customers across all geographical boundaries and time zones. As a result companies have had to deal with an increasing volume of customer contact through the expanding channels of communication. To deal with the growing complexity and magnitude of customer interactions companies are using CRM systems. In a down market, generating more revenue from each customer engagement becomes a higher priority.

13. Content Management Technologies

Content management technologies would take care of requirements of KM initiatives through a process that involves capture, storage, access, selection, document publication of text retrieval, and document management.

14. Business Process Integration

Organizations are using various technologies for their business processes integration. Business-to-business e-commerce has become the true dominant business model where industry as a whole is integrating enterprise applications.

15. Optical Computing

Customers now expect richer content on Web sites. Researchers are working on revamping much of the

Internet core with optical networking technologies in order to meet increasing bandwidth demands driven by the need to deliver richer content on Web sites.

16. Knowledge Management (KM)

Knowledge management seems an obvious imperative in the knowledge economy. Facing competitive pressures and a bleak economic climate, organizations need access not only to technology resources but also to the vast intellectual capital residing within employees and data stores, knowledge that can be capitalized on to bolster decision-making abilities. Early adoptions sharing data via groupware systems and intranets have helped facilitate knowledge sharing but have not gone far enough. Knowledge management is seen to be central to product and process innovation and improvement, to executive decision making, and to organizational adaptation and renewal.

Today, intellectual capital, rather than physical capital, is the driving competitive force for companies in virtually all industries. Businesses are investing in a wide range of KM "solutions" to exploit the power of their intellectual assets and translate them into real value. Knowledge management is all about finding and translating experience, instinct, and values into documented knowledge that can be delivered throughout the supply chain. While the first wave of information technology focused on the creation of the IT infrastructure in the form of networks, the next wave will focus on enabling people to use knowledge to add value to a company's products and services. Globalization and intensifying competition have placed a premium on highly specialized expertise. The challenge lies in using knowledge to create new value. The market for KM products and technologies is increasingly driven by companies seeking to improve their efficiency to save costs, increase revenues, and ensure competitiveness. Many are already implementing successful KM strategies and reaping benefits in terms of greater return on investments, improved productivity, and increased customer and employee satisfaction.

C. Strategies for Information Management

Strategy understands an industry structure and dynamics, determining the organizations relative position in that industry, and taking action to either change the industry's structure or the organization's position to improve organizational results. A strategy is the general direction in which an objective is to be sought. Strategic business planning is a process that uses competitive strategies to allocate its resources to projects that can exploit industry opportunities or defend threats caused by change in the marketplace for the purpose of meeting long-term objectives of the organization. Every organization operates on a Theory of the Business. Strategy converts this Theory of the Business into performance. Its purpose is to enable an organization to achieve its desired results in an unpredictable environment. The Internet provides a better technological platform than previous generations of IT. Gaining competitive advantage does not require a radically new approach to business; it requires building on the proven principles of effective strategy. Alignment between business strategy and IS strategy is widely believed to improve business performance.

The key objectives of 21st century organizations are conducting businesses online, innovating, ensuring low cost production, and striving for customer excellence. Many organizations have state-of-the-art technology but despite that many of those organizations suffer from the following typical technology pitfalls:

- Failure to integrate technology with business strategy
- Technology not linked with customer offerings and services
- Systems remain as fragmented systems

Organizations need to leverage technology for their competitive advantage. Technology solutions should not only provide faster information retrieval and access but should have the ability to leverage information into knowledge over a period of time. However, the return on IT investments should not be measured in just financial returns but should also be measured in improvement of customer retentions and higher satisfaction of customers' expectations.

Information management strategy is about managing the information resources and the IT infrastructure and services that deliver maximum value to the business. It embodies policies, organizational provisions, and activities that help to develop and manage information resources. Information management includes activities of information acquisition, utilization, accessibility and dissemination. Porter suggests that gaining competitive advantage does not require a radically new approach to business; it requires building on the proven principles of effective strategy.

1. Globalization and Information Management Strategy

Due to technology, information revolution, and high competition, companies need to become global in

their thinking and activities. The current wave of globalization in business has encouraged corporations to partner with regional firms to consummate global strategies. The increasing globalization of business has led firms to seek new, and more appropriate, organizational structures, processes, and cultures. This has required the establishment of appropriate information technology platforms to coordinate business processes and to provide coalition mechanisms. Businesses face many challenges in today's rapidly changing environment. Some of these challenges are improving links between information systems strategy, and business strategy, developing and implementing information architecture, implementing knowledge management systems, and reducing IT projects' completion time and budget deviations.

The rapid globalization of business creates significant challenges for information management. Many disparate systems and processes need to be coordinated for effective information management. Information management strategy can be focused in three areas: (1) overall goals and strategy of the organization, (2) the specific requirements for functionality, and (3) the people and processes management. Criteria for information management should flow directly from the organization's business and strategic plans. A business plan defines the goals and objectives of the organization and explains whom the organization serves and how. A strategic plan provides the roadmap or blueprint the business will follow to meet the goals and objectives of the organization. Overall information management strategy will depend upon how information will be captured, maintained, and delivered. The main factors affecting information management strategies can include:

Costs—the economics of developing and maintaining systems
Customer access and service
System integration and customization needs
Security and reliability
Data ownership and access
Heterogeneity of technical infrastructure—software platform and hardware platform
Availability of technologies and vendor services in different countries
The existing IS/IT investments in the different business units
The availability of skills and human resources

However, organizational and political factors can outweigh the above-mentioned logical factors for information management strategy. The rationale behind appropriate information management strategy depends upon how much value addition to the business takes place opting one or the other approach.

2. A Framework for Information Management Strategy

Globalization trends have resulted in a variety of organizational designs that have created both business and information management challenges. As shown in Fig. 1, a framework for information management strategy could be based on the following three major strategic components.

1. Centralized or coordinated data communication architecture on establishing standards
2. Centralized or coordinated data management strategy for creation of corporate databases
3. Alignment of global business and global information systems strategy

Centralized or coordinated data communication architecture on establishing standards describes where applications are executed, where databases are located, and what communication links are needed among locations.

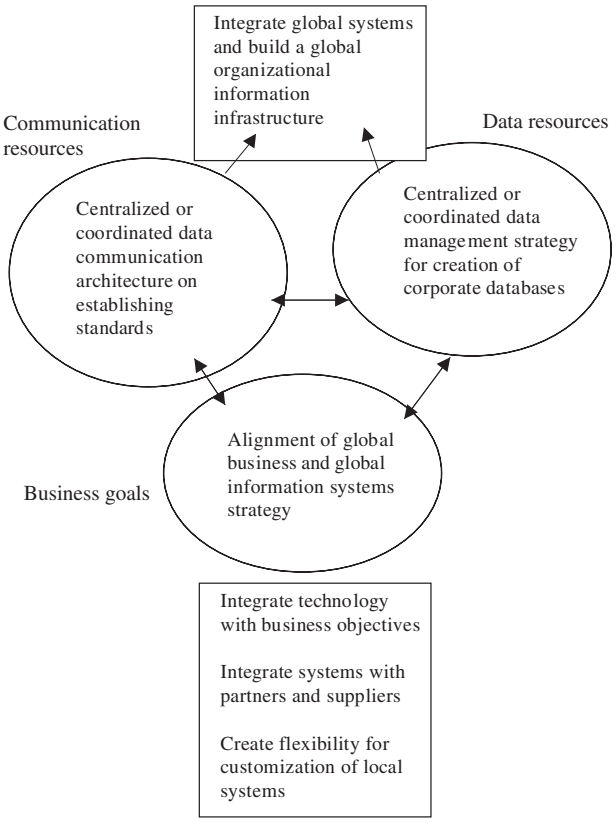


Figure 1 A framework for information management strategy.

This architecture should also set standards for interconnecting various systems.

Data architecture concerns the arrangement of databases within an organization. Organizations need to devise a strategy for creation of corporate databases derived from the firm's value chain activities. Corporate databases should be based on business entities involved in value-chain activities. Therefore, data architecture strategy should consider the flow of data among corporate, functional, and application databases. Organizations need proper linkages between business planning and technology planning. Organizations have to integrate technology with business objectives and also integrate systems with partners and suppliers for organizational effectiveness, efficiency, and performance. Global strategy is defined by Porter as the strategy from which a firm seeks to gain a competitive advantage from its international presence through either a concentrated configuration of activities or coordinating among dispersed activities, or both.

D. Information Strategy and Business Strategy

Twenty-first century organizations have to engage in collaborative, real-time connections with customers, suppliers, and business partners to perform their business activities. The markets demand an entirely new business culture and set of processes and a new business strategy for global competitiveness. Companies may have to create entirely new business processes so that customers, suppliers, and partners can be brought closer to integrating their processes. Companies may use an information management strategy to attack the market, to change products, to improve processes, and to restructure their organization. This use of information management resources to accelerate market responsiveness and to drive higher levels of value creation creates an intensity of competition and increased optimization never before seen in markets. Companies will be able to better leverage financial, human resource, and brand capital to achieve dramatic new results if information management strategy is effective. The successful companies in the New Economy are following entirely new business models and ideas, and reshaping their business strategies with the creation of an effective information management resources infrastructure. Strategy for companies today is a process that can match the dynamism of the market. As more companies use technology to increase their speed and efficiency, e-business tools and techniques are becoming a necessity.

Aligning business strategy and IS strategy, the alignment of global business and information management strategies can be formulated as shown in Table III. Information management strategy needs to address organizational structural issues related to coordination and configuration of value chain activities by proper information system architecture design.

E. Implementing an Information Management Framework

The above framework could be implemented in three types of approaches for information management strategy:

Distributed Approach
Centralized Approach
Hybrid Approach

1. Distributed Approach

In this approach, a company can permit its local business units a significant amount of control over the development of their IT systems and strategies. In this approach, information management is a distributed function. Each business unit contains its own IS capability under its own control. Justification for delegating all information management authority and planning could be that it allows for a fast local decision on applications and priorities while preserving common corporate standards. In this approach, the standards are fixed at central levels and these stan-

Table III Alignment of Global Business and Information Management Strategies

Business strategy structure	IS strategy structure
Multinational decentralization federation	Decentralization/standalone databases and processes
Global/centralized federation	Centralization/centralized databases and processes
International and interorganizational coordinated federation	International/interorganizational linked databases and processes
Transnational integrated network	Integrated/shared databases and processes

dards are passed on to the various offices world wide to follow. Local business units participate within standards and guidelines or use an organization wide template. In this approach, a small core group develops standards, policies, and an evolving template and various offices participate by adopting systems using templates and following standards for look and feel. This strategy for information management requires that each office is responsible for managing and integrating the systems and processes at their local level.

A disadvantage of this approach is that at times the quality of systems and processes varies greatly. Uniformity across the entire organization may suffer; those offices that understand the strategic value to their mission of information resources may have significantly more sophisticated systems than those that do not. Offices may have varying levels of resources they can commit to such an effort, and so there are discrepancies across the offices with regard to quality. Distributed information management also has to rely heavily on a governance structure.

2. Centralized Approach

Another approach to information management is to have a centralized group that has all the responsibilities of creating, coordinating, and managing information resources. In this approach, the focus is to create a centralized control over information resources. Using this approach, the information management team will have complete control over the selection of information systems resources. An advantages of this approach is that all of the technical underpinning systems, even templates and standards, can be applied consistently across the entire organizational set up. The centralized team, as a point of control, can provide more consistently uniform and better quality. In this approach, information systems are a unified function. There may be distributed resources but those are under the control of the central information management group. The centralized approach is justified through features such as economies of scale, critical mass of skills, etc.

There are also some disadvantages. The central team can act as a bottleneck by getting conflicting priorities, forcing it to delay in implementation of important systems. This approach may appear to be more expensive.

3. Hybrid Approach

The hybrid approach is suited for truly global companies that have strong strategic alliances with organizations in many countries. This approach includes both domestic and international information management plans and encourages communication among IT executives in the corporation and its business units or partners.

All three strategies are valid. What is important is identifying the right one for a particular organization. It is necessary for IT managers to determine what type of global organization they are in and to align the IT structure with it.

IV. CHALLENGES AND OBSTACLES TO EFFECTIVE INFORMATION MANAGEMENT

Pursuing an information management strategy in the New Economy means managing business and technology. Organizations need to choose the right mix of technologies and integrate technology infrastructure with organizational business objectives. Managing the technology is often easier than integrating it with business strategy and managing the business end. Many enterprises have not formulated business objectives that depend on their IT infrastructure. There may be many forms of technologies and productivity tools such as newly integrated applications, enterprise resource planning (ERP), or Web-enabling critical legacy applications in the organizations (e.g., word processing, spreadsheets, e-mail, intranet Web sites) but they are not leveraged to attain the maximum business throughput. Given the increased importance of technology in today's business landscape, organizations need to understand the impact of technology on globalization. Whom do they intend to sell products or services to? How will the company reach its market—the channels of distribution: direct sales, technology partners, and services partners?

The top goal for IT managers is to invest wisely in technology, choosing only those initiatives that are closely aligned with core business strategies. This focus is prompting some companies to manage IT projects as if they were in an investment portfolio, sometimes with the help of portfolio management applications. However, there are a number of management issues and challenges related to effective information management:

Fast advancement of technology infrastructure
Location and control of technology infrastructures
Integration challenges
Technology outsourcing
Security and privacy
Capacity planning and disaster recovery

A. Fast Advancement of Technology Infrastructure

Technology is evolving and at times new technological products are brought into markets too frequently. These frequent introductions of new products make it difficult for organizations to invest in a particular technology. It is also difficult to prepare a technology strategy when the technologies themselves change so quickly. Financially, it can be difficult to invest in "leading edge" or state-of-the-art technology which could offer better solutions because the investments in previous technological solutions cannot be written. There has to be a payoff period for technological investments before new technologies are introduced. Even those organizations that are prepared to invest heavily in technology worry about the return on those investments. Assessment and evaluation of emerging and available technologies is another concern.

B. Location and Control of Technology Infrastructure

Global organizations struggle to decide the location and control of technology infrastructure. Considering economies of scale, centralized control could be more effective, to create flexibility and customize solutions locally. However, decentralized or distributed control of technology infrastructures may also be desirable. A number of factors such as economics, enterprise style, innovation, resources, legal requirements, etc., influence the appropriate balance that should be achieved.

C. Integration Challenges

Multinational organizations in their offices in different countries may have a variety of technological infrastructures and systems. Coordinating and integrating these disparate systems and business processes is a difficult challenge. Different business units may have developed applications at different times using different technologies in widely differing ways, running on different platforms. Once the systems are integrated, obviously, it can provide better efficiency and productivity for the organization.

D. Technology Outsourcing

Since technology is becoming the cornerstone of business performance, organizations must rely on technological solutions offered by outside vendors or developed by in-house technical people. Organizations currently suffer from a high turnover of their technical manpower. A crucial decision is whether the organization should outsource the technology or manage it through an in-house approach. The level of support required from suppliers, consultants, and in-house resources differs widely among the different types of technology and will depend upon the criticality of the business applications.

E. Security and Privacy

With the widespread use of technology for business, the issue of security and privacy has become very critical. For global transactions, organizations must communicate by frequently sending data and information through communication mediums that may be vulnerable to security breaches. When organizations share data with their suppliers, vendors, customers, etc., there is a possibility of data misuse thus creating privacy concerns. Organizations incur substantial costs to safeguard their systems. Determining the tradeoff between security and flexibility is a crucial question that organizations debate while spending on technological solutions.

F. Capacity Planning and Disaster Recovery

Organizations find it difficult to grow dynamically and visualize their growth pattern. At times, technological solutions fail because of poor capacity planning. Many virtual storefronts could not sustain their growth because their technical offerings could not handle the volume. A logical extension of capacity planning is contingency planning against a major failure of the technology—disaster recovery. Question such as, "How much back capacity should be provided? What if systems fail? How critical is the response factor?" become important for formulating information management strategy.

V. CONCLUSIONS

The 21st century will be focused toward globalization. Competition among organizations is already intensified due to the globalization process. Over the next few years, the variability and complexity of businesses will increase by a factor of at least 10. Organizations will face a shorter product life cycle for their products

and services. Products and services will be customized for meeting personalized demands. Mass customization will be the main focus and not the mass production. Making customers and retaining customers will be a challenge and organizations will have to exploit technologies for building relationships with customers as well as with suppliers and other partners.

As we move into the 21st century, one of the key challenges that an organization faces is what kind of information management strategy will be appropriate to offer global products and services. On the one hand, organizations have to grapple with introducing state-of-the-art technologies for their businesses, while on the other hand, organizations face the challenge of aligning their technology based systems with a number of partners and collaborators. It is clear that organizations must develop skills for managing new technologies and products creating major shifts in their markets. The creation of knowledge and diffusion of knowledge across the organization's levels will be the prime activity to create a competitive advantage.

SEE ALSO THE FOLLOWING ARTICLES

Developing Nations • Electronic Commerce • Future of Information Systems • Global Information Systems • Globalization • Internet, Overview

BIBLIOGRAPHY

Borck, J. R. (2001). Currency conversion, fraud prevention are hurdles to successful global commerce. *InfoWorld,* Vol. 23, No. 6, 55.

Day, G. S. (2001). The market-driven organization. *Direct Marketing,* Vol. 62, No. 10, 20–23, 37.

DeNoia, L. (2000). Migrating enterprise networks to high-speed. *Business Communications Review,* Vol. 30, No. 2, 40–45.

Drucker, P. (1999). *Management challenges for the 21st century,* p. 43. New York: Harper Business.

Earl, M. (2001). Knowledge management strategies: Toward a taxonomy. *Journal of Management Information Systems,* Vol. 18, No. 1, 215–233.

Galliers, R. D., Leidner, D. E., and Baker, B. S. H. (1999). *Strategic information management—Challenges and strategies in managing information systems.* Oxford: Butterworth.

Gottschalk, P. (2000). Studies of key issues in IS management around the world. *International Journal of Information Management,* Vol. 20, No. 3, 169–180.

Gwynne, P. (2001). Information systems go global. *MIT Sloan Management Review,* Vol. 42, No. 4, 14.

Heichler, E. (2000). A head for the business. *CIO,* Vol. 13, No. 17, 172–184.

Kaounides, L. C. (1999). Science, technology, and global competitive advantage. *International Studies of Management and Organization,* Vol. 29, No. 1, 53–79.

King, W. R., and Sethi, V. (1999). An empirical assessment of the organization of transnational information systems. *Journal of Management Information Systems,* Vol. 15, No. 4, 7–28.

Means, G. E., and Faulkner, M. (2000). Strategic innovation in the new economy. *The Journal of Business Strategy,* Vol. 21, No. 3, 25–29.

Oliver, R. W. (2001). Real-time strategy: What is strategy, anyway? *The Journal of Business Strategy,* Vol. 22, No. 6, 7–10.

Porter, M. E. (2001). Strategy and the Internet. *Harvard Business Review,* Vol. 79, No. 3, 62–78.

Sabherwal, R., and Chan, Y. E. (2001). Alignment between business and IS strategies: A study of prospectors, analyzers, and defenders. *Information Systems Research,* Vol. 12, No. 1, 11–33.

Ward, J., and Griffiths, P. (1997). *Strategic planning for information systems.* New York: Wiley.

Zyl, S.V. (2001). A new order: Value or lost opportunity? *Canadian Underwriter,* Vol. 68, No. 3, 28–32.

Goal Programming

James P. Ignizio *Resource Management Associates*
Carlos Romero *Technical University of Madrid*

GLOSSARY

achievement function The function that serves to measure the achievement of the minimization of unwanted goal deviation variables in the goal programming model.
goal function A mathematical function that is to be achieved at a specified level (i.e., at a prespecified "aspiration" level).
goal program A mathematical model, consisting of linear or nonlinear functions and continuous or discrete variables, in which all functions have been transformed into goals.
multiplex Originally this referred to the multiphase simplex algorithm employed to solve linear goal programs. More recently it defines certain specific models and methods employed in multiple- or single-objective optimization in general.
negative deviation The amount of deviation for a given goal by which it is less than the aspiration level.
positive deviation The amount of deviation for a given goal by which it exceeds the aspiration level.
satisfice An old Scottish word referring to the desire, in the real world, to find a practical solution to a given problem, rather than some utopian result for an oversimplified model of the problem.

GOAL PROGRAMMING, a powerful and effective methodology for the modeling, solution, and analysis of problems having multiple and conflicting goals and objectives, has often been cited as being the "workhorse" of multiple objective optimization (i.e., the solution to problems having multiple, conflicting goals and objectives) as based on its extensive list of successful applications in actual practice. Here we describe the method and its history, cite its mathematical models and algorithms, and chronicle its evolution from its original form into a potent methodology that now incorporates techniques from artificial intelligence (particularly genetic algorithms and neural networks). The article concludes with a discussion of recent extensions and a prediction of the role of goal programming in real-world problem solving in the 21st century.

I. INTRODUCTION

A. Definitions and Origin

Real-world decision problems—unlike those found in textbooks—involve *multiple,* conflicting objectives and goals, subject to the satisfaction of various hard *and soft* constraints. In short, and as the experienced practitioner is well aware, problems that one encounters *outside* the classroom are invariably massive, messy, changeable, complex, and resist treatment via conventional approaches. Yet the vast majority of traditional approaches to such problems utilize conventional models and methods that idealistically and unrealistically (in most cases) presume the optimization of a single-objective subject to a set of rigid

constraints. Goal programming was introduced in an attempt to eliminate or, at the least, mitigate this disquieting disconnect. Conceived and developed by Abraham Charnes and William Cooper, goal programming was originally dubbed "constrained regression." Constrained regression, in turn, was and is a powerful nonparametric method for the development of regression functions (e.g., curve-fitting) subject to side constraints.

Charnes and Cooper first applied constrained regression in the 1950s to the analysis of executive compensation. Recognizing that the method could be extended to a more general class of problems—that is, any quantifiable problem having multiple objectives and soft, as well as rigid constraints—Charnes and Cooper later renamed the method goal programming when describing it within their classic 1961 two-volume text *Management Models and Industrial Applications of Linear Programming*.

B. Philosophical Basis

The two philosophical concepts that serve to best distinguish goal programming from conventional (i.e., single-objective) methods of optimization are the incorporation of flexibility in constraint functions (as opposed to the rigid constraints of single-objective optimization) and the adherence to the philosophy of "satisficing" as opposed to optimization. Satisficing, in turn, is an old Scottish word that defines the desire to find a practical, real-world solution to a problem—rather than a utopian, optimal solution to a highly simplified (and very possibly oversimplified) model of that problem. The concept of satisficing, as opposed to optimization, was introduced by Herbert Simon in 1956.

As a consequence of the principle of satisficing, the "goodness" of any solution to a goal programming problem is represented by an *achievement* function, rather than the objective function of conventional optimization. The goal programming achievement function measures the degree of *nonachievement* of the problem goals. The specific way in which this nonachievement is measured characterizes the particular subtype of goal programming approach that is being employed, and may be defined so as to include the achievement of commensurable as well as noncommensurable goals.

It should be emphasized that, because a goal programming problem is to be satisficed, it is possible that the solution derived may not fit, conveniently and comfortably, into the concept of optimization or efficiency (i.e., nondominated solutions) as used by more conventional forms of mathematical modeling. This is because, in goal programming, we seek a useful, practical, implementable, and attainable solution rather than one satisfying the mathematician's desire for global optimality. (However, if one wishes, it is relatively trivial to develop efficient, or nondominated solutions for any goal programming problem. That matter is briefly described in a section to follow.)

C. A Brief List of Applications

Goal programming's label as the "workhorse" of multiple-objective optimization has been achieved by its successful solutions of important real-world problems over a period of more than 50 years. Included among these applications are:

- The analysis of executive compensation for General Electric during the 1950s
- The design and deployment of the antennas for the Saturn II launch vehicle as employed in the Apollo manned moon-landing program
- The determination of a siting scheme for the Patriot Air Defense System
- Decisions within fisheries in the United Kingdom
- A means to audit transactions within the financial sector (e.g., for the Commercial Bank of Greece)
- The design of acoustic arrays for U.S. Navy torpedos
- As well as a host of problems in the general areas of agriculture, finance, engineering, energy, and resource allocation.

D. Overview of Material to Follow

In this article, the topic of goal programming is covered in a brief but comprehensive manner. Sections to follow discuss the past, present, and future of goal programming as well as the models and algorithms for implementation. The reader having previous exposure to the original goal programming approach will (or should) immediately notice the many significant changes and extensions that occurred during the 1990s. As just one example, powerful and practical hybrid goal programming and genetic algorithm modeling and solution methods will be discussed. Readers seeking more detailed explanations of any of the ma-

terial covered herein are referred to the Bibliography at the end of this article.

II. HISTORICAL SKETCH

As mentioned, goal programming was conceived by Abraham Charnes and William Cooper nearly a half century ago. The tool was extended and enhanced by their students and, later, by other investigators, most notably Ijiri, Jääskeläinen, Huss, Ignizio, Gass, Romero, Tamiz, and Jones.

In its original form, goal programming was strictly limited to linear multiple-objective problems. Ignizio, in the 1960s, extended the method to both nonlinear and integer models, developed the associated algorithms for these extensions, and successfully applied them to a number of important real-world problems, including, as previously mentioned, the design of the antenna systems for the Saturn II launch vehicle as employed in the Apollo manned moon-landing program. During that same period, and in conjunction with Paul Huss, Ignizio developed a sequential algorithm that permits one to extend—with minimal modification—any single-objective optimization software package to the solution of any class of goal programming models (the approach was also developed, independently, by Dauer and Kruger).

Later in that same decade, Ignizio developed the concept of the multidimensional dual, providing goal programming with an effective economic interpretation of its results as well as a means to support sensitivity and postoptimality analysis. Huss and Ignizio's contributions in engineering, coupled with the work of Charnes, Cooper, Ijiri, Jääskeläinen, Gass, Romero, Tamiz, Jones, Lee, Olson, and others in management science served to motivate the interest in multiple objective optimization that continues today.

Goal programming is the most widely *applied* tool of multiple-objective optimization/multicriteria decision making. However, today's goal programming models, methods, and algorithms differ significantly from those employed even in the early 1990s. Goal programming, as discussed later, may be combined with various tools from the artificial intelligence sector (most notably genetic algorithms and neural networks) so as to provide an exceptionally robust and powerful means to model, solve, and analyze a host of real-world problems. In other words, today's goal programming—while maintaining its role as the "workhorse" of multiple-objective decision analysis—is a much different tool than that described in most textbooks, even those published relatively recently.

III. THE MULTIPLEX MODEL

A. Numerical Illustrations

Any single-objective problem, and most multiple-objective ones, can be placed into a model format that has been designated as the multiplex model, and then solved via the most appropriate version of a multiplex (or sequential goal programming) algorithm. For example, consider a conventional (albeit simple) linear programming problem taking on the following traditional form:

$$\text{Maximize } z = 10x_1 + 4x_2 \tag{1}$$

$$\text{Subject to: } x_1 + x_2 \leq 100 \tag{2}$$

$$x_2 \geq 4 \tag{3}$$

$$\boldsymbol{x} \geq \mathbf{0} \tag{4}$$

Ignoring the fact that this undemanding single-objective model can be solved by inspection, let us transform it into the multiplex form for the sake of illustration. To do so, we add a negative deviation variable to, and subtract a positive deviation variable from, each constraint. In addition, we transform the maximizing objective function into a minimizing form by simply multiplying the original objective function by a negative one. The resultant model, in multiplex form, can be written:

$$\text{Lexicographically minimize } \mathbf{U} = \{(\rho_1 + \eta_2), (-10x_1 - 4x_2)\} \tag{5}$$

$$\text{Satisfy: } x_1 + x_2 + \eta_1 - \boldsymbol{\rho_1} = 100 \tag{6}$$

$$x_2 + \boldsymbol{\eta_2} - \rho_2 = 4 \tag{7}$$

$$\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\rho} \geq \mathbf{0} \tag{8}$$

The new variables (i.e., the negative and positive deviation variables, that have been added to the constraints) indicate that a solution to the problem may result, for a given constraint i, in a negative deviation ($\eta_i > 0$), or a positive deviation ($\rho_i > 0$), or no deviation ($\rho_i + \eta_i = 0$). That is to say that we can underachieve a goal (be it a hard or soft constraint), overachieve it, or precisely satisfy it. In the multiplex formulation, the deviation variables that are to be minimized have been shown in boldface, as well as appearing in the first (highest priority) term of the achievement function of function (5).

While this new formulation may appear unusual (at least to those schooled in traditional, single-objective optimization), it provides an accurate

representation of the linear programming problem originally posed. To appreciate this, examine the achievement function, as represented by formula (5).

The multiplex achievement function is a *vector,* rather than a scalar as in conventional single-objective optimization (e.g., linear programming). The terms in this vector are ordered according to priority. The first term [i.e., $\rho_1 + \eta_2$ in function (5)] is reserved for the *unwanted* deviation variables for all *rigid* constraints, or hard goals—restrictions that supposedly *must* be satisfied for the solution to be deemed feasible. Any solution in which this first term takes on a value of zero is thus—in math programming terms—a feasible solution. In goal programming, such a solution is deemed "implementable," inferring that it could actually be implemented in the real-world problem under consideration.

Once the first term has been minimized, the next term (the second term, or $-10x_1 - 4x_2$ in this case) can be dealt with. The algorithm will seek a solution that minimizes the value of this second term, but this must be accomplished *without degrading the value already achieved in the higher priority term.* And this is the manner in which one seeks the *lexicographic* minimum of an ordered vector.

Again, this formulation may appear unusual but it not only accurately represents the linear programming (LP) problem, it also indicates the way in which most commercial software actually solves LP models. Specifically, LP problems are generally solved by the two-phase simplex algorithm, wherein the first phase attempts to find a feasible solution and the second seeks an optimal solution that does not degrade the feasibility achieved in phase 1. Multiplex algorithms simply extend this notion to any number of phases, according to the formulation employed to represent the given problem.

B. Multiplex Form of the Goal Programming Problem

While any single-objective optimization problem can be represented in a manner similar to that described above, our interest lies in multiple-objective optimization and, more specifically, in goal programming (GP). Consequently, let us examine the multiplex model for a specific GP problem.

Consider the illustrative problem represented below, wherein there are two objective functions to be optimized, functions (9) and (10), and a set of constraints to be satisfied. For purposes of discussion, we assume that the constraints of (11) through (13) are hard, or rigid.

$$\text{Maximize } z_1 = 3x_1 + x_2 \quad \text{(profit per time period)} \tag{9}$$

$$\text{Maximize } z_2 = 2x_1 + 3x_2 \quad \text{(market shares captured per time period)} \tag{10}$$

$$\text{Satisfy: } 2x_1 + x_2 \leq 50 \quad \text{(raw material limitations)} \tag{11}$$

$$x_1 \leq 20 \quad \text{(market saturation level, product 1)} \tag{12}$$

$$x_2 \leq 30 \quad \text{(market saturation level, product 2)} \tag{13}$$

$$\boldsymbol{x} \geq \mathbf{0} \quad \text{(nonnegativity conditions)} \tag{14}$$

If the reader cares to graph this problem, he or she will see that there is no way in which to optimize both objectives simultaneously—as is the case in virtually any nontrivial, real-world problem. However, the purpose of goal programming is to find a solution, or solutions, that simultaneously *satisfice* all objectives. But first these objectives must be transformed into goals. To transform an objective into a goal, one must assign some *estimate* (usually the decision maker's preliminary estimate) of the aspired level for that goal. Let's assume that the aspiration level for profit [i.e., function (9)] is 50 units while that of market shares [i.e., function (10)] is 80 units. Consequently the multiplex model for the goal programming problem is shown below, wherein the two transformed objectives now appear as (soft) goals (19) and (20), respectively:

$$\text{Lexicographically minimize } \mathbf{U} = \{(\rho_1 + \rho_2 + \rho_3), (\eta_4 + \eta_5)\} \tag{15}$$

$$\text{Satisfy: } 2x_1 + x_2 + \eta_1 - \boldsymbol{\rho}_1 = 50 \quad \text{(raw material limitations)} \tag{16}$$

$$x_1 + \eta_2 - \boldsymbol{\rho}_2 = 20 \quad \text{(market saturation level, product 1)} \tag{17}$$

$$x_2 + \eta_3 - \boldsymbol{\rho}_3 = 30 \quad \text{(market saturation level, product 2)} \tag{18}$$

$$3x_1 + x_2 + \boldsymbol{\eta}_4 - \rho_4 = 50 \quad \text{(profit goal)} \tag{19}$$

$$2x_1 + 3x_2 + \boldsymbol{\eta}_5 - \rho_5 = 80 \quad \text{(market shares goal)} \tag{20}$$

$$\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\rho} \geq \mathbf{0} \quad \text{(nonnegativity conditions)} \tag{21}$$

The multiplex model for the problem indicates—via the achievement function—that the first priority is to satisfy the hard goals of (16), (17), and (18). Note

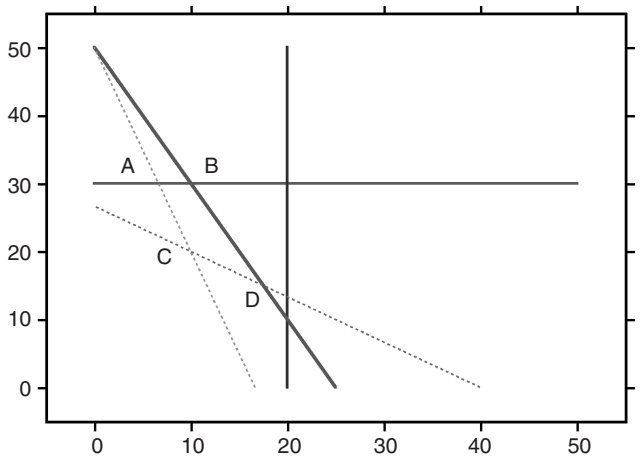


Figure 1 The satisficing region for the example.

that the nonnegativity conditions of (21) will be implicitly satisfied by the algorithm.

Once the deviation variables (i.e., $\rho_1 + \rho_2 + \rho_3$) associated with those hard goals have been minimized (albeit not necessarily to a value of zero), the associated multiplex (or sequential GP) algorithm proceeds to minimize the unwanted deviations (i.e., $\eta_4 + \eta_5$) associated with the profit and market share goals, while not degrading the values of any higher ordered achievement function terms.

Figure 1 serves to indicate the nature of the goal programming problem that now exists. Note that the solid lines represent the constraints, or hard goals, while the dashed lines indicate the original objectives—now transformed into soft goals. It is particularly important to note that those solutions satisficing this problem form a *region,* bounded by points A, B, C, and D, and including all points within and on the edges of the bounded region. This contrasts with conventional optimization in which the optimal solution is most often a single point.

The achievement functions for the linear programming and goal programming illustrations posed previously represent but two possibilities from a large and growing number of choices. We describe a few of the more common achievement functions in the next section.

IV. FORMS OF THE ACHIEVEMENT FUNCTION

The three earliest, and still most common forms of the Multiplex achievement function are listed here and discussed in turn:

1. Archimedean (also known as weighted goal programming)
2. Non-Archimedean (also known as lexicographic, or preemptive goal programming)
3. Chebyshev (also known as fuzzy programming).

A. Archimedean Goal Programming

The achievement function for an Archimedean GP model consists of exactly two terms. The first term always contains all the unwanted deviation variables associated with the hard goals (rigid constraints) of the problem. The second term lists the unwanted deviation variables for all soft goals (flexible constraints), each weighted according to importance. Returning to our previous goal programming formulation, assume that the market shares goal [i.e., function (20)] is considered, by the decision maker, to be twice as important as the profit goal. Consequently, the Archimedean form of the achievement function could be written as follows. Notice carefully that η_5 has now been weighted by 2:

$$\text{Lexicographically minimize } \mathbf{U} = \{(\rho_1 + \rho_2 + \rho_3), (\eta_4 + 2\eta_5)\} \quad (22)$$

Note that, as long as the unwanted deviations are minimized, we have achieved a satisficing solution. This may mean that we reach a satisficing solution, for our example, whether the profit achieved is 50 *or more* units. This infers a one-sided measure of achievement. If we wish to reward an overachievement of the profit, then either the model should be modified or we should employ one of the more recent developments in achievement function formatting. The latter matter is discussed in a forthcoming section. For the moment, however, our assumption is that we simply seek a solution to the achievement function given in Eq. (22), one that provides a satisficing result.

Realize that Archimedean, or weighted goal programming, makes sense only if you believe that numerical weights can be assigned to the nonachievement of each soft goal. In many instances, goals are noncommensurable and thus other forms of the achievement function are more realistic. One of these is the non-Archimedean, or lexicographic achievement function.

B. Non-Archimedean Goal Programming

The achievement function for a non-Archimedean GP model consists of two *or more* terms. As in the case of the Archimedean form, the first term always contains the unwanted deviation variables for all the hard goals. After that, the deviation variables for all soft

goals are arranged according to priority—more specifically, a nonpreemptive priority.

To demonstrate, consider the problem previously posed as Archimedean GP. Assume that we are unable, or unwilling, to assign weights to the profit or market share goals. But we are convinced that the capture of market share is *essential* to the survival of the firm. It might then make sense to assign a higher priority to market share than to profit, resulting in the non-Archimedean achievement function given here:

$$\text{Lexicographically minimize } \mathbf{U} = \{(\rho_1 + \rho_2 + \rho_3), (\eta_5), (\eta_4)\} \quad (23)$$

While the achievement function of Eq. (23) consists of but a single deviation variable for the second and third terms, the reader should understand that several deviation variables may appear in a given term—if you are able to weight each according to its perceived importance.

C. Chebyshev (Fuzzy) Goal Programming

There are numerous forms of Chebyshev, or fuzzy, goal programming but we restrict our coverage to just one in this subsection. The notion of Chebyshev GP is that the solution sought is the one that minimizes the maximum deviation from any single soft goal. Returning to our profit and market shares model, one possible transformation is as follows:

$$\text{Minimize } \delta \quad (24)$$

$$\text{Satisfy: } 2x_1 + x_2 \leq 50 \quad \text{(raw material limitations)} \quad (25)$$

$$x_1 \leq 20 \quad \text{(market saturation level, product 1)} \quad (26)$$

$$x_2 \leq 30 \quad \text{(market saturation level, product 2)} \quad (27)$$

$$\delta \geq (U_1 - z_1)/(U_1 - L_1) \quad (28)$$

$$\delta \geq (U_2 - z_2)/(U_2 - L_2) \quad (29)$$

$$\delta, \boldsymbol{x} \geq \mathbf{0} \quad (30)$$

where:

U_k = the best possible value for objective k (e.g., optimize the problem without regard to any other objectives but objective k)
L_k = the worst possible value for objective k (e.g., optimize the problem without regard to objective k)
δ = a dummy variable representing the worst deviation level
z_k = the value of the function representing the kth objective (e.g., $z_1 = 3x_1 + x_2$ and $z_2 = 2x_1 + 3x_2$).

Given the specific model of (24) through (30), the resulting Chebyshev formulation is simply:

$$\text{Minimize } \delta$$

$$\text{Satisfy: } 2x_1 + x_2 \leq 50 \quad \text{(raw material limitations)}$$

$$x_1 \leq 20 \quad \text{(market saturation level, product 1)}$$

$$x_2 \leq 30 \quad \text{(market saturation level, product 2)}$$

$$\delta \geq (70 - 3x_1 - x_2)/(70 - 60)$$

$$\delta \geq (110 - 2x_1 - 3x_2)/(110 - 70)$$

$$\delta, \boldsymbol{x} \geq \mathbf{0}$$

This model may be easily transformed into the multiplex form by means of adding the necessary deviation variables and forming the associated achievement function. However, it is clear that the Chebyshev model, as shown, is simply a single objective optimization problem in which we seek to minimize a single variable, δ. In other words, we seek to minimize the single worst deviation from any one of the problem goals/constraints.

While the Archimedean, non-Archimedean, and Chebyshev forms of the achievement function are the most common, other, newer versions may offer certain advantages. As mentioned, these newer forms of the achievement function are briefly described in a later section on extensions of GP.

V. GENERAL FORM OF THE MULTIPLEX MODEL

Whatever the form of the achievement function, the multiplex model takes on the following general form:

$$\text{Lexicographically minimize } \mathbf{U} = \{\mathbf{c}^{(1)T}\mathbf{v}, \mathbf{c}^{(2)T}\mathbf{v}, \ldots, \mathbf{c}^{(K)T}\mathbf{v}\} \quad (31)$$

$$\text{Satisfy: } \mathbf{F}(\mathbf{v}) = \mathbf{b} \quad (32)$$

$$\mathbf{v} \geq \mathbf{0} \quad (33)$$

where:

K = the total number of terms (e.g., priority levels) in the achievement function
$\mathbf{F}(\mathbf{v})$ = the problem goals, of either linear or nonlinear form, and in which negative and positive deviation variables have been augmented

$\mathbf{b}$ = the right-hand side vector
$\mathbf{v}$ = the vector of all structural (e.g., x_j) and deviation (i.e., η_i and ρ_i) variables
$\mathbf{c}^{(k)}$ = the vector of coefficients, or weights, of $\mathbf{v}$ in the kth term of the achievement function
$\mathbf{c}^{(K)T}$ = the transpose of $\mathbf{c}^{(k)}$.

We designate functions (31) through (33) as the primal form of the muliplex model. Starting with this primal form, we may easily derive the dual of any goal programming, or multiplex, model.

VI. THE MULTIDIMENSIONAL DUAL

The real power underlying conventional single-objective mathematical programming, particularly linear programming, lies in the fact that there exists a dual for any conventional mathematical programming model. For example, the dual of a maximizing LP model, subject to type I ($\leq$) constraints, is a minimizing LP model, subject to type II ($\geq$) constraints. The property of duality allows one to exploit, for example, LP models so as to develop additional theories and algorithms, as well as provide a useful economic interpretation of the dual variables.

One of the alleged drawbacks of goal programming has been the "lack of a dual formulation." It is difficult to understand why this myth has endured as the dual of goal programming problems was developed, by Ignizio, in the early 1970s. It was later extended to the more general multiplex model in the 1980s. However, space does not permit any exhaustive summary of the multidimensional dual (MDD) and thus we present only a brief description.

We listed the general form of the multiplex model in (31) through (33). Simply for sake of discussion, let's examine the dual formulation of a strictly *linear* multiplex model, taking on the primal form listed below:

PRIMAL: Lexicographically minimize $\mathbf{U}$
$$= \{\mathbf{c}^{(1)T}\mathbf{v}, \mathbf{c}^{(2)T}\mathbf{v}, \ldots, \mathbf{c}^{(K)T}\mathbf{v}\} \quad (34)$$

Satisfy: $$\mathbf{Av} = \mathbf{b} \quad (35)$$

$$\mathbf{v} \geq \mathbf{0} \quad (36)$$

where:

K = the total number of terms (e.g., priority levels) in the achievement function
$\mathbf{v}$ = the vector of all structural (e.g., x_j) and deviation (i.e., η_i and ρ_i) variables
$\mathbf{c}^{(k)}$ = the vector of coefficients, or weights, of $\mathbf{v}$ in the kth term of the achievement function
$\mathbf{Av} = \mathbf{b}$ are the linear constraints and goals of the problem, as transformed via the introduction of negative and positive deviation variables.

If you are familiar with single-objective optimization, you may recall that the dual of a linear programming model is still a linear programming model. However, in the case of a GP, or multiplex model, its dual—the multidimensional dual—takes on the form of a model in which (1) the "constraints" have multiple, prioritized right-hand sides and (2) the "objective function" is in the form of a vector. More specifically, the general form of the multidimensional dual is given as:

DUAL:
Find $\mathbf{Y}$ so as to lexicographically maximize $\mathbf{w} = \mathbf{b}^T\mathbf{Y}$ (37)

Subject to: $\mathbf{A}^T\mathbf{Y} \Leftarrow \mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(K)}$ (38)

$\mathbf{Y}$, the dual variables, are unrestricted *and* multidimensional (39)

Note that the symbol $\Leftarrow$ indicates the lexicographic nature of the inequalities involved (i.e., the left-hand side of each function is lexicographically less than or equal to the multiple right-hand sides). Physically, this means that we first seek a solution subject to the first column of right-hand side elements. Next, we find a solution subject to the second column of right-hand side elements—but one that cannot degrade the solution achieved for the previous column of right-hand side values. We continue in this manner until a complete set of solutions has been obtained for all right-hand side values.

The transformation from primal to dual may be summarized as follows:

- A lexicographically minimized achievement function for the primal translates into a set of lexicographically ordered right-hand sides for the dual.
- If the primal is to be lexicographically minimized, the dual is to be lexicographically maximized.
- For every priority level (achievement function term) in the primal, there is an associated vector of dual variables in the dual.
- Each element of the primal achievement function corresponds to an element in the right-hand side of the dual.
- The technological coefficients of the dual are the transpose of the technological coefficients of the primal.

The development of the MDD for goal programming (or general multiplex) models leads immediately to

both a means for economic interpretation of the dual variable matrix as well as supporting algorithms for solution. A comprehensive summary of both aspects, as well as numerous illustrative numerical examples, are provided in the references.

VII. ALGORITHMS FOR SOLUTION

A. Original Approach

As noted, algorithms exist for the solution of goal programming (as well as any problem that can be placed into the multiplex format) problems in either the primal or dual form. The original emphasis of goal programming was, as discussed, on linear goal programs, and the GP algorithms derived then (by Charnes and Cooper, and their students) were "multiphase" simplex algorithms. That is, they were based on a straightforward extension of the two-phase simplex algorithm.

B. Serial Algorithms: Basic Approach

Assuming that a *serial* algorithm (one in which only a single solution exists at a given time, as is the case with the well-known simplex algorithm for linear programming) is used to solve a goal programming problem, the fundamental steps are as follows:

Step 1. Transform the problem into the multiplex format.
Step 2. Select a starting solution. [In the case of a linear model, the starting solution is often the one in which all the structural variables (i.e., the x_j's) are set to zero.]
Step 3. Evaluate the present solution (i.e., determine the achievement function vector).
Step 4. Determine if a termination criterion has been satisfied. If so, stop the search process. If not, go to step 5. [Note that, in the case of linear models, the shadow prices (dual variable values) are used to determine if optimality has been reached.]
Step 5. Explore the local region about the present best solution to determine the best direction of movement. (For linear models, we move in the direction indicated by the best single shadow price.)
Step 6. Determine how far to move in the direction of best improvement, and then do so. (In linear models, this is determined by the so-called "Theta" or "blocking variable" rule.)
Step 7. Repeat steps 3 through 6 until a termination rule is satisfied.

C. Parallel Algorithms: Basic Approach

The references provide details and illustrations of the use of such serial multiplex algorithms, for linear, nonlinear, and integer goal programming problems. However, with the advent of hybrid multiplex algorithms, the use of *parallel* algorithms is both possible and, in the minds of some, preferable.

A parallel algorithm follows somewhat the same steps as listed previously for serial algorithms. The primary difference is that rather than employing a single solution point at each step, multiple points—or populations of solutions—exist at each iteration (or "generation") of the algorithm. There are two very important advantages to the employment of parallel algorithms in goal programming. The first is that, if the algorithm is supported by parallel processors, the speed of convergence to the final solution is significantly increased. The second advantage is less well known, but quite likely even more significant. Specifically, it would appear from the evidence so far that solutions to *certain types* (more specifically, those employing evolutionary operations) of parallel algorithms are far more stable, and less risky, than those derived by conventional means.

D. Hybrid Algorithm Employing Genetic Algorithms

The basics of a hybrid goal programming/genetic algorithm for solving multiplex models are listed below. Such an approach has been found, in actual practice, to achieve exceptionally stable solutions—and do so rapidly. Furthermore, it is particularly amenable to parallel processing.

Step 1. Transform the problem into the multiplex format.
Step 2. Randomly select a population of trial solutions (typically 20 to a few hundred initial solutions will compose the first generation).
Step 3. Evaluate the present solutions (i.e., determine the achievement function vector for each member of the present population).
Step 4. Determine if a termination criterion has been satisfied. If so, stop the search process. If not, go to step 5.
Step 5. Utilize the genetic algorithm operations of selection, mating, reproduction, and mutation to develop the next generation of solutions (see the references for details on genetic algorithms).
Step 6. Repeat steps 3 through 5 until a termination rule is satisfied.

The parallel algorithm developed via combining goal programming with genetic algorithms offers a convenient way in which to address any single or multiobjective optimization problem, be they linear, nonlinear, or discrete in nature. The single disadvantage is that global optimality cannot be ensured.

VIII. GOAL PROGRAMMING AND UTILITY OPTIMIZATION

Although goal programming is developed within a satisficing framework, its different variants can be interpreted from the point of view of utility theory as highlighted in this section. This type of analysis helps to clarify goal programming variant as well as to provide the foundations of certain extensions of the achievement function.

Let us start with lexicographic goal programming, where the noncompatibility between lexicographic orderings and utility functions is well known. To properly assess the effect of this property on the pragmatic value of this approach, we must understand that the reason for this noncompatibility is exclusively due to the noncontinuity of preferences underlying lexicographic orderings.

Therefore, a worthwhile matter of discussion would not be to argue against lexicographic goal programming because it implicitly assumes a noncontinuous system of preferences but to determine if the characteristics of the problem situation justify or not a system of continuous preferences. Hence, the possible problem associated with the use of the lexicographic variant does not lie in its noncompatibility with utility functions, but in the careless use of this approach. In fact, in contexts where the decision maker's preferences are clearly continuous, a model based on nonpreemptive weights should be used. Moreover, it is also important to note that a large number of priority levels can lead to a solution where every goal, except those situated in the first two or three priority levels, is redundant. In this situation, the possible poor performance of the model is not due to the lack of utility meaning of the achievement function but to an excessive number of priority levels or to overly optimistic aspiration levels (i.e., close to the ideal values of the goals).

Regarding weighted (Archimedean) goal programming, we know that this option underlies the maximization of a separable and additive utility function in the goals considered. Thus, the Archimedean solution provides the maximum aggregate achievement among the goals considered. Consequently, it seems advisable to test the separability between attributes before the decision problem is modeled with the help of this variant.

Regarding Chebyshev goal programming, it is recognized that to this variant underlies a utility function where the maximum (worst) deviation level is minimized. In other words, the Chebyshev option underlies the optimization of a MINMAX utility function, for which the most balanced solution between the achievement of the different goals is achieved.

These insights are important for the appropriate selection of the goal programming variant. In fact, the appropriate variant should not be chosen in a mechanistic way but in accordance with the decision maker's structure of preferences. These results also give theoretical support to the extensions of the achievement function to be commented on in the next section.

IX. EXTENSIONS

A. Transforming Satisficing Solutions into Efficient Solutions

The satisficing logic underlying goal programming implies that its formulations may produce solutions that do not fit classic optimality requirements like efficiency; that is, solutions for which at least the achievement of one of the goals can be improved without degrading the achievement of the others. However, if one wishes, it is very simple to force the goal programming approach to produce efficient solutions. To secure efficiency, it is enough to augment the achievement function of the multiplex formulation with an additional priority level where the sum of the wanted deviation variables is maximized.

For instance, in the example plotted in Fig. 1 the closed domain ABCD represents the set of satisficing solutions and the edge BD the set of satisficing and efficient solutions. Thus, if the achievement function of the multiplex model (15) through (21) is augmented with the term $(-\rho_4 - \rho_5)$ placed in a third priority level, then the lexicographic process will produce solution point B, a point that is satisficing and efficient at the same time. Note that there are more refined methods capable of distinguishing the efficient goals from the inefficient ones, as well as simple techniques to restore the efficiency of the goals previously classified as inefficient. Technical details about this type of procedure can be found in the Bibliography.

B. Extensions of the Achievement Function

According to the arguments developed in Section VIII, from a preferential point of view the weighted

and the Chebyshev goal programming solutions represent two opposite poles. Thus, since the weighted option maximizes the aggregate achievement among the goals considered, then the results obtained with this option can be biased against the performance achieved by one particular goal. On the other hand, because of the preponderance of just one of the goals, the Chebyshev model can provide results with poor aggregate performance between different goals. The extreme character of both solutions can lead in some cases to possibly unacceptable solutions by the decision maker. A possible modeling solution for this type of problem consists of compromising the aggregate achievement of the Archimedean model with the MINMAX (balanced) character of the Chebyshev model. Thus, the example of the earlier section can be reformulated with the help of the following multiplex extended goal programming model:

$$\text{Lexicographically minimize } \mathbf{U} = \{(\rho_1 + \rho_2 + \rho_3), [(1 - Z)\,\delta + Z(\eta_4 + \eta_5)]\} \quad (40)$$

$$\text{Satisfy: } 2x_1 + x_2 + \eta_1 - \boldsymbol{\rho}_1 = 50 \quad \text{(raw material limitations)} \quad (41)$$

$$x_1 + \eta_2 - \boldsymbol{\rho}_2 = 20 \quad \text{(market saturation level, product 1)} \quad (42)$$

$$x_2 + \eta_3 - \boldsymbol{\rho}_3 = 30 \quad \text{(market saturation level, product 2)} \quad (43)$$

$$3x_1 + x_2 + \boldsymbol{\eta}_4 - \rho_4 = 50 \quad \text{(profit goal)} \quad (44)$$

$$2x_1 + 3x_2 + \boldsymbol{\eta}_5 - \rho_5 = 80 \quad \text{(market shares goal)} \quad (45)$$

$$(1 - Z)\,\boldsymbol{\eta}_4 - \delta \leq 0 \quad (46)$$

$$(1 - Z)\,\boldsymbol{\eta}_5 - \delta \leq 0 \quad (47)$$

$$\delta, \boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\rho} \geq \mathbf{0} \quad (48)$$

where the parameter Z weights the importance attached to the minimization of the sum of unwanted deviation variables. For $Z = 0$, we have a Chebyshev goal programming model. For $Z = 1$ the result is a weighted goal programming model, and for other values of parameter Z belonging to the interval (0, 1) intermediate solutions between the solutions provided by the two goal programming options are considered.

Hence, through variations in the value of parameter Z, compromises between the solution of the maximum aggregate achievement and the MINMAX solution can be obtained. In this sense, this extended formulation allows for a combination of goal programming variants that, in some cases, can reflect a decision maker's actual preferences with more accuracy than any single variant.

Other extensions of the achievement function have been derived taking into account that in all traditional goal programming formulations there is the underlying assumption that any unwanted deviation with respect to its aspiration level is penalized according to a constant marginal penalty. In other words, any marginal change is of equal importance no matter how distant it is from the aspiration level. This type of formulation only allows for a linear relationship between the value of the unwanted deviation and the penalty contribution. This corresponds to the case of the achievement functions underlying a weighted goal programming model or to each priority level of a lexicographic model.

This type of function has been termed a one-sided penalty function, when only one deviation variable is unwanted, or V-shaped penalty function when both deviation variables are unwanted. However, other penalty function structures have been used. Thus, we have the two-sided penalty functions when the decision maker feels satisfied when the achievement of a given goal lies within a certain aspiration level interval, or the U-shaped penalty function if the marginal penalties increase monotonically with respect to the aspiration levels. Several authors have proposed improvements and refinements to the goal programming model with penalty functions. Details about this type of modeling are described in the Bibliography.

C. Goal Programming and MCDM

Goal programming is but one of several approaches possible within the broader field of multiple criteria decision making (MCDM). It is a common practice within MCDM to present its different approaches in an independent, disjoint manner, giving the impression that each approach is completely autonomous. However, this is not the case. In fact, significant similarities exist among most of the MCDM methods. In this sense, the multiplex approach presented in Section V is a good example of a goal programming structure encompassing several single- and multiple-objective optimization methods.

Furthermore, goal programming can provide a unifying basis for most MCDM models and methods. With this purpose, extended lexicographic goal programming has recently been proposed. To illustrate this concept, consider the representation provided in Eqs. (49) through (52):

Lexicographically minimize $\mathbf{U}$

$$= \{\lambda_1 \delta_1 + \mu_1 \sum_{i \in h_1} (\alpha_i \eta_i + \beta_i \rho_i)^p, \ldots, \lambda_j \delta_j$$

$$+ \mu_j \sum_{i \in h_j} (\alpha_i \eta_i + \beta_i \rho_i)^p, \ldots,$$

$$\lambda_Q \delta_Q + \mu_Q \sum_{i \in h_Q} (\alpha_i \eta_i + \beta_i \rho_i)^p\} \quad (49)$$

Satisfy: $\lambda_j (\alpha_i \eta_i + \beta_i \rho_i) - \delta_j \leq 0$

$$i \in h_j \quad j \in \{1, \ldots, Q\} \quad (50)$$

$$f_i(\mathbf{x}) + \eta_i - \rho_i = t_i \qquad i \in \{1, \ldots, q\} \quad (51)$$

$$\boldsymbol{\delta}, \boldsymbol{\eta}, \boldsymbol{\rho} \geq \mathbf{0} \qquad \mathbf{x} \in \mathbf{F} \quad (52)$$

where p is a real number belonging to the interval $[1, \infty)$ or ∞. Parameters α_i and β_i are the weights reflecting preferential and normalizing purposes attached to the negative and positive variables of the ith goal, respectively; λ_j and μ_j are control parameters; and h_j represents the index set of goals placed in the jth priority level. The block of rigid constraints $\mathbf{x} \in \mathbf{F}$, can be transferred to an additional first priority level in order to formulate the model within a multiplex format.

If the above structure is considered the *primary model,* then it is easy to demonstrate that an important number of multiple-criteria methods are just *secondary models* of the extended lexicographic goal programming model. Thus, the following multicriteria methods can be straightforwardly deduced just by applying different parameter specifications to the above model:

1. Conventional single-objective mathematical programming model
2. Nonlinear and linear weighted goal programming
3. Lexicographic linear goal programming
4. Chebyshev goal programming
5. Reference point method
6. Compromise programming (L_1 bound) and (L_∞ bound or fuzzy programming with a linear membership function)
7. Interactive weighted Tchebycheff procedure.

The use of goal programming as a unifying framework seems interesting at least for the following reasons. The extended lexicographic goal programming model stresses similarities between MCDM methods that can help reduce gaps between advocates of different approaches. Moreover, this unifying approach can become a useful teaching tool in the introduction of MCDM, thus avoiding the common presentation based upon a disjoint system of methods. Finally, the extended lexicographic goal programming approach allows us to model decision-making problems for which a good representation of a decision maker's preferences requires a mix of goal programming variants. In short, this type of general formulation can increase the enormous potential flexibility inherent to goal programming.

X. THE FUTURE

In the nearly half century since its development, goal programming has achieved and maintained its reputation as the workhorse of the multiple-objective optimization field. This is due to a combination of simplicity of form and practicality of approach. The more recent extensions to the approach have, thankfully, not negatively impacted on these attributes.

We envision a future in which goal programming will continue to utilize methods from other sectors, particularly artificial intelligence. The most significant advances, from a strictly pragmatic perspective, involve combining goal programming with evolutionary search methods, specifically genetic algorithms. Such extensions permit the rapid development of exceptionally stable solutions—the type of solutions needed in most real-world situations.

SEE ALSO THE FOLLOWING ARTICLES

Decision Support Systems • Evolutionary Algorithms • Executive Information Systems • Game Theory • Industry, Artificial Intelligence in • Model Building Process • Neural Networks • Object-Oriented Programming • Strategic Planning for/of Information Systems • Uncertainty

BIBLIOGRAPHY

Charnes, A., and Cooper, W. W. (1961). *Management models and industrial applications of linear programming,* Vols. 1 and 2. New York: John Wiley.

Charnes, A., and Cooper, W. W. (1977). Goal programming and multiple objective optimization. Part I. *European Journal of Operational Research,* Vol. 1, 39–54.

Dauer, J. P., and Kruger, R. J. (1977). An iterative approach to goal programming. *Operational Research Quarterly,* Vol. 28, 671–681.

Gass, S. I. (1986). A process for determining priorities and weights for large-scale linear goal programming models. *Journal of the Operational Research Society,* Vol. 37, 779–784.

Ignizio, J. P. (1963). S-II trajectory study and optimum antenna placement, Report SID-63. Downey, CA: North American Aviation Corporation.

Ignizio, J. P. (1976). *Goal programming and extensions.* Lexington Series. Lexington, MA: D. C. Heath & Company.

Ignizio, J. P. (1985). *Introduction to linear goal programming*. Beverly Hills, CA: Sage Publishing.

Ignizio, J. P., and Cavalier, T. M. (1994). *Linear programming*. Upper Saddle River, NJ: Prentice Hall.

Markowski, C. A., and Ignizio, J. P. (1983). Theory and properties of the lexicographic LGP dual. *Large Scale Systems,* Vol. 5, 115–121.

Romero, C. (1991). *Handbook of critical issues in goal programming*. Oxford, UK: Pergamon Press.

Romero, C. (2001). Extended lexicographic goal programming: A unifying approach. *OMEGA, International Journal of Management Science,* Vol. 29, 63–71.

Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review,* Vol. 63, 129–138.

Tamiz, M., and Jones, D. (1996). Goal programming and Pareto efficiency. *Journal of Information and Optimization Sciences,* Vol. 17, 291–307.

Tamiz, M., Jones, D. F., and Romero, C. (1998). Goal programming for decision making: An Overview of the current state-of-the-art. *European Journal of Operational Research,* Vol. 111, 569–581.

Vitoriano, B., and Romero, C. (1999). Extended interval goal programming. *Journal of the Operational Research Society,* Vol. 50, 1280–1283.

Group Support Systems

Marshall Scott Poole
Texas A&M University

GLOSSARY

anonymity The degree to which the GSS prevents members from ascertaining who made a given entry into the system.
distribution The degree to which members are in different locations.
level of support The type of activities the GSS supports, with higher levels indicating higher degrees of complexity and sophistication in activities.
parallelism The degree to which the GSS enables members to work simultaneously in entering information or making decisions.
restrictiveness The degree to which the GSS limits the freedom of action of the user to follow prespecified procedures.
synchrony The degree to which members must work with each other at the same time versus the capability to use the GSS regardless of whether others are also online.

A GROUP SUPPORT SYSTEM (GSS), or electronic meeting system, combines communication, computer, and decision technologies to support decision making and related group activities. GSSs have been designed to support both small and large groups that are either co-located or distributed. Key dimensions underlying the design and effects of GSSs and their features are synchrony, restrictiveness, level of support, parallelism, and anonymity. Evidence of the effects of GSSs on efficiency, effectiveness, and satisfaction are summarized. Key directions for future development include improved designs for distributed work and automated facilitation.

I. WHAT IS A GROUP SUPPORT SYSTEM?

A group support system (GSS) combines communication, computer, and decision technologies to support activities connected with group work, including meetings, decision making, and subsequent activities. Communication technologies for GSSs include electronic messaging and chat, teleconferencing, document management, and calendaring. Computer technologies include multiuser operating systems, fourth generation languages, and web authoring tools. Decision support technologies include agenda setting, decision modeling methods (such as decision trees or risk analysis), structured group methods (such as the Nominal Group Technique or Interpretive Structural Modeling), and rules for directing group discussion (such as parliamentary procedure).

Groups engage in a wide variety of activities during meetings. The most common is information sharing. Other relevant activities that have been identified include decision making, problem solving, idea generation, planning, project management, leadership and facilitation, conflict and conflict management, negotiation, delegation, role definition, socializing, tension release, fantasizing, and management of the group process. Most GSSs have been designed around decision making, problem solving, facilitation, and project management, but most also incorporate features that support other activities as well. For example, the

SAMM system developed at the University of Minnesota incorporated a number of canned comments such as, "Oh no, not another war story!" that participants could send to the group as jokes.

Group support systems are a type of groupware, and a number of the functions that support meeting processes are discussed in the article on groupware. This article will focus on how GSSs support holistic group processes—decision making, planning, problem solving, creativity, conflict management, facilitation, and structuring the group process—and refer the reader to the groupware article for discussion of support for lower level activities such as scheduling, communication, and coordination and for activities that extend beyond the meeting itself, such as project management, group writing and document management, and shared databases. Typically groupware and group support features are mixed in GSSs, so the two articles should be read together for a fuller understanding of how GSSs operate.

A. History of Group Support Systems

Using computers to support group work has a long history. Douglas Englebart and his colleagues demonstrated the use of computers for collaborative work in the early 1970s. Many of the earliest applications of networked computers took off from a desire to support group work. Murray Turoff used the precursor of the Internet to conduct Delphi sessions involving dozens of authorities in the development of future scenarios. The development of group conferencing was also driven in part by the goal of supporting group decision making and deliberation.

Another strand of activity that influenced the development of GSSs was the extension of decision support systems into group settings. In the early 1980s, a number of researchers had applied decision support systems (DSS) in group settings and by 1984, the idea of developing DSS into Group Decision Support Systems had arisen. Articles by Huber in 1984 and DeSanctis and Gallupe in 1987 defined the parameters of a growing field. The centrality of decision making and its associated processes in the evolution of GSSs is reflected in the name Group Decision Support System which was applied by both Huber and DeSanctis and Gallupe. During the early 1990s the more general terms Group Support System or Electronic Meeting System supplanted the initial designation of these systems. By the early 1990s, over 15 different GSSs had been developed, most stand alone systems that required special servers. Development of these systems was primarily an enterprise of business school faculty rather than computer scientists, which is rather unusual for new technologies. To this day, the pragmatic, management-oriented emphasis still shows in GSSs.

In the 1990s GSSs, like most groupware, attempted to make the transition to the Internet. Most of the specialized GSSs that survived were reconfigured to run on intranets and the World Wide Web. However, as noted below, the need for intensive facilitation has made the transition of all but the most straightforward modules of GSSs difficult.

II. GROUP SUPPORT SYSTEM CONFIGURATIONS

The most common type of GSS is installed in a special decision room (see Fig. 1). Members are provided with a computer and visual display terminal that allows them to enter data and control the operation of

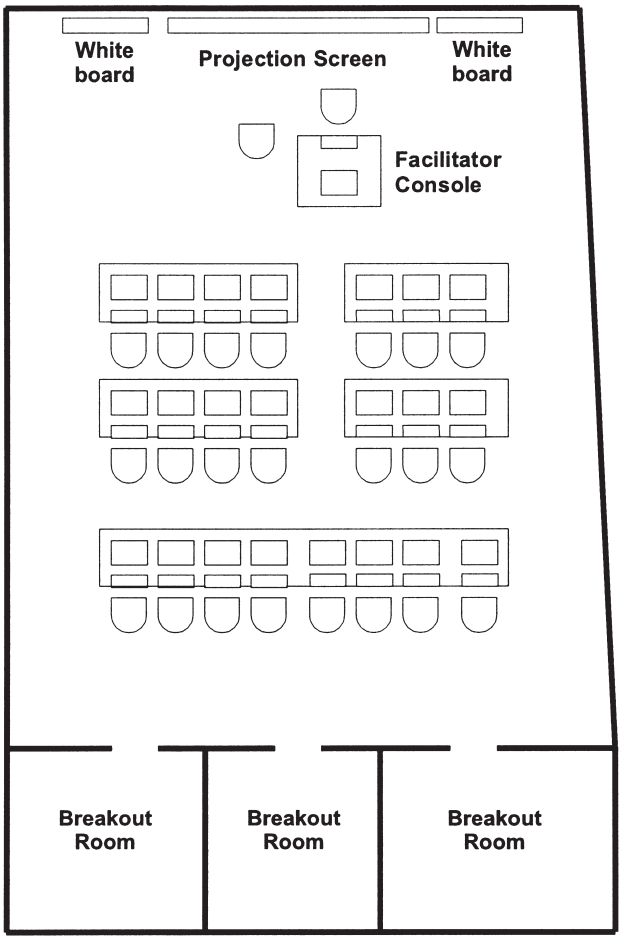


Figure 1 Large facility for a group support system.

the system. The GSS offers a range of procedures, such as agenda-setting methods, idea recording, and voting routines. Specialized decision modeling or structured group methods are usually available. Often there is also a "group" display screen, a large projector which displays common group information, such as lists of ideas or tabulations of votes (this supplements the traditional flip chart or chalk board). In face-to-face meetings, members use the computer system and also talk directly to one another. In dispersed settings, groups may also use a voice or video communication channel. GSSs are being used in strategic planning meetings, for scientific research collaboration, for product design development, and for the management of quality teams.

Though the decision room is the most common GSS setup, a variety of GSSs configurations are possible. Four cases can be defined, shown in Table I (see the article on groupware for a similar typology). Some GSSs support groups whose members are *dispersed,* working in separate conference rooms, offices, homes, or other locations. Other GSSs are designed for use in *face-to-face* meetings in a conference or board room. GSSs may also be distinguished according to whether they support *smaller* working groups or *larger* groups whose members may not know each other well. The Local Area Decision Network supports smaller groups, typically in the same office building and working together on the same project or task. The Computer-Mediated Conference supports large numbers of people who are physically distant from one another but must work on common tasks. The Legislative Session supports larger groups whose members meet face-to-face; in these settings the GSS may regulate member-to-member communication in a hierarchical fashion, allowing members to send messages to only their fellow party members or party chairperson, and meeting proceedings may be electronically recorded and analyzed by interested constituencies.

The bulk of GSS research is centered on the Decision Room, which is the electronic equivalent to the traditional face-to-face meeting. As an example, consider GroupSystems, the most successful and popular GSS. GroupSystems was developed at the University of Arizona and is distributed by Ventana Corporation. GroupSystems can be used with small or quite large groups. Modules available on GroupSystems include brainstorming, several idea evaluation procedures, idea clustering, decision aids such as Stakeholder Analysis and Policy Formation techniques, and capabilities to work with organizational databases.

One of the most interesting features of GroupSystems is "Topic Commenter." Members enter their ideas about a topic and then view and comment on others' entries as these are "randomly" presented on their computer screen. This creates a running "conversation" among members, even though very little verbal exchange occurs. Members read others' entries and enter comments as they occur to them, rotating through their own and others' entries for periods of 30 minutes to over an hour. The effect is rather startling—people huddled over computers, earnestly entering comments as though they were talking to the idea's originator—a wholly new way of conducting group discussion. A record of these "conversations," preserved in the database of ideas and associated comments, commonly serves as the starting point for a decision-making or planning process.

Like most decision room based GSSs, GroupSystems is not intended to replace existing modes of group communication. Instead, it is designed to support and encourage verbal and nonverbal interaction, as well as to provide additional channels for communication and decision support. Consequently, the group might use the GSS only at certain points during a meeting; typically members work at the GSS for a while and then discuss the outputs to the public screen or adjourn to breakout rooms configured for face-to-face work.

The potential of GSSs and other social technologies lies in their ability to enhance human information handling capacity, to provide additional media for interpersonal communication, and to provide data resources and procedural structures for group work. GSSs may also increase the salience of procedures to group members, thus promoting their beneficial effects. For example, if a GSS prompts members to, "Enter your ratings for each of the following options. . .," member activity is synchronized and attention is focused on this step of the decision-making process. By increasing the

Table I **Possible Configuration for Group Support Systems**

	Small group	Large group
Co-located	Decision room	Legislative session
Dispersed	Local area decision network	Computer-mediated conference

salience of procedures, GSSs may also help educate groups and create an awareness of the importance of systematic approaches.

III. DIMENSIONS OF GSSs

Several dimensions underlying GSS designs have been identified. GSSs vary, first, in terms of their *synchrony,* the degree to which members must work with each other at the same time versus the capability to use the GSS regardless of whether others are signed on. Most decision rooms rely on synchronous work, but GSSs can also be used for asynchronous work if coordinated properly. A second dimension along which GSSs vary is *distribution,* the degree to which members can use the system if they are distributed across different locations. Most current GSSs assume that members will be co-located, but some GSSs have been designed for distributed groups. The synchrony and distribution dimensions represent the two dimensions of time and place common in groupware classifications mentioned above.

GSSs in general and their features also vary in their degree of *restrictiveness,* the extent to which they limit the group's activity. Most brainstorming tools are highly restrictive, because they require members to follow closely specified rules about how to enter ideas and register them to the common list and also on how to comment on others' ideas. On the other hand, a group drawing tool such as a whiteboard in NetMeeting is low in restrictiveness, because it enables most members to simply take control and add their inputs to the drawing. GroupSystems, which requires facilitated control in its most typical mode of use, is more restrictive than is a tool such as NetMeeting, which enables members to choose and control the features of the system.

A number of other dimensions apply to features or modules of GSSs and not necessarily to the system as a whole. GSS features differ in terms of *level* of support they offer. DeSanctis and Gallupe in 1987 distinguished three levels of support in GSS features. *Level 1* features provide support for enhanced communication among group members. These features include idea listing, evaluation techniques such as voting or rating, and comment recording. *Level 2* features provide decision support tools, such as multicriteria decision making, stakeholder analysis, and problem formulation. A level 2 GSS supports activities that members could not undertake on their own in a reasonable amount of time and brings advanced decision models into meetings. A major barrier to the use of procedures is members' lack of knowledge and skills. Members often are reluctant to spend precious time and energy researching available procedures and preparing required materials and information. And even with adequate preparation, procedures will not work very well if the facilitator and group do not have the necessary group process skills, skills which require special training and considerable experience. By "automating" procedures, GDSSs can reduce the work involved. *Level 3* features provide guidance for the group through such tools as automated facilitation and expert systems that advise the group on strategies and approaches for making the decision. At this point in time, the majority of GSSs provide level 1 features, while some provide level 2 support. No commercially available systems provide level 3 support at this time, though research on such systems is underway.

GSS features also vary in terms of their level of *parallelism.* Features high in parallelism enable members to work with the system at the same time, whereas those low in parallelism permit only one or a few members from the group to work with the GSS. Electronic brainstorming, for example, is high in parallelism, because all members can enter ideas simultaneously at their respective stations. A tool for group categorization of ideas is typically lower in parallelism, as members must work together to decide on categories and move ideas into them. Parallelism enables GSSs to overcome some of the limitations of discussion in face-to-face groups due to the fact that only one or two members can hold the floor at the same time, blocking the production of other members.

Finally, GSS features differ in terms of the *anonymity* of member entries. Some features provide users with anonymity by mixing their entries up with those of other users. For example, most idea generation tools offer the group the option to turn off identification of the person who contributed each idea. This affords the group the opportunity to short circuit the inhibiting effects of status and power differences on expression of ideas and opinions by members lower in prominence.

IV. RESEARCH ON GSS IMPACTS

A substantial body of research on GSSs has been built over the past 20 years. Fjermestad and Hiltz, McLeod, and Scott provide extensive summaries of the research in this area, which this review will draw on. Much of the existing GSS research is laboratory based, but there are a number of field and case studies as well. Almost all involve the use of nondistributed, synchronous groups. Experience suggests that it is difficult to coordinate complex reasoning and deliberation that

GSSs often attempt to promote in the distributed condition and that making sense of complex outputs of GSS features is problematic in asynchronous conditions. While there is clearly a place for GSSs for distributed groups working asynchronously, designs have lagged in this area.

A. Efficiency

One important issue is whether GSSs increase the efficiency of group decision processes. Studies of GSSs in the field suggest that they enable groups to accomplish more in less time than traditional meetings. Procedures like brainstorming and idea rating are much faster with computers because members can work simultaneously while the computer combines their inputs automatically. Moreover, data management capabilities can greatly enhance group "housekeeping" chores. The GSS can capture ideas and ratings so that they are available to all and provide a historical record. In a comprehensive review, Scott in 1999 found that most laboratory studies showed that unsupported groups were equal to or more efficient than GSSs (15 consistent with this conclusion and only 2 inconsistent), whereas most field studies report that GSS-supported groups were more efficient (17 showing GSSs superior and 6 showing GSSs equal to or less efficient than unsupported conclusions). This discrepancy between lab and field results also occurs for other outcome variables, and we will address it later.

B. Group Performance and Decision Quality

In terms of group performance, over half the studies focused on the impacts of GSSs on idea generation. Laboratory studies have for the most part showed that groups using idea generation procedures in GSSs produce more ideas than unsupported groups. This effect is stronger as group size increases, suggesting that the parallelism provided by GSSs is compensating for group size. Some studies suggest that the quality of the ideas is higher as well. However, Pinsonneault *et al.* 1999 register an important qualification in their finding that unsupported groups using a restrictive nominal group idea generation procedure outperformed GSS-supported groups. This suggests that it is the procedure and not its computerization that makes the difference.

Research on decision quality has yielded mixed results. A number of studies suggest that GSSs yield superior quality compared to unsupported groups, but an almost equal number show that face-to-face groups are equal or superior to GSS groups. Field studies generally support the superiority of GSSs in terms of quality, while the majority of laboratory studies indicate that unsupported groups have superior or equal performance to that of GSS-supported groups. One important variable is the level of the GSS in question. Most studies find that unsupported groups outperform groups using level 1 GSS features. On the other hand, groups using level 2 features consistently outperform unsupported groups.

C. Satisfaction

A final outcome variable is group satisfaction with the decision. This is an important outcome both in its own right and because satisfied users are more likely to continue using the system and valuing its procedures. Results are mixed, but the majority of laboratory studies (10) indicate no difference in satisfaction between GSS and unsupported groups; but in 6 studies GSS groups are more satisfied and in 6 less satisfied than unsupported groups. As with most other outcomes, the majority of field studies suggests that groups using GSSs are more satisfied than unsupported groups.

D. Participation

A number of studies have tested the hypothesis that GSSs equalize member participation. Results suggest that groups using GSSs have higher rates of member participation than unsupported groups. The parallelism available in many GSS features offers members who might find their turns blocked by more talkative members the opportunity to contribute. Anonymity enhances participation as well. These results are in contrast to those for electronic mail and computer conferencing: a number of studies suggest that these less structured media do not necessarily equalize participation. The reviews suggest that although participation may be more balanced for groups using GSSs, influence is not. Status persists in GSS meetings just as it does in electronic mail.

E. Conflict Management

GSSs can also influence how groups manage conflict. There is substantial evidence that GSSs surface differences and conflicts. Level 1 GSSs—as well as computer-mediated communication media such as e-mail and

conferencing—tend to result in higher levels of conflict and greater difficulty in attaining consensus or making decisions than is common in nonsupported groups. Procedures such as mediation help groups manage conflicts because they divulge differences and give groups a way to forge some common ground. Studies have shown that level 2 GSSs that incorporate procedures for helping groups generate options and make decisions result in more effective conflict management than either nonsupported groups or level 1 GSSs.

F. Complexities in the Use of GSSs

The effects of GSS procedures depend on how they are implemented in ongoing groups. Many GSS features are complex and require some guidance or practice to utilize properly. Members do not always understand the rationale behind GSS procedures and existing group norms may conflict with norms built into the GSS. As groups adapt the GSS to their own goals and norms, they may change the procedure in subtle, yet significant ways. In one experiment, providing voting procedures for "straw polls" actually led groups to cut off their discussions and reduce participation in decision making, a use quite different than was intended by system designers.

V. CONCLUSION

The study of the effects of GSSs and collaborative technologies on group processes and outcomes has made good progress in the past decade, but many questions remain. There is little doubt that powerful effects occur and that many realize the potential of systems to structure meetings. More research is needed to ascertain how to promote positive impacts and to ameliorate negative ones.

Two major challenges for the future of GSSs are distributed support and facilitation. For GSSs to achieve widespread use, they must be made available for distributed groups. The growth of the internet and distributed computing and the concomitant increase in virtual groups has created an opportunity for GSSs to show their power and utility. However, reliable GSS designs for distributed groups have proven difficult to develop. While level 1 functions such as idea generation and voting are straightforward to implement and have been incorporated into many conferencing tools, higher level features require guidance and training that are difficult to provide at a distance. Procedures such as multiattribute utility analysis and stakeholder analysis are typically run in co-located groups by specialized consultants, and it has proven difficult to implement them in lean media such as conferencing or chat. The spread of videoconferencing may provide one solution to this problem, since it can be coupled with the GSS and used to guide its use.

Facilitation is necessary for the use of most GSSs. Running an effective meeting takes a good deal of skill derived from training and experience, and many members of organizations do not consciously cultivate meeting skills. As a result, facilitators are needed to help plan and run sessions. Facilitation is particularly important for more complex procedures like multiattribute utility analysis. However, facilitators represent considerable overhead in terms of their cost and the requirement that the meeting convenor must work with them to plan the session. The reluctance to employ and work with facilitators is one barrier to the adoption of GSSs. One way of addressing this problem is to develop level 3 GSSs with automated guidance and facilitation. However, such systems are at this time the subject of research and their realization lies sometime in the future.

The next 10 years will be a critical time for GSSs. Whether key challenges can be addressed will determine whether they remain a useful but marginal tool or win widespread acceptance and use.

SEE ALSO THE FOLLOWING ARTICLES

Computer-Supported Cooperative Work • Global Information Systems • Groupware • Outsourcing • Telecommuting • Virtual Organization

BIBLOGRAPHY

Bikson, T. (1996). Groupware at the World Bank, in *Groupware and teamwork* (C. U. Ciborra, Ed.), pp. 145–183. Chichester, UK: Wiley.

Bostrom, R., Watson, R. T., and Kinney, S. (Eds.). (1992). *Computer augmented teamwork: A guided tour.* New York: Van Nostrand–Reinhold.

Dennis, A. R., and Valacich, J. S. (1999). Research note. Electronic brainstorming: Illusions and patterns of productivity. *Information Systems Research,* Vol. 10, 375–377.

DeSanctis, G., and Gallupe, R. B. (1987). A foundation for the study of group decision support systems. *Management Science,* Vol. 33, 589–609.

Fjermestad, J., and Hiltz, S. R. (1999). An assessment of group support systems experimental research: Methodology and results. *Journal of Management Information Systems,* Vol. 15, 7–150.

Fjermestad, J., and Hiltz, S. R. (2001). Group support systems: A descriptive evaluation of case and field studies. *Journal of Management Information Systems,* Vol. 17, 115–160.

Hiltz, S. R., and Turoff, M. (1978). *The network nation: Human communication via computer.* Reading, MA: Addison–Wesley.

Huber, G. P. (1984). Issues in the design of group decision support systems. *MIS Quarterly,* Vol. 8, 195–204.

Majchrzak, A., Rice, R. E., Malhotra, A., King, N., and Ba, S. (2000). Technology adaptation: The case of a computer-supported inter-organizational virtual team. *MIS Quarterly,* Vol. 24, 569–600.

McLeod, P. L. (1996). New communication technologies for group decision making: Toward an integrative framework, in *Communication and group decision making,* (R. Y. Hirokawa and M. S. Poole, Eds.), 2nd ed., pp. 426–461. Thousand Oaks, CA: Sage.

Nunamaker, J. F., Briggs, R. O., Romano, N., and Mittleman, R. (1997). The virtual office work-space: GroupSystems web and case studies, in *Groupware: Strategies for corporate LANS and intranets* (D. Coleman, Ed.), pp. 231–254. Upper Saddle River, NJ: Prentice Hall.

Pinsonneault, A., Barki, H., Gallupe, R. B., and Hoppen, N. (1999). Electronic brainstorming: The illusion of productivity. *Information Systems Research,* Vol. 10, 110–133.

Poole, M.S., and DeSanctis, G. (1990). Understanding the use of group decision support systems: The theory of adaptive structuration, in *Organizations and communication technology* (J. Fulk and C. Steinfield, Eds.), pp. 175–195. Newbury Park: Sage.

Poole, M.S., Holmes, M. E., and DeSanctis, G. (1991). Conflict management in a computer-supported meeting environment. *Management Science,* Vol. 37, 926–953.

Scheerhorn, D., Geist P., and Teboul, J. C. B. (1994). Beyond decision making in decision making groups: Implications for the study of group communication, in *Group communication in context: Studies of natural groups* (L. Frey Ed.), pp. 247–262. Hillsdale, NJ: Erlbaum.

Scott, C. R. (1999). Communication technology and group communication, in *The handbook of group communication theory and research* (L. Frey, D. Gouran, and M.S. Poole Eds.), pp. 432–472. Newbury Park, CA: Sage.

Groupware

Joey F. George
Florida State University

GLOSSARY

computer-supported cooperative work (CSCW) The study of people working together on a common task, using computer-based tools.

group support system tool Software that supports the performance of a specialized group task, such as idea generation and prioritization. These and similar tasks are typically part of common group approaches to solving problems, especially in the context of meetings.

groupware A class of computer software designed to support the joint work efforts of several individuals on a common task.

videoconferencing Means of communication in which participants in two or more locations send and receive video images along with sound.

I. WHAT IS GROUPWARE?

Groupware is a class of computer software that helps members of a group work better together. Groupware makes it possible for group members to easily share information and to use that information to more easily support working together. Groupware allows group members to communicate clearly with each other, to coordinate their work, and to collaborate with each other. Communication involves sharing information. Coordination involves managing the task the group is working on and the process of completing the task. Collaboration involves the actual process of working together, where sharing information evolves into sharing ideas and producing solutions. As more than one person is involved in group work, groupware requires some type of shared computing platform to work effectively. It is important to remember that groupware tools are designed for multiple people to use. They are not single user tools.

II. THREE CATEGORIES OF GROUPWARE

There are many different software tools that can be called groupware. It is convenient to group these tools into three categories, based on that part of group work the tool best supports. Some tools primarily support communication, while others primarily support coordination, and still others support collaboration. It is rare, however, to find a groupware tool that performs only one of these functions. More typically, a tool will be designed to primarily support one type of group work, but it will also support other group functions to some extent. For example, a groupware tool that primarily supports communication will also provide some support for coordination and collaboration.

A. Communication Tools

Groupware tools that can be classified as communication tools primarily support the sharing of information among group members. These tools allow each group member to put the information they have

out there for the other group members to access. An example is a calendar system, which allows group members to share their personal calendars with each other.

B. Coordination Tools

Coordination tools focus on managing the work the group is doing, helping the group to design the best way to assign responsibility and to monitor progress being made on the group work. An example is a workflow system, which allows a group to break its work into smaller units and establish formal relationships among the smaller units.

C. Collaborative Tools

Collaborative tools focus on taking shared information and using it to create solutions to the problems the group has been assigned to solve. A special case of a collaborative toolset is a group support system. Group support systems provide a series of special purpose tools that help groups find solutions to problems. All of these different types of groupware, calendaring systems, workflow systems, and group support systems, along with other examples, will be discussed in more detail later in this article.

III. COMPUTER-SUPPORTED COOPERATIVE WORK

Computer-supported cooperative work (CSCW) is a related term that is associated with groupware. CSCW does not refer to a particular set of software tools, however, in the same way that groupware does. Instead, CSCW is the study of people working together, using computer-based tools such as groupware. The study of people together using computer-based tools is a broad area of study, involving scientists from many academic disciplines, including computer science, management, psychology, and management information systems. At the end of this article, where we review what we have learned about how groupware has been used in organizations, we will draw on the results of studies conducted by CSCW researchers. It is important to remember, however, that CSCW refers to the study of people using tools together and not to a class of groupware tools.

IV. THE GROUPWARE GRID

Traditionally, whenever a group of people wanted to work together to solve a problem, they had to meet in person at a common time and place. With the advent of the telephone, it became possible for people in different places to meet, but the meeting was restricted to only two people until it became possible to hold conference calls. Videoconferencing provided the ability to transmit pictures as well as sound, but people still met at the same time, even though they were physically dispersed. With computer-mediated communication such as e-mail, it became possible for groups to work together in different places and in different times. Communication no longer had to be synchronous for group work to be accomplished effectively. Looking at groupware in terms of time and place gives rise to a matrix or grid that crosses time and place, resulting in four different quadrants (Fig. 1).

A group meeting can take place in any one of the four quadrants in Fig. 1. The traditional meeting, where everyone meets together at the same time in the same room fits in the same-time, same-place quadrant. The telephone conference call and the videoconference both fit in the same-time, different-place quadrant. However, groups do more in their joint work than meet together. A big component of group work is making information available for all of the members of the group to use, as they need it, when they need it. Groupware tools that allow for this type of information sharing fit in the different-time, different-place quadrant. The fourth and final quadrant, same-place but different-time, is a little harder to visualize than the other three quadrants. A same-place, different-time tool would be a central repository of information that required group members to physically visit a single location in order to access information, although they could go to that location whenever they wanted. Of the four quadrants in the groupware grid, this is the least useful and the least likely to be sup-

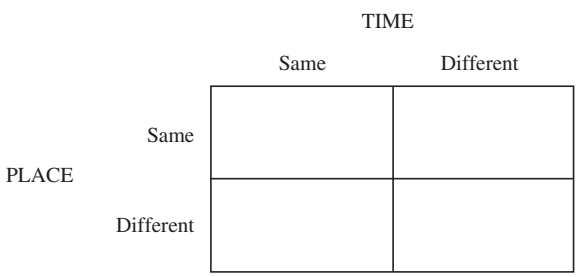


Figure 1 The groupware grid.

ported by a groupware tool. In the next section, different groupware tools will be defined and placed in the groupware grid.

V. SPECIFIC TOOLS AND THEIR ABILITIES

A. Communication Tools

Communication tools allow group members to share information with each other. Some are inexpensive and simple. Others are complicated and expensive and require trained technicians or third parties to operate. They all have in common the ability to allow group members to send information to others they are working with and to access the information they need. There are six different types of groupware tools we will discuss in this section: calendar systems, shared databases, whiteboards, computer conferencing, chat, and videoconferencing. Most groupware products include many of these tools, while some products include all of these tools.

1. Calendar Systems

A calendar system provides an individual, personal calendar for each member of the group. Each person enters appointments into his or her calendar, as would be the case for any other type of calendar or date book. The difference here is that each personal calendar can also be made available to other members of the group. Sharing the information in a personal calendar makes it easier to schedule meetings for groups, as each person can see when the others are free. The alternative is calling or e-mailing each person to find an open day and time and then trying to coordinate everyone's schedules to find a day and time when everyone is available. Usually this will involve contacting everyone again before an agreed upon date can be found. Typically, a personal calendar can be completely shared with others, meaning that others can read all of the calendar and its commitments or only part of the calendar that is accessible or none of it. Calendaring systems can also be configured so that locations have their own calendars. For example, if there is a conference room that is used only for meetings, all meetings scheduled in the room can be posted on the room's calendar. Anyone wanting to call a meeting in the room can consult its calendar to see when it is available.

The type of calendaring systems being referenced here are read-only; that is, group members can read each others' calendars, but they cannot write to them. A group member can find out when someone else is free, but he or she cannot then go and schedule an appointment in that group member's calendar. Once a calendaring system allows writing to as well as reading from calendars, then it is probably better to describe such a system as coordination groupware rather than communication groupware. Other coordination groupware tools are discussed later in this article.

2. Shared Databases

Many professionals keep detailed records about the clients they work for, the particular tasks they perform for clients, and their own business expenses. One reason for keeping such records is that many professionals bill clients by the hour and are required to put in a certain number of billable hours per work week. In many organizations, it would be very helpful if many employees had access to personal records of others in the organization. One benefit would be to identify contacts and leads for new projects. Another benefit would be to discover what approaches worked best in different job situations, helping to create a record of best practices. If knowledge workers could share all that they have learned in their experiences with customers, they could learn from each other about what it takes to do their jobs well. It is just this type of sharing of personal records that shared database groupware makes possible.

3. Whiteboards

Everyone is familiar with the whiteboard, a white drawing surface that you write on with different colored markers. In the groupware context, a whiteboard is a shared drawing surface, where the different participants in a meeting can draw objects that all of the others can see and add to. The interface for a groupware whiteboard usually resembles that for a standalone drawing tool. Users can draw in different colors, using different line widths and types. They can select basic shapes such as squares and circles, and they can label their drawings with text.

4. Computer Conferencing

Computer conferencing is very similar to e-mail. As a participant in a computer conference, an individual makes use of an editor to write a message that he or she then sends to the conference. Instead of going to an individual or set of e-mail addresses, the message

is posted to the conference. The message can be the first one in a stream of messages or it can be a response to a message that someone else has posted. By selecting a topic, the individual can follow the flow of ideas from the initial message through the entire stream of messages to the most current one. These streams of related messages are called threads. As the complete history of idea exchange and argumentation is readily accessible for anyone participating in the conference, it is easier to trace the development of an idea using a computer conference than it is using most e-mail systems.

5. Chat

As is the case with many of the groupware tools introduced in this section, chat is a stand-alone tool that has been incorporated into many groupware products. Like computer conferencing, chat facilities are similar to e-mail in that a user has an editor to type in comments and a window to view others' comments. Unlike either e-mail or computer conferencing, however, chat facilities are synchronous. Users exchange messages in real time, so that comments receive instant responses. Chat facilities, like chat rooms, are often associated with teenagers using the Internet, but chat has become a staple tool in many groupware products. Depending on the product and its configuration, it is possible for a group to use chat, a shared whiteboard, audio conferencing, and even videoconferencing all at the same time.

6. Videoconferencing

Of all the individual groupware tools discussed so far, the oldest may well be videoconferencing. In a videoconference, participants in two or more locations send and receive video images with sound. The effect is like watching live television, but with less demanding production values. Because sending and receiving video images and sound involves a large amount of bandwidth, videoconferencing, until very recently, has been expensive and has involved the use of special purpose facilities and staff. Cheaper and lower quality alternatives, such as desktop videoconferencing, have become widely available in recent years, made possible through the ready availability of small video cameras and other low-cost tools. For many types of equipment and configurations, it is also no longer necessary to have telephone company operators establish connections for videoconferences.

B. Coordination Tools: Workflow Systems

Coordination tools provide group members with the ability to plan and manage their work together. The best example of a coordination tool is a workflow system. Workflow systems are designed to make the flow of work necessary for a process more predictable and more manageable. To do this, most workflow systems take a three-part approach. First, the overall process is broken down into smaller component parts or subprocesses. The process can also be broken down into documents or user roles. Second, the relationships among the different components are formally defined and mapped. The relationships may be temporal or sequential or based on some other type of interdependencies. Third, the identified dependencies are then used as the basis for automating some parts of the workflow completely.

For example, a workflow system would allow workers or even clients to request a service by entering the request in the system. Others in the organization or even outside of it would be notified of the request and the action or response expected from them. Once work on the request has begun, whoever works on the request would update the system with the progress they had made. As the request worked its way through the system, its status could be checked at any time by workers or by the client who initiated the request.

C. Collaborative Tools: Group Support Systems

Collaborative tools go beyond communication and coordination. They provide groups with the support they need to take communicated information and to use it to solve problems, reach solutions, and make decisions. The best example of a collaborative tool is a group support system, which is actually a collection of many different and varied tools. A group support system, or GSS, typically assists groups in the meeting process and in joint efforts such as writing and editing.

1. Meeting Support

Although every meeting is different in terms of who attends and what the topic for discussion might be, many meetings are very similar in the process that is followed. A common goal for a meeting is the prioritization of ideas that are generated during the meeting. The ideas may be part of a strategic planning process, in which executives seek to determine what

the company should be doing in the short term and long term, or the ideas may be part of an effort to identify and make new products. Whatever the particulars may be, the process for determining the most and least important ideas in a set of ideas is fairly standard (Fig. 2). First, participants generate ideas. Second, the group works to consolidate the total number of ideas into a relatively small set. Third, the remaining ideas are prioritized from best to worst. Most GSSs have tools that support this process sequence.

a. Brainstorming

The GSS tool used to help with the idea generation process is usually based on a process called brainstorming. In traditional brainstorming, group members are given a topic and asked to come up with ideas relevant to that topic. Participants shout out their ideas, and a facilitator writes the ideas on a whiteboard or a flip chart. Participants are told not to evaluate each other's ideas, to be as creative as possible, and to piggyback off the ideas of others. The end result is supposed to be a long list of creative and useful ideas. A variant of brainstorming is called brainwriting. It works much the same way as brainstorming, but instead of shouting out their ideas, a participant takes a piece of paper and writes an idea on it. Then he or she puts the paper in the middle of the table where all the participants are seated. He or she then picks up a piece of paper someone else has written on and comes up with an idea that piggybacks off of what has been written on the paper.

A GSS idea generation tool is based on either brainstorming or brainwriting. In either case, a participant sitting at a computer screen reads the topic, enters an idea, and sends it to the server. In the brainstorming tool, the idea appears immediately on the participant's screen. Every other idea being entered appears on the screen too. As more and more ideas are entered, the screen scrolls up so that the latest idea to be entered is always visible. Each participant sees every comment being entered by every other participant as soon as the comment is sent. In the brainwriting tool, each participant gets back a file that contains other ideas entered by other participants. The participant then enters a comment, sends the file to the server, and gets back a new file. As the files are distributed randomly, no single participant is likely to see all of the files being passed around. In either approach, all of the ideas generated are captured by the GSS. Nothing is filtered through a facilitator, who may misunderstand the idea and not write it correctly on the board or flip chart, as is the case with traditional brainstorming. Also, as all of the ideas are captured by the system, no idea is lost. Everyone can enter their ideas in parallel to each other, speeding up the process.

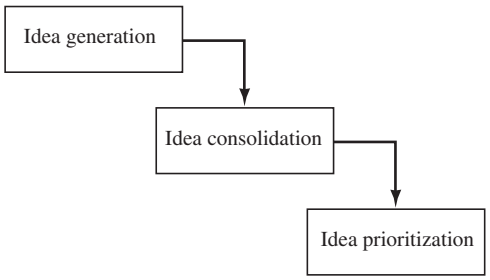


Figure 2 A typical group meeting process.

b. Idea Consolidation

Whether the model is brainstorming or brainwriting, the result is typically a large number of ideas, sometimes hundreds. Usually there are too many ideas to be successfully consolidated into a short list for prioritization. Fortunately, many of the ideas are duplicates, and some are fragments or are otherwise unclear. The GSS provides tools that help the group and a meeting facilitator go through the ideas and eliminate the duplicates and clarify or delete those that are unclear. Also, sets of ideas that are about a particular subtopic can be consolidated into a single, larger idea. This part of the process usually takes a long time and a lot of human intervention, although recent GSS developments have included attempts at idea consolidation using artificial intelligence techniques. The ideal result of the idea consolidation phase is a list of about a dozen ideas.

c. Prioritizing

The final step in this three-part process is prioritizing the list of ideas that resulted from the idea consolidation process. Ideas may be prioritized from the best to the worst, or from the most likely to the least likely to be implemented, or some other agreed upon criteria may be used. Most GSSs have many tools available for prioritization. One tool may allow participants to rank order ideas. Another tool may allow participants to assign scores or weights. Yet another tool may allow participants to each vote for only one idea. After all of the group members are done with their individual prioritizations, they submit them, and the GSS tabulates the group results. These results are then made available to the group so that they can see them and compare their own votes to those of the other group members.

2. Problems with Meetings

One of the reasons that GSSs were originally developed is because of the problems with traditional same-time, same-place meetings. Ideally, a group is created to work on a problem because each group member brings different skills and experience to the effort, and it is thought that the combined efforts of the group will result in a better solution than any individual could have reached alone. Many times, however, the hoped for result is not achieved because of problems with the traditional meeting process. Although there are many problems with the traditional meeting process that have been identified, three of them are worth discussing: (1) evaluation apprehension, (2) production blocking, and (3) free riding.

a. Evaluation Apprehension

Evaluation apprehension occurs in a traditional meeting when a participant has a good idea to share but is scared to say anything. This apprehension may be due to many things: the person may not feel comfortable talking in a meeting; the person thinks the idea is good but is not too sure, so he or she is afraid of getting called stupid for stating the idea in public; or the person may not want to say something in front of his or her boss, who is also attending the meeting, especially if the idea is critical of something the boss has done or is known to believe. The end result is the idea is never stated and may therefore be lost forever.

b. Production Blocking

Production blocking refers to ideas never being shared because of all of the other group activity going on around the person with the idea. As the person never states the idea, its production is therefore blocked. Some of the activity that may be going on around a meeting participant includes other people talking. If one person is talking, it is difficult for others to speak, unless there are clear rules that are established and followed about who speaks when. Even clear rules about speaking turns and times may lead to production blocking, however. If 12 people meet for 1 hour, and it is decided ahead of time that every person will speak for the exact same amount of time, then each person only gets to speak for 5 minutes during the meeting. If each person gets their 5 minutes at once, then anything they think of during the remaining 55 minutes never gets shared. Real meetings do not work this way, of course. Everyone is free to talk whenever he or she has something to say. But in real meetings, some people talk much more than others, and some people never talk at all. It is fair to say that for those who do not talk at all, all of their production is blocked during the meeting.

Other conditions that may lead to production blocking are forgetfulness and irrelevance. Sometimes when one person is talking and a second person thinks of something useful to say, the first person may talk so long that the second person forgets what the good idea was. Ideas become irrelevant when they cannot be introduced at the right time in a conversation. Conversations move and change as different people enter into the conversation and contribute. What may have been seen as relevant early on becomes less so as time moves on. Even though the idea may still be a good one, many people are reluctant to stop the flow of a conversation and pull it back to some earlier point. In this case, and in the case of forgetfulness, the ideas are not shared, so their production is blocked.

c. Free Riding

Free riding is a common occurrence in groups, especially when the group is large and when individuals in the group are rewarded as a group and not as individuals. It is very tempting to be in a group and let others do the work, especially if everyone gets equal credit for what the group accomplishes. That is what happens in free riding. One or more people coast along, not doing their fair share, while others take on the burden of doing their own work plus that of the others.

3. How Group Support Systems Deal with Problems with Meetings

GSSs have been designed to deal specifically with problems such as evaluation apprehension and production blocking. GSSs do not address free riding very well and may actually make it worse. Evaluation apprehension is addressed in a GSS through making input anonymous. The idea is that if the name of the person who has the idea is divorced from the idea itself, others in the group will evaluate the idea on its own merits and not based on any biases they may have about who said it. In such an atmosphere, it may even be safe to enter a comment that is critical of a program created by the boss, because no one can tell who came up with the idea. Anonymity in a GSS is not a blank check to criticize people, however. It is in fact a way to reduce the criticism of people, as the ideas are separated from the people who typed them in. Production blocking is dealt with in a GSS by having everyone enter their ideas simultaneously. Since everyone can type at once, there is no sharing of the meeting time, so every idea should get entered into the

record. There should be little or no production blocking due to the limited time available in a typical meeting. Forgetfulness or perceived irrelevance of ideas could still be issues and could therefore still lead to some production blocking, but this should not be as much of a problem as it is in typical meetings. The anonymity that helps reduce evaluation apprehension, however, may also increase free riding. Without anyone else in the meeting knowing your identity, and therefore not knowing if you have typed in any ideas or not, it is much easier to sit back and not contribute.

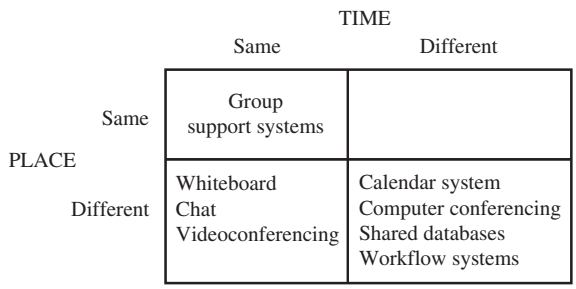


Figure 3 The populated groupware grid.

4. Group Editing and Writing

GSSs usually provide other tools that support other group activities, but the exact type and number of tools will vary from product to product. Another major focus for a GSS toolset is a tool that supports simultaneous group editing and writing. Group editing tools allow different members of the group to each take a different section of a document and work on it exclusively while other members of the group work on other sections. In addition to seeing their own parts of the larger document that they are working on, group members can also see the work being done by others. However, no one can edit a section being worked on by someone else at the time the work is being done. Once an individual releases a section, then someone else can edit it. Group editing and writing are useful when a group is responsible for creating a document, such as design specifications or some type of analysis or any other document that is the product of the group's work.

D. Populating the Groupware Grid

Now that groupware tools have been defined, we can place them in the groupware grid from Fig. 1. GSSs have been placed in the same-place, same-time quadrant (Fig. 3). GSSs have been placed here because the same-place, same-time environment seems to be the most common and the most powerful for GSS use by work groups. Considering that GSSs were designed to address many problems associated with traditional meetings, it makes sense that GSSs would have the most dramatic impact in such a meeting environment. Even though most GSSs do support same-time, different-place meetings, they have been placed in the same-time, same-place quadrant because that is their most common use.

In fact, special purpose facilities called decision rooms have been created to host group meetings supported by GSSs (Fig. 4). A typical decision room has a U-shaped table around which participants sit. Each person has their own computer to use to enter information and to access what others are entering. There is also usually a projector and a public screen which are used to display information useful to the group as a whole, such as aggregate views of their work. A facilitator who is not a member of the group typically leads group sessions in decision rooms. The facilitator has his or her own computer station, from which software tools are started and ended, and the overall process is monitored. Such dedicated facilities are not needed by any of the other groupware tools discussed in this article, except for the more traditional forms of videoconferencing.

Three groupware tools have been placed in the same-time, different-place quadrant. They are whiteboards, chat facilities, and videoconferencing. All three tools allow group members who are physically separated from each other to work together at the same time. Four groupware tools have been placed in the different-time, different-place quadrant. They are calendar systems, computer conferencing, shared databases, and workflow systems. All four tools allow group members to submit and access information whenever they need it, from wherever they happen to be. All four tools act as repositories of information, which is independent of place and time. Most groupware products contain several tools. Typically some tools support same-time, different-place group work, while others in the same toolset will support different-time, different-place group work.

VI. THE INTERNET AND GROUPWARE

As is true for any new class of information systems, when groupware products first appeared on the market, they were very expensive. As more and more people began to understand their usefulness, and as more and more vendors entered the groupware market, the

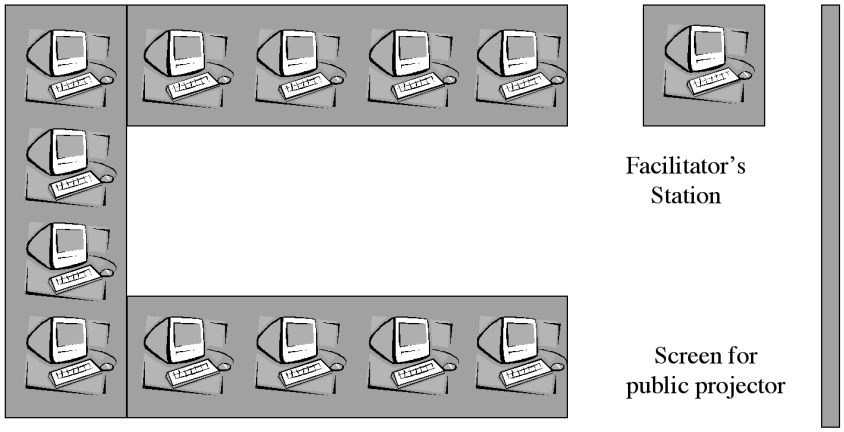


Figure 4 Group support system decision room.

number of available products increased and the costs fell. With the advent of the Internet, it is now possible to find fairly sophisticated groupware products available over the Internet at very low cost and in some cases for free. Many Web portals, for example, offer group calendar systems at no cost. The same is true for chat facilities and computer conferencing. Most of these free Internet tools are stand-alone tools, which do not allow for the advantages of having many tools in a single product, available through a single interface. Some groupware products, which combine many of the standard groupware tools, are included with computer operating systems and allow communication through the Internet. It is even possible to gain access to and use GSSs over the Internet at very low cost or for free. To avoid the overhead of installing and configuring a groupware product to run in-house, several application service providers (ASPs) have emerged to essentially rent groupware use to organizations on an as-used basis. Renting groupware from an ASP may cost as little as $20 per user per session. Given their growing availability and increasingly low cost, groupware products are fast becoming standard tools for group work and communication.

VII. WHAT WE KNOW ABOUT HOW GROUPWARE WORKS IN ORGANIZATIONS

As is the case with many new information system technologies, the use of groupware by people in organizations was very limited when groupware products first became available. With many software products, it is not immediately clear what the functions of the software are and how they would benefit the organization if the software was implemented. Groupware had the additional challenge of being software designed for collaboration among multiple individuals, something software did not do very well before that time.

A. Calendar Systems

When calendar systems were first introduced in the mid-1980s, they were rarely used. While managers and their secretaries made use of group calendars, very few other people did. The main reason for this was that most people did not see the benefit of keeping their personal calendars current on-line. As they did not see the benefits, they tended not to take the time and effort to either put their calendars on-line or to keep their calendars current. Managers tended to keep their calendars current because they often had assistance in doing so, i.e., their secretaries, and they also saw the utility in using calendaring systems. Managers tended to call meetings more often than other employees. Ten years later, calendar systems were more commonly used. One reason was the development of technological improvements, such as graphical interfaces, that made the tools easier to use. Another key factor, however, was social. Researchers found professionals under a great deal of peer pressure to use the software in organizations where group work was the predominant mode of work. When most people used the calendar system, those who did not use it actually hindered group task performance.

B. Shared Databases

A similar history exists for shared databases. When these systems were first available in the late-1980s, peo-

ple in adapting organizations were not quite sure what to make of them. To those who had not been properly educated to understand the communication and collaboration aspects of groupware, the tools seemed to be like other stand-alone tools they were already familiar with, such as e-mail. In situations where people do not understand the possibilities for groupware, they will either use it the same way they use stand-alone software tools or they will not use it at all. Furthermore, in organizations where the reward structure is based on keeping information personal and private, the introduction of groupware alone is not capable of instilling an atmosphere of sharing and trust.

In one interesting case, in a consulting firm where information was highly valued and was therefore kept private, the only groups that were successful in using a shared database system were those at the very top of the organization and those in the technical support area. Those at the top were able to make good use of the product because they were beyond the competition that marked the rest of the firm. They were used to working together, and the shared database system supported their efforts. Similarly, workers in the technical support area were also able to benefit from the product because they were also not subject to the competitive environment that most consultants worked in. Shared-database systems have become very popular in all types of organizations in the past decade, however. Such systems currently have millions of registered users.

C. Desktop Videoconferencing

Videoconferencing was also slow to get started, primarily due to the costs and special equipment needed. Desktop videoconferencing, which is much cheaper and requires relatively little technical expertise to operate, is currently becoming more widely available. Even in organizations that have adapted desktop videoconferencing, use among different members of the organization varies considerably. A 1-year study of an organization that was implementing desktop videoconferencing within the organization revealed several different reasons why people were not using the technology. Some people thought there were not enough other people using the system in the organization to make use worthwhile. They saw no point in trying to initiate a videoconference with others if the others were not using the system. There was some peer pressure to use the desktop videoconferencing system, but not enough to force widespread use.

Many of those who used the system used it more frequently to communicate with people who were physically distant from them. Others reported they did not use the system because it did not fit their jobs. Still others were concerned about privacy. They did not realize the system had built-in features that would have allowed them to control the extent to which others could see them. Finally, some people simply did not like to be seen during calls. In many ways, desktop videoconferencing is still in the early stages of implementation, much as calendar systems were in the mid-1980s. Desktop videoconferencing may well follow the same development trajectory as calendar systems, with use spreading as the technology improves and as peer pressure increases.

D. Group Support Systems

Finally, there has been extensive research into the effects of GSS use, both inside organizations and in the laboratory. Research from the field tends to report dramatic levels of task performance and user satisfaction with GSS products. Research from the laboratory tends to have more conservative effects. On the plus side, GSS use has been found to be associated with improved task focus for groups, more equal participation in the meeting process across group members, and better decision quality. On the minus side, GSS use has been associated with increases in the time necessary for a group to make a decision, lower levels of satisfaction for GSS groups compared to groups with no GSS support, and more difficulty for groups in reaching consensus on what they should do.

VIII. FUTURE RESEARCH

Future research in groupware will proceed on at least two fronts, software development and system implementation and use. In the area of software development, tool and product developers will continue to create and build new groupware tools that will expand the capabilities for groups that need to communicate, coordinate, and collaborate on their work across place and time. Many of these tools will fail to work as hoped, but others will become part of the mainstream of common computer-based tools. Researchers will take new and existing groupware tools and expand their capabilities and their abilities to run on new platforms, such as cellular phones and personal digital assistant devices. They will also make these tools more compatible with the Internet and more widely available over it.

Other researchers will study the implementation of groupware tools in work organizations and how the

tools are used (if they are used) after implementation. The implementation and use of older groupware tools, such as calendar systems and GSSs, has already been extensively studied in organizations. More recent tools, such as desktop videoconferencing, are only just now being studied as organizations decide to implement them and as workers figure out how to integrate the new tools into their daily work routines. The results from these studies will provide feedback to the developers, who will use what has been learned to improve the designs of existing tools and products and to develop new ones.

SEE ALSO THE FOLLOWING ARTICLES

Computer-Supported Cooperative Work • Database Administration • Decision Support Systems • End-User Computing Concepts • End-User Computing Tools • Group Support Systems and Electronic Meeting Systems • Internet, Overview

BIBLIOGRAPHY

DeSanctis, G., and Gallupe, R. B. (1987). A foundation for the study of group decision support systems. *Management Science.* 33(5):589–609.

Diehl, M., and Stoebe, W. (1987). Productivity loss in brainstorming groups: Toward the solution of a riddle. *J. Personality and Social Psychology.* 53(3):497–509.

Ellis, C. A., Gibbs, S. J., and Rein, G. L. (1991). Groupware: Some issues and experiences. *Communications of the ACM.* 34(1):38–58.

Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. In *CSCW '88 Proceedings,* pp. 85–93. New York: ACM Press.

Grudin, J., and Palen, L. (1995). Why groupware succeeds: Discretion or mandate? In *Proceedings of the Fourth European Conference on CSCW,* H. Marmolin, Y. Sunblad, and K. Schmidt, eds., pp. 263–278. Dordrecht/Norwell, MA: Kluwer Academic.

Grinter, R. E. (2000). Workflow systems: Occasions for success and failure. *Computer Supported Cooperative Work.* 9:189–214.

McLeod, P. (1992). An assessment of the experimental literature on electronic support of group work: Results of a meta-analysis. *Human-Computer Interaction.* 7:257–280.

Nunamaker, J. F., Jr., Dennis, A. R., Valacich, J. S., Vogel, D. R., and George, J. F. (1991). Electronic meeting systems to support group work. *Communications of the ACM.* 34(7): 40–61.

Orlikowski, W. (1992). Learning from notes: Organizational issues in groupware implementation. In *CSCW '92 Proceedings,* J. Turner and R. Kraut, eds., pp. 362–369. New York: ACM Press.

Jessup, L. M., and Valacich, J. S. (1993). *Group Support Systems.* New York: Macmillan Co.

Webster, J. (1998). Desktop videoconferencing: Experiences of complete users, wary users, and non-users. *MIS Quarterly.* 22(3):257–286.

Health Care, Information Systems in

Joseph K. Tan
The University of British Columbia

GLOSSARY

asynchronous transfer mode (ATM) A network connection that breaks data down into packets of uniform size that can be sent synchronously and then reassembled in the proper sequence at the other end. An ATM network can connect different platforms at 50–155 mbps.

Bayesian forecasting The Bayesian forecasting techniques are based on the concept of using prior knowledge for calculating conditional probabilities so that "base-rate" probabilities can be continually updated.

clinical decision support system (CDSS) A subclass of health division support systems that provides clinical data banks and algorithms, analytic or pathophysiological models, clinical decision theoretical models, statistical pattern recognition methods, symbolic reasoning, and clinical expert knowledge bases to support diagnostic thinking and cognitive reasoning strategies of expert and less-than-expert clinicians.

cognitive task analysis (CTA) A set of methods that can be used to assess and capture the knowledge, skills, and cognitive processing of experts and less than experts. CTA focuses on high-level cognitive processes in task performance and identification of problems and strategies.

data envelopment analysis (DEA) A analytical, linear programming-based technique used to benchmark and compare performance of single objectives of comparable units. The methodology can easily be extended for problems with multiple objectives.

data mining (DM) The analysis of large database warehouses to discover patterns in the data and to reveal associations between outputs and inputs. Data-mining algorithms cull through medical data to find these associations.

expert decision support system (EDSS) A computer-based consultation program that combines expert methods and decision support technologies to aid the decision-making process of an expert or less than expert.

health decision support system (HDSS) An interactive, user-controlled system designed specifically to assist the decision maker in utilizing data and models for solving semistructured problems.

health maintenance organization (HMO) A network of independent practitioners and provider organizations that is organized to provide managed care and/or comprehensive health care to patients who make periodic payments in advance depending on the extent of coverage being subscribed.

health management information system (HMIS) The application of a total information management systems' perspective in linking relevant theoretical principles with practical methodologies and their applications to improve health service delivery within the context of current and future health care environments.

medical informatics and telematics The study of methods in medical, clinical, and bioinformatic computer-based information processing, storing, managing, transferring, and telecommunications and their applications to improve health service

delivery within the context of current and future clinical health care environments.

multiview A methodology for the health management information system developmental process that brings together various logical world tools and techniques to provide a smooth transition between the logical world stage and the physical world stage in the HMIS modeling process.

neural networks The domain of artificial intelligence that mimics the reasoning and learning systems of the human physical brain structure for data mining.

organization technology interface The boundary between various organization human resources and technological systems, particularly computer hardware and software systems, where all of the organization information and communications are managed.

stochastic model A probabilistic-based statistical model for the analysis of experimental data to obtain a conclusion within confidence limits.

I. INTRODUCTION

The historical evolution and knowledge development of information technology (IT) and information systems (IS) in the health care arena over the last 50 years have been and are still transforming health care services delivery systems and environments.

Specific to the health care industry, administrators, clinicians, researchers, and other health practitioners are facing increasing pressures to adapt to growing public and private sector accountability. Major sources of concerns come from decreased government and third-party funding, increased patient education, participation and expectation, new and emerging forms of health organizational reporting structures, as well as rapid and exciting breakthroughs in computing and networking technologies. The need for more efficient and effective health data sharing, voluminous health information processing, and coordinated health knowledge management and health decision making, therefore, argues for a critical effort to bridge the concepts of information systems and the management of health care services delivery.

In light of these changes and needs, those who understand health information system (HIS) concepts, trends, and challenges and the potential benefits of health IT applications will be better prepared to work collaboratively using computer-based groupware and net appliances during the current knowledge era to achieve greater group productivity, to improve health organizational decision-making effectiveness, and to cultivate more positive interorganizational partnerships in the context of the growing number of electronic health (e-health) services delivery networks. Even so, rapid shifts in the e-health markets and environments have also dictated a growing need for improved ergonomics and more intelligent interfaces in current HIS applications and implementations.

Generally speaking, IS in health care may be viewed as an integrated, multidisciplinary field, bridging key areas of strategic health systems planning and management, corporate, and departmental HIS technologies and applications; tactical IS design and developmental methodologies; health informatics and telematics; health operational analysis and specialized computer modeling; and practical implementation and ongoing evaluation of mostly automated health information management and clinical management systems. More particularly, three major themes underlying this eclectic field identified elsewhere by the author of this article include: (1) HIS technologies and applications, (2) HIS theories and methodologies, and (3) HIS administration and impacts.

Before leading the readers through a bird's eye view of each of these themes, it appears desirable that they have a clear understanding of the fundamental concepts of information and the basic processes of an HIS. Section II of this article therefore focuses on the basic concepts of information and HIS functions. Section III surveys the history of HIS technologies and applications. Section IV highlights cornerstones of HIS theoretical and methodological foundation, while Section V shifts attention to key HIS administration issues and impacts. Section VI concludes the entire exposition with a focus on how the different themes discussed here can be combined to achieve an effective HIS implementation and provide some further considerations on how HIS practice in the coming era may be affected by changing HIS needs, trends, and challenges.

II. INFORMATION CONCEPTS AND HEALTH INFORMATION SYSTEM FUNCTIONS

An HIS may simply be conceived as an integrated health information processing engine that links interrelated human–computer components for the accurate and rapid collection of various patient-related data, information, and knowledge elements to generate aggregated, well-classified, and needed administrative and clinical information and knowledge for aiding users in retrieving and disseminating such information and knowledge for use in decision making,

control, analysis, diagnosis, treatment planning and evaluation, and many other subsequent health-related cognitive activities. Other terms referring to a related or similar conceptualization of HIS that are commonly encountered in the emerging literature on IS in health care include health care information systems (HCIS), health management information systems (HMIS), medical information systems (MIS), medical informatics (MI), as well as health informatics (HI) and telematics.

A. Health Data vs Clinical Information and Knowledge

On the basis of this general HIS conceptualization, health data are therefore differentiated from clinical information and knowledge in specific terms, that is, the emphasis on some form of meaningful processing having taken place to convert health data to useful clinical information and knowledge and the significance of applying readily available, accessible, appropriate, accurate, and aggregated clinical information and knowledge as opposed to raw data gathered at the source to aid insightful and intelligent health-related decision making under increasingly complex and demanding situations. Knowledge, as the term suggests, refers to the cumulative experience of applying information to decisions, thereby producing "wisdom," "rules of thumb," and "associations" to be used for future health-related decision situations.

Health data elements are specific facts and parameters. A good piece of datum is characterized by its accuracy, reliability, completeness, accessibility, timeliness, and security. Accuracy is achieved when the health data recorded are true, correct, and valid about the status of a patient's condition; for example, a temperature of 104°F recorded as 101°F is not accurate. Reliability means that the health data recorded are trustworthy and consistent; for example, if the allergy list of a patient exists in the food services system, the same list should appear in the pharmacy system. Completeness entails that all required health data should (and must) be recorded; for example, a unique identifier must exist in the patient master index (PMI) for each patient recorded in a database to differentiate one patient from the other. Accessibility refers to empowering appropriate personnel with valid access authorization and authentication to view the relevant data wherever and whenever required; for example, physicians should be able to view their patients' electrocardiography (EKG or ECG) or electroencephalography (EEG) reports in their private offices following their rounds in the wards. Timeliness ensures that the available health data are current for the decision tasks at hand, while security and privacy stipulate that only designated persons with valid access rights can view or make changes to any or all relevant aspects of the recorded data. This will ensure patients' data confidentiality and safeguard against data misuse.

Clinical data are unique. Unlike financial and accounting data, clinical data are typically nontransactional. Even so, data about a patient may be entered by different personnel, by different departments, and at different times to show the progress of health recovery of the patient. For example, a nurse may jot down the demographics of a patient when she or he first arrives, then the physician or other specialists will record their observations and diagnoses about the patient. These data are mainly textual in nature. The physician and specialists may also send the patient for laboratory tests and other scans which will give rise to other types of quantitative and qualitative data; that is, these data can range from data derived from statistical and complex simulation models and tabular and graphical presentation data to digitized images to opinions and interpretations of the data. Similarly, when a patient is discharged, the process will further create administrative and financial data (e.g., billing and accounting data).

In the context of a health organization such as a health maintenance organization (HMO), HISs are used primarily for gathering patients' financial and clinical data, following which these various sources of data elements are aggregated and manipulated using necessary and relevant models including built-in knowledge elements as needed to support health providers (e.g., health administrators, clinicians, nurses, and other health care professionals) in making timely decisions in order to improve the efficiency, effectiveness, and efficacy of health services rendered or to be rendered to the patients.

Collected health data must be meaningful and worthwhile. While health databases must be properly organized and made available to their users in a timely fashion, it is critical to consider and understand the needs of their users prior to the data collection process. Otherwise, managing and maintaining inappropriate and unnecessary data, especially in large medical databases, may wastefully drain away valuable health organizational resources. Users of medical and health-related data range from patients to care providers, government agencies, health care planners, judicial agents, educators, researchers, and third-party payors. Different types of users may also require different scope, format, and presentation of data. To

design and build an effective HIS, it is critical to fully comprehend the basic functions of an HIS.

B. Basic Functions of a Health Information System

Basic functions of an HIS may be loosely grouped into three phases: data input, data management, and data output phase, as depicted in Table I.

A major function of an HIS is to gather appropriate and sufficient data from various sources before any user can meaningfully interpret the data to satisfy the administrative and clinical needs of patients and providers. The fundamental value of an HIS lies in well-managed data collection and verification processes. Data collection is not about collecting every piece of information, but only those that are vital to assist patients, providers, and other secondary users (e.g., third parties and researchers).

In the past, clinical and health-related data about a patient were frequently coded differently by different coders, resided in different health information systems and databases, and were also captured in a different format. To improve data timeliness, validity, and integrity, the preferred strategy in data collection method today is to use automated and direct data input at the source, such as the use of a point-of-care bar code scanner, and to warehouse the data either centrally or via on-line distributed network technology. Such approaches require that the acquired input data be first converted into standardized codes. This strategy is preferred to traditional manual and mechanical input and conversion methods that rely chiefly on clerical transcription of patient data from various self-reports and handwritten documents via the keyboard or some other input devices that are also prone to human errors. Regardless of the way patient-related data are gathered, coded, and entered into an HIS, the input data elements should be meticulously verified for accuracy and validity before moving onto the data management phase.

In the next important phase, data management aims to transform the input data into useful and meaningful information in a format that is readily retrievable, comparable, and understood. Three common forms of data management technologies often used during this phase include database management, model-base management and knowledge-base management technologies. Database management enhances data collection and storage activities, improves data integrity, reduces data update anomalies, and promotes preservation and structuring of data for efficient data processing and effective data retrieval activities. Model-base management constructs, establishes, manages, and interrelates models that may be needed by the user to rationalize the data being linked, analyzed, or computed. In fact, not all data collected are directly useful in the health care delivery process. Some information is collected merely to assist in the organization and generation of comparative data and statistics or simply for research. More intelligent health information systems can assist health care providers and clinicians in making more complex decisions that may require expertise within a specialized domain. In this case, apart from the use of database and model-base management technologies, knowledge-base management systems come into play. Knowledge-base management assembles, stores, accesses, updates, and disseminates knowledge elements that may enhance the processing of such specialized decisional processes.

The final phase of data output is concerned with data transfer and data distribution (i.e., data retrieval and transmission activities) and with data presentation and use. A key problem in data output using current HIS technologies is the tendency to provide unnecessary, excessive, or overwhelmingly complex information that is not helpful or needed by the users to complete the task at hand. Accordingly, computerized software and intelligent graphical interfaces can be built to compact a large amount of information conveniently and to support individual users in filtering out the irrelevant information that may not be needed for a particular or specialized task application.

Quality displays and more appropriate representations of health data are important because not only

Table I Basic Functions of an HIS

Data input phase	Data management phase			Data output phase
Data acquisition		Data storage		Data retrieval
Data verification	Data classification	Data computation	Data update	Data presentation
	Data management	Model-base management	Knowledge-base management	Data usage

can inappropriate representations of data slow down the process of data interpretation, but the decisions made arising from poorly represented data could also be error prone and generate unwanted or inappropriate clinical interventions. The application of a poor image compression technique or the use of an inadequate digital image resolution, for example, may not only slow down the reading of a scanned image for a radiologist, but it can easily lead to risky misdiagnosis on a patient's health condition.

III. HISTORY OF HEALTH INFORMATION SYSTEM TECHNOLOGIES AND APPLICATIONS

All students of the HIS field should become admirably familiar with the historical evolution and knowledge development of HIS technologies and applications. Owing to space limitation, the readers are asked to seek out further readings, as provided in the listed references at the end of this article. Accordingly, a more detailed discussion on the early and more recent years of both "failing" and "successful" HIS developments can be easily found in the rapidly expanding literature on health and medical informatics.

A. Genesis of Health Information System Technologies and Applications

The genesis of HIS technologies and their applications date back to the 1950s. This was the time when only mainframes were available; when only the very large and major hospitals of G-7 countries could afford to house and use these machines as they were both very bulky and very expensive; when programming was done entirely by trained professionals at the assembly language level; and when even the processing of a routine batch of health-related information took considerable time, knowledge, and skill to operate successfully. Nonetheless, even with the use of computers, the end results were mostly fraud with mechanical and programming errors. The failures of this first era of IS in health care was due mainly to the lack of interest on the part of hospital administration, the lack of funding, and the lack of knowledge and skill in the design and use of computerized systems.

From the early 1960s through to the 1970s, a new era of IS in health care emerged when a growing group of hospitals throughout the United States (e.g., Akron Children's, Baptist, Charlotte Memorial, Desconess, El Camino, Henry Ford, Latter Day Saints, Mary's Help, Monmouth Medical Center, St. Francis, Washington Veteran's Administration, and others) and in Europe (e.g., Sweden's Danderyd Hospital and Karolinska Hospital, England's London Hospital and Kings Hospital, and Germany's Hanover Hospital) began to agree on the need for advancing the development of a patient information management system prototype. Despite the risk of major HIS failures, these "pioneering" hospitals invested large amounts of money, time, and effort to move toward computerization.

Seeing the sudden interest being expressed by these hospitals and the potential market opportunities arising from this movement, large computer vendors such as Burroughs, Control Data, Honeywell, IBM, and NCR provided support toward making a patient information system a possibility. Lockheed Information Systems Division, McDonnell-Douglas, General Electric (GE), Technicon Corporation, and several other companies that were experienced in the application of computers for managing complex systems also joined in this effort.

Unfortunately, many of these early attempts resulted in almost complete failures, as the perceived complexity of the information requirements of a patient management system was gravely underestimated by most computer vendors, and some companies such as GE and Lockheed had to withdraw from their participation due to lack of continuing funding and management support. Many of these pioneering hospitals also had to fall back on their manual systems to keep their facility running smoothly, and several of the hospital administrators had to make the very difficult choice of abandoning the HIS project altogether at a huge cost.

Yet, the Technicon system, which was initiated by Lockheed for the El Camino Hospital in Mountain View, CA, and later bought over and improved by the Technicon Corporation under the leadership of Edwin Whitehead, became the one successful prototype that laid the foundation for all future working hospital patient information management systems throughout North America and Europe. The major lesson that was learned in the El Camino project was the need to focus on users' information needs and the need to change users' attitudes, in particular, physicians' and nurses' resistance.

Large-scale data processing applications in medicine and health record systems also began to diffuse during the early and mid-1970s as continuing gains in productivity and evidence of increased efficiency could be traced back to computerization. However, successes for many early HIS projects were still few and far between due to their pioneering nature and high costs. Johns Hopkins Oncology Center acquired their first computer system in 1976 for $250,000 and its processing power was only a fraction of today's

personal computers (PC). Among early patient record systems, Computer Stored Ambulatory Record System (COSTAR), Regenstrief Medical Record System (RMRS), and The Medical Record (TMR) were other successful examples. COSTAR, a patient record system developed at the Massachusetts General Hospital by Octo Barnett in the 1960s, was later extended to record patient data relating to different types of ailments, for example, multiple sclerosis (MS-COSTAR). RMRS was a summary-type patient record system implemented in 1972, and TMR was an evolving medical record system developed in the mid-1970s at the Duke University Medical Center. Together with the success of the Technicon patient management system, the use of these automated record systems soon indicated a considerable need for widespread introduction of computing into health care.

When minicomputers were later introduced into organizational life during the late 1970s and early 1980s, health administrators and practitioners soon began to realize the efficiency and data processing power of the computers for harnessing the large and increasing volumes of medical and other health-related data. Medical data include demographics of patients; clinical and health services data; and epidemiological and health population statistics, for example, the prevalence and incidence of a disease such as tuberculosis (TB) or statistics on TB morbidity and mortality. Other health-related data include health administrative and patient financial data, such as inventories of drugs and medical equipment, and routine transactional data including the management of patient billings, insurance copayments, account receivables and payables, and general ledgers.

B. Traditional Health Information System Technologies and Applications

By the early 1980s, trends in size and cost reduction of computers with corresponding increases in powerful processing chips dramatically resulted in the move away from focusing only on massive health data processing using the mainframe or minicomputers to new and more efficient forms of health information management (HIM), office automation (OA), and teleprocessing via PC environments.

Whereas HIM is concerned with the automation of routine management reporting to support administrative and patient care applications, OA is concerned with the automation of health office systems and processes to reduce expenditure of time and effort of health knowledge workers. Teleprocessing is still a relatively new aspect of HIS technology and refers simply to the electronic transmission of health data, text, or voice information from one computer (source) to another (destination). Hence, teleprocessing merely enhances HIM and OA applications. Common examples of traditional HIS applications employing these technologies include admission-discharge-transfer (ADT) systems, medical records, financial and accounting systems, quality control and scheduling systems, resource management and material management systems, contract and claim management systems, as well as hospital and stand-alone clinical information management systems.

Each of these different HIS technologies contains both hardware and software components interacting with the human component as a catalyst to bring about customized, efficient, and effective sought-after solutions (information) to problem tasks (questions) encountered in the different areas of health services delivery. It was soon realized that the key to success for these technologies lies chiefly in the design of the human–computer interface, for example, a move away from mechanical input devices (e.g., keyboards, mouse, or tracking ball) to more convenient and human-friendly interfaces (e.g., optical character recognition, touch screen, light pen, and voice recognition), especially for clinical users such as nurses, general physicians (GPs), and specialists.

Beyond HIM, OA, and teleprocessing applications, increasing interests in health computing technologies and applications among health researchers and practitioners during the late 1980s and early 1990s quickly evolved into areas of expert method applications, clinical decision support systems, nursing decision support systems, and other forms of health decision support systems (HDSS). Researchers have always wanted to add intelligence to computer systems, and its extension to MISs quickly became noted as a most valuable and noteworthy application domain. Researchers believe that intelligent MISs should be able to mimic the thinking processes of clinicians. How such thinking processes may be engineered and expert knowledge programmed into automated systems are issues relating to HIS methodologies. These issues will thus be further explored and elaborated in Section IV.

At this point, current and emerging trends in HIS technologies and applications will be discussed.

C. Current and Emerging Health Information System Technologies and Applications

Apart from the HDSS and expert method applications, new and emerging forms of HIS technologies and strategic HIS applications have also advanced in

many other fronts. These include virtual patient records (VPR), document management (DM), geographical information systems (GIS), group HDSS, executive information systems (EIS), data warehouse (DW), data mining applications, networking and asynchronous transfer mode (ATM) networks, community health information networks (CHIN), the Internet, intranets and extranets, HI, and telemedicine. Owing to space limitation, it is possible to provide readers with only brief glimpses of each of these sometimes overlapping technologies. Readers who are interested in pursuing further details can refer to references provided in the Bibliography.

VPR refers to virtual information technology architecture for housing uniquely identifiable information about an individual patient from various distributed sources. Accordingly, diversified data coded in different format for use in different platforms can be converted into a common format for use in a common virtual platform to aid clinical and other health practitioners who are geographically separated. As such, this is not just the creation of a central depository like a massive traditional database, but the design and development of a common network via an open system for the conversion and transmission of media-rich medical data from multiple distributed sources to support multiple users. TeleMed is a collaborative VPR prototype project initiated between researchers at Los Alamos National Laboratory and physicians from the National Jewish Medical and Research Center in Denver that supports real-time interactive uses of media-rich graphical patient records among multiple users at multiple sites. In this sense, VPR technology is ideal for telemedicine practitioners.

DM technology aims to put clinical and financial data on-line. Among other possibilities, DM applications can include document imaging, workflow, electronic form processing, mass storage, and computer output to laser disk technologies. Many hospitals and health organizations can make use of DM technology to handle the paper-intensive process of collecting and filing patient information, for example, the use of notebook computers and customized software can empower busy nurses to rapidly and accurately update all patient and insurance records electronically rather than having to hand write and later transcribe many new documents and forms in performing their regular visits to patients. This technology essentially frees up time for nurses to focus on patient care.

A GIS is a powerful HIS tool for collecting, recording, storing, manipulating, and displaying spatial data sets, specifically mapping information. In other words, a GIS uses spatial data such as digitized maps and is capable of representing a combination of text, graphics, icons, and symbols on two- and three-dimensional maps. An application of GIS technology may be the digital mapping of a certain epidemic, for example, HIV infection among a subpopulation group across various counties in a Canadian province, to depict the spread of the prevailing ailment. This knowledge can then be used to target interventions effectively for specific population groups rather than focusing treatments on isolated individuals.

A group HDSS combines analytic modeling, network communications, and decision technology to support group strategic thinking, problem formulation, and goal seeking solutions. It aims essentially at easing the group decision-making processes, and among other things, its use will not only reduce the cognitive burden, but will also reduce the mental effort associated with group meetings. Therefore, a major benefit of this technology is its potential for increasing efficiency, effectiveness, and productivity of group interactions through the application of asynchronous board meetings, on-line forums, or special group meetings conducted electronically where board members and executives can network and share information with each other without being constrained by separation in time and geographical distances.

In the context of health computing, an EIS collects, filters, and extracts a broad range of current and historical health-related information from multiple applications and across multiple data sources, both external and internal to a health provider organization, to provide health executives and other key stakeholders with the appropriate information to identify key problems and strategic opportunities. A common example of an EIS application in health care is its use by HMO executives in strategic planning sessions to determine the challenges and potentials of various business strategies going forward. One popular feature is the capability of an EIS to drill down information from one level to another. Such capability enables these executives to retrieve answers conveniently to special and ad hoc queries. Another important feature is the ease of integrating EIS technology with other related technologies such as a GIS, an expert system (ES), an HDSS, or a group HDSS.

DW architecture for integrated information management is simply a technology for providing an integrated source of aggregated, organized, and formatted data. In this sense, the data in a DW are aimed at supporting management decision and strategic planning. Accordingly, these data may sometimes be stratified and most likely already aggregated and filtered from legacy systems. Again, the use of a DW may be combined with an EIS, an ES, an HDSS, a group HDSS, and a GIS to increase data analytic and processing

power and also to develop new and complex forms of HIS technologies.

Today, the most prominent use of a DW in health care is the collection of massive linked data from diverse sources for the application of data mining (or sometimes referred to as data dipping) techniques to discover new knowledge. Such data mining techniques explore the data in the DW for hidden trends and patterns. Examples of data mining tools include artificial neural networks, case-based (analogical) reasoning, statistical methods, genetic algorithms, and explanation-based reasoning. The ability of an HMO to explore and discover the best practices by comparing and contrasting physician practice patterns for different treatment protocols corresponding to specific case mix groups based on an analysis of historical data captured over time is an application of the DW and data mining technologies. The unraveling of the human genome for deciphering underlying genetic codes to provide treatments for various challenging ailments is another noble example of DW and data mining technology applications.

Health networking is essentially solving the underlying problem of virtual or e-health care by quickly moving massive amounts of media-rich information from one point of a network to another. The architecture of such a network may be a hub-and-wheel communication configuration; an open system configuration, including the use of the Internet, electronic data interchange (EDI), or extranet configuration; a groupware; or an intranet. CHIN may be conceived as a network that links health care stakeholders throughout a community or a region. More specifically, it is an integrated collection of telecommunication and networking capabilities that facilitate communications of patient, clinical, and financial information among multiple providers, payers, employers, pharmacies, and related health care entities within a targeted geographical area. Central to its success is the practical implementation of a computerized patient record system at the community or regional level. Figure 1 shows the Wisconsin Health Information Network (WHIN) as a practical example of a participating CHIN.

The Internet may generally be viewed as a complex web of networks. Currently, HIS technology has become an important interactive research and communication tool for aiding both medical professionals and health consumers in search of health-related information and knowledge. Extending the concept of the Internet are the intranets and extranets. These use the same hardware and software to build, manage, and view Web sites. Unlike the Internet, however, these private virtual networks are protected by security software known as "firewalls" to keep unauthorized users from gaining access. Essentially, an intranet is a private computer network to support Internet-based services only to inside organizational members, whereas the extranet

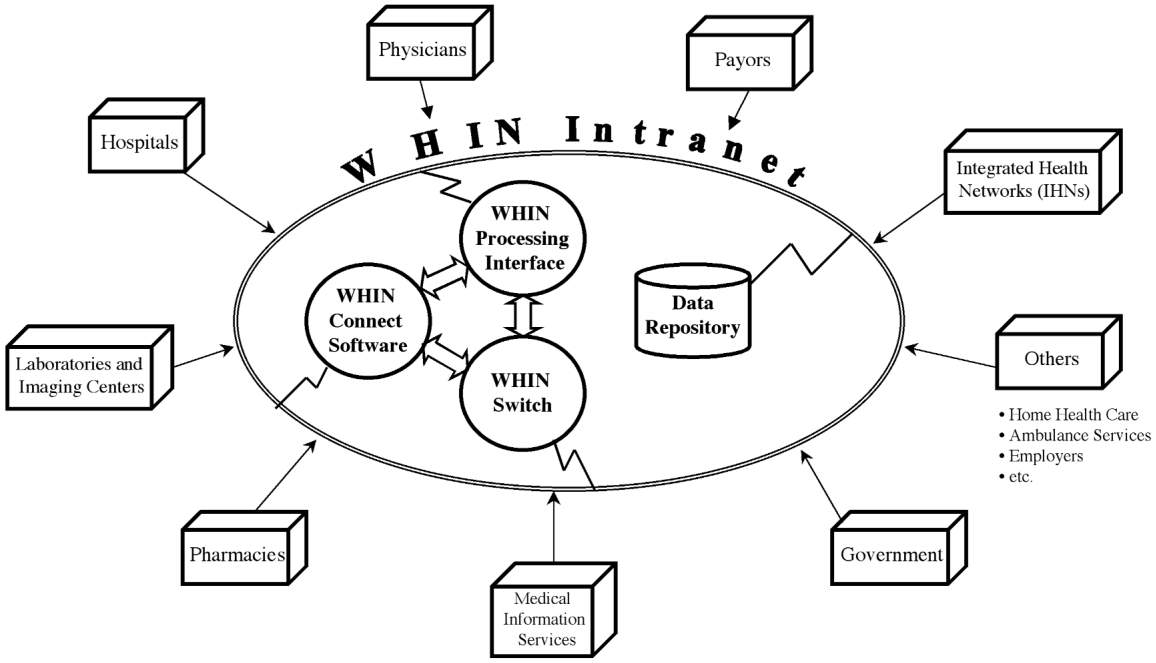


Figure 1 Participants of Wisconsin Health Information Network (WHIN). Copyright 2002, Joseph K. H. Tan.

extends network access privileges to certain partners, giving them access to selected areas inside the private virtual network and thereby creating a secure customer or vendor network. Examples of Internet, intranet, and extranet use in health care abound. A simple case is that of providing users such as patients, physicians, hospitals, and others with access to on-line insurance service data. The benefits of electronic filing of insurance benefits and claims via an extranet would dramatically cut costs such as agency and other labor costs while increasing the accessibility of its network to both patients and providers and would provide insights to health care trends and medical best practices.

The chief emphasis of HI and telemedicine is on clinical and biomedical applications of the different HIS technologies reviewed so far. At the clinical level, HDSS, clinical decision support systems (CDSS), and expert decision support systems (EDSS) are being developed to assist physicians and other medical specialists in diagnosis and treatment. An example of a CDSS is an interactive video disk system that helps enter personal health data to weigh the pros and cons of surgery. Other examples include systems for monitoring and alerting care providers to abnormal heart rates, for guiding prescribing pharmacists to potential adverse drug interactions, and for educating patients on preventive health care and health-promoting activities.

Finally, telemedicine is the use of digital networks to perform virtual diagnoses of disease and disorders. Teleradiology is among the first successful applications of telemedicine where X-rays or scanned images of patients are digitized and stored electronically for sharing among multiple health providers at geographically distant sites. Other examples of telemedicine applications include teleconsultation such as teledermatology and telepathology, telesurgery including telegastroscopy, robotics and virtual reality, telelearning such as videoconferencing and on-line medical education, and telecare.

IV. CORNERSTONES OF HEALTH INFORMATION SYSTEM THEORIES AND METHODOLOGIES

To date, considerable attention has been given by HIS researchers to etch an identity for the HIS field, in particular, the building of a cumulative tradition for HIS theories and methodologies. On the one hand, prominent HIS theoretical foundations include general systems theory, information and communications theory, decision theory and decision-making models, and end-user and organizational behavior theories. On the other hand, some of the key HIS methodological foundations include information processing methods, IS development methodologies, and methods for knowledge elicitation. In this section, due to very limited space, only a few of these more popular and interesting HIS theoretical and methodological foundations will be highlighted.

A. General Systems Theory

The general systems theory, also known as "Cybernetics," is a major cornerstone of HIS theoretical and methodological foundations. Systems theory has played a vital role in contemporary science in the information era. The theory begins with the empirical observation that all "systems," regardless of their disciplinary domain, share some important similarities in their underlying structure. They also exhibit some common behavioral patterns, such as statistical constancy, growth and decay trends, and rhythmic or oscillatory behavior.

Broadly defined, all systems have objects and attributes. Objects constitute the components of a system, whereas attributes are the properties associated with these objects; that is, an attribute is an abstract descriptor that characterizes and defines the component parts of the system. In considering a hospital bed allocation system, for example, objects of the system may include the actual beds; patients; health service providers who attend to the patients; and the computer that stores, analyzes, and provides the bed allocation information. The attributes describing the object "patient" in this case may include the patient's name, gender, age, medical insurance number, and a brief description of the diagnostic case mix group for which the patient is considered to have been admitted. Moreover, the number of beds to be allocated daily may vary considerably depending on whether or not there is a job action among unionized workers or a sudden outbreak of a certain epidemic, for example, mumps or measles among children in the community. Over time, however, it is expected and usual for this bed allocation system to reach some statistical stability or equilibrium if it is to continue functioning efficiently and effectively within a hospital setting.

A system combines all the objects and their attributes and defines the relationships among these objects, thereby enabling the different parts to add up to some greater unity than all of its individual parts. For example, the emergent property of an HIS is more than its individual hardware, software, application task, and user components because of the many interfaces between its human and machine components.

More generally, a system is a set of interrelated elements. An open system is one that interacts with its environment, whereas a closed system does not. The structure of a system may involve a hierarchy of embedded subsystems, each having its own unified purpose that contributes jointly to the functioning of the larger system. The functioning of these subsystems can also vary in complexity. The simplest process involves a triad: an input, a process, and an output. More complicated functioning may involve a series of conversion processes, positive and negative feedback mechanisms, and the channels through which the environment can exert its influence. Therefore, viewing the health service delivery industry from an open systems perspective will provide valuable insights into the functioning and structure of the contextual system for HIS design and development. As such, understanding the application of general systems theory on health services delivery systems can better define the role and function of the HIS to be designed and developed.

Another major application of general systems theory in the context of HIS planning and design is the development and use of computerized models to obtain valuable insights into the behavior of complex, real-world systems. Since models are attempts to imitate systems from a particular viewpoint, and health provider organizations are systems that run mostly through rational decision-making mechanisms, the close relationship between computerized modeling and decision making cannot be overemphasized. Therefore, such computerized models are sometimes simply referred to as decision-making or decision support models. This brings attention onto another cornerstone of HIS theoretical and methodological foundation, that is, decision support models.

B. A Taxonomy of Decision Support Models

The term "model" is difficult to define because it has been used to mean many different things, events, or even ideas in varying contexts. As noted previously, a simple conceptualization of a model is that it is a representation of reality. However, from a health data processing and decisional support perspective, a model may be thought of as the context or structure that provides meaning to the data. For example, if people are given a set of data that contains a series of names associated with a string of ten digits formed by a separate grouping of an initial three digits that are bracketed, followed by another three and four digits with a dash inbetween such as John (604) 327-0268, they are likely to infer that these numbers are the phone numbers of the respective individuals. This is because their experience with the phone number model for use in North America is typically one of (XXX) XXX-XXXX structure or schemata, with the bracketed digits referring to area codes.

In terms of computerized models for designing and developing HIS applications, however, it is possible to span a taxonomy of models along a decision complexity continuum. These include, but are not limited to, decision analysis techniques, mathematical programming, computer simulation models, heuristic programming, and nonquantitative (qualitative data) modeling. In general, these models do not or are not intended to replace the decision makers, but they serve to aid these users in rationalizing their decision-making process and justifying their final choices. Hence, the term "decision support models" is used here.

Decision analysis is a popularly used modeling technique to aid decision making under uncertainty and risk. The computation essentially generates the expected outcome of each alternative among a given set of alternatives on the basis of available or unavailable information about the environment and converts the information on uncertainty into risk estimates. Two simple examples are the use of decision tables and decision trees. A specific application of decision analysis to medical decision making, the CDSS, is discussed in Section VI.

In situations where there can be many more alternatives and it is not possible to generate a manageable set of alternatives, mathematical programming takes the approach that reality can be simplified and represented as a set of mathematical equations or relationships. These relationships represent the constraints and limitations on the number of inputs as well as the relationships between inputs and the outcome variables. The commonly employed linear programming technique used in providing optimal solutions to many well-structured, mostly single, goal (criteria) problems is an example of mathematical programming. Other more sophisticated mathematical programming techniques used to solve complex, semistructured, multicriteria problems include nonlinear programming, dynamic programming, 0-1 programming, and, more recently, data envelopment analysis.

If the complexity of the problem situation increases to the extent that the relationships among the variables cannot be conveniently simplified into a series of mathematical relationships, then computer simulations, certainty factors, and stochastic modeling may be used. In a computer simulation, either the real dis-

tribution of variables can be used or a probabilistic distribution may be applied to model the variable distribution to be simulated. In this respect, using simulation to model reality can allow the relationships between variables to be kept either very simple or closer to reality. Moreover, data that have been collected in the past may be used to test and validate the simulation model. When the validity of the simulated model can be demonstrated, further experiments can be constructed to compare various alternatives. These experiments have the advantage that time can be compressed significantly, allowing several months or years to be modeled quickly within a single simulation run that may last only a few minutes. In contrast to mathematical programming, however, only good enough ("satisficing") solutions rather than optimizing solutions can be expected from the use of computer simulation models.

In highly complex situations where the problems are considered somewhat ill-structured and even simulation cannot be applied, heuristics or rules of thumbs are often employed by decision makers to aid problem solving. These heuristics may be incorporated into a computer model of the situation, thus, the term heuristic programming. An example of heuristic programming is the use of expert methods such as neural computing (networks). Neural networks are experimental computer designs that purport to build intelligent algorithms. Neural networks operate in a manner modeled after human brains, in particular, the cognitive ability to recognize patterns. Another class of heuristic models is that of genetic algorithms. Genetic algorithms randomly generate initial solutions to specific procedures, which can then be further recombined and mutated at random just as in an evolutionary process to produce offspring solutions that may yield better offspring and parent solutions or new algorithms. For example, a set of generic operators can be used to generate specific procedures for developing routing and scheduling heuristics for a transportation problem. These generic operators can then be stored for generating new algorithms. In this way, new customers can be added, routes can be merged, and the sequence of customers can be modified using different sets of generic operators. A form of visual interactive modeling can then be used to allow the user to see the results and intervene to change the procedure if the results are not as experienced. Other examples of heuristic programming include the use of fuzzy logic, case-based reasoning, and rough-set methods, since these techniques often incorporate experts' heuristics in generating problem solutions.

At the farthest end of the decision complexity spectrum lie the nonquantitative (qualitative analysis) models. As the field of nonquantitative analysis is still very young, there is a need for considerable research to examine the applications of different approaches and the use of computerized models for such analysis.

C. Information Systems Development Methodologies

Methodologies for the design, development, and implementation of HISs or, more simply, systems development methodologies (SDM) refer to the deployment of systematic approaches to HIS planning, analysis, design, development, and implementation. Of these, the three major phases emphasized in most SDM include systems analysis (SA), systems design (SD), and systems implementation (SI).

SA relates to activities involving the review of current information architecture and the organization environment. In contrast, SD relates to activities involving the specification of new information architecture and systems requirements. Finally, SI relates to activities involving the selection and commissioning of new systems architecture and applications.

The early period of HIS development was characterized by the absence of any formalized SDM. Typically, it was the programmer who bore the major responsibility for HIS design and development. Over the years, as user needs became increasingly complex, the need for a formal development process became evident; subsequently, various SDM and methodological approaches were championed to help minimize the problems of uncoordinated HIS development. Each methodology has been based on a philosophical view, which can range from an exclusive focus on the technical aspects to a focus on the humanistic side of developing an HIS. Most prominent among these SDM are the systems development life cycle (SDLC) model, the structured techniques, and prototyping and contemporary models. Given the space limitation and the fact that much of the discussion provided here will be elaborated in other parts of this book, it appears only logical to provide a very brief overview of this important aspect of the HIS field.

First, the classical SDLC model follows a rigid step-by-step process beginning with a feasibility study and continuing with systems investigation, SA, SD, and SI. The final two steps are systems maintenance and systems review and the entire life cycle is then repeated. While this methodology was a significant improvement over ad hoc approaches, it nevertheless failed to

meet user needs and expectations because of the lack of user participation.

Structured methodologies brought an entirely new approach to systems development, that is, greater emphasis on systems analysis and systems design phases. Systems analysis and design technique (SADT) diagrams, for example, depict the data flows between functions, support both data-oriented and process-oriented perspectives, portray the control under which each function operates, and show the mechanisms responsible for implementing the function. In contrast, SA/SD methodology uses transformational analysis and transactional analysis to emphasize the transition between SA and SD processes. Finally, structured systems analysis and design methodology (SSADM) is a methodology that extends the soft system methodology (SSM) and emphasizes the analysis and design stages of the SDLC model.

Both SDLC-based and structured SDM required that users be able to articulate in advance the information requirements in the HIS. However, users are often unable to specify what they want, and even if they do, their wants often may not reflect their real needs. This is evidenced by the many revisions most newly developed systems must undergo before these are finally accepted. The prototyping technique addresses this problem precisely.

Two opposing views of prototyping have emerged. In evolutionary prototyping, the proven SDLC and structured methodologies are essentially incorporated into prototyping merely to fine-tune the development process, to engage greater user participation, and to enhance user acceptance of the final HIS product. In revolutionary prototyping, the designer often uses a creative trial-and-error process to generate an initial prototype rapidly. Fourth-generation languages (4GL) are often applied to produce these prototypes. The users as well as programmers are then encouraged to test, validate, and fine-tune these prototypes. Such iterations are then repeated until a final acceptable HIS product is achieved.

Today, the evolution of SDM has moved from the rigid step-by-step SDLC-based and structured approaches to prototyping to contemporary models. These include computer-aided system engineering (CASE) tools, object-oriented analysis and design (OOAD), and multiview methodologies. CASE tools automate different parts of software or systems development and can assist with any or all aspects of the SA and SD processes. OOAD focuses on the objects that are acted upon in the development process. The methodology is based on the premise that software should be developed out of standard, reusable components wherever possible. Finally, multiview is a nonprescriptive methodology that strives to be flexible in that it allows any preferred methodologies, techniques, and tools to be combined for the greatest benefits within the multiview framework. As shown in Table II, the multiview framework comprises five major steps, including analysis of human activity, analysis of information (information modeling), analysis and design of sociotechnical aspects, design of human–computer interface, and design of technical aspects.

Together, these five stages provide a practical approach to HIS design and development that integrates the human and technical components within the larger context of changing technologies, user demands, and organizational needs.

D. Knowledge Elicitation Methods

Expertise in medicine and in other health care domains (e.g., health management, nursing, and pharmacy) can generally be grouped along a continuum ranging from laypersons to experts. A layperson may

Table II The Multiview Stages

Stage	Description
1. Analysis of human activity	Problems within the organization are identified and potential HIS solutions are suggested
2. Analysis of information	Data flows are mapped to model and analyze the problem(s) of Stage 1
3. Analysis and design of sociotechnical aspects	Balances technical objectives with social objectives, as well as ranks and chooses among these alternatives
4. Design of human–computer interface	Designs the technical interface based on user input to support acceptance of the HIS
5. Design of technical aspects	Formulates the technical specifications of the HIS

Adapted from Tan, J. (2001). *Health management information systems: Methods and practical applications.* Gaithersburg, MD: Aspen Publishing, Inc.

simply be considered as someone who has only common sense or everyday knowledge of the domain, while an expert is someone who has specialized knowledge of the domain. In between these two individuals are beginners, novices, intermediates, and subexperts. Beginners are those with prerequisite knowledge assumed by the domain; novices are those with knowledge equivalent to beginners or laypersons; subexperts are those with generic, but not substantive, in-depth knowledge of the domain; and intermediates are, by default, those whose knowledge of the domain is just above the beginner level but just below that of the subexpert level. In this section, approaches to the knowledge extraction problem encountered in designing and developing ES in health care, intelligent HDSS, and other equally intelligent forms of integrated and emerging HIS technologies and applications are discussed.

A wide range of techniques have evolved from studies in diverse fields and disciplines (e.g., medical cognition, cognitive psychology, artificial intelligence, organizational science, computer science, and linguistic) in terms of eliciting knowledge from humans for the purpose of incorporating such expertise into specialized computer programs. Among the more established techniques discussed in the literature are interviews, computer-based interactive techniques, methods involving rating and sorting tasks, protocol analysis, and, more recently, cognitive task analysis (CTA). Owing to limited space, each of these techniques and their implications for HIS design and development will be discussed briefly.

Apparently, interviews, structured or unstructured, are direct means of acquiring knowledge from experts. In structured interviews, experts or nonexperts may be probed, based on a structured protocol, to describe in sequence how specific cases are normally dealt with when they are performing a certain task, particularly under difficult or complicated situations. In unstructured interviews, the same experts or nonexperts may be asked similar type and related questions in no particular order, depending on answers previously provided. It is also possible for the interviewer to probe for further clarification or to lead the expert or nonexpert (i.e., the study subject) to talk on particular aspects of the problems if the interviewer feels that previous answers were inadequate. The disadvantage of using interviews is that it is difficult to expect anyone to be able to articulate precisely the hidden knowledge that is to be extracted because they say what they think they do in performing a certain task instead of what they actually do.

Computer-based interactive approaches to knowledge extraction involve having the study subjects use interactive tools or computer programs known as knowledge-based editors to assist them in directly generating computer-assisted knowledge acquisition. OPAL is an example of a graphical knowledge acquisition system for a cancer therapy management program, whereas INTERMED is a collaborative project which uses experimental tools for extracting knowledge of medical practice guidelines based on experts' interpretations of written guidelines. Another application of computer-based interactive approach is the use of software designed to analyze case data, thereby automatically inducing the inference rules. In this case, the interactive approach used is essentially an indirect means of knowledge acquisition, that is, a method for which inferences are made about the nature of the expert knowledge from computer analysis of case data.

Psychological research in judgment and personality studies has contributed to the elicitation of knowledge via use of rating and sorting tasks. Here the attempt is to create a classification scheme, thereby identifying the domain elements along certain meaningful taxonomies. For instance, experts can be asked to sort concepts printed on cards into meaningful clusters. Similarly, these experts may be asked to rate concepts along a certain continuum or among different categories. As an illustration, experts may be asked to rate different whiplash cases involving rear-ended motor vehicle accidents into "mild," "intermediate," and "severe" categories based on varying reported symptoms. In this way, the hidden knowledge based on the experts' general opinions as well as interpretations of these cases can be elicited.

Protocol analysis or thinking aloud may be considered a critical direct approach for knowledge elicitation. This technique has received considerable attention in cognitive psychology studies. The question has been whether experts are better able to articulate the knowledge they possess in thinking aloud when asked to solve a problem than less than experts and what are the notable differences between the thinking strategies of experts vs. less than experts with regards to solving the same problem in specialized domains. One application of this method in the field of medical cognition is to record the interactions of experienced physicians (or residents) with patients in terms of diagnosing the patients' problem. The analysis of these differing interactions would provide researchers with insights into the different thinking strategies of residents vs. expert physicians when faced with similar diagnostic problem cases. The intent is to use the extracted knowledge for programming intelligent computer software that can serve as useful decision

support tools for the residents in similar real-world settings. Just as with the interviews, a major problem with protocol analysis is the ability of the experts (or non-experts) to accurately verbalize what may be hidden in their respective thinking and reasoning processes.

Finally, CTA extends most, if not all, of the above traditional task analysis approaches to include ways of capturing higher level cognitive processes in task performance as well as physical behaviors. CTA refers to a set of methodologies, including the use of structured interviews, video analysis of work situations, and protocol analysis or other approaches, that can be applied separately or jointly to capture the knowledge, skills, and processing strategies of experts vs. less than experts when given complex tasks to solve. CTA generally involves six steps: (1) identifying decision problems to be investigated in the analysis, (2) generating decision tasks or cases, (3) obtaining a record of expert problem solving for the task(s), (4) obtaining a record of the problem solving of novices and intermediates for the same task(s) that was (were) presented to the experts in Step 3, (5) analyzing the performance of experts vs. the less than experts, and (6) recommending the systems requirements, design specifications, and knowledge base contents for HIS development. It is only after many repeated and careful investigations, as well as rigorous analyses of the data derived from the application of these several steps, that proper and valuable insights toward achieving the right mix of information and knowledge elements needed to support decision making and complex problem solving for those who are considered less than experts will be gained. In fact, even experts themselves will find an intelligent HIS developed from the CTA approach useful as the system can serve as a sounding board to situations involving critical life-and-death decisions.

Aside from understanding the evolution of HIS technologies and applications, as well as the ability to critically appraise the foundational stones of HIS theories and methodologies, a third major theme of this rapidly expanding HIS field relates to key issues of HIS administration and impacts.

V. KEY ISSUES OF HEALTH INFORMATION SYSTEM ADMINISTRATION AND IMPACTS

Among the most critical aspects of HIS administration for health organizations are issues related to strategic HIS planning, technology management, and systems implementation issues. The first group of activities is concerned with building a strategic alignment of goals between senior management and HIS personnel and between HIS personnel and other users at lower levels of the organizational hierarchy, whereas the second and third groups of activities have to do with managing the diffusion of HIS innovations, redesigning work practices, and addressing issues of user acceptance or failure arising from the implementation of a HIS.

As for HIS impacts, not only is it important to underscore the impacts of HIS on individuals, groups, and organizations, but it is critical to relay how the implementation of an HIS may ultimately affect the larger context of our health services delivery system and society at large. As we move toward the future expansion of the HIS field, knowledge of HIS impacts will determine the next directions to be taken for HIS technology development and growth.

A. Strategic Health Information System Planning and Management

Strategic HIS planning and management entail many concepts and activities, including data processing and information management concepts, strategic technology planning methods and sound methodologies for growing new and complex HIS applications, applied systems theory and applied decision theory, as well as the search for specific and practical methodologies to create effective HIS design. Most importantly, the mission of the HIS group should reflect the vision of the health organization and should serve as a thematic rationale for integrating departmental HIS goals and objectives.

A major trend in HIS planning is the shifting of responsibilities and power from traditional data processing professionals to end users and, more appropriately, to top management. This trend is justified because of the growing acceptance of the notion that information is a corporate and strategic resource and should be properly managed just like any other organizational assets including land, labor, and capital. Most current approaches to HIS planning concentrate on drawing a blueprint for a total HIS architecture based on past and available HIS technologies rather than emerging ones. Given the rapid shifts in the IT marketplace and the lengthy delays often experienced inbetween HIS strategic planning sessions, it may be wise to revisit the decisions made during these planning sessions before actual HIS implementation. Therefore, more recent approaches call for a shorter time span between HIS strategic

planning sessions and more attention to the changing HIS marketplace. Regardless of the time span inbetween strategic planning sessions, there is always the need to conduct environmental assessment before moving on to the formulation of a strategic HIS plan. One approach to HIS strategic planning is scenario planning. Here, competing multiple futures are first envisioned and strategies are then developed and tested against these possible futures. The IT vision is then set within these possible futures and further reconciled with current reality, that is, the status of HIS development currently existing within the organization. For example, if the IT vision is one in which handheld devices are to be employed by all health organizational workers while current reality involves only the use of connected PCs, then the transition to the new IT environment will not only call for changes in work practices and habits, but new ways of conducting the health organization's businesses may also have to be developed. All in all, no single approach to planning is considered superior; instead, a blending of different approaches is often advisable because the various approaches deal with different aspects of HIS planning.

Successful HIS planning also requires effective HIS management. Briefly, HIS management is the management of the entire HIS capacity of the health organization. Top management must work to ensure that information resources and HIS technology are best adapted to meet the needs of organizational functions and activities. In many cases, the changes needed to develop the necessary organizational restructuring for efficient information processing and effective decision making will not happen without a significant amount of leadership at the executive level, including directives from the chief information officer (CIO). Moreover, the increased complexity of the health care environment and the rapid rate of change in the capabilities of HIS technology will make the process of HIS management increasingly complex and difficult.

Put together, the primary goal of effective HIS administration for a health organization is to ensure that adequate attention is given by top management to ensure a smooth interface between the technological and human elements. Such interfaces occur at two levels, that is, between the individual users and HIS applications or the human–computer interface (HCI) and, in the broader concept, between the organization as a whole and HIS technology or the organization–technology interface (OTI). In this case, OTI focuses on the overall alignment of technology and human resources in pursuit of the organizational goals, while HCI emphasizes the building of HIS applications to augment user capabilities. Many of the problems faced in HIS planning therefore have to do with poor OTI configuration, whereas many of the challenges faced in HIS management have to do with inadequate HCI design.

B. Health Information System Implementation Issues

HIS implementation includes the responsibility for both the integration of HIS technology and the incorporation of accompanying HIS administrative issues into the organization. Not only should HIS technology be integrated with old and new equipment within the organization's existing technical configuration, but HIS administrative procedures must also be integrated into existing administrative procedures. These administrative issues may include policies dealing with privacy, security, and confidentiality of patient records; legal and ethical considerations in data collection, analysis, and distribution; and organizational policies regarding hardware, software, and data standards. Overall, the primary objectives of HIS implementation are reduced operational costs and better quality patient care because, after all, the promise of computerization is to make information handling more efficient by reducing the number of costly errors and unnecessary delays and to project an improved professional image by relieving health care workers of tedious reporting activities to concentrate on patient care.

The challenges of HIS implementation are therefore interwoven with many other factors and organizational challenges, including the integration of quality planning, quality control, and quality improvement processes to evolve a secure, well-managed, and quality-focused HIS environment; the integration of information management technology, organization management technology, and user-interface technology for building an efficient, organization-wide HIS infrastructure; and the integration of data, model, and knowledge elements for designing effective HIS applications. Together, these key challenges typically point toward the need for an integration of environmental, technological, and organizational components for driving and directing the implementation of various HIS technologies and applications successfully within the larger intraorganizational or interorganizational systems context.

Finally, any HIS implementation will bring about some form of organizational change, for example, changes in the organizational structure, changes in

the level of computing competence required of current and future employees, and changes in the information flow processes and reporting behavior of the organization. To ensure that these workers will have the appropriate knowledge, skills, and attitudes critical for addressing concerns arising from these HIS related changes, it is important that health organizations also address staffing issues such as having an aggressive recruitment program for attracting valuable technical staff from competitors, creating opportunities for staff training and development, and employing winning strategies for the retention of knowledgeable and well-trained workers.

C. Health Information System Impacts on Individuals and Organizations

The HIS has had an impact on the individual user, the work group, and the organization in many ways, such as productivity and competitiveness, work habits, span of control, information processing efficiency and networking capability, decision-making effectiveness, as well as intelligence and expertise.

At the individual level, for instance, it is critical to know if the introduction of an HIS will result in better productivity and decision-making effectiveness of the user. For example, it may be argued that a traveling health manager who is equipped with a personal data assistant (PDA) that acts both as a cellular phone with an automated directory and a Net appliance with the capability to access e-mails and Web sites will be able to better perform his or her duties irrespective of his or her whereabouts. At the work group level, the HIS will impact on the ability of group members to share data, coordinate activities, and network effectively. A VPR system for use by multiple care providers, for example, is one that will integrate all of the information provided by the different care providers regarding the patient at any time.

At the organizational level, the HIS will impact changes on many fronts, for example, organizational structure and culture. An HIS will improve managerial productivity, increase span of control, flatten organizational hierarchies, increase power of the decentralized units, change the power and status of individual workers, and open up new possibilities for new organizational units and services. Health organizations will put on a different culture with the HIS diffusion phenomenon, for example, one organization may completely change the way it performs health care services because of automated intelligence, online training capabilities, and virtual networking.

D. Health Information System Impacts on Health Services Delivery System and Society

These same impacts that an HIS have on health organizations may now be extended to the entire health services delivery system and to society at large. For example, the use of the Internet to transfer massive amount of media-rich patient data and the availability of knowledge systems such as robots and automated intelligent systems may induce many legal and ethical questions about privacy, security, and individual and institutional property rights. One question that is frequently asked, for example, is, "Who owns all the different pieces of the stored medical information about a particular patient?" Another question may be, "What information should or should not be kept on-line about an individual and who has access to the information?" Similarly, a follow-up question could be, "How accurate and secure is the information being stored to prevent the information from being misused?"

Other societal impacts of advances in HIS technology include changes in employment levels for health workers and how work may be performed (e.g., telecommuting); changes in the role of the disabled, women, and minority workers in the health workplace; new opportunities for cyber crime and misuse of power; new ways of purchasing health services for consumers; new ways to prevent injuries for work in hazardous environments; new gadgets and automated devices for helping seniors and the disabled; and the construction of healthy "smart" houses and general improvements in health care, lifestyles, and the quality of one's life.

VI. CONCLUSIONS

As the new millennium begins, hospitals and other health services organizations alike will face increasing pressures to change due to changing demographics, changing governments, changing HIS technology marketplace, and changing health services environments.

Notwithstanding these changes, we have seen how a young discipline such as the HIS field can grow and expand quickly to affect every aspect of daily living, in particular, health and the health care system. In bringing this chapter to a close, the example of building a CDSS will show how the different major facets of HIS

technology and application, HIS theory and methodology, and HIS administration and impact can be combined to achieve an effective CDSS design.

First, the CDSS illustrated here is an HIS technology and application that most clinicians will be able to use to aid their diagnostic reasoning. Essentially, this CDSS mimics the hypothetico-deductive approach used by most physicians in deriving clinical decisions. Essentially, it is an iterative process of data collection, interpretation, and hypothesis generation and refinement until a satisfied level of certainty is reached. Within a single iteration, hypotheses are often refined and these drive the next set of questions to be asked (or the next set of examinations to be conducted). Thus, the basic functions and concept of the CDSS technology and application is no different from any of the many evolving forms of HIS technologies and applications that have been discussed previously.

Second, for most of these clinical decisions, sensitivity and specificity are the two methodological bases for hypothesis generation. Sensitivity is the likelihood that a known symptom is observed in a patient with a given disease or condition, whereas specificity is the unique characteristic of a disease or condition, though possession of such characteristic is not an absolute conclusion for the disease. When specificity has an absolute certainty, the observation becomes pathognomonic; for example, a Pap smear with abnormal cells is a pathognomonic indicator of cervical or uterine cancer. Interpretation of a hypothesis is then made against the prevalence of the disease in the population of interest.

For the CDSS to "think" like clinicians, the systems must be taught the different likelihood (i.e., sensitivity and specificity) of specific symptoms to a disease. Such likelihood, better expressed in terms of probability, is a measurement derived from health statistics. The cutoff point in which a patient is diagnosed with a particular disease is the result of continuous collection and analysis of medical data. For example, a 50-year-old female patient is diagnosed with diabetes when her blood glucose reading reaches 5.5. If this cutoff point is reduced, then the sensitivity of the blood glucose test increases while the specificity decreases. Thus, if the cutoff point for diagnosing diabetes for the above clinical group is reduced to 5.4, then more patients will be diagnosed with diabetes. The cutoff point is then a guideline to diagnosing diabetes. However, a patient with a blood glucose reading of 5.6 does not necessarily have diabetes. This is because every clinical test has false positives as every clinical test result is made up of four subgroups: (1) true positives (TP), that is, patients' test results are positive and the patients truly possess the disease; (2) true negatives (TN), that is, patients' test results are negative and the patients do not possess the disease; (3) false positives (FP), that is, patients' test results are positive, but the patients do not possess the disease; and (4) false negatives (FN), that is, patients' test results are negative, but the patients possess the disease. The true-positive rate (TPR) is the sensitivity probability of a test result and measures the likelihood that the patient being suspected of having the disease does have a positive test result. Conversely, the true-negative rate (TNR) is the specificity probability of a test result and measures the likelihood of a nondiseased patient having a negative test result.

The initial hypothesis of a patient having a certain disease or condition is the pretest probability (i.e., prevalence). This judgment may be based on prior experience or on knowledge of the medical literature. When tests and examinations are conducted, the initial hypothesis is verified, thereby yielding the predictive value or resulting in the posttest probability. Predictive value is the probability that a disease is present based on a test result. The posttest probability then becomes the pretest probability of the next hypothesis. This leads to the Bayes' theorem to calculate the posttest probability using the pretest probability and the sensitivity and specificity of the test. Herein lies the foundation of HIS theory and methodology on which the clinical reasoning model is built upon and programmed into the CDSS to support clinical and diagnostic decisions. It is possible, of course, to apply other methodologies such as neural networks and case-based reasoning, but the intent here is merely to illustrate a common and specific HIS methodology based on a well-known decision theory, the Bayes' theorem.

In terms of HIS administration, the CDSS illustrated here must be supported and accepted by the intending clinical users in order to be effective and to have a positive impact on the organization. Further, most hospitals or health organizations have numerous HISs with varying computer platforms and data storage structures. How this CDSS can be integrated or embedded into the larger organizational HIS infrastructure is a key challenge. In other words, rather than having patient data stored redundantly in the CDSS as well as other separate systems such as patient records, it may be possible to network these different systems so that the same data are stored only once and can be shared virtually among the different subsystems. Other related challenges include the need to align the goals of the different subsystems, to coordinate data collection and standardization, to improve computer–user interfaces, and to train users and to

encourage their active participation in managing and harnessing the technology.

Finally, the greatest impact of a system like the CDSS discussed here will be felt when it can be made available to any clinician worldwide and not just to those within the health organization. In this sense, the Internet is an ideal platform for HIS development. When software applications are run through the Internet, the implementation and upgrade costs are reduced. When the access interface is through the Internet, the PC from which the clinician accesses the medical information does not need to have the software applications installed. Notwithstanding, security is always the major issue when applications are accessed through the Internet. Clinical data must not be compromised when HISs development moves in this direction.

SEE ALSO THE FOLLOWING ARTICLES

Data Mining • Decision Support Systems • Decision Theory • Ethical Issues • Human Side of Information, Managing the Systems • Medicine, Artificial Intelligence in • Neural Networks • Service Industry

BIBLIOGRAPHY

Austin, C. J., and Boxerman, S. B. (1998). *Information systems for health services administration,* 5th ed. Chicago, IL: AUPHA Press/Health Administration Press.

Lindberg, D. A. B., *et al.* (1987). *Proceedings of ACM Conference on History of Medical Informatics,* November 5–6, Bethesda, MD; www.acm.org.

Shortliffe, E. H., Perreault, L. E., *et al.,* eds. (2001). *Medical informatics: Computer applications in health care* (Ball, M. J., and Hannah, K. J., Series Eds.). New York, New York: Springer-Verlag.

Tan, J. (2001). *Health management information systems: Methods and practical applications,* 2nd ed. Gaithersburg, MD: Aspen Publishing, Inc.

Tan, J., and Sheps, S., (eds.) (1998). *Health decision support systems.* Gaithersburg, MD: Aspen Publishing, Inc.

vanBemmel, J. H., and Musen, M. A., eds. (1997). *Handbook of medical informatics.* Heidelberg, Germany: Springer-Verlag.

Human Resource Information Systems

Michael Bedell
California State University, Bakersfield

GLOSSARY

client-server A configuration of two computers where one computer (the client) relies on the other computer (the server) to provide instructions and store data that may be needed for processing.

database management system A software product that stores information in an organized, easy to retrieve format.

enterprise resource planning (ERP) system An integrated suite of software applications designed to manage the "enterprise-wide" resource needs of an organization. Resources managed may include production scheduling, financial management, logistics, and human resources.

event rules A set of instructions in an information system that enables the "intelligent agent" within the computer software to make decisions and to process information automatically.

human resource information system (HRIS) Also known as a human resource management system (HRMS). The human resources application in an ERP suite.

intelligent agent A software module that is given instructions, in the form of event rules, about how to process a particular piece of information.

process mapping Process by which an organization's operations are examined, charted, and reengineered.

prototyping Process by which a version of the information system is configured to allow for testing prior to being put into production.

reengineering Process when an organization's mapped-out processes are examined for waste and inefficiency. The process is then rebuilt without the wasted or inefficient steps.

I. INTRODUCTION

The Human Resource Information System (HRIS) is a software application designed to support the human resources (HR) function within an organization. The HRIS stores employee, applicant, and other "people-related" data so that HR professionals can make accurate and timely decisions. As the competitive environment changes, the HR function will increasingly rely on the HRIS to help meet new challenges and opportunities.

"The competitive reality is that organizations will have to develop capabilities that will better serve their customers while differentiating the organization from its competitors" (Ulrich, 1997, p. 1). The maintenance and development of human capital is necessary to sustain an organization's competitive position. Quite simply, the organization needs to have a human capital mix (e.g., knowledge, skills, and abilities) that will facilitate goal achievement year after year. Organizations that fail to take a proactive approach to developing and maintaining human capital will be less competitive.

The management and development of human capital is but one challenge presented by an increasingly complex marketplace. Other challenges to an

organization may include global competition growth, capability awareness and development, change management, technology, and the employee life cycle (Ulrich, 1997, p. 2). Ulrich (1997) suggests that the HR function may be ideally positioned to help an organization manage these challenges.

Many of the other service providers/business partners/resource managers (e.g., finance, accounting, and logistics) identify and meet customer needs through the management and analysis of data. In a similar fashion, many HR functions have taken the first step toward becoming service providers by adopting information systems that manage the data necessary to meet customer needs. These information systems have not always been available. HRISs have evolved over the last decade into a real-time data storage and retrieval tool that enables the HR professional to meet their customers' needs quickly.

To understand the modern HRIS, this article discusses the typical roles of the departments within the HR function. This will develop a common framework of HR activities from which information system requirements can be developed. Also, the evolution of HR from a purely administrative function into a function with strategic value provides insight into the development of HRISs. Then, the components of an HRIS and a rough sketch of how an HRIS is implemented are illustrated. This article concludes with a discussion of how an HRIS supports HR.

II. OVERVIEW OF HUMAN RESOURCES

The HR function (once known as the personnel department) has often been perceived as a business function without much more purpose than to perform routine administrative tasks. Most of the administrative tasks that the HR function engaged in were largely preventative in nature. Typical administrative tasks might include managing processes to avoid legal challenges (e.g., discrimination in hiring and unfair terminations), managing benefits to minimize costs, and monitoring employee behavior to minimize accidents and absenteeism.

The nature of HR activities defined the data requirements of the HR function. While the legacy HR information systems (pre-1990s) were very successful at storing the data necessary to fulfill administrative HR tasks, rarely could the data stored in these systems be utilized for other decisions or tasks. Because the data was often perceived to be company information that would save it from a legal challenge, the following limitations often existed: (1) the type of information stored was very specific, (2) the quantity of information about each individual was limited to what would be necessary in a lawsuit or to run a benefits program, and (3) accessibility of the information was restricted to a few specialists. Uses for this information beyond what the system was designed for would require a specialist to develop a custom report, if possible. These limitations often made activities, such as succession planning, impossible to perform without a substantial investment in a separate database or paper-based system. For example, many of the legacy systems tracked when an employee started with the organization and the employee's current position, exactly what was required for legal purposes. Information that would regularly be stored about that employee for promotions or succession planning (e.g., competencies developed and job experience) would have to be stored in the employee's paper file or in a separate database system.

As HR issues became more complex, many organizations began to separate their HR function into departments that could specialize in specific HR activities (e.g., legal, hiring, training, and compensation). The departments evolved from the primary responsibilities of the HR function, such as recruiting, compensation and benefits, training and development, and organizational development/planning. As each department developed, so did the information requirements for HR decision making. In order to better understand these information requirements, a brief tour of the HR function is provided.

III. THE HUMAN RESOURCES FUNCTION—A BRIEF TOUR

In order to better understand the HR function and HRIS needs, a brief tour of the contemporary HR function is provided here. The discussion visits recruiting, compensation, training and development, and organizational development. While by no means intended to be a comprehensive list of duties, each subsection provides illustrations of the most common responsibilities of each department. Examples of those responsibilities and samples of each department's information requirements are also discussed. In addition, each section of the tour will take a brief look at the legal issues facing HR departments and what information is required from an information system to successfully manage the legal environment.

Please note that many of these departments have similar information requirements. For example, each department could require the same job analysis in-

formation as the foundation for decision making. Job analysis could be performed in any department; however, it is most often the realm of one department to complete the formal job analysis and then to share the relevant information. In this example, the task of job analysis has been placed in the compensation and benefits area.

A. Recruiting

1. Responsibilities of Recruiting

Recruiting is perhaps the first set of responsibilities that come to mind with regard to the HR function. The recruiting department is responsible for developing an applicant pool; selecting new hires through the use of interviews, assessment, and testing; training and scheduling interviewers; extending offers; and performing new employee orientation. Recruiting often assists with the staffing planning process which identifies employees to develop for future promotions as key positions become available within the organization. Recruiting is also involved in the development and validation of selection methods. Selection methods must be statistically validated to ensure that selection methods are job related and that the best employees are being chosen and to verify that discrimination is not occurring.

To fill a position with an external or new hire, the recruiting department places an ad for an open position, receives and tracks applicant resumes, schedules interviews *and* interviewers, schedules applicant testing, performs background checks, and extends offers. To fill a position with an internal candidate (i.e., lateral transfer or promotion), the recruiting department posts the open position, receives notification of applicant interest, schedules interviews and interviewers, examines past work history, and extends offers.

2. Information Requirements of Recruiting

Recruiting must be able to store and easily retrieve information. From the standpoint of the job that is being filled, information about the job, such as the job description, job specification, and compensation range, is required. The job description provides a comprehensive list of the task requirements for the position. The job specification contains a list of the knowledge, skills, and abilities that an employee should have to maximize the probability that they will be successful in this position. The compensation range connects the job to a competitive pay range which is used when extending a job offer and often provides the organization with a competitive advantage for attracting and retaining the best people.

Applicant information must also be easy to store, retrieve, sort, and compare. Ideally, recruiting departments need to have a comprehensive data storage and retrieval system that can make manageable the process of screening and tracking hundreds of applicants. Applicant information may include a resume, reference letters, interview notes, test scores, background checks, and offer/acceptance information. Easy access to this information will enable the recruiting function to easily compare applicants to ensure that the person who best fits the position is hired, promoted, or transferred.

Third, the legal environment demands that the recruiting department also track applicants on the basis of individual differences (e.g., race, gender, national origin, and disability status) so that annual Equal Employment Opportunity (EEO) reports may be filed. Legal precedent has also established that the organization should track hiring rates and know the job relatedness and validity of selection methods to ensure that the best candidates are hired and to protect from lawsuits.

B. Compensation

1. Responsibilities of Compensation

The compensation and benefits department is responsible for ensuring that all employees are paid for time worked and that benefits programs are implemented correctly. Changes to employee jobs and personal lives must be tracked to ensure that employees receive the salary and benefits due to them. The compensation and benefits department also conducts periodic job reviews, in the form of job analysis and evaluations, to identify the compensable factors of a job. Compensable factors are usually defined in terms of the knowledge, skills, abilities, and other factors that are required for job success (job description information is often shared with other departments). The compensation and benefits department also gathers salary information from salary surveys and other sources to ensure that the wages or salary paid to a particular job is equitable and competitive with other organizations in the industry.

For example, when a new position is created, the compensation and benefits department performs a job analysis to determine what tasks are assigned to the new position. From this task information, the

knowledge, skills, and abilities required for an employee to be successful can be identified. Then salary survey information for similar positions within the organization and across the industry is collected to develop a competitive salary range (high, low, and midpoint) for each position. This information is usually shared with the recruiting department so that potential employees can be interviewed and competitive offers can be extended.

2. Information Requirements of Compensation

The compensation department requires access to a substantial amount of information to make timely and accurate decisions. Job description and specification information in the form of competencies must be accessible to support decision making. Wage rates, number of hours worked, base salary, and employee benefits deduction information are all required for accurate payroll processing (do not forget routing and account numbers for direct deposit). Salary ranges, salary survey information, information about employee seniority, tenure with the organization, employee history, and performance appraisal information are all required whenever raises or promotions are planned. To ensure that the benefit package that the employee selected is delivered in a timely and accurate fashion, data about the employee's benefit choices and personal information are required (i.e., which health plan, dependent information, address, etc.). The compensation and benefits department is also required to track information for legal reporting purposes (raises, promotions, initial offers, benefits problems, etc.).

C. Training and Development

1. Responsibilities of Training and Development

As noted in the opening of this article, organizations that have highly trained, flexible workforces are able to react faster to changes in the marketplace and to outperform their competition. The highly capable and flexible workforce is the result of successful training and development programs. The first purpose of the training and development department is to identify the training needed to ensure that the workforce has the knowledge, skills, and abilities (KSAs) necessary to fulfill organizational objectives in the form of daily operations. For example, a new employee may go through a training program to learn how to complete the tasks that are assigned to him or her. At the conclusion of each training program, outcomes data, in the form of tests, attitude measurement, and behavior, is collected and reviewed to help the training and development department to determine the effectiveness of each training program. Employee performance appraisal data can be used to determine areas for employee improvement and to develop training.

The second purpose of the training and development department is to identify, offer, and coordinate employee development programs. The purpose of the development programs is to increase the collective KSA and other factors of the workforce. The increase in KSAs is critical to the organization's long-term competitive position. Regardless of what business the organization decides should be part of its business portfolio, the human capital within the organization will enable the organization to be successful in that business. Since it is impossible to predict exactly what market forces and what strategic decisions will be made, excellent development programs tend to focus on continuously improving the human capital within the organization along a path that would enable the organization to meet strategic needs. For example, an employee may be encouraged to enroll in a college statistics course to develop statistical quality control skills that may be required in future work roles. Perhaps a 5-year goal of the organization is to expand overseas. This may lead to development programs that teach language and culture for the most likely host countries.

2. Information Requirements of Training and Development

The training and development department draws on information developed by several other HR departments to develop accurate and effective training. With regard to training, the goal is to provide training classes that will ensure that employees have the skills necessary to complete their job duties. This training must be provided in a timely fashion and convenient location for each employee. To be successful at this task, information about employee job requirements, in the form of job description data, is used to develop appropriate training. Once appropriate classes have been developed, information about instructor availability, equipment needed, facility availability, acceptable vendors, and a class roster facilitate scheduling. At the conclusion of training, information about employee attitudes toward the training, learning, and transfer of training back to the workplace is collected

to measure training effectiveness. In addition, competencies gained by the employee are stored in the employee data set for future use by recruiting and compensation.

To fulfill the developmental needs of the organization, the training and development department has to be able to make sure that the developmental activities match the organization's longer term strategic objectives. Specific data needs focus on tracking employee KSA development in terms of courses taken, roles specific employees have held, and other developmental activities. Also, development program vendors, employee enrollment/completion, and tuition reimbursement information need to be tracked so as to manage the process. For example, if a number of students are attending the local university for a language course and are unable to converse in that language as promised, this developmental activity may not be worthwhile.

D. Organizational Development

1. Responsibilities of Organizational Development

The organizational development department is responsible for developing succession plans, performing periodic headcount, budgeting, and defining the organization's structure. Some organizational development groups also assist with developing training programs, selection devices, and performing job analyses. Succession planning is a critical activity for the organization's long-term strategic objectives. The purpose of succession planning is to plan for both anticipated and unexpected vacancies by identifying potential candidates for key positions. Information about the succession plan is usually shared with other HR departments so that the plan can be implemented. For example, information would be shared with the training and development department so that the appropriate KSAs could be developed. When a vacancy occurs, an assessment of the individuals in the "succession queue" for that position would be performed and one of those individuals would be promoted. This process enables the organization to develop outstanding management talent and to ensure that the organization always has effective leaders.

Headcount and budgeting focus on managing the number of employees to ensure that the functional areas have enough people to complete their assigned tasks while also managing salary constraints. An organization might decide to reengineer operations in order to focus on smoother service delivery. The organizational development department assists in the development of the new organizational structure, implements that structure, and then determines how successful the change has been.

2. Information Requirements for Organizational Development

The organizational development department may use data developed by other departments to complete its responsibilities. For example, the process of succession planning requires information from compensation, recruiting, and training and development. Succession planning requires the following information about the job being examined: job analysis, job specification, incumbent employee information, salary information, and a list of potential replacements. Information about potential replacements would include job roles held, training and development activities, prior work experience (outside of the organization), performance appraisal data, and a list of acquired competencies (in terms of KSAs). Data about the position being planned for can be matched with potential replacements to determine which candidate is most likely to be successful if promoted.

The task of tracking the organization's structure, headcount, and budgets also requires information from several sources. This information includes accurate counts of employed individuals, an accurate picture of the number of positions available, up-to-date information about the progress of ongoing recruiting activities, aggregate compensation data, individual status in terms of full time vs. part time, family and medical leave (FMLA) status, and anticipated employee separations. Another task that organizational development might be involved in is the validation of a new selection method. To validate a selection method, performance data and the selection method data are statistically regressed upon one another to determine if the selection tool predicts as anticipated. This information is also examined for patterns of discrimination.

E. Information Sharing across Human Resources

As the discussion of the HR function demonstrates, information accessibility and information sharing are important components of an HRIS. For example, upon completion of a job analysis the data is used by the compensation department to develop an appropriate

salary range, by the training and development department to develop a training plan for new hires in the job, and by the recruiting department to select an individual who will provide the "best fit" for the organization. Each department requires access to the same data as a foundation for many of its individual tasks.

The continually increasing complexity of the HR legal environment has caused an increased need for access to more sophisticated data for periodic EEO reporting. Much of this reporting is done in the form of statistics about hiring practices, turnover, training, and compensation. As each department uses the HRIS, data about their activities is entered which can be easily accessed by the legal counsel who completes the report.

From another perspective, the information requirements presented above demonstrate that HR uses data for two types of decisions: administrative and strategic. Administrative decisions are considered to be routine and use employee and position information to keep the organization functioning. Employee information might include address changes, benefit elections, and current position. Position information might include salary range, reporting relationship, and job specification.

As noted previously, a special administrative need for information is the legal requirement. The nature of labor legislation requires periodic reporting of labor statistics and practices. Thus, the HRIS has to have the ability to aggregate statistics about labor practices, such as selection.

Strategic decisions also use employee and position information, but instead the focus is on the long-term direction of the organization. Employee information of interest includes KSAs developed from positions held in the current organization and from prior work experiences. This information is queried in both individual and aggregate forms. Position information might include future human capital requirements and succession planning.

IV. AN OVERVIEW OF THE HUMAN RESOURCE INFORMATION SYSTEM

The information systems that have historically been used by HR functions were designed as administrative compensation and benefits systems. These systems were designed to track employee benefit choices and related costs. Some of these systems also handled the data entry for payroll since a portion of the benefit cost would be deducted from the employee's paycheck. These systems were mainframe based and came in nonrelational database form, flat files, spreadsheets, or proprietary databases. Access was usually through simple "dumb" terminals or terminal emulators. Major limitations of these systems were that (1) they had limited reporting capabilities; (2) they only tracked employee compensation and benefits history; (3) the database could not easily be modified to track additional information; (4) the database was limited to the information within and could not be integrated with other data sources; (5) they often did not have a uniform interface, thus requiring specialized knowledge just to access the data; and (6) information retrieval could only be performed by employees with database expertise.

The biggest liability of these systems was their inability to keep pace with the increasing need for information within the HR function. As HR evolved from a purely administrative role and took on a competitive and strategic role, additional HR information requirements, such as competency management, developed. The inability to modify the database compounded the information requirement issue and lead organizations to develop supplemental databases to store the information that was needed to be competitive. The problem with using multiple databases was, at a minimum, twofold. First, there was the obvious information consistency issue. Second, electronic methods of integrating data were nonexistent. Data integration was completed by querying each database separately and then combining the data by hand.

The modern HRIS differs from older systems in that these systems were designed to meet the needs of the entire HR function and not just the compensation and benefits department. New technology in database systems enabled information generated by the HR function about each employee, applicant, or position to be tracked and, more importantly, integrated with other data for decision-making purposes.

Take, for example, the HR generalist trying to determine which employee to transfer into a marketing job. The generalist will use the performance appraisal table, the training and development table, and the employee interest data table to develop a short list of employees that are eligible for the position. Rather than performing a query on each table and then manually combining the data to determine which employee to transfer, the generalist can use key employee data (e.g., employee number) to link tables and query all of the relevant information at once.

When these systems are integrated with Internet/intranet technology, HR data becomes available to HR staff anywhere in the world. As HR information

needs grow, the modern HRIS is easily modified and more flexible than previous generations of HRISs.

V. MODERN HUMAN RESOURCES INFORMATION SYSTEMS

A. Capabilities of the Modern Human Resource Information System

The modern HRIS has been designed to assist the HR professional in all facets of his or her job. Whether performing routine administrative tasks or making fairly high-level decisions governing the future direction of an organization, the HRIS is designed in an integrated fashion to provide the HR professional with easy access to *all* the information needed to support decision making. The software may also be "intelligent" enough to remind the user of relevant legislative and regulatory issues. System capabilities will be discussed in terms of administrative capabilities, automated features, intelligent features, strategic HR needs, query capabilities, and the ability to work with other HR software products.

Administrative capabilities of any HRIS focus on the ability to make sure that routine processes are completed in a timely fashion and with no errors. For example, the compensation and benefits department tracks hours worked, vacation used, sick days used, and any payroll changes to determine gross payroll outflows. Information about each employee's benefit elections is used to compute the appropriate deductions to each employee's check. Collectively this information forms the raw payroll data. When integrated by an automated payroll feature, the data that has been identified as necessary is transferred to the module that completes the payroll process.

The "automated" nature of the HRIS essentially allows the HR professional to determine what "rules" are going to be followed with regard to both routine and nonroutine transactions. For example, the data that is required to complete the monthly payroll process can be identified in advance. Payroll processing is then scheduled at the beginning of the year and the automated rules gather that data and complete the payroll process as scheduled.

The intelligent capabilities of the modern HRIS enable the information system to automatically react to changes in one part of an employee's record and then update information elsewhere. The integrated nature of the HRIS and the development of various rules tell the HRIS "if X happens, do Y." For example, organizations that function in multiple states may have different health insurance providers and plans for their employees. If an employee moves from Illinois to California, his or her home address zip code would change. When the address change is entered, the HRIS would notice the move to a different provider area and either (a) enroll the employee in the comparable plan in the new coverage area, (b) query a benefit specialist to find out what program to enroll the person in, or (c) automatically send an e-mail or print a letter informing the employee that they need to make a new health program choice. Another intelligent capability enables the recruiting department to identify the position that they need to fill and to ask the HRIS to provide a list of suitable internal and external candidates. The HRIS intelligent agent scans the applicant and/or employee database for individuals that meet the competency needs of the position and returns a list of potential candidates to interview.

To meet the strategic needs of the HR function, information needs to be accessible quickly and accurately in both individual employee and aggregate form. For example, an organization may have a strategic goal to open retail outlets in a foreign country. The HRIS would enable the HR generalist to quickly develop a query or run a report to identify and recommend individuals that are qualified for leadership positions in the new country.

Most modern HRISs have query and reporting capabilities that enable the HR generalist to simply choose from common reports or query capabilities. Once the information is retrieved it can be printed in a raw data format; organized using a delivered reporting tool (e.g., Crystal Reports, Visio); or transferred to an external database package, spreadsheet, or word processor for manipulation and formatting. Many of the modern HRIS share their security protocols with the reporting tools that are packaged with the information system so that confidential information remains secure. Also, the Internet-capable nature of most HRIS enable HR generalists in even the most remote facility to query the main database and receive accurate information immediately.

The final capability required is the ability to interface with other software. The HRIS modules that are part of large enterprise resource planning (ERP) software packages are designed to interface with a growing number of other modules such as financial, distribution, logistics, customer information, e-commerce, e-mail, reporting tools, input tools (e.g., resume scanning), Internet/intranet, and external vendor packages (e.g., payroll and benefits). However, this interface capability is not limited to the ERP class of information

systems. The HRIS programs designed for the small or midsize organization are also able to integrate with other software/vendors. Indeed, it is probably more critical that these smaller HRIS packages be able to interface with outside software as these organizations are more likely to outsource some of their HR needs.

B. How the Modern Human Resource Information System Works

1. Client–Server Architecture

The client–server architecture is a distributed computing system where tasks are split between software on the server computer and the client computer. The client computer requests information from the server, initiating activity or requesting information. This is comparable to a customer who orders materials from a supplier who responds to the request by shipping the requested materials. A strength of this architecture is that distributed computing resources on a network share resources, in this case a single database, among many users. Another strength of this architecture is that additional hardware can be easily added to increase computing power. In the case of the HRIS, the HR professional uses their client computer to request information appropriate to their security clearance from the server. The HRIS server computer houses the database which contains the organization's data.

2. Database

Most modern HRISs use a relational database to store data. Each relational database consists of a set of tables, each of which contains a predefined category of information. Each table contains one or more data categories in columns. Each row in the table corresponds to an individual employee. A typical database would contain an employee table that contains information about each employee, such as address, phone number, position/jobs held, benefits program enrollment, and individual competencies.

The employee table would be linked to a job table which contains specific information about the position/jobs that the employee has had in their history. The job table consists of columns that contain information about competencies required (KSAs), pay grade, current salary, and organizational reporting relationships. Another link from the employee table would be established to the benefit program table. The benefit program table stores the choices each employee has made about benefits coverage so that the organization can fulfill each employee's benefits coverage.

For example, a recruiting specialist may be looking to fill a position internally and identifies a set of employees as potential candidates. If additional information about each employee's skill set is required, the recruiting specialist can have the information system provide a report about what competencies each employee has acquired during their careers.

Another advantage of a relational database is that new data can be added relatively easily; that is, an additional table could be added with modifications to the user interface to reflect the additional table. However, the greatest strength of a relational database is that information from several tables can be combined to generate custom reports and "what-if" scenario analyses and to perform other queries. For example, data from a new employee selection method and performance appraisal data can be combined to determine the utility of that selection method.

3. Hardware and Software

The modern HRIS is a comprehensive yet scalable software solution that is designed to be run on many platforms. To fully take advantage of a modern HRIS, three major hardware resources are required: (1) a database server, (2) an application/file server, and (3) a client computer (see Fig. 1). The database server usually runs some type of database software which provides structure to the data being stored and interfaces with both the application/file server and the client workstation. The application server provides the HRIS interface for the client and engages in some of the processing required. Finally, the client workstation runs a kernel of the application and the user interface. When the user requests data the software kernel on the client workstation requests that data from the database server and simultaneously completes the processes necessary to view and process that data from the application/file server.

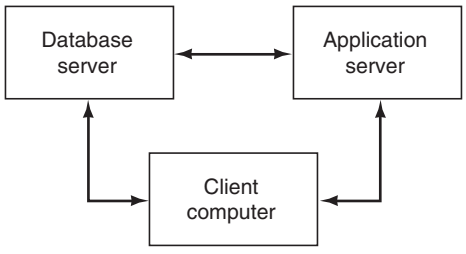


Figure 1 Computing arrangement in an HRIS.

The modern HRIS is a complex combination of software and hardware requirements (Table I). For example, the PeopleSoft HRIS system, which is a major module in the PeopleSoft. ERP system, is designed to run on several different platforms, such as IBM RS6000, Digital (Compaq) Alphas, HP 9000s, Sun Sparc, Intel platforms, and others, as the application and database servers. Possible operating systems include UNIX variants, Windows NT, OS/400, and others. Examples of database software that can be used with PeopleSoft include Oracle, SQL Server, Sybase, DB2/MVS, and others. Other software requirements include telecommunications program/Internet protocol (TCP/IP) and database connectivity software. Options also exist when choosing or configuring reporting software to include Crystal Reports and Microsoft Office products (www.PeopleSoft.com, 1999).

4. Cost

There are a variety of products that are listed as HRISs. Some of these products are not full-blown HRISs since they are designed to help manage only one facet of

Table I Vendors of Human Resource Software Products

Company	Web site	Product purpose
Achievement Tec	http://www.personalitytesting.com	Testing
Anchor Software, Inc.	anchor_fcl@msn.com	HR solutions
Ascentis Software	http://www.ascentis.com	HRIS
Automatic Data Processing, Inc.	http://www.adp.com	Payroll and benefits
Auxillium West	http://www.auxillium.com	Virtual HR department
Baan	http://www.baan.com	ERP class HRIS
Best Software	http://www.bestsoftware.com/	HRIS
Ceridian Employer Services	http://ces.ceridian.com	Comprehensive HR solution
Computer Associates	http://www.cai.com	ERP
The Computer Psychologist	http://www.computerpsychologist.com	Applicant testing
Criterion Incorporated	http://www.criterionhr.com	HR planning software
Cyborg Systems, Inc.		Client–server HRIS
DATAIR Employee Benefit Systems, Inc.	http://www.datair.com	Benefit administration software
Frank Gates Software Systems, Inc.		Electronic HR interface system
G. Neil Companies	http://www.gneil.com	HR tools
GEAC Commercial Systems	http://www.commercial.geac.com	Payroll/HR software
HireSystems, Inc.	http://www.hiresystems.com	Resume data system
HRSoft, LLC	http://www.hrsoft.com	Various enterprise-wide HR solutions
Infinium Software	http://www.s2k.com	HR/payroll solution
JD Edwards	http://www.jdedwards.com	ERP class HRIS
Lawson Software	http://www.lawson.com	HR system that integrates to ERP modules
Oracle	http://www.oracle.com	ERP class HRIS
People-Trak	http://www.people-trak.com	Full-featured HRMS
PeopleSoft	http://www.peoplesoft.com	ERP class HRIS
PeopleSoft Select	http://www.peoplesoft.com	Small/mid-class HRIS
Personic Software, Inc.	http://www.personic.com	Staffing software
Ross systems	http://www.rossinc.com	ERP class HRIS
SAP	http://www.sap.com	ERP class HRIS
SCTC Corp	http://www.sctcorp.com	ERP class HRIS
Spectrum HR Systems Corporation	http://www.spectrumhr.com	HRIS

the HR function. Many of the smaller HRIS products are designed to store information for organizations with 25–500 employees and cost between $300 and $5000. Midrange products are aimed at organizations with 25–10,000 employees with an implementation cost of up to a few hundred thousand dollars. The ERP class of HRIS is targeted at organizations with more than $250 million in revenues, although at least one of these providers has developed a version of their product for the midrange market. The cost of implementing an ERP class of HRIS may cost as much as $10 million.

C. Human Resource Information System Implementation

Simply purchasing a software package will not guarantee that the new system will meet the needs of the organization. The selection and implementation process of any HRIS requires a thorough understanding of the organization's needs. For the large organization that requires an HRIS that can track thousands of employees and applicants, the implementation will be time consuming and hundreds of tasks will have to be managed. For the smaller organization, using an HRIS that is designed to have a turnkey implementation, the process mapping and prototyping phases will be much shorter and less thorough. However, the small organization would be advised to consider process mapping and prototyping efforts, as many of the benefits associated with large-scale HRIS implementations are the result of these efforts.

1. Process Mapping

The process of an HRIS implementation begins with a step that is very similar to reengineering. To get the most out of an HRIS or any ERP system, an organization must commit to completing a thorough analysis of organizational processes known as process mapping. Process mapping is a thorough analysis of how the HR function "does business." While the HR function might be difficult to pin down in terms of tangible outputs, this function has potentially hundreds of processes that are completed on a daily, monthly, or quarterly basis. Other processes are completed on an as-needed basis. For example, if an employee moves and changes his or her address, there is a process by which the information is collected and entered into the database, and benefits and tax information are checked to ensure that the address change does not alter the benefits provider or the local tax rate.

The analysis of how the HR function completes its tasks has to go beyond simply interviewing a few employees or managers. Their perception of how processes are completed may not match what really happens. Good process mapping is often the result of using a variety of process mapping tools. Process mapping tools are very similar to the tools used for reengineering and total quality management efforts (e.g., flowcharts and process diagrams). For the best results, an organization should engage in a thorough process mapping strategy so as to eliminate later customizations.

The process mapping phase is often best completed by assigning a functional area expert to work with an information technology professional and/or an experienced consulting team. The consulting team is important as they are paid to question "sacred cows" and other "untouchable" processes. Many of the initial benefits that come out of any ERP or HRIS implementation come from the process mapping and redesign phase.

2. Prototyping

The purpose of the prototyping phase is to determine how the information system will be configured. Prototyping is best performed by the organization's designated functional experts, HRISs specialists, with involvement/guidance from the consulting team if applicable. The question that is solved during the prototyping phase is how the information requirements of the organization's processes match up with what the plain or "vanilla" version of the information system provides. Each of the organization's processes are examined, beginning with how information will be entered into the system, what information needs to be stored, how information will be stored, the steps that information will follow through the system as it is processed, and how that information will be accessible in the form of reports and queries. In addition, information that needs to be passed between separate modules of the information system or to other software needs to be identified so that interfaces can be activated or built.

While prototyping the data entry steps the contents of the prepackaged look-up tables need to be examined as to whether they meet the organizations needs. Data entry sequences are checked against these tables. Examples of look-up tables are a location table that contains a list of states and their abbreviations. Another table might contain a list of commonly received degrees, while a third would contain the list of EEO reporting categories as defined by Equal Employment

Opportunity Commission (EEOC) regulation and/or federal law. For example, if the organization is a private organization, some entries in the EEO table can be ignored as certain EEO categories only have to be reported on by certain governmental and public entities. The location table might need to be modified as it may initially only reflect in the United States and possessions while the organization actively recruits individuals from Canadian universities.

3. Testing and Daily Operations

After prototyping is completed, the information system is configured and tested. Ideally, testing should be completed before the previous system, either computerized or paper and pencil, is eliminated. Rigorous testing allows data entry methods, processing steps, and output results to be checked for accuracy by comparing the results to previous methods. Also, if something was missed during the prototyping stage, realistic testing should enable the organization to rectify any omissions that may have occurred during prototyping. This will allow daily operations to go forward, hopefully without major flaws, as the changeover to the new information system begins.

4. Customization

Every HRIS can be customized to more closely meet the needs of the HR function. Customization occurs by adding tables to the database, modifying the user interface, modifying reporting capabilities, or any other change that may be desired. Some customization is inevitable as most HR functions will require reports that are different from those delivered by the system. Obviously, the closer the system matches the organization's needs, the more successful the implementation will be.

One drawback of customization is that an organization may be using customization as a "Band-Aid®" for poor processes or to hide poorly reengineered processes. A more significant drawback is the cost that customization adds to the software. Every customization requires an information technology expert to modify and document the software. Documentation is critical as future upgrades will require that each customization be performed again or updated.

Among the user groups of the major software vendors, there is an ongoing debate as to whether customization is wise. The software vendors suggest that the customer leave the system as plain vanilla as much as possible to ease the upgrade process and minimize support issues. Various customers have argued that their implementations would not be successful if they had not performed their customizations. The solution that is most accepted is that customization should be performed if (1) there is a solid business reason for the customization or (2) the customization can be created and maintained for a cost that is less than the benefits received. For example, a customization that was determined to be cost-effective during an implementation moved all the legacy system data to a table. This table was accessed only via query and reporting methods. The customization lessened the time needed for data transfer and enabled the organization to very quickly move forward with a clean slate.

D. How Human Resource Information System Facilitates the Human Resource Process

The modern HRIS facilitates the HR process through information accessibility, information accuracy, sharing of information across boundaries, automation, and employee self-service. The consolidation of information to a centralized relational database eliminates the need to maintain multiple databases and therefore increases information accuracy, accessibility, and consistency. For example, use of a centralized database makes the employee's (or applicant's) entire HR file easily accessible so that the HR professional can fulfill customer needs quickly, whether the HR professional is in the corporate headquarters or in a manufacturing facility 4000 miles away.

The ease of accessibility and increased information accuracy provided by a centralized database also facilitates the strategic planning process. By consolidating employee data to one location, human capital needs can be matched with strategic planning efforts to determine corporate strengths and weaknesses. From this strength and weakness information, the HR function can determine how to better meet the organization's needs.

Within the HR function, departments can use information collected by other areas of HR to better achieve their goals. For example, the training and development department can examine the most recent performance management data to determine what training programs should be offered to meet both strategic and departmental goals. The recruiting function can use strength and weakness data from strategic planning efforts to determine what competencies need to be recruited to achieve the desired human capital mix.

A number of HR processes can also be automated to free the HR professional to spend more time working

with the customer and fulfilling strategic needs. For example, benefit events that require simple actions, such as an employee's dependent having their 18th birthday, can be coded in the form of event rules. The event rules would notify the employee of the pending change and then have the system remove the dependent from the parent's health care coverage on the 18th birthday. In another example, an employee could decide that they cannot afford the time for a training class and cancel their enrollment. The event rules would automatically select an employee from the wait list, enroll the employee, and notify the previously wait-listed employee that they are now enrolled. A final example would have the organization receiving an applicant resume via the organization's Web site in the middle of the night. The event rules would automatically categorize the resume by competency, generate a form letter acknowledging receipt, and have the letter waiting on the laser printer for a signature in the morning.

Another way that HRISs facilitate HR processes is through employee self-service. Employee self-service is a concept that is new to HR. The employees are given limited access to the HR database via corporate intranet or secure Web server. For example, to change an address the employee can pull up the on-line form and make the change. This saves HR from needing address change forms and data entry personnel and lessens the potential for mistakes as there are fewer steps in the process. The customer is satisfied because their HR needs are immediately fulfilled. Other self-service opportunities that have been deployed include training enrollment, changes to personal information such as benefits and taxes, on-line resumes and competency management, internal job postings and application, and posted answers to frequently asked questions.

VI. EXAMPLES FROM COMPANIES WITH SUCCESSFUL HUMAN RESOURCE INFORMATION SYSTEM IMPLEMENTATIONS

There are thousands of companies that have implemented modern HRISs. Of these companies, a majority of these completed PeopleSoft implementations (PeopleSoft is the major player in the HRIS market). Anecdotal evidence from these companies suggests that HR departments are better able to meet employee needs, streamline operating procedures, reduce operating expenses, and increase information accessibility.

Case studies of PeopleSoft implementations (from the PeopleSoft Web site) validate this anecdotal evidence. Sears implemented PeopleSoft and anticipates a "46% reduction in operating expenses because of streamlined operations" (PeopleSoft, 1999). While this 46% may sound too good to be true, note that Sears is "the fifth largest retailer in the U.S., with HR transaction volume approximately equal to 160,000 new hires and 14 million paychecks annually" (PeopleSoft, 1999). BP Oil reports savings of $2.5 million a year from their PeopleSoft implementation. Of this $2.5 million, approximately $2 million of it is from reducing HR function headcount (PeopleSoft, 1999). There are many other examples of empirical evidence that suggest that an HRIS implementation, if done well, can save the organization time, money, and other resources.

Some organizations have even suggested that the implementation of an HRIS (like PeopleSoft) may benefit the organization in ways that are not obvious. For example, PeopleSoft is designed around HR best practices so the implementation of PeopleSoft may force the organization to improve their processes to match the information system. If the HRIS is built around best practices, any change in processes should be for the better. Also, the organization will benefit from a thorough process mapping/reengineering effort.

VII. CONCLUSIONS

The modern HRIS has been designed to meet the needs of the HR function. The HRIS is a complex collection of software, hardware, and database technology that is designed to be flexible and adaptable to the organization's needs. The HRIS provides the HR function with the means to collect, store, organize, and access information. The information system sustains the organization's competitive position through data-driven human capital development and acquisition efforts. The needs of the employee are met as the information system facilitates the accurate delivery of HR services such as payroll, training, and benefits. Increasing automation and self-service capabilities are freeing the HR professional to pursue more strategic activities and may also increase employee retention and satisfaction. Self-service capabilities allow employees to check the balance of their 401k plan or the contributions to a dependent care account. The addition of Internet-based capabilities will increase the consistency and accessibility of HR services. The modern HRIS provides has enabled HR to evolve into a competitive service function within the organization.

SEE ALSO THE FOLLOWING ARTICLES

Corporate Planning • Human Side of Information • Intelligent Agents • Management Information Systems • Operations Management • Outsourcing • Prototyping • Reengineering • Service Industry • Staffing the Information Systems Department

BIBLIOGRAPHY

Best Software. (1999). On-line information at www.bestsoftware.com.

Boyett, J., and Boyett, J. (1999). Human resources most effective practices across best companies. *PeopleSoft white paper series.* Pleasanton, CA: PeopleSoft.

Byars, L., and Rue, L. (2000). *Human resource management* 6th ed. New York: Irwin-McGraw-Hill.

Cameron, B. (1998). Is ERP in trouble? *Computerworld online.*

Darter, K. (1999). Personal communication regarding a PeopleSoft implementation.

ERP Supersite. (1999). On-line resource center.

Floyd, B. (1999). Expanding HR Education Through PeopleSoft: Understanding the Challenge. Unpublished article.

Knorr, E. (1999). ERP's rough waters. *Upside today: The tech insider.* Electronic article.

Noe, R., Hollenbeck, J., Gerhart, B., and Wright, P. (2000). *Human resource management: Gaining a competitive advantage.* New York: Irwin-McGraw-Hill.

PeopleSoft. (1999). On-line information at www.PeopleSoft.com

People-Trak. (1999). On-line information at www.people-trak.com.

SAP. (1999). On-line information at www.SAP.com.

Spectrum, H. R. (1999). On-line information at www.spectrumhr.com.

Spencer, L. M., Jr. (1995). *Reengineering human resources.* New York: Wiley.

Ulrich, D. (1997). *Human resource champions.* Cambridge, MA: Harvard Univ. Press.

Ulrich, D., Losey, M., and Lake, G. (1997). *Tomorrow's HR management.* New York: Wiley.

Human Side of Information, Managing the Systems

Carmen de Pablos
Rey Juan Carlos University, Spain

GLOSSARY

information and communication technologies Elements of hardware, software and communications that allow the electronic transport of data among places.
information systems Group of persons, procedures, and resources that collect, transform, and distribute data in the organization.
managing the change The process of driving the needed organizational fits.
people Persons working in a firm.
systems strategy Main objectives to take into account in relation to the way persons, procedures, and resources work in a firm.

INFORMATION SYSTEMS are today considered important capabilities in the organizations. Managing information systems is a basic business function. This implies the need to develop an information system's strategy, analogous to others. Systems strategy serves as the basis for fundamental business strategy decisions. It helps answer questions; for example, which systems should be used to implement product design concepts, or how technology should be managed in a firm.

I. INTRODUCTION

Information systems strategy is an instrument of business and corporate strategies. From a competitive strategy's point of view, Porter affirms in 1985 that information systems can be used defensively to sustain achieved advantage in established lines of business or to develop new products and markets.

In 1992, Stalk and colleagues distinguished core competence from a firm's strategy capabilities, "whereas core competence emphasizes technological and production expertise at specific points along the value chain, capabilities are more broadly based, encompassing the entire value chain." They define a capability as "a set of business processes strategically understood. . . .the key is to connect them to real customer needs." Thus, technological competences and capabilities are complementary concepts, and Porter's value chain analysis provides a useful tool for examining their interrelationships.

Information system's strategy can be discussed in terms of:

1. The deployment of the system in the firm's product-market strategy to position itself in terms of differentiation and delivered cost and to gain technology-based competitive advantage
2. The use of information technology, in the various activities in the firm's value chain
3. The firm's resource commitment to various areas of systems
4. The firm's use of organization design and management techniques to manage the systems, especially since the human side of use

As Clark in 1987, and Henderson and Clark later, in 1990 note, there must be a distinction between design concepts and their physical implementation.

Core design concepts can be implemented in various ways to become components. For instance, establishing an insurance contract from an insurance company could be achieved through using face-to-face interaction or via the Internet. Each of these implementations, in turn, refers to an underlying technological knowledge base. Each of the core concepts of a product thus entails systems choice. In addition to components, a product has also an architecture that determines how its components fit and work together.

System choices require careful assessments of technical as well as market factors and identify an array of targets for technology development. The high costs of investments in systems make the choice and target for systems development an especially outstanding dimension of a firm's strategy.

II. PEOPLE IN INFORMATION SYSTEMS

An information system is a group of people, practices, values, and technologies in a particular local enviroment. In information systems, the emphasis should not be on technology, but on human activities that are served by technology. Technology becomes therefore a facilitator in the information system. People are essential factors in the information systems, this way Meister in 1987, writes in terms of human-machine, systems as "an organization of man or woman and the machines they operate and maintain in order to perform assigned jobs that implement the purpose for which the system was developed."

Nardy and colleagues considered in 1999 that diversity is necessary for the health of the system itself, to permit the system to survive continual and perhaps chaotic change. Information systems should be crowded with diferent kinds of people and ideas and technologies. A diverse information system must be composed of many different resources and materials to achieve some organizational goals.

Information technologies are rapidly changing, and the information systems in which these technologies play a role must take part in the changes. We have to take into account that the social and technical aspects of an environment co-evolve. People's activities and tools should adjust to the changes. This is part of the dymanic balance achieved in systems. An information system as a persistent structure over time acquires its own history and must continually evolve.

When we add new technologies to our own information systems, we sometimes try to work in the absence of essential resources. Often such resources are skilled people, whose presence is necessary to support the effective use of technology. People, as users of tools, are responsible for integrating these resources in such a way that they make sense for us. They act locally in a committed way, but, as Ellul documents in 1964, they choose to respond with initiative based in local main assumptions and values.

When evaluating an information system, why questions become very important, since they allow the exploration of motivations, objectives, and values: why this particular technology seems best, why it fits well with our current practices or why it does not.

The most important idea to remember about why questions is that local knowledge is required both to formulate and respond to questions, and local knowledge is distributed throughout an information system. No single person can know enough to ask all the right questions. A diverse set of perspectives is needed to develop a useful information system. This means that everybody should be encouraged to ask questions, not just those with highly visible technical knowledge or management responsibility.

Diversity should be preserved in the information system. In 1992 Wilson, by referring to the ecosystems, says diversity should be preserved for at least two reasons, First, every species alive today is a natural survivor that may offer as yet unknown benefits for our own health and productivity. Second, and most important, an ecosystem as a whole is threatened by the loss of certain species. If we do not respect diversity, we may unintentionally damage the entire ecosystem.

III. DIVERSITY OF INFORMATION SYSTEMS

Computer users can do many things with the communication capability that information systems give them. They can, for example, send electronic mail messages to one another or forecast market trends just by searching some data from their computers or distant computers.

For example, some electronic commerce (EC) have found ways to take advantage of some of the special atributes of the Internet: the ability to communicate in both directions between buyers and sellers or update their inventory frequently.

The biggest problem here is to attempt to convince people that information systems are primarly a means to bring together different parts, for example, consumers and producers. Today the Internet is a diverse and decentralized place, and there has been an important creative response to its possibilities. Ellul spoke in 1964 of "autonomous technology" as though

it were a personified force that somehow gave birth to itself. At the organizational level we have to count on people who have developed some technical and managerial skills that allow them to develop a firm's information system in a proper way according to its main objectives in the organization.

Nardy and colleagues assumed in 1999 that the characteristics of an information ecology share much with biological ecologies: diversity, locality, systemwide interrelationship, keystone species, and co-evolution. What makes information systems different is the need to apply human values to the development of the practices and technologies within the systems. Tools and practices co-evolve.

Our way into the process of technological change is to adopt a gesture of participation and compromise with technology. Resistance takes place sometimes as a part of the strategy, but using technology according to thoughful values in the firm seems to be the most viable approach for the proper use of an information system.

IV. FROM MANAGING SYSTEMS TO MANAGING NETWORKS

Information systems have progressively been evolving toward business networks with the implementation of new information technologies. A network is broadly defined by Lewis in 2001 as a collection of devices and circuits that provide means for transferring data from one device to another (e.g., from one computer to another, from one phone to another), where such data transfer is the basis for applications such as telephone, e-mail, videoconferencing, manufacturing, inventory, billing, and accounting systems.

An information system's main purpose is to support the informational and operational requirements of the business.

Information systems management is then the practice of first monitoring and controlling an existing information system, planning for system extensions and modifications in order to meet increasing demands on network operations, and incorporating new elements into the network.

First there is a network infrastructure, that it is to say, transmission devices that receive traffic from and forward traffic to, other transmission devices (routers, hubs, switches, access devices for wireless networks, cable modems, and satellite stations).

Second, there is the transmission medium over which traffic flows—copper wire, coaxial cable, fiber-optic cable, telephone lines and airwaves. Third, there are computer systems that reside on the network, desktop computers, workstations, mainframes computers, and laptops. Fourth, there are software applications, that run on the computers, documenting writing applications. Fifth, people who use the information systems that are obviously components in the network.

Finally, there are services supported by software applications. A service is something that arises in virtue of the structure and operation of transmission devices, transmission media, computer systems, and applications. Daily examples of information systems in a network are EC or distance learning.

Integrated systems management means the management of the devices, the data and flows, and the people that interact with it.

There has been an evolution in the management of the systems that is shown in Fig. 1. It can be observed that as information technologies have evolved, more elements have been added to an information systems and because of that, the management of the systems today is more complex.

Lewis mentions the five-layer telecommunications management network model (TMN), as seen in Table I.

The first layer is concerned with the overall management of the telecom carrier business. It covers aspects relating to business processes and strategic business planning. It seeks to capture information to determine whether business objectives and policies are being met.

The service management layer has to do with the management of services provided by a service provider to a customer or to another service provider. Examples of such services include billing, order processing, and trouble ticket handling.

The third layer is concerned with a network with multiple elements. As such it supports network monitoring and remote configuration. In addition, this supports issues such as bandwidth control, performance, and quality of service.

The element manager is concerned with the management of individual network elements, for example, switches, routers, bridges, and transmission facilities.

The element layer refers to bare elements that are to be managed.

A. Roles in Managing Information Systems

In the end a deployed management system must answer the ideas of the executive who situated the requirements in the first place. It is a common phenomenon that system developers lose track of business requirements as they get involved in the engineering intricacies of system development. That should be avoided.

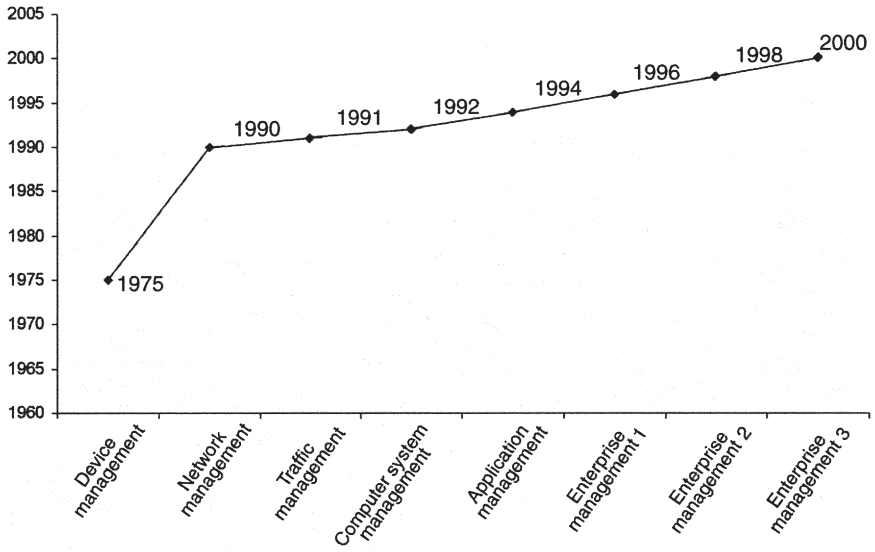


Figure 1 History of data communications and network management. [Data from Lewis, L. (2001). *Network and Systems Series* (Manu Malek, ed.), Dordrecht: Kluwer Academic.]

There are "agents" that focus on methods of monitoring the service parameters. We can see them in the following categories:

- *Network agents*—dedicated to connect nodes in the network infrastructure, bridges, hubs, routers
- *Traffic agents*—focused in the traffic that follows over transmission media in the network infrastructure
- *System agents*—computer systems that reside on the network; they reside in the system, read the system log files, and perform system queries
- *Application agents*—focused on business applications distributed over networks and computers
- *Special purpose agents*—built to monitor parameters that are not covered by any of the above; for example, an EC insurance agent that monitors the reliability and response time of web page retrievals from some geographical domain to a web server farm in a second geographical domain
- *Enterprise agents*—who own a wide view of the network infrastructure, including connection modes, systems, and applications that make up the network

Table I The Five-Layer Telecommunications Management Network Model

Business management level information
Service management level information
Network management level information
Element management level information
Element level information

[From Lewis, L. (2001). *Network and Systems Series* (Manu Malek, ed.), Dordrecht: Kluwer Academic. With permission.]

As Meister indicates in 1987, the human functioning in information systems can be described in terms of perception, information processing, decision making, memory, attention, feedback, and human response processes. There are some tasks related with the interaction of humans with information technologies that must be controlled.

In the interaction with information systems, seen in Table II, there are some subject interests of technical groups of the human factors.

B. Phases in the Systems Management

Integrated management systems means the need to develop some kind of organized structure in order to consider the appropriate ongoing of all the elements that conform to the system all over time. Here there are some examples of these phases:

Phase 1: understand the initial requirements of a new technology from user and business perspectives
Phase 2: develop a conceptual architecture from user and business requirements
Phase 3: transform it into a physical architecture

Table II Interaction of Technical Groups and Human Factors in Information Systems

Communications	All aspects of human to human communication, with an emphasis on multimedia and collaborative communications, information services and technologies and infrastructure technologies in education, medicine, business productivity and personal quality of life
Computer systems	Human factors aspects of (1) interactive computer systems, especially user interface design issues (2) the data-processing environment, including personnel selection, training and procedures and (3) software development
Organizational design	Improving productivity and the quality of life by an integration of psychosocial, cultural and technological factors and with user interface factors (performance, acceptance needs, limitations) in design of jobs, workstations and related management systems
System development	Concerned with research and exchange of information for integrating human factors into the development of systems. Integration of human factors activities into system development processes in order to provide systems that meet user requirements

[Adapted from Lewis, L. (2001). Network and Systems Series (Manu Malck, ec.) Dordrecht: Kluwer Academic. With permission.]

Phase 4: implement the physical architecture
Phase 5: test the implementation
Phase 6: control the implementation and get user feedback
Phase 7: return to phase 1

Although many firms try to follow a structured vision when managing their systems, some find it difficult, even imposible due to the appearance of some of the following problems:

- Lack of top management commitment to the project
- Misunderstanding the requirements
- Lack of adequate involvement
- Failure to manage end-user expectations
- Changing scope/objectives
- Lack of required knowledge/skills in the project personnel
- Introduction of new technology
- Conflict between user departments

The lack of user involvement is one of the main factors that causes an unsuccessful management information system. A related factor is the lack of a clear establishment and understanding of user requirements.

One study in 1995 of systems projects in England examined the opinions of information system managers who had experienced so-called runaway projects (projects that threatened to spiral out of control). When they were asked what they have done in the past to try to gain control of runaway projects, the consensus was as shown in Fig. 2.

When they were asked what they intended to do in the future to improve projects, their general consensus was as seen in Fig. 3.

There are some useful software packages to monitor and control information systems, and they have a special emphasis over business processes, services, applications, transmission devices, computer systems, and traffic in small to very large multivendor networked domains.

Most of this develops some specific actions in the following points:

- Discover the components of the networked domain
- Represent the relations among components in a centralized or distributed data repository
- Collect performance data over time
- Perform traditional fault, configuration, accounting, performance, and security management
- Integrate with complementary management systems
- Manage the networked domain with a web browser

C. Developing Positive Attitudes toward the Information System

Companies, by seeking strategic advantage through significant technological innovation, need to recognize how this process operates and design specific organization and management practices to motivate people and guide them in the use of information systems.

Companies must develop internal management systems to plan and control time spaces and their corporate cultures must accept these timetables as given.

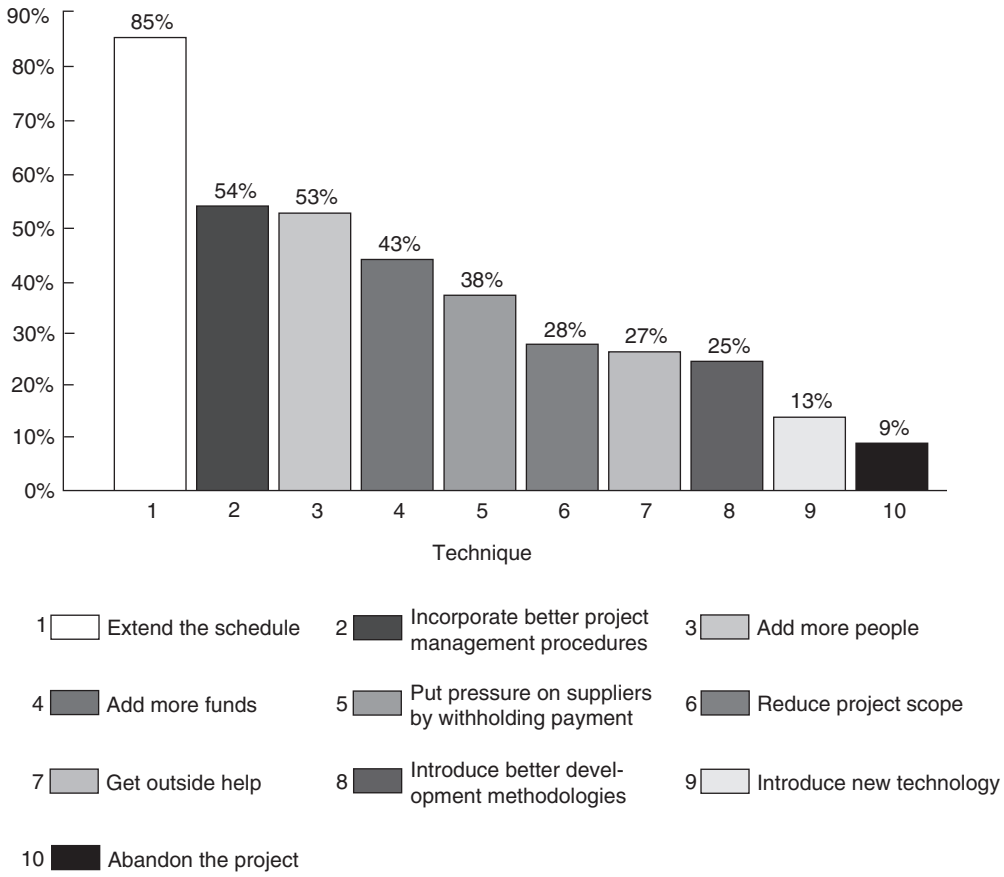


Figure 2 What managers try to do to control runaway projects. [Data from Lewis, L. (2001). *Network and Systems Series* (Manu Malek, ed.), Dordrecht: Kluwer Academic. With permission.]

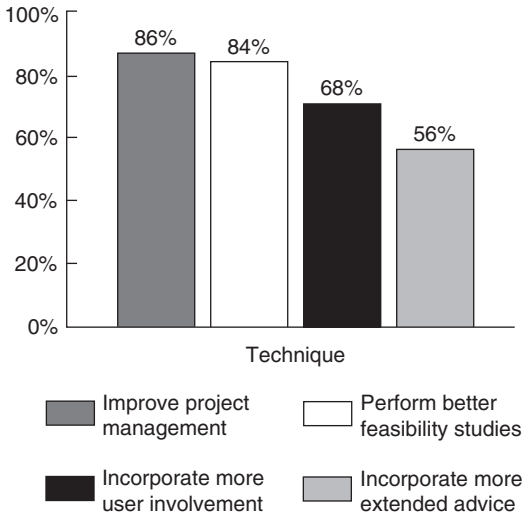


Figure 3 What managers try to do to improve projects. [From Lewis, L. (2001). *Network and Systems Series* (Manu Malek, ed.), Dordrecht: Kluwer Academic. With permission.]

For this it is necessary to integrate research, technology production, marketing, innovation, and people in activities searching for corporative goals. As Quinn indicates in 1986, the most successful enterprises seem to:

- Establish broad but challenging overall company objectives in light of long-term sociological, economic, and technological developments
- Determine what specific unique strategy the overall company and each division will be using in effecting these objectives
- Rank and balance research programs and anticipate threats and opportunities
- Develop clear mechanisms and supporting motivational environment for using information systems

To adopt maximum motivational and information benefits, these companies consciously structure pur-

poses between competing approaches only after they reach advanced prototype stages.

To further enhance motivation, companies must also emphasize small teams working in a relatively independent environment.

V. FACING CHANGES

Because of the development of new information technology, firms need to change the way their information systems work. Sometimes evolution in systems is not slow and gradual, but abrupt. But information technology can never substitute an essential human function. Nardy and colleagues in 1999 use the example of librarians to illustrate this idea: the librarian understands that there may be a difference between what the client literally says and what the client actually wants. If clients' requests were always carried out exactly as presented, they would often yield poor results.

The characteristics they find most important in the library information systems are the diversity and complementarity of technological and human resources, and the presence of librarians as a keystone species giving unique shape and strengh to the library information systems.

In 1990, Munter finds advantages and disadvantages in the use of different resources in some information systems as shown in Tables III and IV.

The key to the continued success rates these companies search is to be found in their careful management of structure. This includes a reexamination of existing formal structures, in light of changing technology and markets needs. Structure is to be managed and adapted to changed conditions.This, in turn, requires a willingness to invest valuable time in structure too.

A. The Process of Learning

The implementation of a new information system requires a process of learning. In 1990, Leonard-Barton distinguishes among two different kinds of learning, (1) vicarious learning, from the experience of others, and (2) experimental learning, from one's own experimentation with the innovation. Both should occur in the organization as well in the distributed places the company works with.

If the new technology is going to be dispersed in different sites, the most important criterion for selection of the pilot sites is representativeness rather than receptivity. If the first site is atypical of the later ones in some important way, less technical and organizational learning can be transferred from this early user to later ones, and potentially costly problems with the new technology may not surface soon enough to be efficiently solved.

Peers who have already implemented a new technology have what Rogers, in 1982, called "safety credibility."

In 1990, Leonard-Barton conceptualizes managerial strategies for successful implementation under conditions of high organizational change, as shown in Table V.

Even well-rewarded user-experts can be expected to eventually tire of the role, of course. The original user-experts must train others to take their places. This strategy can ensure a hierarchy of expertise and a more equitable distribution of people's duties over time.

The implementation of new systems is a process of learning. Participants even at low-cost, successful sites felt this point was ill-understood. Some systems are more simple than others, but any system being used purposely to alter the way an organization operates likely involves a great deal of stressful change. Managers who understand that they are managing organization change, not just technical change, are well positioned to direct the learning process.

Table III Advantages and Disadvantages of Groupware Meetings

Advantages	Disadvantages
Working with dispersed groups, meeting different times, different places	Lack richest nonverbal cues of body, voice, proximity and touch simultaneously
For speeding up meeting follow-up activities because decisions and action items may be recorded electronically	Not as effective when establishing new group rapport and relationships that are crucial
	May be more difficult to use and more likely to crash than low-tech equipment

[From Munter, M. (1999). *Managing Complexity in the High Technology* (Glinow, V., ed.), pp. 19–26; 19–33. Oxford: Oxford University Press. With permission.]

Table IV Advantages and Disadvantages of E-mail Meetings

Advantages	Disadvantages
Increase participation because it overcomes dominance by overly vocal and quick to speak participants	Decrease attention to the person receiving the message and to social context and regulation
Increase communication across hierarchical frontiers	Be inappropriately informal
Decrease writing inhibitions, using more simple sentences, brief paragraphs	Consist of quick and dirty messages, with grammar errors and lack of logical frameworks for readers
Decrease transmission time when you are circulating documents	Increase use of excessive language and other irresponsible and destructive behavior
And speed up meeting follow-up activities because all decision and action items are recorded electronically and can distributed electronically	Overload receivers with trivia and unnecessary information
	The sender can not control if and when a receiver chooses to read a message and that, because the messages is electronic, it is less private than hard copy

[From Munter, M. (1999). *Managing Complexity in the High Technology* (Glinow, V., ed.), pp. 19–26; 19–33. Oxford: Oxford University Press. With permission.]

In 1992, Stalk and colleagues distinguished core competence from a firm's strategy capabilities, "whereas core competence emphasizes technological and production expertise at specific points along the value chain, capabilities are more broadly based, encompassing the entire value chain." They define a capability as "a set of business processes strategically understood. . . .the key is to connect them to real customer needs."

VI. ORGANIZATIONAL EFFECTS OF HUMANS MANAGING INFORMATION SYSTEMS

In 1991 Sproull and Kiesler considered a two-level perspective that emphasizes how technologies can have both efficiency effects and social system effects. They stress how some analyses have no way to recognize that the most important effects of a new technology may be not to let people do old things more efficiently but instead do new things that were not possible or feasible with the old technology. They put some examples with today's most popular computer-based communication technologies and applications; for example, networks and electronic mail change the ways people interrelate to each other. Some of these effects are:

- **First level efficiency effects—**Customers may be reluctant to leave messages with the secretary because its contents are too complicated, or they may be unwilling to reveal private information to an intermediary. But leaving an electronic message is a very practical option. Through this tool, people can work more efficiently than they could do otherwise.
- **Second-level effects—**Electronic mail makes it possible to have fast, asynchronous group communication, as well as one-to-one communication. Electronic group mail can decrease group coordination costs just as electronic one-to-one decreases one-to-one coordination costs. In the nonelectronic world,

Table V Managerial Strategies for Successful Implementation

	Vicarious learning	Learning through experience
Corporate level	• Facilitate horizontal transfer of knowledge among sites	• Select first sites to be representative of later user sites
Plant level	• Actively seek implementation know-why and know-how from previous users	• Create local user-experts to absorb and transmit implementation knowledge • Adjust pace of change to match available resources

[From Leonard-Barton, D. (1990). *Managing Complexity in the High Technology Organizations* (Glinow, V., ed.), Oxford: Oxford University Press. With permission.]

groups use face-to-face meetings to link members with one another and time outside meetings to buffer members from one to another. During meetings members become mutually aware of other's attitudes and problems.

A. Firm's Communication Effects

When technological change creates new social situations, traditional expectations and norms can lose their power. People try new ways of working.

Deindividuation occurs when people have anonymity. These situations can inspire agitation, feelings of being "part of something else," and freedom from social or moral structures—feelings that in turn lead to suggestibility—or do whatever a leader or strong cue suggests. Firms develop explicit rules and policies for information systems' use and they differ from one organization to another.

Every relationship is both informational and emotional. Electronic communication may offer peripheral employees new opportunities to initiate connections within the organization to reduce the information gap and increase motivation. If conectivity is high, there are potentially many people accesible via the network. Because of the social processes described, employees should feel somewhat uninhibited about "meeting" new people electronically.

B. New Human's Role in Organizations

The remote worker is a new role which appeared due to the implementation of information and communication technologies in the firm.

A simple technology introduced to speed up the transmission of information led to changes in who had control over information and expertise. These changes made old social relationships much more complicated. Changes in control over information led to changes in work performance.

The relative lack of status in some information systems means that some strategic choices affect control over information, performance, and social influence.

C. New Ways of Organizing

Recent developments in the management of information systems suggest that more substantial structural change may be possible. The foundations will be computer networks: telecommunications, transport and data networks that reach throughout the organization and beyond it to customers or clients and suppliers. The beforementioned second-order effects are more related to changing patterns of attention, social contact, and interdependencies than they are to speeding up information flow.

The concept of information systems' psychological switching costs can become very important at a firm level. The concept means that, even if economic costs are equivalent, if people from different organizations have developed satisfying personal relationships with one another, they will be reluctant to forego those relationships by switching to another supplier. People naturally attend to what is close at hand and ignore what isn't.

VII. SOME ETHICAL ISSUES

Information is a commodity and a very useful one at a firm level. Information systems will help us to decide which information is really essential to help us with the decisions that we need to make. We should not question what to do with the next innovation in our information system, but how to manage the possibilities it offers us in our daily work.

In 1995 Leonard-Barton admits that ethical decisions related to acquiring, processing, storing, disseminating, and using information can affect workers' quality of life. As computer-based work affects more relationships with customers and users, crucial issues such as privacy, information access, and protection of intellectual property appear.

Firms should take into consideration some ethical issues, especially the ones related with:

1. The lack of use of information systems dedicated to enhance business processes
2. An inappropriate use of information systems, for example, using a company computer for driving personal business
3. Destruction or change of information, for example, "hacking" and trespassing into systems, including introducing software viruses
4. Computer crime, for example, using systems for theft and fraud

Key ethical issues include the requirement for individual privacy and the confidentiality of information. Significant ethical issues have to do with the fact that managers must seek a balance between their temptation to acquire some private data and their obligation to respect the privacy and autonomy of others. The

accessibility and reliability of available information is another critical ethical issue. It is an ethical responsibility for the manager to ensure that their organization's information systems are both reliable and accurate. People rely on information to make decisions, decisions that materially affect their lives and the lives of others and they must properly use this information.

VIII. SUMMARY

Useful information systems, as parts that fit together well in terms of social values and policies, must be properly managed in a social matrix consisting of services, norms, and conventions.

If the practices that evolve in the managing of an information system are efficient and productive but fail to uphold the ideals or ethics of the people involved, the system will be subject to considerable stress.

What is really interesting about information systems is their potential to foster diversity. Information systems are accesible to individuals and small local groups as well as large organizations. Communication and information sharing can take place between one person or another, among limited groups of people, or throughout the whole world of the information system users.

Information systems management means first of all monitoring and controlling information systems and second, planning modifications and incorporating new elements. Due to the important role people play as the most outstanding resource in the information system, there are some human roles in relation with information systems that allow the proper organizing of those functions mentioned above. It is needed to develop a structured approach to integrate the different elements in an information system as it evolves.

Organizations first need to gain experience and be comfortable with information system's groups and be able to deal with information procedures before they attempt electronic restructuring.

Our vision puts attention on the relationship among people. A user implies a view of technology as a discrete commodity whose attributes and functions can be specified by the designer. For many firm's reasons people need to communicate with each other by means of an information system of which they are part. Technology is just a facilitator.

Organizations best achive benefits from new technology when they make complementary changes in organization and management. Often there is no argument here; new technology is viewed as an opportunity to make other organizational changes, and the only question is what changes must be made.

These two sets of objectives are contradictory but together make a point: computer-based communication allows people to work somewhat more efficiently, but the realized benefits depend ultimately on the policies, designs, and vision of people who want to organize work in new ways.

SEE ALSO THE FOLLOWING ARTICLES

Computer-Supported Cooperative Work • Digital Divide, The • Ergonomics • Human Resource Information Systems • Organizations, Information Systems Impact on • People, Information Systems Impact on • Psychology • Resistance to Change, Managing • Sociology • Staffing the Information Systems Department • Virtual Organizations

BIBLIOGRAPHY

Bird, C., Schoonhoven, and Jelinek, M. (1990). Dynamic tension in innovative, high technology firms: Managing rapid technological change through organizational structure. In *Managing complexity in the high technology organizations* (Glinow, V., ed.), pp. 90–118, Oxford: Oxford University Press.

Burgelman, R. A., and Rosenbloom, R. S. (1989). Technology strategy: An evolutionary process perspective. In *Research on technological innovation, Management and Policy,* Vol. 4 (Rosenbloom, R. S. and Burgelman, R. A., eds.), pp. 1–23. Greenwich, CT: JAI Press.

Burgelman, R. A., and Rosenbloom, R. S. (1999). Designing and implementation of technology strategy: An evolutionary perspective. In *The technology management handbook* (Dorf, R., ed.), pp. 16–1; 16–15, Chap. 16. Boca Raton, FL: CRC Press.

Clark, K. B. (1987). Managing technology in International competition: The case of product development in response to foreign entry. In *International competitiveness* (M. Spence, and H. Hazard, eds.), pp. 27–74. Cambridge, MA.

Ellul, J. (1964). *The technological society.* New York: Vintange Books.

Evenland, J. D., and Bikson, T. K. (1988). Work group structures and computer support: A field experiment. *Transactions on Office Information Systems,* Vol. 6, No. 4, 354–379.

Hampson, K. D. (1993). Technology strategy and competitive performance: A study of bridge construction. Doctoral dissertation. Standford University, Standford, CA.

Henderson, R. M., and Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms, *Administrative Science Quaterly,* No. 35, pp. 9–30.

Kiesler, S., and Sproull, L. (1982). Managerial response to changing environments: Perspectives on problem sensing from social cognition. *Administrative Science Quaterly,* No. 27, pp. 548–570.

Leonard-Barton, D. (1990). Implementing new production technologies: Exercises in corporate learning. In *Managing complexity in the high technology organizations* (Glinow, V., ed.), pp. 160–187. Oxford: Oxford University Press.

Leonard-Barton, D. (1995). *Wellsprings of knowledge: Building and sustaining the sources of innovation*. Boston: Harvard School Business Press.

Lewis, L. (2001). Managing business and service networks. In *Network and Systems Series* (Manu Malek, ed.), Dordrecht: Kluwer Academic.

Meister, D. (1987). Systems design, development and testing. In *Handbook of Human Factors* (Salvendy, G., ed.), pp. 17–42. New York: John Wiley & Sons.

Munter, M. (1999). Effective meeting management. In *Managing complexity in the high technology organizations* (Glinow, V., ed.), pp. 19–26; 19–33. Oxford: Oxford University Press.

Nardy, B. A., and O'Day, V. L. (1999). *Information ecologies, using technology with heart*. Cambridge, MA: The MIT Press.

Porter, M. E. (1985). *Competitive advantage: Creating and sustaining superior performance*. New York: The Free Press.

Quinn, J. B. (1986). Innovation and corporate strategy. In *Technology in the modern corporation* (Horwitch, M., ed.), pp. 167–183. Boston, MA: Pergamon Press.

Rogers, E. M. (1982). *Diffusion of innovations*. New York: The Free Press.

Rosenbloom, R. S. (1985). Managing technology for the longer term: A managerial perspective. In *Uneasy alliance: Managing the productivity knowledge dilemma* (Clark, K. B., Hayes, R. H., and Lorenz, C., eds.). Boston, MA: Harvard Business School Press.

Sproull, L., and Kiesler, S. (1991). Connections. Cambridge, MA: The MIT Press.

Stalk, G., Evans, P., and Shulman, L. E. (1992). Competing on capabilities: The new rules of corporate strategy. *Harvard Business Review,* pp. 57–69.

Wilson, E. O. (1992). The diversity of life. New York: Norton and Company.

Hybrid Systems

Imre J. Rudas
Budapest Polytechnic

GLOSSARY

artificial neural networks (ANNs) or simply neural networks (NNs) are large-scale systems involving a large number of special type, nonlinear, coupled processors called "neurons." These processing elements can be connected together according to well-defined typical structures fit to solve different typical problem classes. Certain networks can be taught by given patterns of data; others can learn or classify patterns automatically. NNs can recognize not only the patterns used for their training, but also similar patterns by generalization as well.

computational intelligence umbrellas and unifies neural networks, fuzzy systems, and evolutionary computing.

expert system knowledge-based systems that reproduce the behavior of a human expert in a restricted application area. Knowledge can be represented in many different ways, such as frames, semantic nets, rules, etc. Knowledge is processed in inference engines, which normally perform symbol processing, i.e., truth values of antecedents, conclusions, etc.

fuzzy sets theory was introduced by Lotfi A. Zadeh as an extension of binary crisp logic, that is, a mathematically rigorous framework to deal with uncertainty of not necessarily statistical nature. It applies the generalization of the "characteristic function" of the classical sets (crisp logic) according to which an element either belongs to a set or does not belong at all, as this function takes the value of 1 or 0, respectively. Fuzzy logic allows a gradual transition from membership to nonmembership via the "membership function," expressing to what extent the elements of any classic subset can belong to a "concept" represented by the fuzzy set. Fuzzy sets are especially advantageous means for mathematical representation of semi-qualitative/quantitative linguistic concepts as "big," "small," "very much," "scarcely," etc.

genetic algorithms (GA) first proposed by John Holland in early 1970s. A GA can be described as being a parallel global search technique that mimics the evolution mechanism of organisms. The basic element that is processed by a GA is a finite bit string (gene) which is composed by joining together some substrings, each of which is the binary coding of a parameter of the search space. Each string thus represents a possible solution to the optimization problem. Let N be the population of a group of such strings. Let N(0) be the randomly generated initial population and N(t) be the population at time t. The main loop of a GA is the generation of a new population N(t + 1) from N(t). This is done by applying some so-called genetic operators on the population. The first thing that is done on the population is an evaluation. Each string of the current population is decoded into the corresponding parameters. A fitness function then takes these parameters as inputs and returns a fitness value. This

value indicates a measure of the parameters performance on the objective function. The next operation is reproduction, in which the strings make copies of themselves. The strings with a higher value of fitness have a greater probability of producing one or more copies. Then crossover and mutation operators operate on the population. Crossover is the exchange of a portion of strings among the population, and mutation randomly changes some bits of a string. The overall effect is to move the population N toward the part of the solution space with higher values of fitness function.

soft computing based on fuzzy logic, artificial neural networks, and probabilistic reasoning, including genetic algorithms, chaos theory, and parts of machine learning, and has the attributes of approximation and dispositionality.

HYBRID SYSTEMS are those systems in which fuzzy logic, neural networks, and genetic algorithms are used in combination. Each of these techniques allows the combination of domain knowledge and empirical data to solve real-world complex problems. Concerning hybrid systems, Lotfi A. Zadeh's expectation is: *"in coming years, hybrid systems are likely to emerge as a dominant form of intelligent systems. The ubiquity of hybrid systems is likely to have a profound impact on the ways in which man-made systems are designed, manufactured, deployed and interacted with."*

I. COMPUTATIONAL INTELLIGENCE AND SOFT COMPUTING AS PILLARS OF HYBRID SYSTEMS

Recent advances in emerging technologies; in knowledge representation and processing; and in sensor technology, sensor data processing, and sensor fusion, with the availability of powerful low-cost microprocessors, predicate the era of intelligent systems.

The conventional approaches for understanding and predicting the behavior of such systems based on analytical techniques can prove to be inadequate, even at the initial stages of establishing an appropriate mathematical model. The computational environment used in such an analytical approach may be too categoric and inflexible to cope with the intricacy and the complexity of real-world industrial systems. It turns out that in dealing with such systems one has to face a high degree of uncertainty and tolerate imprecision, and trying to increase precision can be very costly.

In the face of difficulties stated previously, fuzzy logic, artificial neural networks, and evolutionary computing techniques were integrated as hybrid systems. Soft computing and computational intelligence are two approaches to the symbiosis of these techniques.

A. Computational Intelligence

The term "computational intelligence" started to be known worldwide in 1994. Since then not only have a great number of papers and scientific events been dedicated to it, but numerous explanations of the term have been published. In order to have a brief outline of history of the term, the founding and most interesting definitions will be summarized in this section.

1. The Definition of Bezdek

Based on the 1994 World Congress on Computational Intelligence, a book of invited papers was published under the name *Computational Intelligence: Imitating Life* in which J. C. Bezdek gave the following definition.

A system is *computationally intelligent* when it deals only with numerical (low-level) data, has a pattern recognition component, does not use knowledge in the artificial intelligence sense, and additionally, when it (begins to) exhibit

- Computational adaptivity
- Computational fault tolerance
- Speed approaching human-like turnaround
- Error rates that approximate human performance

Bezdek considers an artificially intelligent system as a computational intelligence system whose "added value comes from incorporating knowledge in a nonnumerical way."

2. The Definition of Marks

R. Marks in 1993 gave the following definition, which probably forms the basis of most of the subsequent definitions.

> Neural networks, genetic algorithms, fuzzy systems, evolutionary programming, and artificial life are the building blocks of computational intelligence.

Mark's definition is rather practical and most authors use it with slight modifications, by accepting usually that computational intelligence umbrellas and unifies neural networks, fuzzy systems, and genetic algorithms.

According to the view of L. T. Kóczy, hard and soft computing represent just subsequent steps in the evolution of machine intelligence by utilizing additional structural ("sub-symbolic") information, a more efficient way is presented to cope with computational complexity. Further steps are more advanced approaches in soft computing, such as rule interpolation in sparse knowledge bases, and hierarchically structured interpolative approaches, where a strongly articulated role is given to hybrid technologies especially in model identification.

B. Soft Computing

Zadeh proposed a new approach for machine intelligence, separating hard computing techniques based on artificial intelligence from soft computing techniques based on computational intelligence (Fig. 1).

- *Hard computing* is oriented towards the analysis and design of physical processes and systems and has the characteristics of precision and formality. It is based on binary logic, crisp systems, numerical analysis, probability theory, differential equations, functional analysis, mathematical programming, approximation theory, and crisp software.
- *Soft computing* is oriented towards the analysis and design of intelligent systems. It is based on fuzzy logic, artificial neural networks, and probabilistic reasoning, including genetic algorithms, chaos theory, and parts of machine learning, and has the attributes of approximation and dispositionality.

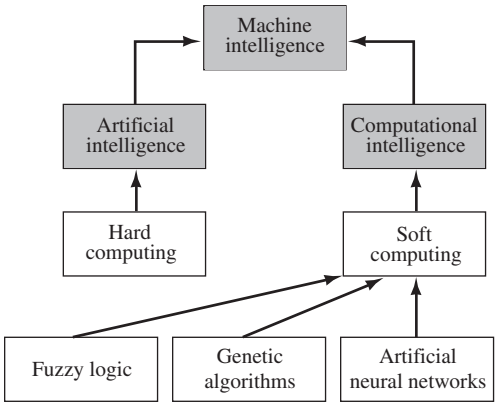


Figure 1 Artificial intelligence vs. computational intelligence.

Although in hard computing imprecision and uncertainty are undesirable properties, in soft computing the tolerance for imprecision and uncertainty is exploited to achieve an acceptable solution at a low-cost, tractability, high machine intelligence quotient (MIQ). Zadeh argues that soft computing, rather than hard computing, should be viewed as the foundation of real machine intelligence.

Soft computing, as he explains, is (1) a consortium of methodologies providing a foundation for the conception and design of intelligent systems and (2) is aimed at a formalization of the remarkable human ability to make rational decisions in an uncertain and imprecise environment.

The guiding principle of soft computing is: *Exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution cost, and better rapport with reality.*

The constituents and the characteristics of hard and soft computing are summarized in Table 1.

The connection between approximation theory in hard computing and approximation by soft computing approaches is obvious when considering various interpolation techniques applied in fuzzy and artificial neural network (ANN) models in the case of sparse knowledge.

Fuzzy logic (FL) is mainly concerned with imprecision and approximate reasoning; neurocomputing with learning and curve fitting; genetic computation with searching and optimization; and probabilistic reasoning with uncertainty and propagation of belief. The constituents of soft computing are complementary rather than competitive.

The experiences gained over the past decade have indicated that it can be more effective to use them in a combined manner, rather than exclusively. For example, an integration of FL and neurocomputing has already become quite popular (neuro-fuzzy control) with many diverse applications, ranging from chemical process control to consumer goods.

Systems in which FL, neural networks, and genetic algorithms are used in combination are referred to as hybrid systems.

In the following sections hybrid systems will be called according to the order of the leading and supporting role in the systems, i.e., a neuro-fuzzy system is a neurosystem supported by FL.

II. FUZZY-NEURO SYSTEMS

Systems where neural networks are used to improve the fuzzy systems are referred as fuzzy-neuro systems. The neural nets are mainly used to increase system

Table I The Constituents and the Characteristics of Hard and Soft Computing

Hard computing		Soft computing	
Based on	**Has the characteristics**	**Based on**	**Has the characteristics**
• Binary logic	• Quantitative	• Fuzzy logic	• Qualitative
• Crisp systems	• Precision	• Neurocomputing	• Dispositionality
• Numerical analysis	• Formality	• Genetic algorithms	• Approximation
• Differential equations	• Categoricity	• Probabilistic reasoning	
• Functional analysis		• Machine learning	
• Mathematical programming		• Chaos theory	
• Approximation theory		• Evidental reasoning	
• Crisp software		• Belief networks	

performance and to reduce developing time and cost. Due to the great number of approaches, only some typical fuzzy-neuro systems are discussed.

A. Learning of Rule Base

In many applications there is no or limited a priori knowledge about the rules. Clustering is one of the methods used to create the rule base in case of insufficient knowledge about rules or fuzzy sets. The generation can be performed by fuzzy clustering or by neural networks. Pedrycz and Card in 1992 introduced the linguistic interpretation of Kohonen's self-organizing map that can be used successfully for the clustering. For clustering, some special cases of the cluster-oriented approaches can be applied, such as the hyperbox method or the structure-oriented approach.

B. Design of Membership Functions

Two major methods can be distinguished to design membership functions by using neural networks (NNs), such as

- Generation of nonlinear multidimensional membership functions by NNs
- Tuning of parameters of the membership functions by NNs

The first NN approach to generate membership functions was NN-driven fuzzy reasoning. This approach uses an NN to represent multidimensional nonlinear membership functions and uses the NN as a membership generator in a fuzzy system. In conventional fuzzy systems, one-dimensional membership functions are independently designed and then combined to generate multidimensional membership functions. The advantage of the NN-based method is that it can generate nonlinear multidimensional membership functions directly.

The design process of NN-driven fuzzy reasoning has three steps:

- Clustering the given training data
- Fuzzy partitioning the input space by NNs
- Designing the consequent part of each partitioned space

Another NN approach to design membership function is the tuning of the parameters which define the shape of the membership functions in order to reduce error between output of the fuzzy system and supervised data. Two methods have been used for this purpose:

- Gradient-based methods
- Genetic algorithms

The genetic algorithm-based parameter tuning will be discussed in the next section.

The steps of the gradient-based methods are:

- Determination of the parameters of the membership functions
- Tuning of the parameters to minimize the actual output of the fuzzy system and the desired output using gradient methods

Usually the center position and width of the membership functions are used as shape definition parameters.

Sugeno and Yasukawa (1993) proposed a very efficient way for learning trapezoidal fuzzy rule bases us-

ing fuzzy clustering methods. Kóczy *et al.* (1999) proposed an extension of this approach for the identification of hierarchically structured interpolative fuzzy models, which might lead to a major step in overcoming the difficulties of exponential complexity in systems with a large number of state variables.

C. Adaptive Network-Based Fuzzy Inference System, a Typical Fuzzy-Neuro System

A typical NN for tuning fuzzy logic systems is the adaptive network-based fuzzy inference system (ANFIS) proposed by Jang in 1993. ANFIS represents (Fig. 2) a Takagi-Sugeno type fuzzy model with six layers in a special feedforward network. The rules are of the form

R_i: IF x_1 is A_1^i AND x_2 is A_2^i AND... AND x_n is A_n^i
THEN $y_i = c_0^i + c_1^i x_1 + ... + c_n^i x_n$, i = 1, 2, ..., m

The first layer is the system input, while in the second layer each neuron is connected to one of the input variables and stores the membership function. The neurons in the third layer perform a suitable t-norm representing the antecedents of the fuzzy rules. The fourth layer computes the relative degree of fulfillment for each rule. The polynomial coefficients of the right-hand side of each Takagi-Sugeno rule are computed in the fifth layer. The sixth layer computes the output value.

For learning the combination of backpropagation, the least-mean square estimation is used. To update the parameters of the fuzzy sets backpropagation, while updating the polynomial coefficients, the least-mean square algorithm is used.

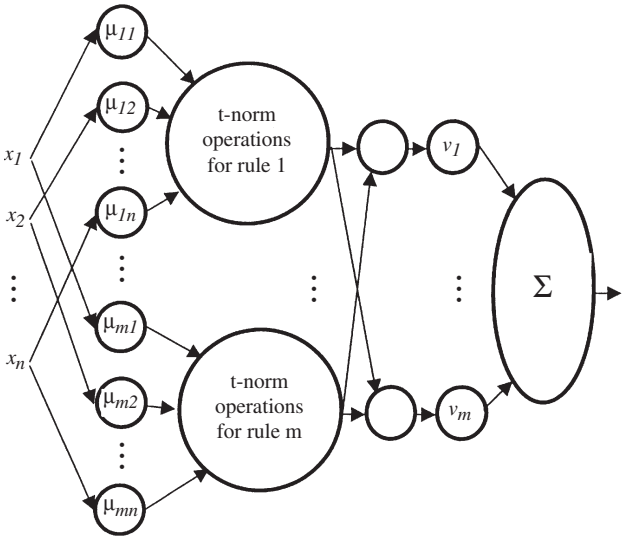


Figure 2 The structure of ANFIS.

III. NEURO-FUZZY CONTROLLERS

In recent years (since about 1990) we have seen a growing use of neuro-fuzzy controllers. In general, NNs, in such applications, are not user trainable and their function is one of the four depicted in Fig. 3.

In Fig. 3a, the use of the NN is shown to be a development tool. Designing and fine-tuning of the membership functions are carried out by the NN, as was discussed previously.

In some cases, a NN is used completely independently, (Fig. 3b) mainly for nonlinear function interpolation. For example, in a Matsushita air conditioner, a NN is used, completely independent of the fuzzy system that controls the heat pumping, to derive the

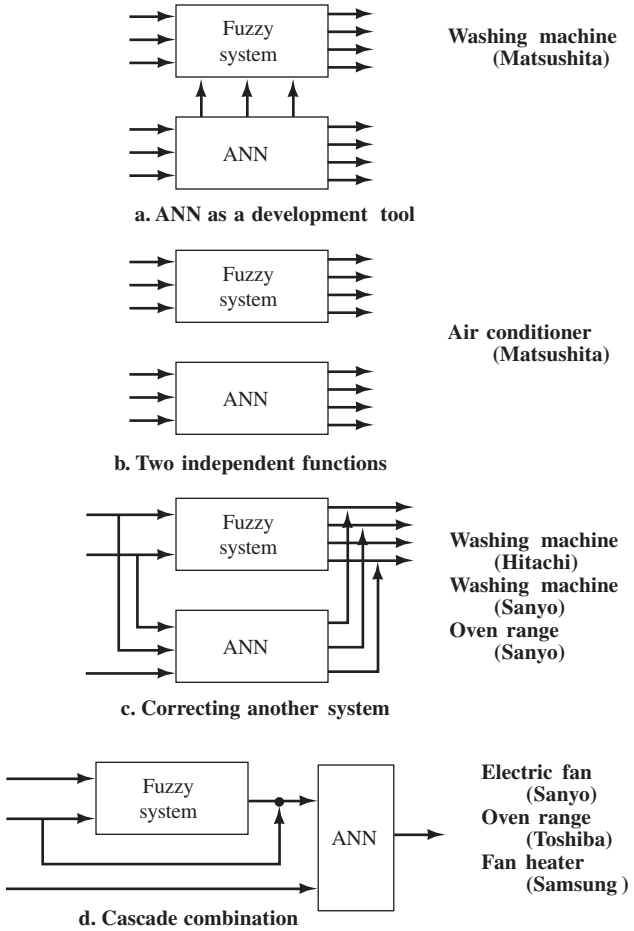


Figure 3 Neuro-fuzzy combinations.

predictive mean vote as defined by ISQ 773 as a measure of comfort level. This requires a mapping from 6D space to 6D space, which is done by the NN.

In some other applications, the role of the NN is corrective, as shown in Fig. 3c. This is the approach used to improve the performance of an already marketed product. For example, the later models of a washing machine can incorporate a NN to handle some extra input that was not considered in the original fuzzy controller.

In cascaded systems (Fig. 3d), FS accomplishes a part of the task and then passes it to a NN. For example, in a Sanyo neuro-fuzzy fan, the fan must rotate toward the user, which requires the determination of the direction of the remote controller.

In some consumer products marketed recently, the standard system modifies itself automatically according to the usage style of the owner and/or the owner can modify the basic operational parameters of a system according to his or her personal preferences. The determination of the preheating time in a kerosene fan heater and the cooling level in an air conditioner can be cited as examples.

IV. FUZZY-GENETIC SYSTEMS

Since the first application of genetic algorithms in FSs was reported in 1989 by C. Karr, L. Freeman, and D. Meredith, a great number of papers have been published in this area. Genetic algorithms are used to design the membership functions in antecedents, consequents parameters, and the number of rules.

The application of genetic algorithms involves two main issues:

1. Genetic coding of the parameter set
2. Generation of the fitness of the parameter set

In order to decide the shape of membership function, the number of rules, and consequent parts, the parameterization of the FS has to be done which involves the choice of the parameters of the shapes of the membership functions, the representation of the number of rules indirectly by considering the boundary conditions of the application and the center positions of membership functions, and the representation of the consequent parts. Then these parameters are genetically coded as concatenated binary strings.

The determination of the fitness function used to represent the goodness of a solution depends on the particular application; the general method is not available.

V. GENETIC-FUZZY SYSTEMS

The parameters of the genetic algorithm (GA), such as population size, crossover, and mutation rates, have great influence on the performance of the algorithm. For run-time tuning these parameters, Lee and Takagi (1993) developed a genetic-fuzzy system. The proposed fuzzy controller has the following input variables:

- Average fitness/best fitness
- Worst fitness/average fitness
- Best fitness

The output variables modifying the GA parameters are:

- Population size
- Crossover rate
- Mutation rate

The parameters should be remained within given operational ranges, and the changes of the parameters cannot be more than 50%. The method provides high improvement in run-time computational efficiency.

Some other methods can be found in the literature for tuning the basic GA parameters, for example, the meta-level GA proposed by Grefenstette in 1986. The tuned parameters are population size, crossover rate, mutation rate, generation group, selection strategy, and scaling window.

VI. NEURO-GENETIC SYSTEMS

In neuro-genetic systems, usually GAs are used to design the ANN. The most typical applications are:

1. Optimization of weights of ANNs by GAs instead of the well-known gradient methods
2. Decision of ANN configuration by GAs, where the ANN is an evolutional system

Neuro-genetic systems are applied, for example, in electric consumer products to provide adaptability to the user environment or for costumizing the mass products on the user's side. A typical kind of application is the air conditioner produced by LG Electric. The input variables of the system are room temperature, outdoor temperature, reference temperature, time, and the temperature set by the user. The control target is to keep the temperature set by the user. The control is performed by an ANN, but if the user changes the required temperature then a GA modifies the number of neurons and the weights.

Another type of neuro-genetic system allows the learning process of the ANN instead of backpropagation by the GA. Backpropagation-based ANNs usually converge faster than GAs because of their exploitation of local knowledge, but on the other side this local search can cause the ANN to frequently become stuck in a local minimum. The GA is slower but results in a global search. To take advantage from both techniques a hybrid learning algorithm can be used, in which first the GA is used to find a suitable parameter region and the final tuning is performed by backpropagation.

VII. HYBRID HIERARCHICAL INTELLIGENT CONTROL

Fukuda and his co-workers proposed a hierarchical intelligent control (HIC) system to control intelligent robots. The HIC system is a typical hybrid system of ANNs, FL, and GAs and consists of three levels, namely, a *learning level,* a *skill level,* and an *adaptation level* as shown Fig. 4.

The learning level is based on an expert system and provides the recognition and planning to develop control strategies. In the recognition level, NNs and fuzzy NNs are used. The inputs of the NNs are numeric values provided by sensors, while the outputs are a symbolic quality indicating the process states. The inputs and outputs of the fuzzy NN are numeric quantities, and the fuzzy NN clusters input signals by using membership functions. Both the NN and the fuzzy NN are trained with the training data sets of a priori knowledge obtained from human experts.

The planning level reasons symbolically for strategic plans or schedules of robotic motion, such as task, path, trajectory, force, and other planning in conjunction with the knowledge base. The GA optimizes the control strategies for robotic motion and optimizes structures of NN and FL connecting each level.

At the skill level the fuzzy NN is used for specific tasks following the control strategy produced at the learning level in order to generate appropriate control references.

Compensation for nonlinearity of the system and uncertainties included in the environment are dealt with in the NN. Thus, the NN in the adaptation process works more rapidly than that in the learning process It is shown that the NN-based controller, the Neural Servo Controller, is effective to the nonlinear dynamic control with uncertainties. Eventually, the NNs and the fuzzy NNs connect neuromorphic control with symbolic control for HIC while combining human skills.

This HIC can be applied not only to a single robot, but to multi-agent robot systems as well.

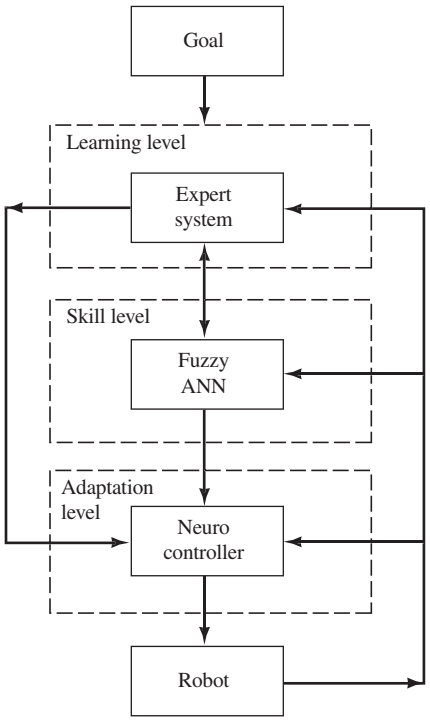


Figure 4 Hierarchical hybrid controller.

VIII. SOME OTHER APPLICATIONS OF SOFT COMPUTING TECHNIQUES

A. Fuzzy-Neural Decision Making System

Fuzzy-neural systems can be used in engineering systems also. For example, in distributed control a number of machines or devices work together to reach a specific objective. The individual elements of the process (agents) can work cooperatively where an upper level supervisory controller can redistribute the tasks. On the other hand, the agents may operate separately in a competitive and reactive manner. In each case the performance within the distributed control environment can be significantly improved by incorporating intelligence into the agents themselves, with or without a higher level intelligent supervisory controller.

A fuzzy-neural system, given by De Silva, can be applied for a decision making problem associated with goal seeking by multiple agents. The knowledge base consists of fuzzy IF-THEN rules. Each rule has a "belief value," which represents the level of the validity of that rule. The belief values and the parameters of the membership functions are initially determined off-line

using a static, feedforward NN employing a training set of data obtained from the process. These parameters are updated on-line during operation of the decision system. The output of the knowledge-based system will be a decision.

The decision making problem can be formulated as the optimization of a cost function. The cost function itself expresses the cost of making a particular decision under a given set of conditions and for a specified set of process objectives. The cost weightings may be assigned through experience and expertise. Once the knowledge base, the membership functions, and the belief values for the rules are all known, the on-line decision making may be carried out by applying the compositional rule of inference. The steps involved would be the following:

1. Fuzzification of sensory information.
2. Supply the fuzzified sensory information to the knowledge base, and apply the compositional rule of inference.
3. Interpret the goal-seeking decision obtained in Step 2, and execute it.

B. Fuzzy Expert Systems

Expert systems are knowledge-based systems that reproduce the behavior of a human expert in a restricted application area. Knowledge can be represented in many different ways, such as frames, semantic nets, rules, etc. Knowledge is processed in inference engines, which normally perform symbol processing, i.e., truth values of antecedents, conclusions, etc.

The increasing importance of handling uncertainties led to the development of fuzzy expert systems. The main issues of using FL in expert systems are the following:

- To include uncertainty in the knowledge base
- To use approximate reasoning algorithms
- To apply linguistic variables

The knowledge base of an expert systems is built up from human knowledge which is usually imprecise, hence the application of the fuzzy rule base is a much more realistic modeling of a real-world situation than using crisp logic.

Uncertainty in the knowledge base results in uncertainty in the conclusion, so as a consequence of the representation of uncertain knowledge adequate reasoning methods should be used. This can be either the compositional rule of inference suggested by Zadeh or any of the numerous reasoning algorithms.

The interface modules of the expert system both on the user and on the expert side can communicate with human beings in a more natural way by using linguistic variables.

SEE ALSO THE FOLLOWING ARTICLES

Decision Support Systems • Engineering, Artificial Intelligence in • Evolutionary Algorithms • Expert Systems Construction • Industry, Artificial Intelligence in • Intelligent Agents • Knowledge Acquisition • Knowledge Representation • Machine Learning • Neural Networks • Robotics

BIBLIOGRAPHY

De Silva, C. W. (1994). Automation intelligence. Engineering Application of Artificial Intelligence. 7(5):471–477.

Fukuda, T., and Shibata, T. (1993). Hierarchical Control System in Intelligent Robotics and Mechatronics. *Proc. of the IECON'93, Int. Conf. on Industrial Electronics, Control and Instrumentation*. Vol. 1, pp. 33–38.

Fuller, R. (2000). Introduction to neuro-fuzzy systems. In *Series of Advances in Soft Computing*. Heidelberg: Physica-Verlag.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Grefenstette, J, (1986). Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybernetics*. SMC 16:122–128.

Kaynak, O., Zadeh, L. A., Türksen, B., and Rudas, I. J., Eds. (1998). *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*. Springer NATO ASI Series. Series F: Computer and Systems Sciences. Vol. 192.

Kóczy, L. T. (1995). Algorithmic aspects of fuzzy control. Int. J. Approximate Reasoning. 12:159–219.

Kóczy, L. T., Hirota, K., and Muresan, L. (1999). Interpolation in hierarchical fuzzy rule bases. Int. J. Fuzzy Systems. 1:77–84.

Kohonen, T. (1984). *Self-Organization and Associative Memory*. Berlin/New York: Springer-Verlag.

Lee, M. A., and Takagi, H. (1993). Dynamic control of genetic algorithm using fuzzy logic techniques. In *Proc. of the 5th Int. Conf. on Genetic Algorithms,* ICGA'93, Urbana-Chamaign, II, S. Forrest, ed., pp. 76–83. San Mateo, CA: Morgan Kaufmann.

Ruspini, E., Bonisone, P. P., and Pedrycz, W., Eds. (1998). *Handbook of Fuzzy Computation*. Bristol and Philadelphia: Institute of Physics Publishing.

Sugeno, M., and Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. IEEE Trans. on Fuzzy Systems. 1:7–31.

Takagi, H. (1993). Cooperative system of neural networks and fuzzy logic and its application to consumer products. In *Industrial Industrial Applications of Fuzzy Control and Intelligent Systems,* J. Yen and R. Langari, eds., New York: Van Nostrand-Reinhold.

Takagi, H. (1993). Fusion Techniques of Fuzzy Systems and Neural Networks, and Fuzzy Systems and Genetic Algorithms. Proc. SPIE Proc. Technical Conference on Applications of Fuzzy Logic Technology. Vol. 2061, pp. 402–413.

Hyper-Media Databases

William I. Grosky
University of Michigan, Dearborn

GLOSSARY

anglogram A histogram of angles, where each bin of the histogram corresponds to a particular angle range.
content-based image retrieval Retrieving particular images from an image database based on their content.
data model An abstract approach to modeling real-world entities inside a database.
Delauney triangulation A triangular mesh constructed over a set of points.
feature point A feature (particular property of a media object) which can be represented by a point. For example, a corner point is an example of an image shape feature.
histogram An integer vector, where each integer represents the number of particular objects having values in a certain range. For example, a color histogram is an integer vector, where each integer is the number of pixels in an image having a value in a given range of gray-levels.
image database A collection of images organized for efficient searching.
point feature map A collection of point features of a media object.
semantic gap The conceptual distance between the way an automated system represents media objects (at the level of features) and the way a user thinks of media objects (at the level of concepts)
semcon An image region having some particular semantics.

I. INTRODUCTION

In the past 15 years, the database field has been quite active, whether in discovering more efficient methods for managing classical alphanumeric data, in bringing application-dependent concepts, such as rules, into the database environment, or in managing such new types of data as images and video. When new types of data are first brought into a database environment, it is quite natural that this data is transformed so as to be representable in the existing database architectures. Thus, when images were first managed in a database, researchers developed numerous techniques concerned with how to represent them, first in a relational architecture and then in an object-oriented architecture.

If this representation is done in a way compatible with the types of queries and operations that are to be supported, then the various modules that comprise a database system ostensibly don't have to be changed to work with this new type of data. After all, if an image or its contents can be represented as a set of tuples over several relations, then why shouldn't the classical relational techniques developed for indexing, query optimization, buffer management, concurrency control, security, and recovery work equally well in this new environment? Historically, this is what indeed occurred. It is only after some experience working with new types of data transformed in such a way as to be part of existing database systems that one comes to the conclusion that there is an inherent weakness with this approach. There is a mismatch

between the nature of the data being represented and the way one is forced to query and operate on it.

Queries and operations based on classical approaches just won't do for multimedia data, where browsing is an important paradigm. The importance of this paradigm is illustrated by the fact that multimedia databases are sometimes referred to as *hypermedia databases*. Standard indexing approaches won't work for annotation-independent, content-based queries over multimedia data. Other modules of a database system likewise have to be changed in order to manage multimedia data efficiently. At the present time, we realize that this must be done, but there is no agreement on how to proceed. Commercially, the object-relational database systems are state of the art for implementing hypermedia database systems, but even these systems leave much to be desired.

The process of managing multimedia data in a database environment has gone through the following historical sequence:

1. Multimedia data were first transformed into relations in a very ad hoc fashion. Depending on how this was done, certain types of queries and operations were more efficiently supported than others. At the beginning of this process, a query such as *Find all images containing the person shown dancing in this video* was extremely difficult, if not impossible, to answer in an efficient manner.
2. When the weaknesses of the above approach became apparent, researchers finally asked themselves what type of information should be extracted from images and videos and how this information should be represented so as to support content-based queries most efficiently. The result of this effort was a large body of articles on multimedia data models.
3. Since these data models specified what type of information was extracted from multimedia data, the nature of a multimedia query was also discussed. Previous work on feature matching from the field of image interpretation was brought into a database environment and the field of multimedia indexing was initiated. This, in turn, started the ball rolling in multimedia query optimization techniques.
4. A multimedia query was realized to be quite different than a standard database query, and close to queries in an information retrieval setting. The implications of this important concept have still not played themselves out.
5. It was only after the preceding events that improvements in other database system modules were considered. These fields of research are still in their infancy.

In this article, we discuss the basics of hypermedia information management. We start by examining the nature of multimedia data and the area of multimedia data modeling, followed by a discussion of content-based retrieval.

II. THE CHARACTERISTICS OF MULTIMEDIA DATA

Multimedia data are quite different from standard alphanumeric data, both from a presentation as well as from a semantics point of view. From a presentation viewpoint, multimedia data is quite huge and has time-dependent characteristics that must be adhered to for a coherent viewing. Whether a multimedia object is preexisting or constructed on the fly, its presentation and subsequent user interaction push the boundaries of standard database systems. From a semantics viewpoint, metadata and information extracted from the contents of a multimedia object are quite complex and affect both the capabilities and the efficiency of a multimedia database system. How this is accomplished is still an active area of research.

Multimedia data consist of alphanumeric, graphics, image, animation, video, and audio objects. Alphanumeric, graphics, and image objects are time-independent, while animation, video, and audio objects are time-dependent. Video objects, being a structured combination of image and audio objects, also have an internal temporal structure which forces various synchronization conditions. A single frame of an NTSC quality video requires (512 $\times$ 480) pixels $\times$ 8 bits/pixel = 246 kb, while a single frame of an HDTV quality video requires (1024 $\times$ 2000) $\times$24 bits/pixel = 6.1 mb. Thus, at a 100:1 compression ratio, an hour of HDTV quality video would take 6.6 gb of storage, not even considering the audio portion. Utilizing a database system for presentation of a video object is quite complex, if the audio and image portions are to be synchronized and presented in a smooth fashion.

Besides its complex structure, multimedia data require complex processing in order to extract semantics from their content. Real-world objects shown in images, video, animations, or graphics, and being discussed in audio are participating in meaningful events whose nature is often the subject of queries. Utilizing state-of-the-art approaches from the fields of image interpretation and speech recognition, it is often possible to extract information from multimedia objects which is less complex and voluminous than the multimedia objects themselves and which can give some clues as to the semantics of the events being represented by these objects. This information consists of

objects called *features,* which are used to recognize similar real-world objects and events across multiple multimedia objects.

How the logical and physical representation of multimedia objects are defined and relate to each other, as well as what features are extracted from these objects and how this is accomplished are in the domain of multimedia data modeling.

III. HYPERMEDIA DATA MODELING

In a standard database system, a data model is a collection of abstract concepts that can be used to represent real-world objects, their properties, their relationships to each other, and operations defined over them. These abstract concepts are capable of being physically implemented in the given database system. Through the mediation of this data model, queries and other operations over real-world objects are transformed into operations over abstract representations of these objects, which are, in turn, transformed into operations over the physical implementations of these abstract representations. In particular, in a hypermedia data model, the structure and behavior of multimedia objects must be represented. What makes this type of data model different from a standard data model is that multimedia objects are completely defined in the database and that they contain references to other real-world objects that should also be represented by the data model. For example, the person *Bill* is a real-world object that should be represented in a data model. The video *Bill's Vacation* is a multimedia object whose structure as a temporal sequence of image frames should also be represented in the same data model. However, when *Bill* is implemented in a database by a given sequence of bits, this sequence is not actually *Bill,* who is a person. On the other hand, the sequence of bits that implements the video *Bill's Vacation* in the database *is* the actual video, or can be considered to be such. In addition, the fact that *Bill* appears in various frames of the video *Bill's Vacation* doing certain actions should also be represented in the same data model. Thus, the types of information that should be captured in a hypermedia data model include the following:

1. The detailed structure of the various multimedia objects
2. Structure-dependent operations on multimedia objects
3. Multimedia object properties
4. Relationships between multimedia objects and real-world objects
5. Portions of multimedia objects that have representation relationships with real-world objects, the representation relationships themselves, and the methods used to determine them
6. Properties, relationships, and operations on real-world objects

For images, the structure would include such things as the image format, the image resolution, the number of bits/pixel, and any compression information, while for a video object, items such as duration, frame resolution, number of bits/pixel, color model, and compression information would be included. Modeling the structure of a multimedia object is important for many reasons, not the least of which is that operations are defined on these objects which are dependent on its structure. These operations are used to create derived multimedia objects for similarity matching (e.g., image edge maps), as well as various composite multimedia objects from individual component multimedia objects (e.g., multimedia presentations).

An example of a multimedia object property is the name of the object; for example, *Bill's Vacation* is the name of a particular video object. A relationship between a multimedia object and a real-world object would be the *stars-in* relationship between the actor *Bill* and the video *Bill's Vacation.*

Suppose that *Golden Gate Bridge* is a real-world object being represented in the database and that a particular region of frame six of the video *Bill's Vacation* is known to show this object. This small portion of the byte span of the entire video is also considered to be a first-class database object, called a *semcon,* for i*con*ic data with *sem*antics, and both the *represents* relationship between this semcon and the *Golden Gate Bridge* object and the *appearing-in* relationship between the *Golden Gate Bridge* object and the video *Bill's Vacation* should be captured by the data model. Attributes of this semcon are the various features extracted from it that can be used for similarity matching over other multimedia objects. Semcons can be time-independent, as above, or time-dependent, in which case they correspond to events. See Fig. 1 for an illustration of some image semcons.

IV. INTELLIGENT BROWSING

A multimedia database with the addition of an intelligent browsing capability is known as a hypermedia database. How to accomplish intelligent browsing in a multimedia collection can best be understood

Figure 1 Some image semcons.

through the definition of a *browsing-schema,* which is nothing more than an object-oriented schema over nonmedia objects, which has undergone a transformation that will shortly be explained.

In the ensuing discussion, let us restrict ourselves to images; similar operations would apply to objects of other modalities. To transform our original object-oriented schema into a browsing-schema, we first add a class of images. Each image is actually a complex object, comprising various regions having semantic content (semcons). Similarly, each such region itself may be decomposed into various subregions, each having some semantic content. This decomposition follows the complex object structure of the nonmedia objects represented by the given regions. That is, if nonmedia object o_2 is a part of nonmedia object o_1, and o_1 has a representation r_1 appearing in some image (as a particular region), then, cases exist where r_1 would have a component r_2 that is a representation of object o_2. (This would not be the case where r_2 is occluded in the scene.) For example, a window is part of a building. Thus, the region of an image corresponding to a building may have various subregions, each of which corresponds to a window.

To the resulting schema, we now add a class of semcons. Attributes of this class are based on various extracted features such as shape, texture, and color, which are used for determining when one semcon is similar to another, and thus represents the same nonmedia object. We note that semcons as well as their attributes are considered as metadata.

To each nonmedia class, we then add a set-valued attribute *appearing-in,* which leads from each instantiation of that class to the set of images-locations where its corresponding semcon appears. We also add an attribute *represents* to the class of semcons which leads from each semcon to the nonmedia object which that semcon represents. The resultant schema is then defined as the browsing schema corresponding to the original object-oriented schema. It is now possible to view an image, specify a particular semcon within this media object, and determine information concerning the nonmedia object corresponding to this particular image region. For example, viewing an image of Professor Smith, it is now possible to navigate to a set of images containing representations of the students of Professor Smith.

Whenever viewing a particular image, the user can choose a particular semcon, *r,* for further examination. One of the actions the user can carry out is to view the value of any attribute, *a,* defined over the nonmedia object which *r* represents. This is accomplished in the browsing schema by calculating *represents(r).a.* If the value of this attribute is of a simple data type (e.g., integer, real, or string), it is textually presented to the user. If, however, this attribute's value is another (nonmedia) object, the user is allowed to browse through a set of images, each of which contains a representation of this latter nonmedia object. This approach easily generalizes to set-valued attributes. In a similar fashion, the user can follow an association (relationship). For example, if semcon, *r,* is chosen by the user and the nonmedia object *represents(r)* participates in a binary relationship with a collection, *S,* of other nonmedia objects, then the user is

allowed to browse through a set of images consisting of images which contain a representation of a nonmedia object from the collection *S*.

When a particular semcon is chosen, the user can view a scrolling menu of choices, which includes each attribute and relationship in which the nonmedia object represented by the particular semcon participates. Through the use of filtering commands, the user will be able to navigate through paths composed of many relationships and attributes and restrict the collection of media objects at the final destination. For example, choosing a particular semcon which is an image of a particular Mayan artifact, a filtering command of the form *self.type.artifacts, where self.type.artifacts.discovered = '1923',* will take the user to a collection of images which represent artifacts of the same type as the given Mayan artifact, but which were discovered in 1923.

A very important use of this approach is to navigate along a *similarity path.* Such a path proceeds from a given semcon to the set of images containing semcons similar to the given semcon. An illustration of this sort of navigation would be to proceed from an image containing some flowers to the set of all images in the database that also contain such flowers. This browsing path is not, however, mediated by the relationships *represents* and *appearing-in,* but by content-based retrieval techniques. After this is done, the user can choose to update the relations *represents* and *appearing-in,* so that future browsing can be done more efficiently. As different users view the resultant output of a content-based query in different ways, what is acceptable for one user may not be acceptable for another user. Thus, rather than globally update these two relations for all users, each user will have his own version of these relationships.

An important problem arises as to how the initial state of the browsing schema is constructed. At present, this must be done manually. Given a particular image collection, we assume the existence of a preexisting database schema that captures the various entities and their relationships. Then, for each image, semcons and their corresponding database entities must be identified. We note that some images may also be inserted into the system without manual labeling and rely on similarity path browsing to identify the semcons appearing in them.

V. IMAGE AND SEMCON MATCHING

Most existing techniques match entire images against one another. An alternative technique is to extract semcons from the query and database images and perform matching at the semcon level. This latter methodology is much more difficult, however, as finding semcons automatically is a difficult task. As mentioned later on in this section, a way around these difficulties is to decompose the image into using various fixed partitioning strategies.

Historically, image and semcon matching has consisted of developing representations for the image features of shape, color, and texture, along with appropriate distance measures. Throughout the years, different approaches have been developed for these features. This section illustrates existing techniques, while in the next section, we will present a generic approach that we have developed that captures the spatial relationships of an image's point feature map.

Shape retrieval can be categorized into exact match searching and similarity-based searching. For either type of retrieval, the dynamic aspects of shape information require expensive computations and sophisticated methodologies in the areas of image processing and database systems. So far, similarity-based shape retrieval is the most popular searching type. Extraction and representation of object shape are relatively difficult tasks and have been approached in a variety of ways. We view shape representation techniques as being in two distinct categories: *measurement-based methods,* ranging from simple, primitive measures such as *area* and *circularity* to the more sophisticated measures of various *moment invariants;* and *transformation-based methods,* ranging from functional transformations such as *Fourier descriptors* to structural transformations such as *chain codes* and *curvature scale space feature vectors.*

Jagadish in 1991 introduced the notion of a rectangular cover of a shape. Since it is restricted to rectilinear shapes in two dimensions such that all of the shape angles are right angles, each shape in the database comprises an ordered set of rectangles. These rectangles are normalized, and then described by means of their relative positions and sizes. The proposed shape representation scheme supports any multidimensional point indexing method such as the grid-file and K-D-B tree. This technique can be naturally extended to multiple dimensions. In addition to the limitations mentioned previously, the process of obtaining good shape descriptions of rectangular covers is not straightforward.

One of the first image retrieval projects was QBIC, initiated at IBM. Provided with a visual query interface, a user can draw a sketch to find images with similar sketches in terms of color, texture, and shape. A union of heuristic shape features such as area, circularity, eccentricity, major axis orientation, and some algebraic moment invariants are computed for content-based image retrieval. Since similar moments

do not guarantee similar shapes, the query results sometimes contain perceptually different matches.

Techniques exist to enable the retrieval of both rigid and articulated shapes. In these schemes, each shape is coded as an ordered sequence of interest points such as the maximum local curvature boundary points or vertices of the shape boundary's polygonal approximation, with the indexed feature vectors representing the shape boundary. To answer a shape retrieval query, the query shape representation is extracted and the index structure is searched for the stored shapes that are possibly similar to the query shape, and the set of possible similar shapes is further examined to formulate the final solution to the query.

There are techniques to recursively decompose an image into a spatial arrangement of feature points while preserving the spatial relationships among its various components. In these schemes, quadtrees can be used to manage the decomposition hierarchy and help in quantifying the measure of similarity. These schemes are incremental in nature and can be adopted to find a match at various levels of details, from coarse to fine. These techniques can also be naturally extended to higher dimensional space. One drawback of these approaches is that the set of feature points characterizing shape and spatial information in the image has to be normalized before being indexed.

One of the earliest image retrieval projects utilizing spatial color indexing methods was QBIC, mentioned above. Provided with a visual query interface, the user can manually outline an image object to facilitate image analysis in order to acquire an object boundary, and then request images containing objects whose color is similar to the color of the object in the query image. In the QBIC system, each image object is indexed by a union of area, circularity, eccentricity, major axis orientation, and some algebraic moment invariants such as its shape descriptors, along with color moments such as the average red-green-blue (RGB) values, as well as a k element color histogram. Other research groups have also tried to combine color and shape features for improving the performance of image retrieval. The color in an image has been represented by three one-dimensional color histograms in (RGB) space, while a histogram of the directions of the edge points has been used to represent the general shape information. An interesting image representation scheme is that of *blobworld*. This technique attempts to recognize the nature of images as combinations of objects so as to make both query and learning in the blobworld more meaningful to the user. In this scheme, each blob (region) in the image is described by the two dominant colors, the centroid for its location and a scatter matrix for its basic shape representation.

Though it is more meaningful to represent the spatial distribution of color information based on image objects or regions, various fixed image partitioning techniques have also been proposed because of their simplicity and acceptable performance. Images can be divided into five partially overlapped, fuzzy regions, with each region indexed by its three moments of the color distribution. One can also use the interhierarchical distance (IHD), which is defined as the color variance between two different hierarchical levels (i.e., an image region and its subregions). Based on a fixed partition of the image, an image can be indexed by the color of the whole image and a set of IHDs which encode the spatial color information. One can also partition an image into N*N blocks with each block indexed by its dominant hue and saturation values.

Instead of partitioning an image into regions, there are other approaches for the representation of spatial color distribution. Techniques exist which partition histogram bins based on the spatial coherence of pixels. A pixel is coherent if it is a part of some *sizable* similar-colored region, and incoherent otherwise. Color correlograms, as described by Huang and colleagues in 1997, have been used to index images. A color correlogram is actually a table containing color pairs, where the kth entry for $<i, j>$ specifies the probability of locating a pixel of color j at a distance k from a pixel of color i in the image.

We note that both the histogram refinement and correlogram approaches do not recognize the nature of images as combinations of objects. As for meaningful region-based image representations, two image objects are usually considered similar only if the corresponding regions they occupy overlap. Along with the position dependence of similar image objects, the fixed image partition strategy does not allow image objects to be rotated within an image. In addition, in order to check whether these image objects are in the requisite spatial relationships, even 2-dimensional-strings and its variants suffer from exponential time complexity in terms of the number of concerned image objects.

Our *anglogram-based* approach to feature-matching, described in the next section, is, we believe, a generic approach. We have used it for shape matching and color matching.

VI. A GENERIC IMAGE MODEL

Humans are much better than computers at extracting semantic information from images. We believe

that complete image understanding should start from interpreting image objects and their relationships. Therefore, it is necessary to move from image-level to object-level interpretation in order to deal with the rich semantics of images and image sequences. An *image object* is either an entire image or some other meaningful portion of an image that could be a union of one or more disjoint regions. Typically, an image object would be a semcon (iconic data with semantics). For example, consider an image of a seashore scene shown in Fig. 2, consisting of some seagulls on the coast, with the sky overhead and a sea area in the center. Examples of image objects for this image would include the entire scene (with textual descriptor *Life on the Seashore*), the seagull region(s), the sand regions(s), the water region(s), the sky region(s), and the bird regions (the union of all the seagull regions). Now, each image object in an image database contains a set of unique and characterizing features $F = \{f_1, \ldots, f_k\}$. We believe that the nature as well as the spatial relationships of these various features can be used to characterize the corresponding image objects.

In 2-D space, many features can be represented as a set of points. These points can be tagged with labels to capture any necessary semantics. Each of the individual points representing some feature of an image object we call a *feature point*. The entire image object is represented by a set of labeled feature points $\{p_1, \ldots, p_k\}$. For example, a corner point of an image region has a precise location and can be labeled with the descriptor *corner point,* some numerical information concerning the nature of the corner in question, as well as the region's identifier. A color histogram of an image region can be represented by a point placed at the center-of-mass of the given region and labeled with the descriptor *color histogram,* the histogram itself, as well as the region's identifier. We note that the various spatial relationships among these points are an important aspect of our work.

Figure 2 An image of a seashore scene.

Effective semantic representation and retrieval requires labeling the feature points of each database image object. The introduction of such feature points and associated labels effectively converts an image object into an equivalent symbolic representation, called its *point feature map*. We have devised an indexing mechanism to retrieve all those images from a given image database which contain image objects whose point feature map is similar to the point feature map of a particular query image object. An important aspect of our approach is that it is rotation, translation, and scale invariant when matching images containing multiple semcons.

A. Shape Matching

The methodology of our proposed shape representation for image object indexing is quite simple. Within a given image, we first identify particular image objects to be indexed. For each image object, we construct a corresponding point feature map. In this study, we assume that each feature is represented by a single feature point and that a point feature map consists of a set of distinct feature points having the same label descriptor, such as *Corner Point*. After constructing a Delaunay triangulation of these feature points of the point feature map, we then compute a histogram that is obtained by discretizing the angles produced by this triangulation and counting the number of times each discrete angle occurs in the image object of interest, given the selection criteria of what bin size will be, and of which angles will contribute to the final angle histogram. As the nature of our computational geometry-based shape representation consists of angle histograms, we call the shape index a *shape anglogram*. For example, the shape anglogram can be built by counting the two largest angles, the two smallest angles, or all three angles of each individual Delaunay triangle with some bin size between 0 and 90°. An $O(max(N, \#bins))$ algorithm is necessary to compute the shape anglogram corresponding to the Delaunay triangulation of a set of N points.

Our idea of using an anglogram to represent the shape of an image object originates from the fact that if two image objects are similar in shape, then both of them should have the same set of feature points. Thus, each pair of corresponding Delaunay triangles in the two resulting Delaunay triangulations must be *similar* to each other, independent of the image object's position, scale, and rotation. In this study, corner points, which are generally high-curvature points located

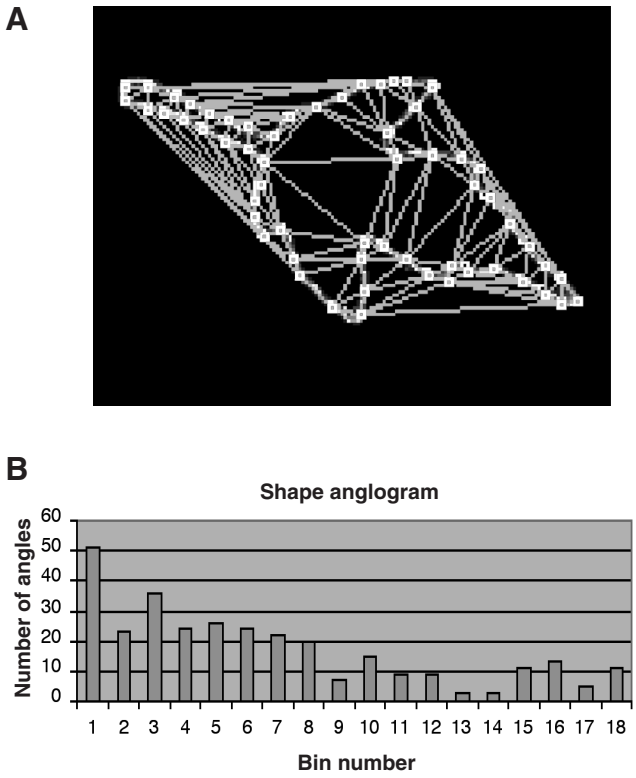


Figure 3 (A) Delaunay triangulation of a leaf and (B) resulting shape anglogram.

along the crossings of an image object's edges or boundaries, will serve as the feature points for our various experiments. We have previously argued for representing an image by the collection of its corner points. This approach proposes an interesting technique for indexing such collections provided that the image object has been normalized. In our present approach, which is histogram-based, the image object does not have to be normalized. This technique also supports an incremental approach to image object matching, from coarse to fine, by varying the bin sizes.

Figure 3a shows the resulting Delaunay triangulation produced from the point feature map characterizing the shape of the image object, *leaf,* in which corner points serve as the feature points. Figure 3b shows the resulting shape anglogram built by counting all three angles of each individual Delaunay triangle, with a bin size of 10°.

B. Color Matching

Digital images can be represented in different color spaces such as RGB, HSI, YIQ, or Munsell. Since a very large resolution of millions of colors is unwanted for image retrieval, the color space is usually quantized to a much coarser resolution. For example, hue-saturation-intensity (HSI) color space is designed to resemble the human perception of color in which hue reflects the dominant spectral wavelength of a color, saturation reflects the purity of a color, and intensity reflects the brightness of a color. It is noted that humans are less sensitive to differences in either saturation or intensity than to differences in the hue component, so that, in general, hue is quantized more finely than the saturation or intensity component for image retrieval when HSI is used for image representation. As the process of grouping low-level image features into meaningful image objects and then automatically attaching semantic descriptions to these image objects is still an unsolved problem in image understanding, our work intends to combine both the simplicity of fixed image partition and the nature of images as combinations of objects into spatial color indexing so as to facilitate image retrieval.

Based on the assumption that salient image constituents generally tend to occupy relative homogeneous regions within an image, we expect that one or more meaningful image constituents may be composed of some image blocks with a particular color. Regardless of whether these image blocks are connected or not, they approximate the composition of the nature of images as combinations of objects. In our spatial color-indexing scheme, an image is first divided evenly into a number of *M*N* nonoverlapping blocks. Then each individual block is abstracted as a unique feature point labeled with its spatial location and dominant colors. After we adjust all two neighboring feature points to a fixed distance, all the normalized feature points form a *point feature map* of the original image for further analysis.

By representing an image as a point feature map, we capture not only the color information of the image, but also the spatial information about color. We can flexibly manipulate sets of feature points instead of dealing with image blocks. In order to compute our spatial color index of an image, we construct a Delaunay triangulation for each set of feature points in the point feature map labeled with the identical color, and then compute the feature point histogram by discretizing and counting the angles produced by this triangulation. An O*(max(N, #bins))* algorithm is necessary to compute the feature point histogram corresponding to the Delaunay triangulation of a set of *N* points. The final image index is obtained by concatenating all the feature point histograms together. We note that in our spatial color-indexing scheme, feature point histograms are not normalized, as a

drawback of normalized histograms is its inability to match parts of image objects. For example, if region A is a part of region B, then, in general, the normalized histogram H_A is no longer a subset of the normalized histogram H_B.

An example is shown in Fig. 4. Figure 4a shows a pyramid image of size 192*128; by dividing the image evenly into 16*16 blocks, Fig. 4b and Fig. 4c show the image approximation using dominant hue and saturation values to represent each block, respectively. Figure 4d shows the corresponding point feature map perceptually, and we note that the distance between any two neighboring feature points is fixed, as images of different sizes undergo normalization. Figure 4e highlights the set of feature points labeled with hue 2, and Fig. 4f shows the resulting Delaunay triangulation. Figure 4g shows the resulting Delaunay triangulation of a set of feature points labeled with saturation 5, and Fig. 4h shows the corresponding feature point histogram obtained by counting only the two largest angles out of each individual Delaunay triangle with bin size of 10°. Our work has concluded that such a feature point histogram provides a sufficient and effective way for image object discrimination.

Histogram intersection was originally proposed for comparing color histograms of query and database

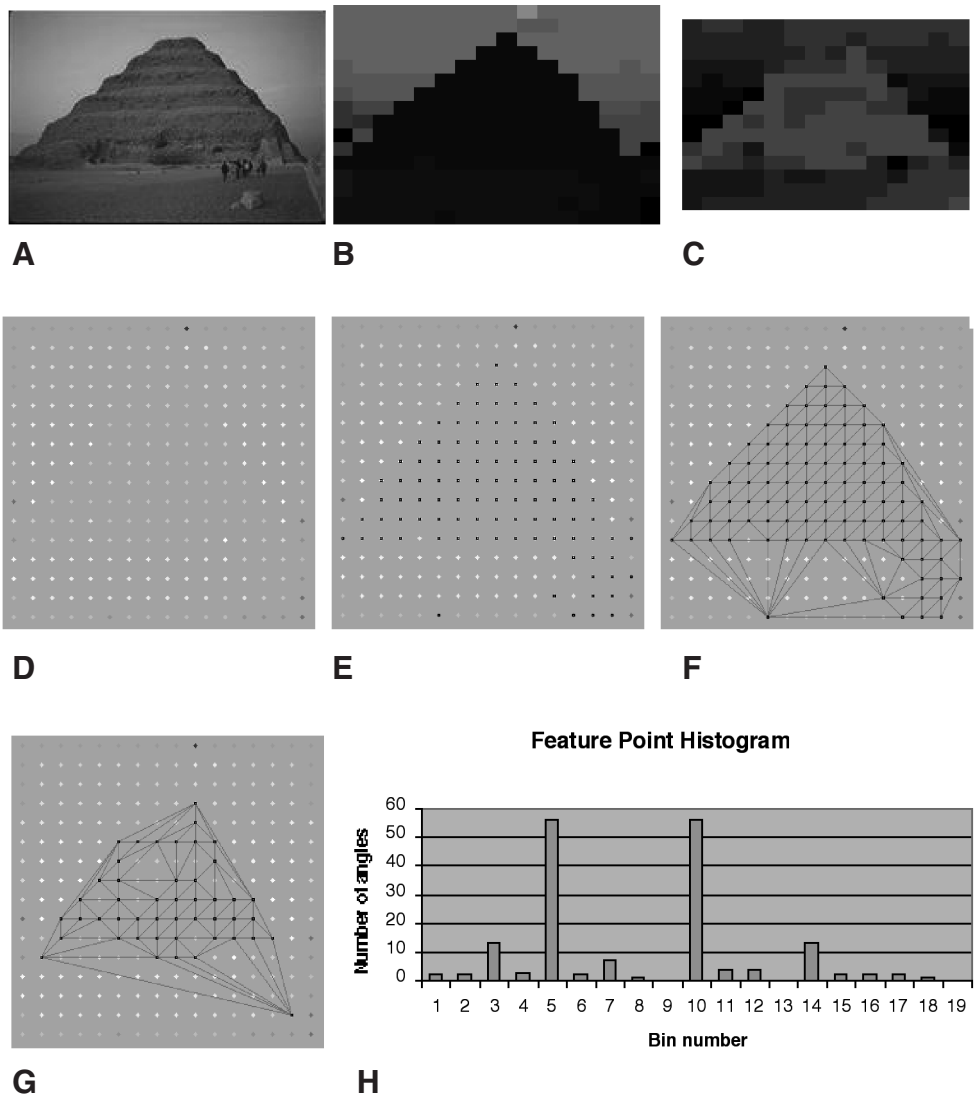


Figure 4 (A) A pyramid image; (B) hue component; (C) saturation component; (D) point feature map; (E) feature points of hue 2; (F) Delaunay triangulation of hue 2; (G) Delaunay triangulation of saturation 5; and (H) resulting feature point histogram of saturation 5.

images. It was shown that histogram intersection is especially suited to comparing histograms for recognition. Additionally, histogram intersection is an efficient way of matching histograms, and its complexity is linear in the number of elements in the histograms. The intersection of the histograms I_{query} and $M_{database}$, each of n bins, is defined as follows.

$$D(I_{query}, M_{database}) = \frac{\sum_{j=1}^{n} \min(I_j, M_j)}{\sum_{j=1}^{n} I_j}$$

Suppose that Q is the query image index consisting of m color-related feature point histograms, Q_1, $Q_2, \ldots, Q_m$, DB is the database image index with corresponding m color-related feature point histograms $DB_1, DB_2, \ldots, DB_m$ and w_i is the jth of m of variables which define the relative importance of color-related feature point histograms in our similarity calculation. For example, if HSI is used for image representation, hue-related feature point histograms are often assigned a larger weight value than saturation-related ones, as humans are more sensitive to hue variation. The similarity measure function used in this study is histogram intersection-based; it is given below.

$$Distance(Q, DB) = \frac{\sum_{i=1}^{m} w_i D(Q_i, DB_i)}{\sum_{i=1}^{m} w_i}$$

Each $D(Q_i, DB_i)$ uses histogram intersection to obtain a fractional value between 0 and 1. Before being normalized by the number of angles in the query image, the result of histogram intersection is the number of angles from the database image that have the same corresponding angles in the query image. Therefore, we can meaningfully think about the spatial color index of an image. Any nonzero feature point histogram represents some image objects of a particular color, while any all-zero feature point histogram, called an *empty* histogram, means that there are no image objects of that color. Based on the histogram intersection-based similarity function, the comparison of query and database images using spatial color indices can be taken as a query-by-objects-appearing.

VII. THE SEMANTIC GAP

Existing management systems for image collections and their users are typically at cross purposes. While these systems normally retrieve images based on low-level features, users usually have a more abstract notion of what will satisfy them. Using low-level features to correspond to high-level abstractions is one aspect of the *semantic gap* between content-based system organization and the concept-based user. Sometimes, the user has in mind a concept so abstract that he himself doesn't know what he wants until he sees it. At that point, he may want images similar to what he has just seen or can envision. Again, however, the notion of similarity is typically based on high-level abstractions, such as activities taking place in the image or evoked emotions. Standard definitions of similarity using low-level features generally will not produce good results. For all users, but especially for the user who doesn't know what he wants until he sees it, the efficiency of the system will likely be improved if it supports intelligent browsing so that the user will be satisfied in the shortest amount of time. It is our belief that intelligent browsing should be mediated by the paradigm of image similarity as well as by an appropriate organization of metadata, including annotations and self-describing image regions.

We characterize content-based retrieval systems that try to capture user semantics into two classes: *system-based* and *user-based*. System-based approaches either try to define various semantics globally, based on formal theories or consensus among domain experts, or use other techniques, not based on user-interaction, to get from low-level features to high-level semantics. User-based approaches, on the other hand, are adaptive to user behavior and try to construct individual profiles. An important component of most user-based approaches is the technique of relevance feedback.

Numerous examples of system-based approaches exist, from allowing the user to compose various features which evoke certain emotions, to using textual information close to an image on a web page to derive information regarding the image's contents, to exploring a heterogeneous clustering methodology that overcomes the single-feature matching drawback of having images that are similar have different semantics.

Approaches that depend on some form of user interaction include those that define a set of queries that correspond to a user concept, a system that learns how to combine various features in the overall retrieval process through user feedback, and an exploration paradigm based on an advanced user interface simulating 3-dimensional space. In this space, thumbnail images having the same user semantics are displayed close to each other, and thumbnails that are far from the user's semantic view are smaller in size

than thumbnails that are closer to the user's semantic view. The user can also convert images that are close to each other into a concept and replace the given set of thumbnails by a concept icon.

There have been many papers that generalize the classical textually based approach to relevance feedback to the image environment. Using the vector-space model for documents and queries, textually based relevance feedback transforms the *n*-dimensional point corresponding to a query based on user feedback as to which of the documents returned as the query result are relevant and which are nonrelevant. While the query is changed, the similarity measure used remains the same.

A similar approach can be implemented for content-based image retrieval using several techniques. These approaches differ in the way the query vector is changed. In one approach, positions in the vector representation of an image correspond to visual keywords. This approach is similar to that used for text. In another approach, the query vector changes, either because different feature extraction algorithms are being used for the same features, or different features are being used altogether. For example, color features can be extracted using many different approaches, such as global color histograms, local color histograms, and anglograms. Based on user feedback, the system can discover that one approach is better than the others. It may also discover that texture features are better for a particular query than color features. Then, there is a completely different approach, where the matching function is changed to give different weights to the given features. For example, through user feedback, the system may decide to give more weight to color than to texture. The MARS project, described by Rui, Huang, Ortega and Mehrotra in 1998, has examined many of these approaches throughout the last few years.

For textual information, the technique of latent semantic analysis has often been applied for improved semantic retrieval. This technique reduces the dimensionality of the document vectors by restructuring them. Each new attribute is a linear combination of some of the old attributes. Based on the co-occurrence of keywords in documents, this technique forms *concepts* from the collections of the old attributes. The result is that when a keyword, *kw*, is included in a query, documents which have the keywords from the same concept as *kw* may also be retrieved, whether *kw* is mentioned in the query or not.

Various techniques for latent feature discovery have been developed for text collections. These include latent semantic indexing and principal component analysis. There has not been much work on using these techniques for image collections. The only work of which we are familiar that intentionally uses such dimensional reduction techniques for images and text, that of La Cascia, Sethi, and Sclaroff in 1998, does so to solve a completely different problem. The environment of this work is that of web pages containing images and text. Instead of a term-document matrix, they define a term-image matrix, where the terms are taken from the text that appears close to the given image. Terms that appear closer to the given image are weighted higher than terms appearing further away. It is this term-image matrix that is used to discover latent features. An image feature vector is then comprised of components, one component representing various image features and another component representing the column vector corresponding to the given image from the transformed term-image matrix. This does not, however, solve the problem of trying to find different image features that co-occur with the same abstract concept, which would be of tremendous help in discovering the underlying semantics of images.

VIII. COMMERCIAL SYSTEMS FOR HYPERMEDIA INFORMATION MANAGEMENT

In the past, there were some heated discussions among researchers in the multimedia computing and database communities as to whether the then current database systems were sufficient to manage multimedia information. On balance, people in multimedia computing were of the opinion that advances needed to be made in the database arena in order to manage this new type of data, whereas people in databases seemed to feel that the newer database architectures were sufficient to the task. Database architectures have surely changed from then to now, but there should be no argument that no existing database system contains all of the advanced options discussed in this article. Be that as it may, currently, there are at least three commercial systems for visual information retrieval (Excalibur Technologies, www.excalib.com; IBM, www.ibm.com; Virage, www.virage.com) and several commercial database systems at various levels on the object-relational scale (DB2 Universal Database, www.ibm.com; Oracle, www.oracle.com) that can manage multimedia information at an acceptable level. However, what is acceptable by today's standards will surely not be acceptable by tomorrow's standards.

In order for database systems to handle multimedia information efficiently in a production environment, some standardization has to occur. Relational systems are efficient because they have relatively few standard operations, which have been studied by many database researchers for many decades. This has resulted in numerous efficient implementations of these operations. Blades, cartridges, and extenders for multimedia information are at present designed in a completely ad hoc manner. They work, but no one is paying much attention to their efficiency. Operations on multimedia must become standardized and extensible. If the base operations become standardized, researchers can devote their efforts to making them efficient. If they are extensible, complex operations can be defined in terms of simpler ones and still preserve efficiency. Hopefully, the efforts being devoted to MPEG-7 will address this concern.

ACKNOWLEDGMENT

Portions of this article are reproduced with permission from Rudas and Kaynak (2000), Techniques in soft computing and their utilization in mechatronic products. In *Mechatronic Systems Techniques and Applications,* Vol. 5, (C.T. Leondes, ed.), Gordon and Breach Science Publishers, Copyright Overseas Publishers Association N.V.

SEE ALSO THE FOLLOWING ARTICLES

Database Administration • Database Development Process • Desktop Publishing • Distributed Databases • Multimedia

BIBLIOGRAPHY

Berry, M. W., Drmac, Z., and Jessup, E. R. (1999). Matrices, vector spaces, and information retrieval. *SIAM Review,* Volume 41, Number 2, 335–362.

Deerwester, S., Dumais, S. T., Furnas, G. W., *et al.* (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science,* Volume 41, Number 6, 391–407.

Grosky, W. I. (1994). Multimedia information systems. *IEEE Multimedia,* Volume 1, Number 1, 12–24.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image indexing using color correlograms. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* San Juan, Puerto Rico, pp. 762–768.

Jagadish, H. V. (1991). A retrieval technique for similar shapes. *Proceedings of the 1991 ACM SIGMOD,* Denver, Colorado, May 29–31, pp. 208–217.

Jain, R. (1993). NSF Workshop on visual information management systems. *Sigmod Record,* Volume 23, Number 3, 57–75.

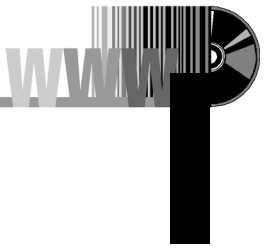
La Cascia, M., Sethi, S., and Sclaroff, S., (1998). Combining textual and visual cues for content-based image retrieval on the World Wide Web. *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries,* Santa Barbara, CA, 24–28.

Niblack, W., Barder, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., Faloutsos, C., and Yaubin, G. (1993). The QBIC Project: Querying images by content ising color, texture, and shape. *Proceedings of SPIE Storage and Retrieval for Image and Video Databases,* Vol. 1908, 173–181.

Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S. (1998). Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology,* Volume 8, Number 5, 644–655.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *Transactions on Pattern Analysis and Machine Intelligence,* Volume 22, Number 12, 1349–1380.

Tao Y., and Grosky, W. I. (2001). Spatial color indexing using rotation, translation, and scale invariant anglograms. *Multimedia Tools and Applications,* Volume 15, Number 3, 247–268.



Industry, Artificial Intelligence in

Lakhmi C. Jain
University of South Australia

Zhengxin Chen
University of Nebraska, Omaha

GLOSSARY

expert systems Knowledge-based systems that demonstrate expert-level knowledge for problem solving in a specific knowledge domain.
heuristics Problem-solving techniques using subjective knowledge, hunches, trial and error, rule of thumb, and other such informal but generally accurate methods.
heuristic search Searching for a solution in the state space using heuristics to improve efficiency.
knowledge representation Representing knowledge in the problem domain using certain schemes (such as using predicate logic) so that reasoning can be performed.
soft computing Computing that is tolerant of imprecision, uncertainty, and partial truth. It refers to the discipline situated at the confluence of distinct methodologies, including fuzzy logic, neural network, and probabilistic reasoning (including evolutionary algorithms, chaos theory, causal networks).
self-adaptive system Computer software that is able to evaluate its own behavior and change its behavior when the evaluation indicates that is not accomplishing what the software is intended to do, or when better functionality or performance is possible.

ARTIFICIAL INTELLIGENCE (AI) in industry is a typical showcase of how AI has evolved and contributed to the modern technology and economy. Much of the research in AI and industry is sponsored at least in part by government organizations. In this article, we examine the contribution and role of AI in industry. Our examination roughly consists of three parts: we start with an overview of the relationship between AI and industry; we examine how basic AI concepts are used in industry; and how industry has stimulated applied research in AI. Specific issues discussed include constraints. In the second part of this article, we present several case studies. This includes a discussion on the methodology used in AI in industry, as well as an examination of several typical sectors within AI industry where AI techniques have played a significant role. Besides a brief discussion on expert systems, we discuss neural networks, genetic algorithms, as well as some other soft computing techniques. In the third part of this article, we discuss some ongoing research projects as well as new frontiers of AI in industry. The last part of this article contains some comments, observation, some future research directions, and discussions on AI in industry as a whole. We notice that applied AI has been an important driving force for AI research in dealing with real-world problems, which is in sharp contrast of early AI research where toy problems were used to examine basic concepts. The focus of this article is not to provide a complete coverage of AI in industry, rather, the representative techniques underlying many industry AI applications.

I. OVERVIEW

A. Basics AI Concepts in Industry Applications

Some basic concepts of AI have been reviewed in the overview article of this volume. Basic concepts of AI,

such as symbolic reasoning, heuristic search, and knowledge representation have been widely applied in industry. Machine learning techniques founded on heuristic search and knowledge representation have also been widely applied. However, in many cases, AI in industry should not be viewed as direct applications of AI principles. For example, typical languages used in AI, such as LISP and Prolog, are not necessarily adopted in AI projects in industry, for realistic concerns. Rather, C, C++, or higher-level languages or tools (such as OPS5 or CLIPS) are widely used in many applications. Nevertheless, AI in industry does apply basic principles of AI and has sent numerous feedbacks to motivate academia for applied research.

A very important issue that needs to be addressed in industry applications is uncertainty, as encountered in many real-world situations, such as dealing with missing, inconsistent, and incomplete data. Techniques based on traditional reasoning methods based on symbolism may not be powerful enough to deal with these situations. An important category of reasoning under uncertainty in AI employs artificial neural network techniques, which follows the philosophy of subsymbolic reasoning. Neural networks have widened the scope of AI in that emulation of human intelligence is no longer restricted at a psychological level; rather, emulation at a biological level has largely extended the vision of AI research. Genetic algorithms are another important example of simulating human intelligence at the biological level. In addition, practitioners in industry have also absorbed nutrition from areas outside of AI as well. A typical example is fuzzy logic. Although not born as an AI technique, fuzzy logic has been successful as an effective tool for aiding reasoning under uncertainty in many industry applications. It is interesting enough to point out that these methods (neural networks, genetic algorithms, fuzzy logic, and other reasoning techniques) are sometimes referred to as soft computing. Another term, computational intelligence, has also been used to emphasize many new aspects which are not emphasized in the original symbolic reasoning as studied in traditional AI.

An important remark that must be made here is related to scale-up. In the early history of AI, various state space search techniques have been demonstrated in various toy problems, but failed to deal with the huge amount of states generated in practical applications. AI practitioners had to turn to power of knowledge. Knowledge-based systems and expert systems have made significant progress in industry applications. By taking advantage of domain-specific knowledge, heuristic search techniques have arrived at a new level of importance. In addition to symbolic reasoning traditionally discussed in AI, subsymbolic reasoning and other techniques (including those developed earlier under the umbrella of pattern recognition, operations research, as well as other related fields) have proliferated.

From a practical perspective, we agree with Drew McDermott's observation that AI's success arises not from sophisticated representation and reasoning methods, but rather from simple representations and tractable algorithms, where complexity is mastered in the process of formulating a problem solving to which such methods can be applied. Frigenbaum has further turned the focus around from knowledge representation to knowledge itself, arguing that when AI has had success in complex reasoning tasks, it has typically depended on encoding and exploiting knowledge of the domain.

We discuss several aspects to illustrate the theoretical foundation behind AI applications in industry.

B. Some AI Techniques Successfully Used in Industry Applications

Below we examine some AI techniques which have found successful applications in industry.

1. Constraint-Based Search

As a particular case of the application of heuristic state-space search, let us take a look at *constraint-based reasoning*. Many industrial scheduling problems use constraints, especially problems that require fine-grained scheduling (which combines AI with operations research). Personnel scheduling and transportation problems are two industrial applications where constraint technology is increasingly the technology of choice. Constraint-satisfaction methods can successfully navigate enormous search spaces. In some cases, we can apply formal complexity results to the constraint model to prove tractability. However, in general, constraint problems are formally intractable, and we have to solve them by a judicious combination of algorithms and heuristics. Specialized languages and packages can greatly reduce programming effort. Constraint programming is a method of solving highly combinatorial problems given a declarative problem description and a general constraint-propagation engine. This technology, developed in the 1990s, has very successfully solved production scheduling and resource-assignment problems.

As an example of constraint-based reasoning, constraint logic programming can help detect buildings

from the air. The recognition system developed in this application uses constraint relaxation and variable elimination to handle uncertainty and unobservability of building parts. The application is the identification of building in aerial images. Heuristics have been applied in the two (successive) reasoning phases of constraint placement and variable instantiation. In both phases, the heuristics determine the order in which the constraints and variables are processed. Comparing search control heuristics is used in other applications as well.

2. Building Intelligent Agents

An intelligent agent is a software entity which functions continuously and autonomously in a particular environment, often inhabited by other agents and processes. Software agents can also be viewed as intelligent agents due to their intelligent behavior. Frequently cited aspects of intelligent agents include proactive (the ability to take the initiative), reactivity (the ability to selectively sense and act), autonomy (goal-directness, proactive, and self-starting behavior), collaborative behavior, "knowledge-level" communication ability, inference capability, temporal continuity, personality, adaptivity, and mobility, as well as others. Historically, intelligent agents have been motivated from simplifying distributed computing and overcoming user interface problems. The need for diverse, highly sophisticated, and rapidly changing skills and knowledge, as required in many industrial applications, makes the multiagent paradigm particularly appropriate for knowledge-based design. An agent communication language (ACL) refers to four different key components: the performative, service, content, and control add-ins levels. As for agent functionality, there are also four categories: user-centric agents, problem-solving agents, control agents (exclusive to multiagent systems, primarily provides control services to other agents), and translation agents (which provide a bridge between systems using different data standards). In addition, web-based agents show a great potential for design and engineering applications. To integrate industry-applications-oriented agents into the Web, various issues should be resolved, including resolving the conflict between HTTP's client/server protocol and the peer-to-peer protocol required by agents.

3. Qualitative Reasoning

Broadly speaking, qualitative-reasoning research aims to develop representation and reasoning techniques that will enable a program to reason about the behavior of physical systems, without the kind of precise quantitative information needed by conventional analysis techniques such as numerical simulators. Inferences can be made with much less information than detailed quantitative analysis would require. Advantages over conventional numerical simulation include coping with incomplete information, imprecise but correct prediction, easy exploration of alternatives, and automatic interpretation. Basic qualitative reasoning techniques include qualitative arithmetic, qualitative simulation, knowledge representation, and model formulation and abstraction. As an example of real-world applications, a gas turbine monitoring system has been developed in Scotland. It uses a qualitative model of the turbine system dynamics for prediction and diagnosis.

II. TYPICAL AI TECHNIQUES USED IN INDUSTRY

A. Methodology and Techniques Useful in Industry

1. Expert Systems (and Knowledge-Based Systems) in Industry

Expert systems (and more generally, knowledge-based systems) solve problems by taking advantage of domain-specific knowledge. As a useful aid for expert system development, CLIPS (an acronym for C language integrated production system) is a multiparadigm programming language that provides support for rule-based, object-oriented, and procedural programming. Further, CLIPS is a forward-chaining rule-based production system language based on the Rete algorithm for pattern-matching. A command-line interpreter is the default interface for CLIPS. The CLIPS programs are expressed with commands, functions and constructs. In CLIPS a fact is an ordered list of fields; it also supports template (or non-ordered) facts. Rules allow the user to specify a set of conditions to CLIPS such that, when the conditions in the left-hand side (LHS) are satisfied, a set of actions in the right-hand side (RHS) are executed. A brief discussion on CLIPS and its applications can be found in Chen (2000), and a simple example will be discussed in a later section of this article.

2. Soft Computing Techniques

In the following we provide a brief discussion on some soft computing techniques. For more detailed discussion on these issues, see Chen (2000).

a. Fuzzy Logic

Fuzzy sets deal with *graduality* of concepts and describe their boundaries and have nothing to do with frequencies (repetition) of an event. The term fuzzy logic was originally coined to refer to multivalued logic (in contrast to standard logic which is two-valued—true or false, nothing else). A fuzzy set may be regarded as a class in which there is a graduality of progression from membership to non-membership or, more precisely, in which an object may have a grade of membership intermediate between unity (full membership) and zero (nonmembership). Fuzzy logic, when applied to decision-making problems, provides formal methodology for problem solving and incorporates human consistency, which are important characteristics required by fuzzy decision-making systems. Such systems should possess the following functionality:

1. Explain the solution to the user
2. Keep a rigorous and fair way of reasoning
3. Accommodate subjective knowledge
4. Account for "grayness" in the solution process

Some recent research results have been collected in Jain et al. (1999) from some of the more recent applications of knowledge-based signal processing. Mechanisms have been developed to overcome difficult or previously intractable problems in industries where the payoff can be significant. In general, the problems described are those of large, complex, distributed systems and the techniques that have been developed to deal with them are oriented to obtaining results. In general, the primary task has been to make automated systems more human-like; the primary technique employed has been fuzzy logic.

b. Neural Networks

The basic feature of neural networks is that they de-emphasize the use of symbols to denote objects and relations; intelligence is viewed as arising from the collective behavior of large numbers of simple, interacting components. The terms subsymbolism and connectionism are used to describe this basic feature. The nodes and weights in a neural network demonstrate a distributed knowledge representation. The architecture of neural networks makes itself suitable for machine learning. In fact, in a neural network, learning is carried out by adjusting weights.

c. Evolutionary Algorithms

Evolutionary algorithms refer to learning algorithms patterned after the processes underlying evolution—shaping a population of individuals through the survival of its most fit members. The power of selection across a population of varying individuals has been demonstrated in the emergence of species in natural evolution, as well as through the social processes underlying cultural change. As an example of evolutionary algorithms, let us take a look at genetic algorithms. A genetic algorithm (GA) is one that seeks to improve upon the quality of a problem solution through a process that mimics that of natural selection and adaptation of species in nature. The method begins with a set of initial solutions (called an initial population) to a complex problem. It then crosses and mutates selected solutions from this initial set to develop a new set of solutions. The procedure is repeated to create successive generations of solutions until a predefined stopping criterion is met. This criterion may be based on a threshold value for error expended, solution quality, or a combination.

The philosophy behind the method is that if we conduct a search for a better answer simultaneously from multiple locations within a complex space of solutions, we stand a better chance of locating the globally optimal solution. Consequently, the GA approach has generally been advocated for complex, multimodal solution space where there is high likelihood of a search strategy being trapped at a local optima. The probability of entrapment is generally higher for methods that localize their search efforts to a specific region of the solution space.

d. Rough Sets

The following is a very brief sketch on the rough set approach which is becoming increasingly important in soft computing, machine learning, and data mining. The basic idea of rough set theory can be outlined as follows. For the rows, the rough set theory employs the notion of *indiscernibility* class to group similar tuples (rows) together; while for the columns, it employs the notion of *indispensable* attributes to identify the significance of attributes.

The rough set approach starts from the analysis of limits of discernibility of a subset of objects belonging to the domain. For a set X, we define its lower approximation $\underline{R}X$ (which is a union of Xs all containing subsets and upper approximation $\overline{R}X$ (which is a union of all subsets in which X is contained). In rough set approach, a set X is defined in terms of definable sets in A by using $\underline{R}X$ and $\overline{R}X$. Thus we can decide if x is in X on the basis of a definable set in A rather than on the basis of X; we deal with $\underline{R}X$ and $\overline{R}X$ instead of X itself. Furthermore, based on these two concepts, the concept of *rough set* can be defined as the family of all subsets of U having the same lower and upper approximations in A.

It is important to study the dependency of attributes. An issue in the analysis of dependencies among attributes is the identification and information-preserving reduction of redundant conditions. Another concept is the *minimal set* of attributes: each minimal set can be perceived as an alternative group of attributes that could be used to replace all available attributes. The main challenge is thus how to select an optimal reduct.

In some practical problems, it is often necessary to find the subset of attributes contained in all reducts, if one exits. The attributes contained in all reducts are also contained in the reduct that represents the real cause of a cause-effect relationship. The intersection of all minimal sets is called the *core*.

When the rough set approach is applied on decision tables, production rules (of the "if . . . then" format) can be induced from the reduced set of condition and decision attributes. In general, the computation involved in the lower approximation will produce *certain* rules while the computation involved in the upper approximation will produce *possible* rules.

B. Types of AI Techniques and Sectors of AI Applications in Industry

1. Types of Some Common AI Techniques

There are various types of problems across industries where AI can provide valuable help. Consequently, various AI techniques have been used in various sectors in industry:

a. Intelligent Control

Control is the act of affecting a dynamic system to accomplish a desired behavior. Intelligent control refers both to the control design approach or philosophy and to implementation techniques that emulate certain characteristics of intelligent biological systems. Intelligent control ranges from control architectures, hierarchies, and distribution to learning, expert systems, neural networks, fuzzy systems, and genetic algorithms. Applications include various kinds of robotic problems, automated highway systems, submarine control, space-based process control, and the use of adaptive networks to model and control drug delivery, as well as others.

b. Intelligent Scheduling

Scheduling complex processes, such as chemical manufacturing or space shuttle launches, is a focus of substantial effort throughout industry and government. In the past 20 years, the fields of operations research and operations management have tackled scheduling problems with considerable success. Recently, the AI community has turned its attention to this class of problems, resulting in a fresh corpus of research and application that extends previous results.

c. Product Configuration

The manufacturing trend toward mass customization has awakened great interest in automatic product configuration techniques. Informally, configuration is a special case of design activity with two key features:

1. The artifact being configured is assembled form instances of a fixed set of well-defined component types.
2. Components in the artifact interact with each other in predefined ways.

The advantage of configuration is obvious. Producing a specific design for each customer is not economical. Instead, producers use standardized sets of parts that can be configured into products satisfiying a wide range of requirements. Configurations must be correct, produced quickly, and must be optimal. The salesPLUS product-configuration tool effectively solves complex configuration problems, reducing costs while meeting customer expectations in real-world applications. The product has been successfully applied to many applications, such as the discrete manufacturing of pumps, robots, and vehicles; telecommunications; and high-tech services. More discussions can be found in Faltings and Freuder (1998).

2. Sectors of AI Applications in Industry

The entire field of industry is divided into several sectors for comparative studies, such as: planning and scheduling; project management; system integration; and diagnosis and troubleshooting.

The following is a categorization on AI applications in industry adopted and revised from Rauch-Hindin (1985). Note that some overlaps may exist between different categories.

1. *Planning and scheduling.* Tasks in this category include a series of steps starting with customer or company specification, and going through product design, planning of the manufacturing process, making and assembling a product, and selling and distributing it.
2. *Project management, factory monitoring, long-term planning, and knowledge management.* Tasks in this

category include to automatically sense information, model the organization, provide expert assistance, schedule, monitor, answer status questions, and analyze how the structure and processing of the organization should be changed to optimize such criteria as cost, throughput, and quality. Recent development of knowledge management has significantly enhanced the integration of these tasks. Here we take a brief look at knowledge-based project management. Most often, project management is performed manually by people, with aid from computerized techniques. AI offers manufacturers the ability to conserve human resources, especially as they become less available, and to amplify and leverage the abilities of those who are available. For example, production and managerial supervisors need to monitor and control all the machines, orders, and activities in a factory. Factory monitoring knowledge systems have been designed to give supervisors dynamic access to the factory floor.

3. *Design, simulation and manufacturing*. These are related, but different tasks:
   - Knowledge-based systems and other AI techniques can be constructed to act as a designer's assistant or *design* consultant.
   - AI-assisted *simulation* techniques can make design processes better achieve their goals.
   - Three characteristics have made *manufacturing* activities a prime area for the application of AI: manufacturing is a semantically rich domain; large amounts of data are readily available (thus making interesting connection with database management systems possible); and the increase in hardware speed and advances in algorithms have made it possible to solve problems in a fraction of the time it previously took. For example, a constraint-oriented cooperative scheduling approach for aircraft manufacturing has been studied. This is a human-machine cooperative framework where computers check the consistency of thousands of constraints and humans select relevant choices.
4. *Diagnosis and troubleshooting*. The tasks related to diagnosis are difficult, because they require knowledge of the equipment and how it operates normally, knowledge about failed equipment and its fault symptoms, knowledge of explaining the faults, and ability to form necessary hypotheses and perform some tests to confirm, revise or deny the hypothesis. Knowledge-based systems offer an effective approach to preserve and protect a troubleshooter's expertise and make that kind of expertise available to many people. The idea is to encode as much as possible of the expert's past experience, judgments, and decision-making techniques in a knowledge-based representation that can be understood and manipulated by a computer that is accessible by other troubleshooters. In this manner, the knowledge and skills of the expert can be used to provide advice to, amplify the knowledge of, and leverage the more average and less experienced repair personnel.

3. AI in Civil and Structural Engineering

Designing a structure involves many iterations of several knowledge-intensive steps, due to its development cycle:

- Designers determine the structure's functional requirements.
- Designers make an effort to understand the environments.
- Designers model various alternatives for the structure and make predictions.
- Designers develop more detailed design of a few alternatives.
- Construction companies use their expertise to predict issues related to physically realize the design.
- Construction company managers deal with a dynamic construction process (such as to deal with unexpected problems and changing economic conditions).
- Owners monitor the structure for signs of deterioration, maintain it when deterioration occurs, and modify it to accommodate new functions.

AI techniques have been widely used in civil and structural engineering. Applications including knowledge-based assistance for finite-element modeling, use of shared graphics for conceptual and collaborative building design, development and use of neural networks for predicting the amount of springback in a reinforcement bar, development of decision support systems to help monitor dams and monuments and to assess the seismic risk of buildings, and development of system to help process-plant owners perform plant maintenance only as needed.

4. AI in Steel Industry

Steel production involves a number of states, such as melting, casting, rolling, and forging, that entail

complex chemical and thermic reactions as well as intricate mechanical operations. The steel industry must deal with incomplete and uncertain data. Nearly all steelmakers worldwide now use expert systems, fuzzy logic, and neural nets to improve quality assurance and production efficiency. For many years, the steel industry's main objective has been to maximize production by automating processes and streamlining plan organization. Ongoing research into new steel qualities has produced a broad range of products, which present many new control problems. Although other industries reflect the same tendency toward processing in smaller lot sizes, the steelmaking environment shows more diversity than most because of the particular characteristics of its material and manufacturing technology. In addition, the capital-intensive nature of the industry can make unanticipated violations of technological constraints extremely costly. Some typical factors are:

- Most steelmaking processes are temperature-sensitive. Since chemical reactions depend on temperature, any loss of heat during processing may degrade the steel's quality.
- Steelmakers apply expert systems instead of conventional software because the controlling software has to reason with existing uncertainties and master the inherent complexity of typical control problems. For example, the main focus of the blast furnace quality improvement effort is hot-metal silicon variability. This is controlled by the heat levels inside the furnace. Existing expert systems address two problems: predicting abnormal situations and keeping the thermal condition stable.

Expert systems have also been developed for other purposes in steel industry, such as scheduling high-grade steel production which cuts planning time, allows for greater what-if experimentation, and improves quality control.

5. AI in Power Plants

The maintenance objective for any manufacturing or power plant is that plant systems should always be available to support plant function, without ever limiting plan production. An intelligent real-time maintenance management system has been developed to help process-plant engineers and owners perform value-based plant maintenance. With the system, they can inspect subsystems, identify component operating parameters, and review and make notes regarding component performance or operational and maintenance history.

C. Case Studies

1. Fusion of Intelligent Agents for the Detection of Aircraft in SAR Images

Receiver operating curves are used in the analysis of 20 images using a novel Automatic Target Recognition (ATR) Fusion System. Fuzzy reasoning is used to improve the accuracy of the automatic detection of aircraft in Synthetic Aperture Radar (SAR) images using *a priori* knowledge derived from color aerial photographs. The images taken by the two different sensors are taken at different times. The results of the experiments using real and generated targets with noise for a probability of detection are 91.5% using the ATR fusion technique, the false alarm rates have been improved by approximately 17% over using texture classification. To achieve this, receiver operating curves have been used in the analysis of 20 images to investigate the use of a fuzzy rule based fusion technique to automatically detect targets such as ground-based aircraft, in SAR images with greater certainty by using all available information possible. To verify the robustness of the algorithm, numerous additional aircraft targets (with gray level values with varying amounts of added noise and spatial positions of pixels similar to that of the real targets) have been generated in two types of backgrounds, one with more natural vegetation, the other with more man-made structures. To improve the results some *a priori* information has been incorporated in the form of an aerial photographic image taken of the background (i.e., taxiways, buildings, roads etc.) many months before the SAR image was taken.

ATR involves extraction of critical information from complex and uncertain data for which the traditional approaches of signal processing, pattern recognition, and rule-based AI techniques have been unable to provide adequate solutions. Target recognition of one or more fixed signatures in a stationary background is a straightforward task for which numerous effective techniques have been developed. If the target signatures and the background are variable in either a limited or known manner, more complex techniques such as using rule-based AI (i.e., expert systems) methods can be effective. However, rule-based AI systems exhibit brittle rather than robust behavior (i.e., there is great sensitivity to the specific assumptions and environments). When the target signatures or backgrounds vary in an unlimited or unknown manner, the traditional approaches have not been able to furnish appropriate solutions.

A number of ATR systems, based on various methods, have been developed and tested on a very limited

data set, and good classification performance has been reported. However, in practice these efforts have been only partially successful and have produced high false alarm rates. Reasons for this are the nonrepeatability of the target signature, competing clutter objects having the same shape as the actual targets, experience with a very limited database, obscuration of targets, and a limited use of *a priori* information. Many previous algorithm approaches have been proposed. The major categories of algorithms, functions, and methods previously applied to ATR include classical statistical pattern recognition, syntactical pattern recognition, and rule-based AI approaches.

Neural network technology provides a number of tools which could form a basis for a potentially fruitful approach to the ATR problem. For ATR one needs methods to recognize targets and backgrounds that are both sufficiently descriptive yet robust to signature and environmental variations. One also needs the ability to adapt the procedure for additional targets and environments. The existence of powerful learning algorithms is one of the main strengths of the neural network approach. It has been demonstrated that ATR performance can be enhanced by use of *a priori* knowledge about target signatures and backgrounds, in this case *a priori* knowledge of aircraft size and the type of ground surfaces on which aircraft were more or less likely to be found.

The advantages of using statistical methods of ATR are that is was satisfactory for patterns with well-behaved distributions, but one of the critical issues was that it required large representative training sets. The limitation of the statistical approach to the ATR system led to the hope that better performance might be achieved by using rule-based AI or expert-system/knowledge-based techniques. The intrinsic deficiencies of applying expert-systems/knowledge-based systems of ATR include the following: (1) the difficulty in extracting the necessary data from the sensor to support the inference procedures; (2) the difficulty in acquiring, representing, and exploiting prior knowledge (such as terrain data); and (3) the present lack of an overall knowledge representation structure applicable to ATR. The ATR architecture shown in Fig. 1 addresses the first difficulty in previous ATR systems by combining knowledge-based systems with the fusion of sensors and processing techniques. Hence if target identity attribute information is unavailable from one sensor or processing technique the system has the flexibility in the rules derived from the expert to compensate by using the information from another sensor/processing technique to make a decision on the likelihood of the target based on the way we humans fuse information from our eyes, nose, ears, and other sensors.

Some current target detection techniques on SAR images have been using background discrimination algorithms and the use of texture and pixel intensity analysis with a trained classifier. A fusion system is presented which improves the detection process by using all available information to improve target detection and reduce false alarm rates. The ATR fusion system block diagram shown in Fig. 1 illustrates the fusion of identification attribute information in SAR and color aerial images to identify aircraft targets. The five processing windows are displayed next to the five registered images they are simultaneously processing. The five processing windows are calculating:

1. Attribute "A" using an 8×8 window to calculate the area of the target in the SAR texture classification image I1
2. Attribute "B" using a 9×9 window to calculate the positions of man-made targets using the background discrimination algorithm on the SAR image
3. Attribute "C & D" which are the outputs of the recurrent neural classifier trained on the average spectral values from three 5×5 windows on the red, green, and blue bands of the aerial image

Several (three or four) sample applications are presented. The cases are taken from recently published articles or from the authors' own work. For each application, several typical features are examined.

2. A Rough Set Machine Learning Preprocessor

Rough set theory is applied to diagnostic classification of mechanical objects. Examples are shown which are concerned with evaluation of diagnostic capacity of symptoms, comparison of different methods of defining symptom limit values, reduction of the set of symptoms to a subset ensuring satisfactory evaluation of the object's technical state, and creation of the classifier of the technical state. In the following we use a simpler example, a rough set machine learning preprocessor, to illustrate the idea of using rough sets.

As an expert system development tool, the rules in CLIPS are manually coded by knowledge engineers. To make CLIPS more useful, a preprocessor has been developed. It is able to produce rules to feed the CLIPS rule base. The preprocessor reduces training examples contained in decision tables and produces results in a form which can be further processed by

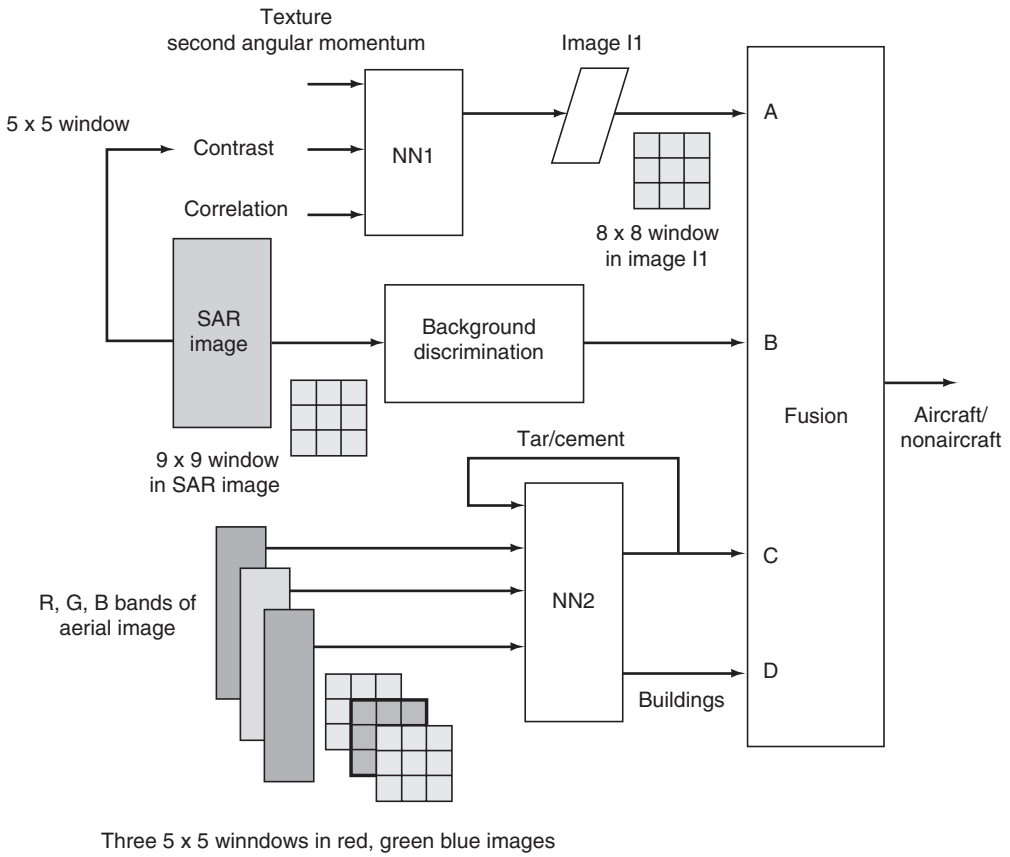


Figure 1 An intelligent ATR fusion system block diagram.

CLIPS programs. By this way, machine learning techniques can be used to assist in the update of knowledge bases as used in expert systems. In this section, we provide a brief description for this preprocessor. The particular learning method used in our experiment is a version of rough set learning algorithm on decision tables, but other machine learning algorithms can also be used. Decision tables are flat tables containing attributes and decisions as columns and actual data elements as rows. For example, in engineering applications, attributes of a decision table may be related to various symptoms while decision variables are fault diagnosis.

For instance, Table I depicts a simple troubleshooting system where the diagnosis (D) is affected by symptoms S1, S2, S3, and S4. S1 has three values: very low, low, and high, while S2, S3, and S4 have three values: absent and present. In addition, the decision variable D has two values: p1 (for problem 1) and p2 (for problem 2). A decision table involving 10 examples is shown in Table I. Each line (except for the top line which indicates the attribute names) in this table denote a training example. For instance, the first example in the table indicates "If S1 is very low, S2 is absent, S3 is absent and S4 is also absent, then the problem is p1."

The algorithm used by the preprocessor mainly consists of three parts:

1. Identify indispensable attributes (i.e., remove attributes that are not needed for rule construction)
2. Generate certain rules (creates sets partitioned to the different decisions)
3. Computing decision-cores of each elementary condition category (i.e., check which values of condition attributes are indispensable in order to discern the values of the decision attribute)

Finally, simplification may be needed to remove superfluous decision rules and handle the "don't care" values of attributes.

To illustrate how rough set approach can be used in engineering applications, consider Table I again. Several rules which can be learned from Table I, including rules R1 and R2:

Table I Simple Troubleshooting System

Symptom 1	Symptom 2	Symptom 3	Symptom 4	Decision
Very low	Absent	Absent	Absent	Problem 1
Very low	Absent	Present	Present	Problem 1
Low	Absent	Present	Present	Problem 2
Low	Present	Absent	Absent	Problem 1
Low	Present	Absent	Absent	Problem 2
High	Absent	Absent	Absent	Problem 1
High	Present	Absent	Absent	Problem 1
High	Present	Absent	Absent	Problem 2
High	Present	Present	Present	Problem 2
High	Present	Present	Present	Problem 2

R1: "If S1 is very low then the problem is p1."
R2: "If S2 is present and S4 is present then problem is p2."

As a side note, we point out that a strength of the rough set approach is its capability of dealing with inconsistent data (as exemplified by the examples 4 and 5 in Table I).

The output of the preprocessor takes the form of CLIPS rules, which can be added to the knowledge base in any expert system constructed using CLIPS. When the produced CLIPS file is loaded, it will prompt the user for an attribute value for each of the given attribute fields. Once a rule has been matched, it will be fired and the decision found will be displayed.

The generated CLIPS file can then be executed in a CLIPS environment. For example, if we feed in the 10 examples shown in Table I, several generated rules will be displayed on the screen, including the following two, which are corresponding to rules R1 and R2 appeared after Table I:

<000> S1(very low) ->p1
<008> S2(present) S4(present) ->p2

The above discussion explained how the CLIPS rule base can be expanded by adding rules which are generated through machine learning techniques. Furthermore, knowledge update can be supported in the CLIPS environment itself, as shown in a case study which involves an aircraft diagnostic system. The B-1B defensive avionics system (DAS) provides protection for the aircraft from enemy threat systems, allowing more effective penetration of hostile territory. At the heart of the system is a component referred to as ALQ-161 which automatically monitors and displays received emitter ratio frequency (RF) data along the flight path to the operator (via a set of controls and displays) and automatically (or manually) assigns deceptive jamming techniques to confuse enemy radar systems. It is difficult to maintain such a complex system. What is needed is a low-cost and effective approach to an expert maintenance advisor system, based on heuristic knowledge, to augment existing support equipment and built-in-test (BIT) data. The test results obtained during causal reasoning would be captured, maintained, and organized, by original symptom into a knowledge base consisting of rules of malfunction repairs supplemented with relevant supporting data.

III. NEW DEVELOPMENT OF AI IN INDUSTRY

A. Some Current Research Projects Applying Existing Techniques

The Knowledge-Based Intelligent Engineering Systems Centre (KES Centre) in University of South Australia is involved in researching techniques for the design and implementation of knowledge-based intelligent information systems.

a. Land Mine Detection

One of the aims of this research is to provide a quantitative demonstration of the benefits of using multiple sensors for the detection of surface land mines from a distant platform. In particular it will show a reduction in the false alarm rates (FAR) through the fusion of two sets of imagery from an infrared sensor with a rotating polarizer attached and a digital multispectral camera. The results show using a multilayer perception (MLP) neural network and an

adaptive theory resonance (ART2) (with novel modifications) neural network classifiers on the input textural and spectral characteristics of selected multispectral bands (using a GA tool) we obtained false alarm rates at around 3%. Using polarization-resolved images only we obtain false alarm rates of 1.15%. Fusing the best classifier processing the multispectral images with the polarization-resolved images FAR's have been obtained as low as 0.03%. This result has shown the large improvements gained in the fusion of sensors and over the commercial systems.

b. A Neural-Based Automatic Target Recognition System

The objective of this research is to implement 2-D visual ATR capable of self-organized real-time learning, and memory-guided search in cluttered backgrounds with rotation invariance. This new feature is achieved by providing more input synapses to each cell in the input layer of the neural network. This permits the cell to test for the presence of the target shape in a wider range of locations.

c. Knowledge-Based Decision Aids System for Management in Tactical Environment

The aircraft system has multiple sensors to collect data from dynamic environment. These sensors provide a large amount of information to the system operator. The purpose of this project is to design a knowledge-based system that will help an operator to make decisions easily.

d. Knowledge-Based Performance Evaluation System using Simulation Training and Live Training Practices

The main goal of this research is to use knowledge-based techniques to develop a system that will evaluate Naval Cadet trainees as Officers of the Watch (OOW). The system will behave as an evaluation analysis tool for the trainee's supervisor, examining the strong and weak points of trainee performance. A knowledge-based expert system has been constructed for this task and mechanisms have been developed to represent the knowledge which is derived from the evaluation sheets of the trainees.

e. Decision and Tracking of Distant Airborne Targets in Cluttered Background in Infrared Image Sequences

The detection and tracking of dim targets by scanning and staring image registration (IR) systems are under investigation using novel parallel processing algorithms based on the associative processing paradigm. The main aim of this project is the design and development of a set of novel intelligent image-processing algorithms for detection, classification, and tracking of small low contrast aerial targets in highly cluttered environments in real time. The ongoing work to date includes the development of a novel method of calculating cloud motion using mathematical morphology operations.

B. New Frontiers: Some Recent Development and Trends

In addition to research work applying existing techniques, some recent development and trends of development of AI in industry can be found in recent workshops sponsored by the American Association for Artificial Intelligence (AAAI).

1. Exploring General Aspects of AI and Manufacturing

Specific topics of AI interest in manufacturing life cycle activities including design, engineering, production planning, scheduling and control, process diagnosis and control, recycling and remanufacturing; AI and business process reengineering; AI and issues of enterprise integration, including enterprise modeling, architectures for coordination and collaborative, distributed decision-making; and the role of AI in supporting new manufacturing concepts such as agility, virtual manufacturing, etc.

Over the past decade, AI concepts and techniques have been productively applied to diverse aspects of manufacturing decision making, ranging from product and process development, to production management, to process diagnosis and quality control. It is no longer a question of whether AI technologies will have an impact on manufacturing but one of better understanding and exploiting the broad potential of AI in this domain. New manufacturing concepts and philosophies such as lean manufacturing, agile manufacturing, and virtual manufacturing place increasing emphasis on the need for more intelligent manufacturing systems, and there is general consensus that AI technologies will play a key role in the manufacturing enterprise of the future.

One continuing obstacle to more rapid development and application of AI in manufacturing is the low bandwidth of communication (of problems, approaches, and solutions) between the manufacturing and AI research disciplines. From an AI standpoint, manufacturing is seen as a rich application area and

research driver. However, too often work has proceeded without good understanding of the actual problems faced by manufacturing organizations, and the solutions developed are consequently seen to offer only marginal practical gain. From the manufacturing side, there is general recognition that AI has important contributions to make, but there is also limited understanding of AI technologies and their relevance to manufacturing problems.

One focus is to build greater mutual understanding of important research challenges and technological potential in this field, break down the cultural barriers that currently exist between these two disciplines, and foster future interaction and collaboration with these two communities toward the realization of intelligent manufacturing systems.

Some specific issues are

- AI in manufacturing life-cycle activities' including design, engineering, production planning, scheduling and control, process diagnosis and control, recycling and remanufacturing
- AI and business process reengineering
- AI and issues of enterprise integration, including enterprise modeling, architectures for coordination and collaboration, distributed decision-making
- The role of AI in supporting new manufacturing concepts such as agility, virtual manufacturing, etc.

2. From Self-Organization to Self-Adaptive Software

Self-organization is an important feature and has been demonstrated in several AI techniques, such as neural networks and GAs. It has also been applied in various research projects. A research project reported in Jain et al. (1999) discussed self-organizing manufacturing systems in which each module self-organizes effectively according to other modules. A module is defined as a process for decision making concerning manufacturing systems. Each module decides the interaction among modules. However, combinatorial optimization problems present a challenge in manufacturing systems. Genetic algorithms have been applied to deal with related problems, such as scheduling, path planning, and resource allocation. Additional challenges also exist, such as modules may not share complete information concerning other modules, or information received from other modules is often ambiguous or incomplete. Fuzzy set theory has been applied. As a result is a virus-evolutionary GA to fuzzify flow shop scheduling problems with fuzzy transportation time. Simulation results are encouraging.

A related (but different) concept is developing a *self-evolving tool* for knowledge acquisition, as discussed in Chen (1999). Ideally, it would be nice to build a knowledge acquisition tool which is able to self-evolve; namely, it is able to improve its behavior for knowledge acquisition. The tool allows the user (a domain expert) not only to enter new knowledge, but also to tell the tool some meta-knowledge, namely, what kind of knowledge should be acquired and what kind of questions should be asked. By this way, an expert in a knowledge domain can always recast the tool for his own need so that "everybody gets whatever he wanted." Traditionally self-evolution or self-organization has been an important notion in cybernetics and AI. Starting with a generic "empty" structure, such a tool allows the user to recast it into a form which is suitable for knowledge acquisition in some specific knowledge domain. By this way, the tool can build itself (namely, self-evolving).

An important trend in AI in industry is the development of robust software through *self-adaptation*. As noted, self-adaptive software requires high dependability, robustness, adaptability, and availability. An infrastructure is proposed to support two simultaneous processes in self-adaptive software: system evolution, the consistent application of change over time; and system adaptation, the cycle of detecting changing circumstances and planning and deploying responsive modifications.

Self-adaptive software modifies its own behavior in response to changes in its operating environment (which is anything observable by the software system, such as end-user input, external hardware devices and sensors, or program instrumentation). Application developers have faced numerous issues, such as under what conditions does the system undergo adaptation, should the system be open- or closed-adaptive, what type of autonomy must be supported, under what circumstances is adaptation cost-effective, how often is adaptation considered, and what kind of information must be collected to make adaptation decisions, etc.

An interesting application of self-adaptive software in robotics is exemplified in a work of gesture-based programming. Systems supporting gesture-based programming capture, in real time, the intention behind fleeting, context-dependent hand motions, contact conditions, finger poses, and even cryptic utterances. The systems retain previous acquired skills, which facilitates gesture interpretation during training and provides feedback control at run time.

3. Topological Representation and Reasoning in Design and Manufacturing

As an example of new development of a specific technique, the new area of research involves the integration of topological properties in a wide variety of design-related issues and activities, including:

- Descriptive and computational modeling of design knowledge, organization, and process
- Geometric representation (including reasoning about tolerances); design entity similarity measurement
- Geometric/topological/physical integrated modeling of physical behavior
- Design to manufacturing transformation modeling
- Topological optimization
- Qualitative spatial reasoning

Topological design and manufacturing, as an area of a unifying design abstraction, embraces many aspects, such as:

1. *Representation models for design knowledge, and conceptual design process.* This includes casting design in the framework of set theory, set-point topology or metric spaces; representing structure of abstract design concepts and their relationships; and developing intelligent systems for automating the topological mapping of functional to attribute space. A related aspect is representation models for the conceptual and preliminary design process with topological knowledge structure.
2. *Searching techniques for an optimal topology during design synthesis.* An example of this aspect is truss design optimization.
3. *Computational models for measuring the similarity of function concepts.* Sample issues related to this aspect include calculating a metric/distance between two design space entities; directing the search for components that meet the required functionality; and integrating metric/case-based reasoning techniques.
4. *Computer-aided geometric design.* Issues related to this aspect include solid and nonmanifold modeling; feature recognition; feature-based design; geometric abstractions for reasoning about shape; topology/algebraic interaction; topology-based models for reasoning about adjacency relations among vertices, edges, and faces; as well as developing tools for reasoning under uncertainty to deal with imprecise and approximate geometry that may fail to accurately represent the topology of the object.
5. *Qualitative spatial reasoning using topology.* Issues related to this aspect include reasoning about properties of points or point sets in space; detecting intersection relations among combinations of point sets; developing methods where topological queries can be solved by topological computation without geometry; and topological-based reasoning for finding consistent paths through point set combinations.
6. *Models of physical behavior.* Issues related to this aspect include computational modeling that combines physical behavior and geometry; applying algebraic topology based on cell complexes, chains, and topological operations on chains; developing computer languages for engineering physics; and applying integrated function-geometry models for analysis, simulation, and automated synthesis.
7. *Representation and reasoning of geometric tolerances.* Issues related to this aspect include developing topology-based computational tools to address the following problems: (1) how to construct a locality around the boundary of the nominal object in which geometric variations are allowed; and (2) how "topologically" similar the geometry of the object within the tolerance is to that of the nominal object.
8. *Models of design-manufacturing mapping.* Issues related to this aspect include developing AI tools for mapping a design form into its corresponding manufacturing representation; and reasoning about situations where there are close points in the design space for which their manufacturing representations are very far from each other.

4. Global Optimization Techniques

As yet another example of newly developed specific techniques, let us briefly examine the issue of global optimization. Problems of industrial systems design and management have large solution space, i.e., the number of solutions is very large. Many local optima exist. Efficient methods for global optimization are required.

Topics for developments and applications concerning global optimization techniques for industrial engineering include:

- Greedy random adaptive search procedures
- Natural evolutionary computation
- Nonmonotonic search strategies
- Partitioned search methods
- Statistical methods
- Simulated annealing
- Tabu search
- Threshold algorithms and their hybrids

Much attention has been paid in methods which can achieve some impressive tasks, such as: (1) to combine intelligently different concepts for exploring and exploiting the search space, and use learning strategies in order to find efficiently near-optimal solutions; (2) to couple optimization techniques and discrete events simulation models; and (3) to apply global optimization approaches for multiobjective optimization.

IV. CONCLUSIONS

In this article, we have presented a brief review of some basic AI concepts used in industry applications. We discussed some successful AI techniques used in industry applications, such as constraint-based search, building intelligent agents, and qualitative reasoning. We then examined several typical AI techniques used in industry, such as expert systems and soft computing techniques. We then studied types of AI techniques, such as intelligent control, intelligent scheduling, and product configuration. We examined several sectors within AI, including AI in civil and structural engineering, AI in the steel industry, and AI in a power plant. Two case studies were then presented, including fusion of intelligent agents for the detection of aircraft in SAR images and a rough set machine learning preprocessor. Finally, we discussed some new developments of AI in industry, including some current research projects applying existing techniques as well as new frontiers of research trends, such as exploring general aspects of AI and manufacturing, and self-adaptive software, as well as others. Our discussion has indicated that industry applications have been an important driving force for applied AI research.

We conclude this article with some other issues related to AI and industry applications,

- Industrial applications should continuously address traditional concerns within AI. For example, scale-up has always been a hurdle for applying AI techniques in industry. The plain notation of state space search is not appropriate in many applications.
- Ontology is concerned with describing the world and has important role in AI applications.
- Computational creativity plays an important role in design.
- The use of A* algorithm in physical implementation of data warehouses developed in industry is an example of the relationship between AI and other disciplines and integration with other forms of information technology (such as integration with database management systems).
- Earlier we have already briefly examined web-based agents. Researchers should continuously pay attention to e-commerce and related applications.
- There are other promising AI techniques that have been, or could be used in industry applications. One such area is data mining.

SEE ALSO THE FOLLOWING ARTICLES

Engineering, Artificial Intelligence in • Evolutionary Algorithms • Expert Systems • Expert Systems Construction • Intelligent Agents • Knowledge Representation • Medicine, Artificial Intelligence in • Neural Networks

BIBLIOGRAPHY

Buckley, S. J., and Murthy, S. S., eds. (1997). AI in manufacturing. *IEEE Expert,* 12(1), 22–56.

Chen, Z. (1999). Knowledge acquisition assisted by CLIPS programming. *Engineering Applications of Artificial Intelligence,* Vol. 12, 379–387.

Chen, Z. (2000). *Computational intelligence for decision support.* Boca Raton, FL: CRC Press.

Dorn, J., ed. (1996). AI in steelmaking. *IEEE Expert,* 11(1), 18–35.

Faltings, B., and Freuder, E. C., eds. (1998). Configuration. *IEEE Intelligent Systems,* pp. 32–33.

Faltings, B., and Freuder, E. C., eds. (1998). Configuration. *IEEE Intelligent Systems,* 13(4), 32–85.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., eds. *Advances in knowledge discovery and data mining.* Menlo Park, CA: AAAI/MIT Press.

Filippidis, A., Jain, L. C., and Martin, N. (1999). Fusion of intelligent agents for the detection of aircraft in SAR Images. *IEEE Transactions PAMI.*

Filippidis, A., Jain, L. C., and Martin, N. N. (1999). The application of knowledge-based techniques in surface landmine detection. *IEEE Transactions on Signal Processing,* 47 (1), 176–186.

Freuder, C., ed. (2000). Constraints. *IEEE Intelligent Systems,* 15(1), 24–44.

Garrett, Jr., J. H., and Smith, I. F. C., eds. (1996). AI in civil and structural engineering. *IEEE Expert,* 11(3), 20–53; 11(4), 24–49.

Goel, A. K. (1997). Design, analogy, and creativity. *IEEE Expert,* 12(3), 62–70.

Hearst, M., and Hirsh, H., eds. (2000). AI's greatest trends and controversies. *IEEE Intelligent Systems,* 15(1), 8–17, 2000.

Hendler, J., ed. (1996, 1999). Intelligent agents. *IEEE Experts,* 11(6), 20–63, 14(2), 32–72.

Huhns, M. N., Singh, M. P., and Gasser, L., eds. *Readings in agents,* San Francisco: Morgan Kaufmann.

Iwasaki, Y., ed. (1997). Qualitative reasoning. *IEEE Expert,* 12(3), 16–61.

Jain, L. C., Johnson, R. P., Takefuji, Y., and Zadeh, L. A., eds. (1999). *Knowledge-based intelligent techniques in industry.* Boca Raton, FL: CRC Press.

Kasabov, N. K. (1998). *Foundations of neural networks, fuzzy zystems, and knowledge engineering.* Cambridge, MA: MIT Press.

Labio, W. J., Quass, D., and Adelberg, B. (1997). Physical database design for data warehouses. *Proceedings of the international conference on data engineering.*

Laddaga, R., ed. (1999). Self-adaptive software. *IEEE Intelligent Systems,* 14(3), 26–62.

O'Leary, C., ed. (1998). Knowledge management. *IEEE Intelligent Systems,* 13(3), 30–48.

Ostrognay, M., Jain, L. C., Cropley, A., Puri, V., and Filippidis, D. (1999). Intelligent data processing in training evaluation practices. *Proceedings of the third international conference on knowledge-based intelligent information engineering systems,* pp. 21–25. New York: IEEE Press.

Passino, K. M., and Ozguner, U., eds. (1996). Intelligent control. *IEEE Expert,* 11(2), 28–55.

Pedrycz, W., and Zadeh, L. A. (1995). *Fuzzy sets engineering.* Boca Raton, FL: CRC Press.

Rauch-Hindin, W. B. (1985). *Artificial intelligence in business, science, and industry,* Vol. II (Applications). Englewood Cliffs, NJ: Prentice Hall.

Slowinski, R., ed. (1992). *Intelligent decision support: Handbook of applications and advances of the rough sets theory.* Boston, MA: Kluwer.

Swartout, W., and Tate, A., eds. (1999). *Ontologies. IEEE Intelligent Systems,* 14(1), 18–54; 14(3), 63–81.

Information Measurement Theory (IMT)

Dean Kashiwagi
Arizona State University

I. INTRODUCTION

In 1948, Claude Shannon was credited with discovering "information theory." Information theory had the following precepts:

1. Fundamentally all communication systems are the same.
2. They all have a speed limit, measured in terms of binary digits per second.
3. Above the speed limit, the information cannot be perfectly transmitted.
4. Below the speed limit, the perfect transmission of information was possible, regardless of the strength of signal or the static or noise of the environment.
5. The limitation or constraint was the transmission speed or the medium and not the noise of the environment.
6. All mediums could therefore pass "perfect information" in digital bits.

Shannon realized that the constraint of communicating, or moving information, was the transmission speed of the medium and not the noise of the environment. Dean Kashiwagi took the concept a step further and postulated that it was not the lack of information, but the processing speed of an individual that creates the perception of lack of information and use of bias that is the obstacle to understanding perfect information. If true, human processing could be replaced by a faster processor, which would generate information. This information would be acceptable to all people due to the fact it was generated without bias. Kashiwagi proposed that the technology of "information" would not only be used to communicate information, but to create the information that would be understood by people of different levels of processing speed.

Information measurement theory or IMT was formulated in 1991 by Dean Kashiwagi at Arizona State University as the structure to use an artificial intelligent processor, in a performance information procurement system (PIPS). By 2001, the process had been implemented over 300 times, and the cost of delivery in the prototype tests was reduced by over 50%. PIPS is a procurement process; however, it is the structure for any decision making process. The decision making process, once it is implemented, will allow one person to do the work of five, by minimizing decision making and functions that do not add value.

The discussion will be in the following order:

1. Information measurement theory logic
2. IMT applications in different fields of study
3. The artificial intelligent decision model, the mathematics, and the application
4. Performance information procurement system (PIPS)

II. INFORMATION MEASUREMENT THEORY

The purpose of IMT is to set an information structure that allows the use of artificial intelligent processors to optimize decision making and minimize risk (not getting the desired outcome). IMT is defined as a deductive logical explanation of the structure of an "event," and the use of the measurement of relative and related data in terms of "information" that defines the conditions of an event or event object at a specific time and predicts the future outcome of the event.

The objectives of IMT include:

1. Identifying future outcomes by measuring relative differences of data
2. Minimizing decision making and the use of personal bias by identifying (predicting) the most likely outcome
3. Identifying and relating the use of information to performance levels
4. Creating a structure where a "nonbiased" decision maker, or artificial intelligent system can predict the future outcome, replacing the use of personal decision making, minimizing the risk of inaccuracies caused by personal bias or subjectivity
5. Overcoming the obstacles to understanding by creating an information system that allows everyone to access the information in terms of relative differences which minimize subjectivity

The results of the IMT structure include:

1. Reducing the number of translations of data
2. Identifying the best possible options in terms of relativity
3. Setting the requirements in terms of information level
4. Minimizing decision making by use of an AI decision-making tool which prioritizes to a specific requirement

III. DEDUCTIVE LOGIC

There are two major methods of problem solution or logic accepted by the scientific arena: inductive logic and deductive logic. Inductive logic also known as the scientific method follows the following steps:

1. Setting up a hypothesis which defines an outcome
2. Devising an experiment that tests the hypothesis
3. Conducting the experiment to discover previously unknown information
4. Identifying whether the hypothesis is true or false
5. Hypothesizing where the results of the experiment can be used

Deductive logic is defined as the redefining or reordering of existing information to define an outcome. Deductive logic differs from inductive logic in the following ways:

1. There is no new information or theories.
2. There is no experimentation to identify the results.
3. It is faster and simpler.
4. It requires less technical or "specialized" information which is not understood by the average layperson.

IV. IMT THEORETICAL FOUNDATION

IMT is based on deductive logic. It is also deductive in nature and allows individuals with no specialized knowledge to understand and implement. The first definition in IMT is of the laws of physics that represent the physical environment: "Laws predict the future outcome in every state and at every time period. Examples of laws include gravity and combustion."

Therefore, the number of laws stays consistent over time (Fig. 1). Scientists continue to discover more of the existing laws over time. It is also possible that science may unknowingly incorrectly identify a law at one period of time, and find out at a later time period that the law was defined incorrectly or incompletely. It is important to understand that laws are not created, but discovered. This definition is used in Hawking's "no boundary" theory.

The second definition in IMT is of information. Information is defined as "the combination of laws and data (measurements of the conditions) which represent the existing conditions that can be used to accurately predict a future outcome."

Therefore, "information" is not what an individual may perceive, but an explanation of what "actually exists." Therefore, a difference between two individuals would be the amount of information perceived. The constraint is not that the information does not exist, but the information cannot be perceived.

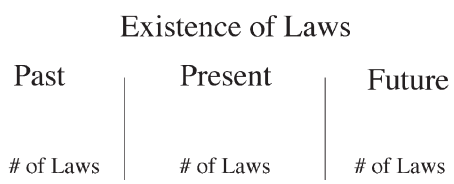


Figure 1 Number of laws of physics.

IMT then defines an event as "anything that happens that takes time."

The event has initial conditions, final conditions, and changing conditions throughout the event (Fig. 2). The number of laws stays consistent throughout the event. The following are characteristics of events:

1. Every event has an unique set of initial conditions and an unique set of final conditions. Two of the easily defined conditions are the time and place of the event.
2. The number of governing laws remains constant throughout the event.
3. Every unique event can only happen one way.
4. Two individuals with different levels of perception may look at the same event and perceive a different event. However, the event is still one event and will have one outcome.
5. There has been no event where the final conditions or outcome is not affected by the initial conditions or previous state. Everything is cause and effect.
6. Every event is constrained by initial conditions and laws and is predictable if all information can be perceived.
7. The change in the event can be identified in terms of differential.
8. There has been no event that has been documented where the event has happened by chance (the final conditions were not predictable or related to the initial conditions).
9. Randomness and probability are merely methods to estimate the final outcome when there is a lack of information about the initial conditions and laws.
10. True randomness does not exist. The only reason the Heisenberg Theory is valid is that we do not have methods or means to accurately measure two linearly related characteristics of particles at the same time. Einstein was criticized for not accepting the premise of randomness, but today it is understood that randomess is caused by our inability to measure and there is no actual random number, event, or object.
11. Every person and every factor impact an event to a relative degree. IMT does not explain why a person is in an event, but it states that the person is a part of the event and will impact the event. The person, the person's decision making, and the person's environment all impact the event. Because every person is predictable or constrained (constraints make everyone unique), the person's decision making is predictable, and therefore any environment with a person is also predictable with "all" information.

Longer or more complex events require more information about the initial conditions and laws to predict the event outcome (Fig. 3). Event C is easier to predict than Event A, due to the minimized change between the initial and final conditions of Event C. It is easier to predict the weather 5 seconds into the future instead of 1 year into the future. However, Event A can be also be divided into finite elements and the last element will look like an Event C (Fig. 4). The final conditions of the new Event C are the same conditions of the final conditions of the Event A that it came from. Therefore, since the final conditions of Event C can happen only one way, and the conditions are the same as the conditions of Event A, all events can only happen one way. This agrees with what we have perceived in reality that at one time, in one location, in one environment, there is a fixed condition that is different from every other condition based on time and location.

Because the initial conditions and laws exist independently of a person's understanding, an individual's ignorance of the existing conditions or laws will not impact the event outcome. The individual will not be able to predict the future outcome. If an individual had more information about the initial conditions and laws of event C, the outcome would be predictable, and Event C would look like Event A to that individual. As previously discussed, it is possible that

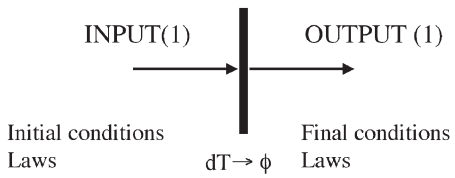


Figure 2 Event.

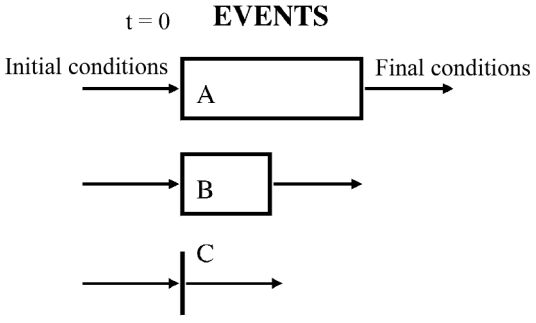


Figure 3 Comparison of events.

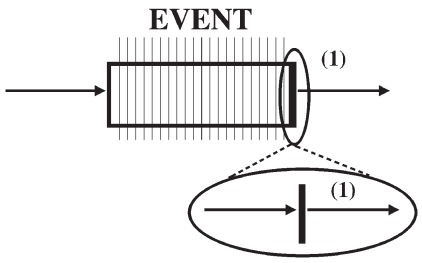


Figure 4 Finite element analysis of event.

two individuals could look at the same event, and one will see it as Event A (easily predicted) and another as Event C (complex and unpredictable).

V. PERCEPTION OF THE EVENT

Every individual is different (location in time and space being the most obvious). Every individual exists in an environment with all information. Individuals perceive existing information, process the information, and apply it if they understand the information. The application of "newly perceived" information causes change, and by observation change leads to the perception of more information. This is the Cycle of Learning (Fig. 5). By observation, the more information perceived, the faster the speed of the cycle. This leads to the following conclusions and the Rate of Change graph (Fig. 6):

1. Application of information and change can be measured more easily than perception and processing.

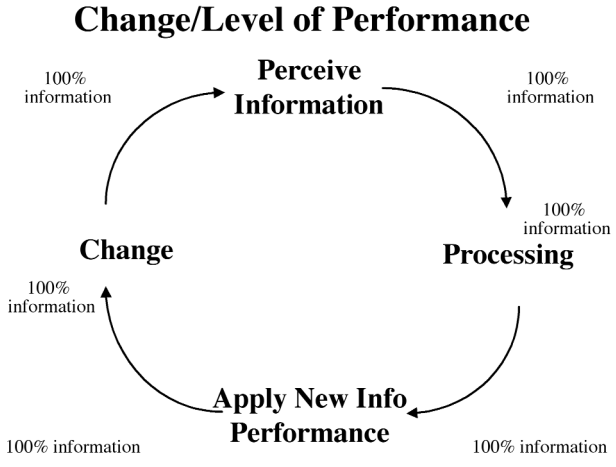


Figure 5 Cycle of learning.

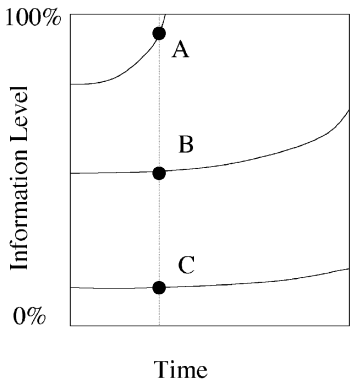


Figure 6 Rate of change.

2. The more information perceived, the faster the rate of change.
3. Those who do not change have difficulty perceiving "new information."

At every time, every person is at a specific level of perception of information. IMT identifies the differential (criteria) between individuals. No two individuals will have the same combination of values for the following factors. The differences include:

1. Education level
2. Salary and financial status
3. Location
4. Time of birth
5. Type of government in environment
6. Race
7. Culture
8. Financial level
9. Family
10. Birth order
11. Level of perception
12. Genetic makeup
13. Religion
14. Occupation type and performance
15. Talents
16. Hobbies
17. Language

The values for the criteria make a unique performance line for the individual. All the above factors have a relationship or impact on the level of information, the ability to process information, or the opportunity to access information. Figure 7 shows the Rate of Change Chart with two-way Kashiwagi Solution Models (KSM). The KSMs have the following characteristics:

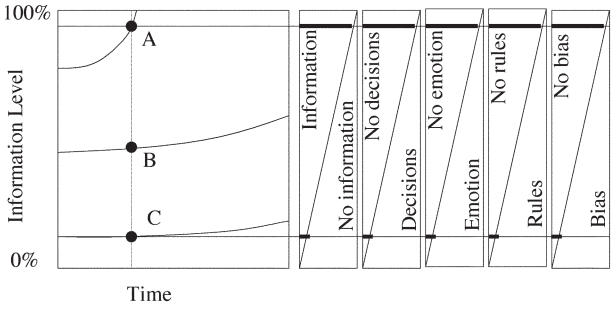


Figure 7 Rate of change and KSMS.

1. They represent two sides of each characteristic in terms of information.
2. As a person moves from a lower level to a higher level, the characteristic must represent an increase or decrease in information.
3. All factors are relative and related.
4. The slope of the line separating the two opposites is not significant when the amount of information is not an issue.

Figure 8 shows two individuals (B1 and B2) who are very similar in terms of processing speed, amount of information perceived, identified level of performance, and change rates. To differentiate and predict the difference in a future time period would require too much information (data that differentiates). This would require, within our current methodologies and measuring tools, extensive statistical analysis with large amounts of data from representative, random sampling. IMT is interested in the movement from one level to another, and in identifying the top and bottom levels of the graph. The assumption of the KSM is that all factors are related and relative. To avoid using extensive statistical sampling, the environments of study will be the extremes to validate the top and bottom areas of the KSM and not the middle sections.

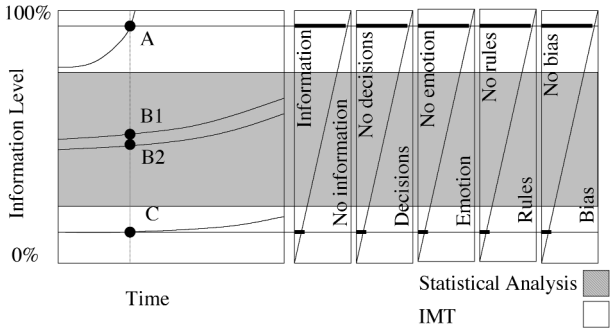


Figure 8 Analysis of extreme environments.

The important issue is not the slope of the line representing the change in the degree of relativity of a factor, but proving that on opposite ends there is a predominance of one factor relating to either a high level of information or low level of information.

The Type A person is labeled as one with more information. According to the Rate of Change model, the Type A person will perceive more information, process faster, apply more correct principles, and change faster than the Type C person. The KSM information model shows information on the left side and no information on the right side. As someone moves from a lower level to a higher level, they will increase in the amount of information perceived and used. The KSM decision model is next. Decision making is defined as occurring when a person does not know the outcome to an event, and therefore thinks that there are two or more possible outcomes. Those who make decisions are therefore defined as not having enough information to predict the event outcome.

Therefore, decision making is on the right side of the KSM. As someone gains information, they will make fewer decisions. This is supported by the following:

1. The more information one perceives, the easier the decision becomes.
2. If someone perceives all information, they know the event outcome.
3. People make decisions when they don't have all information.
4. If a person knows the event outcome, they do not make a decision.

The next KSM model is an emotion model. Emotion goes on the right hand side, because as people gain more information, they are more focused and less emotional. People get emotional when they have an event they are relatively unfamiliar with or lack sufficient experience (information). This is easily recognized in sports, management, politics, and leadership.

The KSM places rules on the right hand side. Rules stop people from changing and direct people to do the same thing for slightly different events. Rules are also for people who may have less information and are used as a guide on how to perform their job.

The next KSM model shown in Fig. 8 is "control over others." As people perceive more information, they realize that people cannot be controlled. The following are examples of people not being controllable:

1. People learning that their spouse and children cannot be controlled. The longer a couple is married, the more understanding a spouse has

that their spouse is not controllable or very predictable.
2. The penal system's inability to change hardened criminals.
3. Failure of the welfare system to increase the productivity of the jobless.
4. The United State's inability to transform the Soviet Union, Vietnam, Haiti, and other countries into democracies.
5. Sports teams that are successful are teams that are able to integrate player's talents. Teams that attempt to change their players are less successful.

The last KSM model is a bias model. The more information one perceives, the less subjectivity or bias the person will use. Bias can be defined as using one's limited perception of information. As a person increases in the perception of information, they will make fewer decisions, need fewer rules, become less emotional, control others less, and use less bias. IMT states that no one has all information. Therefore, each individual will make decisions, need some rules, show a degree of emotion, try to control others and the event, and use their own bias.

IMT states that the only use of information is to predict a future outcome. Any environment is an event. Because the event only happens one way, all elements of the event are required. Thus no part of the event becomes more important than any other part of the event. There is no technically "wrong" part of the event, because that would assume that the event was wrong and should not have happened. Therefore, labels such as wrong are subjective based on an individual's biased set of standards. Therefore, the less information an individual has the more the chance that they will implement standards. As Edward Deming reiterated over and over, standards do not help anyone. Some exceed them, and some can't meet them. Standards are therefore used by individuals with very little information (Fig. 9). It also gives little information. People either pass or do not pass the standards.

IMT also states that every individual is comfortable at a different level of information, and the level of information of their environment will match their comfort level of information. Then by using the KSM, the characteristics that correspond to the information environment can be identified. The KSM also allows an individual to identify other characteristics once a few major characteristics are identified. Using the rate of change chart and the KSM chart, the future outcomes of an individual can be predicted. This is confirmed by the Myers–Brinker and numerous other derivations of individual personality profiling and research work by Buckingham and Coffman in 1999 which show over time that individuals do not change their natural tendencies. The more information, the more accurate the prediction. Because of the complexity of the human being, people are the most difficult to predict. The future state of inanimate objects is less complex and easier to predict.

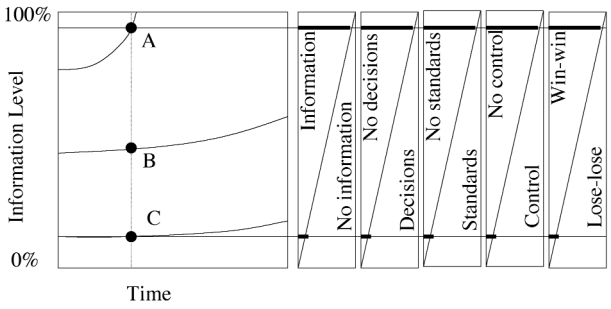


Figure 9 Standards, control, noninformation environment.

The principles of IMT are used in artificial intelligence, psychology, psychiatry, sociology, the study of genetics, business management physics, and metaphysics. IMT principles may be used for predictions, analyses, assistance in decision making, and optimization of performance.

VI. MINIMIZATION OF THE TRANSLATION OF DATA

To minimize subjectivity and the translation of data, the following rules should be followed:

1. Accept all potential criteria to differentiate.
2. Allow the individual or group receiving the service to identify their requirement.
3. Allow potential alternatives to respond to a requirement and also to identify the differentiating performance criteria and sources that best represents them.
4. Allow the alternatives to identify their perception of the future environment.

The rules are simple: compete alternatives based on the requirement and pick the one that is closest to the future outcome. The IMT does not only measure the past performance differential, but the ability to identify the future capability by allowing the alternatives to identify the risk (difference between expectation and future outcome, how to minimize the risk, and differential in terms of risk minimization). This future capability is

measured in relative terms by the receiving group. These rules follow successful business rules:

1. Push decision making to the lowest level.
2. Empower individuals. Empowered individuals have two opportunities: making decisions and accepting responsibility.
3. Recognize that the ability to perceive comes from the individual.
4. Make people successful by giving them the opportunity and the environment that matches their capability.

VII. ARTIFICIAL INTELLIGENT DECISION MAKING MODEL

IMT sets the environment. The KSMs give a description of the environment. The alternatives identify themselves with data descriptors. A multicriteria decision making model was required to prioritize the alternatives against a requirement. For this purpose two models were considered:

1. The Analytical Hierarchical Process (AHP)
2. The Displaced Ideal Model (DIM)

The DIM was selected due to its simplicity and ability to take a "biased" requirement regardless of "level of consistency" of the relationship of the weighting between criteria. Its use of the entropy equation and the natural log function provides a minimization of risk that is required for the procurement model. In early testing, this choice was validated, as one of the requirements of facility owners was to quickly set the weights. Unknown to the authors, later testing would also validate this choice, as many of the criteria would show very little relative difference, making the weights on the majority of criteria a nonissue. The greatest obstacle for using the AHP was the requirement for pairwise comparison of all the variables. A typical decision may have over 500 criteria. Pairwise comparison would make setting the weights a lengthy process. The strengths of the DIM model were:

1. Simplicity. It considered the distance from the best number for each criterion, the weight given the criteria, and an information factor, which was the reciprocal of the entropy equation based on the natural log function.
2. Congruence to information measurement theory (measurement of differences and logic processes). It measures differences and minimizes risk.
3. Ability to take a "biased" set of weights.
4. Measurement of the relative relationship among objective measured criteria, subjective performance criteria, and risk related factors such as the number of different references (individuals and jobs).
5. Minimization of the use of subjectivity by measuring objective and subjective values without translation of the data.

VIII. DISPLACED IDEAL MODEL

The Displaced Ideal Model (DIM) has the following general steps:

1. It identifies the optimal value of each attribute from all the alternatives.
2. It divides each value of each attribute by the optimal attribute value to make all values relational to each other and divides each alternative's relational value by the sum of all the alternatives' relational values for that attribute. This normalizes all the model's attributes, giving a relation of values within and between attributes.
3. The model then uses the entropy equation to identify the entropy of each value of each attribute and then the entropy is summed for each attribute.
4. In order to normalize the entropy between attributes, the sum of each attribute's entropy is divided by the maximum possible entropy for each attribute, which is the natural log of the number of alternatives.
5. The information factor for the attribute is then calculated as the reciprocal of the entropy of each attribute by subtracting the entropy value from one.
6. The model then multiplies the normalized distance of each attribute's value by the attribute's information factor and the weight factor.
7. The alternative with the lowest total relative distance is the best available option.

A more detailed explanation using mathematical equations and symbols is presented below. The process can be broken up into two modules: identifying membership functions (distances) and the identification of the importance of attributes that describe the alternatives. We start with following set of pairs,

$$\{x_i^k,\ d_i^k\} \quad i = 1, \ldots, m$$

$$k = 1, \ldots, n$$

where $d^k{}_i$ is a membership function mapping the scores of the ith attribute into the interval [0,1]. Hence the degree of closeness to $x^*{}_i$ for individual alternatives could be computed as:

1. If $x^*{}_i$ is a max, then $d^k{}_i = x^k{}_i / x^*{}_i$.
2. If $x^*{}_i$ is a min, then $d^k{}_i = x^*{}_i / x^k{}_i$.
3. If $x^*{}_i$ is a feasible goal value or Coomb's ideal value, for example, $x^*{}_i$ is preferred to all $x^k{}_i$ smaller and larger than $x^k{}_i$, then
$$d^k{}_i = [1/2 \{ (x^k{}_i / x^*{}_i) + (x^*{}_i / x^k{}_i) \}]^{-1}.$$
4. If, for example, the most distant feasible score is to be labeled by zero regardless of its actual closeness to $x^*{}_i$, we can define x_{i^*} = Min $x^k{}_i$ and $d^k{}_i = [(x^k{}_i - x_{i^*})/(x^*{}_i - x_{i^*})]$.

The above four functions $d^k{}_i$ indicate that x^j is preferred to x^k when $d^k{}_i < d^j{}_i$.

To measure the attribute importance, a weight of attribute importance λ_i is assigned to the ith attribute as a measure of its relative importance in a given decision situation which is directly related to the average intrinsic information generated by the given set of feasible alternatives through the ith attribute, and, in parallel, to the subjective assessment of its importance, reflecting the decision maker's cultural, psychological, and environmental history. The more distinct and differentiated are the scores, i.e., the larger is the contrast intensity of the ith attribute, the greater is the amount of decision information contained in and transmitted by the attribute. The vector $d_i = (d^1{}_i ... dmi)$ characterizes the set D in terms of the ith attribute. Let $D_i = \Sigma^m{}_{k=1} d^k{}_i$, $i = 1...n$. Then the entropy measure of the ith attribute contrast intensity is

$$e(d_i) = -K\Sigma^m{}_{k=1}(d^k{}_i/D_i) \ln(d^k{}_i/D_i),$$

where $k>0$, ln denotes natural logarithm, and $0 \leq d^k{}_i \leq 1$, $e(d_i) \geq 0$.

If all the $d^k{}_i$ became identical for a given i, then $d^k{}_i/d_i = 1/m$, and $e(d_i)$ assumes its maximum value, that is, $e_{\max} = \ln m$. Thus by setting $K = 1/e_{\max}$ we achieve $0 \leq e(d_i) \leq 1$ for all d_i's. Such normalization is needed for comparative purposes.

We shall also define total entropy as

$$E = \Sigma^n{}_{i=1} e(d_i).$$

Because the weights $\tilde{\lambda}_i$ are inversely related to $e(d_i)$, we shall use $1 - e(d_i)$ rather than $e(d_i)$ and normalize to assure that $0 \leq \tilde{\lambda}_i \leq 1$ and $\Sigma^n{}_{i=1}\tilde{\lambda}_i = 1$: $\tilde{\lambda}_i = [1/(n-E)][1-e(d_i)]$, where n = number of criteria.

Both w_i and $\tilde{\lambda}_i$ are determinants of importance in parallel fashion. If $w_i = 0$, then even $\tilde{\lambda}_i = 1$ does not justify making the ith attribute salient. Similarly, if $\tilde{\lambda}_i = 0$, then even the attribute with $w_i = 1$ becomes irrelevant for making a decision. The most important attribute is always the one having both w_i and $\tilde{\lambda}_i$ at their highest possible levels. The overall importance weight λ_i can be formulated as

$$\lambda_i = \tilde{\lambda}_i \cdot w_i,$$

or after normalization,

$$\lambda_i = [\tilde{\lambda}_i \cdot w_i] / \Sigma n_{i=1} [\tilde{\lambda}_i \cdot w_i], \quad i = 1,...,n.$$

Calculation of the relative distance R_i of each variable would then be

$$R_i = \lambda_i [1 - d^k{}_i], \quad i = 1,...,n.$$

IX. MODIFICATION OF THE AI DECISION MAKER

Modifications to the DIM were required to improve the effectiveness of PIPS. The following modifications were made to the DIM:

1. Increase further the differentials of performance
2. Make the value of an alternative more understandable to the construction industry
3. Change the weight scheme to minimize subjectivity

In some cases the model could not identify a significant difference between scores. Often, raters are fearful of rating in the extreme areas. The ratings are usually in a range that is "acceptable" to the rater. To identify the differences in any criteria where relative subjective decisions are being made, and to stress the importance of the criteria, the lowest score is subtracted from every alternative's score. This creates a range from the lowest score to the highest score, instead of a percentage of the highest score, which did not provide sufficient impact.

To assist the contractors understanding of "value," and to meet state laws, a new method was designed to understand the value of the performance by both the alternatives and the user who set the requirements. Price was moved from the performance criteria and moved to after the performance was identified. The following steps were created:

1. The performance was identified as a minimum distance.
2. Weights were put on price and performance.
3. Prices were given a relative value of the low price.

4. Performances were given a relative value of the best performance (low distance to the best numbers).
5. The relative values for price and performance were then related by the relative weighting on both.

The last change to the model operation is the changing of weights to minimize the user's subjectivity. The initial weights are set by the owner's subjective definition of performance. The DIM uses three inputs: the relative distance from the best number, the weight factor, and the information factor (relative difference of criteria). The information factor is unknown until the prioritization is complete. Once the prioritization is complete, the model will identify which criteria have made the biggest impact. To force continuous improvement, the weight factors on the criteria with the largest impact are reduced by a preset amount. This process of weight changing on a continuous basis moves control of the systems from the rater to the rated, which matches the acceptance of risk.

X. EXAMPLE OF THE MODEL APPLICATION

Table I shows a simple example of buying a roofing system for a building (price is being used as a criteria for explanation purposes only). Alternative C is the most inexpensive, but also has the shortest proven service period. If Alternative A had a price of $1.80, there would be no decision to make; Alternative A would be the best value (lowest cost, highest proven service period, and greatest percentage of installed roofs that did not leak). However, because the price of Alternative A is the highest there is a decision. Table II shows the relative distance away from the best line.

The prices have been changed in Table III. Even though the price of Alternative A is not the lowest, it is not significantly different based on the relative difference of the other factors. There is no information given by the Price criteria.

The processor eliminates the Price criteria and makes the decision based on the other criteria. There is no decision because Alternative A has the best of all values (Table IV).

Table V shows the distances and confirms that Alternative A is the best value. All the information is given back to all the alternatives. They then must make a decision on how to become more competitive based on performance and price.

In the past, experts would identify which criteria were important in making the decision. They would tell the alternative how the job should be done. They would review data submitted by alternatives and subjectively rate the alternatives in the criteria. A decision would be made that is highly subjective. The expert would then ensure that the alternative provided the service as directed. In this environment, the risk of nonperformance is with the expert, and it is more important for the selected alternative to know the expert (decision maker) than to improve the performance of the service.

The IMT approach is to:

1. Identify the requirement in terms of using information and minimization of risk (past performance ratings by references, number of references, similarity of event).
2. Allow the alternatives to identify the differences that they offer.
3. Allow the alternatives to pick their best references.
4. Motivate the alternatives to select their "best personnel" for the critical tasks (people are the risk in the alternatives).
5. Allow the alternatives to dictate how they perceive the task, identify the risk, and minimize the risk.
6. Rate the alternatives on their ability to identify future risk, minimize the risk, and describe how they will add value.
7. Give the alternative the responsibility to provide the service on-time, on-budget, and meeting quality expectations.

Table I Roofing Procurement Example

	Alternatives			
Performance criteria	A	B	C	Owner weights
1. Cost	$2.25	$2.00	$1.80	0.60
2. Service period	25	12	10	0.20
3. Percentage of roofs not leaking	99%	98%	80%	0.20

Table II Relative Distancing

Performance criteria	Alternatives A	B	C	Owner weights
1. Cost	0.024	0.012	0.000	0.60
2. Service period	0.000	0.433	0.500	0.20
3. Percentage of roofs not leaking	0.000	0.001	0.009	0.20
Total	0.024	0.465	0.509	

8. Rate the selected alternative on the performance of the task after the tasking is complete.

A. Implementation of the Performance Information Procurement System (PIPS)

The implementation of PIPS is moving from a "noninformation" (decisions made subjectively by all parties, Fig. 10), to an environment where PIPS, a system that measures the data, transforms it into information that is understood by all (Fig. 11). The PIPS minimizes subjectivity, the function of any party that tries to use subjective data to increase value, and surprises or false expectations. The measurement of data is the implementation of artificial intelligence. The Performance Based Studies Research Group (PBSRG) at Arizona State University has conducted over $3M of research, developing and testing the PIPS over 300 times in the past 7 years on over $150M of construction procurement. The following are some of the results of the tests:

1. Largest construction project—$50M Olympic Village Phase II, State of Utah.
2. No contractor generated cost change orders on $100M of construction.
3. Customer satisfaction with procured construction—99%.
4. Average construction rating—9.5 (Out of 10).
5. On-time, on-budget—99%.
6. Major users—Motorola, Honeywell, IBM, Boeing, United Airlines, the states of Wyoming, Utah, Hawaii, and Georgia, University of Hawaii, and the Dallas Independent School District.
7. Manpower change using the process—One person can do the work of five (80% decrease in manpower requirements).
8. The State of Hawaii design and construction group has implemented IMT/PIPS as an information interface, minimizing their decision making and allowing the designers and contractors to fulfill their responsibilities.

The PIPS is made up of the following elements:

1. Performance lines representing various contractor alternatives
2. A full information environment on the Internet where performance lines, decision making matrices with distancing, and end of project ratings are posted (www.eas.asu.edu/pbsrg)
3. The IMT that sets up the performance criteria, the information environment, and the modeling
4. The PIPS process
5. The AI decision making model discussed above

PIPS is defined by the following process:

1. Contractors generate their performance criteria and select their references.

Table III Revised Price Example

Performance criteria	Alternatives A	B	C	Owner weights
1. Cost	$201	$200	$199	0.60
2. Service period	25	12	10	0.20
3. Percentage of roofs not leaking	99%	98%	80%	0.20

Table IV Revised Options

	Alternatives			
Performance criteria	A	B	C	Owner weights
2. Service period	25	12	10	0.20
3. Percentage of roofs not leaking	99%	98%	80%	0.20

2. PBSRG then gets the referenced past owners to rate the contractors based on the criteria.
3. Key personnel also are rated.
4. No contractors are eliminated. Contractors decline to participate if they are not capable.
5. Contractors are then asked to submit a management plan that identifies the risk to the owner (not on time, not on budget, and not meeting the quality expectations due to the previous two), how the project will be completed, a detailed cost breakout, and a construction schedule.
6. The management plan and key personnel are rated on a relative basis on risk reduction.
7. The past performance, management plan rating, and interview rating are inserted into the artificial intelligence model.
8. The model identifies which criteria make a difference and prioritizes the alternatives.
9. The best alternative is then asked to reverify their proposal with all the key elements, resolve any differences, and sign a contract that they have described.

The results of the 300 tests when compared with the existing delivery structure of construction is significant. The implementation of IMT resulted in:

1. Creating an information environment (the performance information is posted on the Internet)
2. Minimizing decision making by the user
3. Allowing decisions to be made by the parties at risk
4. Minimizing management requirements, paperwork, and inspection
5. Minimizing construction problems
6. Increasing performance and customer satisfaction
7. Providing the first documentation of increased performance over time in the construction industry

It is interesting to note that not all users of the process were receptive to the reduction of workload. Many participants were uncomfortable about not controlling the environment. However, after the process was complete, it was identified that the additional management control functions were not required. This identified one of the main challenges of implementing a information measurement system: the social adaptation to allowing the information to control the event.

XI. CONCLUSIONS

The measurement of information provides the method of decision making in the information age. Information technology will not only carry information from one point to another, but also create the information. The information will then replace decision making. The measurement of data by artificial intelligent decision makers can increase the value and performance of services. It can minimize the risk of false expectations. It can identify the value of alternative solutions. It can also prioritize the alternatives and select the alternative with the lowest risk (not

Table V Revised Relative Distancing

	Alternatives			
Performance criteria	A	B	C	Owner weights
2. Service period	0.000	0.493	0.569	0.20
3. Percentage of roofs not leaking	0.000	0.001	0.010	0.20
Total	0.000	0.494	0.579	

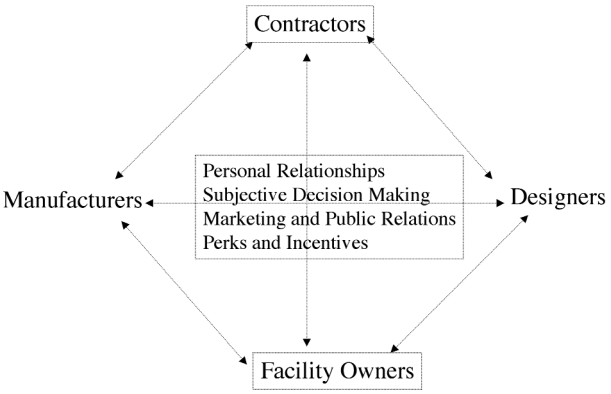


Figure 10 "Status quo" information environment.

meeting expectations). It will be easier to understand by people with different processing speeds, the constraint which causes subjectivity and miscommunication. The major impact of the measurement of information is the decision making by artificial intelligent systems.

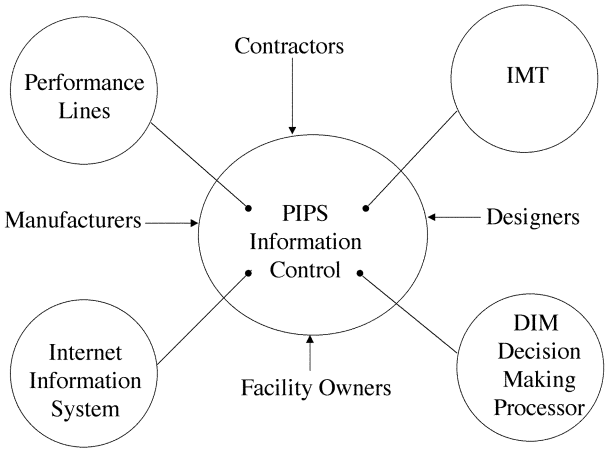


Figure 11 New information environment.

SEE ALSO THE FOLLOWING ARTICLES

Data, Information, and Knowledge • Decision-Making Approaches • Decision Theory • Expert Systems Construction • Future of Information Systems • Hybrid Systems • Information Theory

BIBLIOGRAPHY

Bennett, D. J. (1998). *Randomness,* Cambridge, MA: Harvard Univ. Press.

Buckingham, M., and Coffman, C. (1999). *First, break all the rules.* New York: Simon & Schuster.

Davies, P. (1992). *The mind of God.* New York: Orion Production.

Deming, W. E. (1986). *Out of the crisis.* Cambridge, MA: MIT Press.

Feynman, R. P. (1995). *Six easy pieces: Essentials of physics.* Reading, MA: Addison–Wesley.

Hawking, S. (1988). *A brief history of time.* New York: Bantam Books.

Herrnstein, R., and Murray, C. (1994). *The bell curve—Intelligence and class structure in American life.* New York: Free Press.

Kashiwagi, D. (2001). *Information measurement theory.* Tempe, AZ: Arizona State University.

Kosko, B. (1993). *Fuzzy thinking, the new sciencce of fuzzy logic.* New York: Hyperion.

Laplace, P.-S. (1886). *Oeuvres completes de Laplace,* Vol. 7, book II. Paris: Gauthier–Villars.

Maugh, T. H. (1998). New advice for better marriage. *The Arizona Republic,* Issue A, No. 10, February 22.

Neumann, J. von. (1951). Various techniques used in connection with random digits, in *John von Neumann, collected works,* Vol. 5. New York: Macmillan.

Penrose, R. (1989). *The emperor's new mind: Concerning computers, minds and the laws of physics.* Oxford: Oxford Univ. Press,

Pitino, R. (1998). *Success is a choice: Ten steps to overachieving in business and life.* New York: Broadway Books.

Walsh, B. (1998). *Finding the winning edge.* Champaign, IL: Sports Publishing, Inc.

Whitman, D. (1997). A reality check on welfare reform. *US News Online,* www.usnews.com/usnews/ISSUE/WELF.HTM.

Zadeh, L. (1993). *Fuzzy logic for the management of uncertainty.* J.B. Wiley & Sonsal Acquisition Regulation (27 Dec 1999), FAC 97-15, 15.101.

Information Theory

Patrick Verlinde
Royal Military Academy, Belgium

GLOSSARY

bit The unit of *information* if the logarithm is taken in base 2.
capacity The maximal value of the *transinformation*.
discrete Markov source of order "n" A source for which the emission of a symbol at each step depends on the symbols emitted during the *n* previous steps.
discrete memoryless source A source for which all the source symbols are produced with a fixed set of probabilities, and for which successive symbols are statistically independent.
entropy of a source The expected value of the *information* delivered by the source.
information (of an event) Minus the logarithm of the probability of occurrence (of that event).
information rate The product of the *entropy* of the source with the average number of symbols emitted by the source per unit of time.
mutual information (between two events) The *information* delivered by the outcome of one event on the occurrence of the other one.
redundancy (of a source) Characterizes the deviation between the real *entropy of a source* and its maximal entropy (which is obtained when all the symbols are equiprobable).
transinformation The expected value of the *mutual information* between two events.

I. INTRODUCTION

While the first half of the 20th century was characterized by the arrival of telecommunication systems that allow the transmission of signals in an analog form (audio signals, television, etc.), the second half of that century was characterized by the tremendous development of systems in which the transmitted information is coded in a digital form.

By this coding the real nature of the information signal is pushed to the background, which means that the same system can transmit simultaneously an information flux of very different nature: data, audio signals, television, etc. This development has obviously only been made possible by the use of more and more powerful integrated circuits.

Although it is mainly during the last 20 years that the truly operational digital systems have been developed, the theoretical foundations that were at the base of these developments date back to the mid century. It was in 1948 that Claude Shannon published his famous articles on the mathematical theory of communications. Shannon was interested in the mathematical formulation of two dual problems, namely:

1. How can one realize an optimal conversion from analog to digital signals?
2. How can one realize an error-free transmission of digital signals over a transmission channel which is subject to perturbations (interferences, perturbations, industrial noise, electronic noise, etc.)?

These two problems can be described more accurately by referring to Fig. 1. Figure 1 represents the basic elements of a digital telecommunication system that connects two points (an information source and a destination for this information).

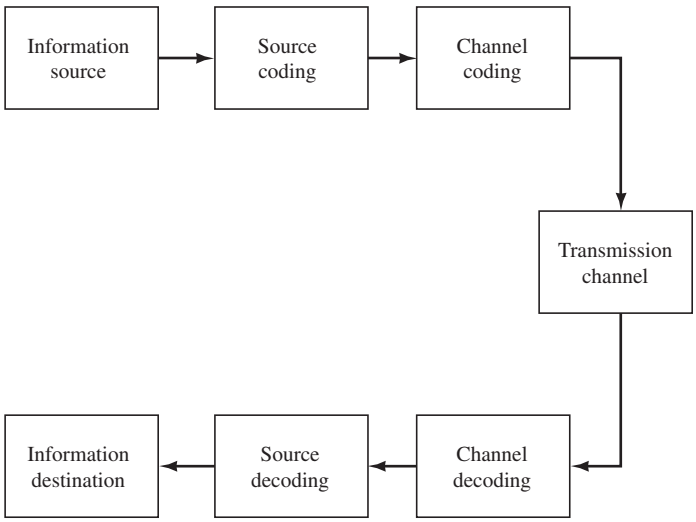


Figure 1 Basic elements of a digital telecommunication system.

The first problem concerns on one hand the set of source and source coding and on the other hand the set of source decoding and destination, which is linked with the first set. This link is realized through an *idealized* transmission channel, in the sense that it is supposed to be free of noise.

The information source generates information in a digital form (i.e., using a discrete source alphabet denoted by A) or in an analog form.

In the first case the source coding consists of realizing a one-to-one correspondence between the elements of the source alphabet A and a set of words (called *codewords*), constructed by means of a code alphabet B. The coding is called *binary* if the alphabet B contains only two symbols (noted 0 and 1). In general the coding is called q-ary if the alphabet B consists of q elements.

The Morse code, for instance, uses a ternary alphabet:

$$B = \{\cdot, -, \text{space}\}$$

Since the coding is one-to-one and since the channel is noise free, the decoding can happen unambiguously and without errors by performing at the receiver end the inverse operations of those executed at the transmission end.

The real problem is to *reduce* maximally the "representation" of the transmitted information. This reduction (or data compression) is calculated by the symbol rate which is required for the complete representation of the source signal and thus finally for its restitution during the source decoding. This minimal rate with which the stochastic series emitted by the source can be transmitted through a noise-free transmission channel to be consequently reconstructed without errors at the destination is linked with a fundamental parameter of the source, called the *entropy* of the source. This notion will be defined further.

If on the contrary the source is of the analog type (discrete or continuous in time), then the signal generated by the source cannot be represented by a finite series of digital symbols coming from a finite alphabet. In other words, the analog-to-digital conversion (which is a form of coding in which the source alphabet A contains an infinite number of elements while the code alphabet is finite, which excludes every one-to-one correspondence) will without any doubt give rise to a loss of information. This means that the source decoding will never allow us to reconstruct the signal emitted by the source. There will exist a *distortion* between the source signal and the reconstructed signal.

Here the problem can be stated using the following terms: If a maximal value of the distortion is given, how can one then reduce maximally the symbol rate at the output of the source coder, with the guarantee for the destination that the distortion will be smaller than the imposed maximum? This problem is not dealt with in this article.

The second problem concerns the operations of channel coding and decoding, in which we now assume that the transmission channel is noisy.

This supplementary coding–decoding has as its goal the removal of the effects of the noise in the transmission channel. Without this supplementary coding–decoding, this noise would undoubtedly cause decoding errors at the receiver side.

In the absence of this supplementary coding it would be impossible to reconstruct at the destination

side without error the message emitted by the source, even in the case of a digital source, which is in conflict with what we have said above in the case of the ideal transmission channel. In other words, the goal of this channel coding is to realize a non-noisy transmission channel, which is composed of the real transmission channel preceded by the channel coder and followed by the channel decoder.

The study of this problem by Shannon has led him to formulate a theorem in which a fundamental parameter of the transmission channel appears, namely, the *capacity* of the transmission channel. This theorem says in essence that, on condition that the symbol rate at the input of the transmission channel is smaller than the capacity of this transmission channel, it is possible to find coding and decoding operations which lead to an error-free reconstruction of the emitted sequence in the receiver. Shannon however did not specify the precise nature of these coding and decoding operations.

From the theories on source and channel coding it follows that, for a digital source, it is possible to transmit without errors data between a source and a destination over a noisy transmission channel, under certain conditions on the data rate and if one uses suitable codes.

For an analog source this stays valid, but only within the limits of an acceptable (determined beforehand) distortion which determines a maximum rate at the output of the source coder. This maximum rate has to be smaller than the capacity of the transmission channel.

These fundamental theories have given birth to a new branch in mathematics and in telecommunication theory, known under the name of *information theory*.

Besides the original articles of Shannon, other often cited references in information theory are given in the Bibliography.

II. INFORMATION AND HOW TO MEASURE IT

A. (Self-)Information Related to an Event—Entropy

The problem that needs to be answered is this: Why do some messages contain more information than others? Once it is determined what the information delivers, we will have to find out how to quantify it. Let us compare the two following messages:

"Yesterday the sun did shine in Las Vegas."

"Canada declared war on the United States."

Knowing the climate in Las Vegas it is clear that the second message (which contains approximately the same number of symbols as the first one) conveys more information than the first one. This is due to the unexpected nature of the corresponding event. Furthermore the announcement of a certain event (the milk is white) contains no information at all. This means that each message that contains information will have to be described by random quantities.

The quantities that measure the information will need to have the following properties:

1. The quantity of information of an event (e.g., a message) depends on the degree of uncertainty of that event.
2. The more unlikely an event, the more information it contains.
3. If two events are independent then it seems reasonable to demand that the knowledge of the two messages leads to a delivery of information which is equal to the sum of the information carried by each message individually.

These three properties did lead to the following expression being chosen as a measure for the information of an event α—denoted $I(\alpha)$—which is likely to occur with a probability $p(\alpha)$:

$$I(\alpha) = -\log p(\alpha)$$

Property 3 is obviously met. To see this let there be two independent events α and β with respective probabilities $p(\alpha)$ and $p(\beta)$. One obtains then:

$$\begin{aligned} p(\alpha \cap \beta) &= p(\alpha)p(\beta) \\ I(\alpha \cap \beta) &= -\log [p(\alpha)p(\beta)] \\ &= I(\alpha) + I(\beta) \end{aligned}$$

where $\alpha \cap \beta$ denotes the logical intersection of the two events α and β.

The choice of the base of the logarithm is not very important; this is only a scaling problem. Most of the time the base chosen is 2 and the unit of information is then called the *bit* (sometimes also called *logon*). When base 10 is chosen, the corresponding unit is called the *Hartley* and when base e is chosen (natural logarithm), this unit is called the *Shannon* or sometimes also the *nat*. In what follows we will always use base 2 without explicitly repeating it.

Example 1

A student has a 50% chance of passing an examination. The announcement that he passed gives information equal to 1 bit.

Example 2

A source has a binary alphabet, which is noted {0,1}. Every T seconds the source delivers a symbol of the alphabet. The two symbols are equally likely. Let E be the following event: "the source delivered a sequence of n symbols, alternatively 0 and 1." One then has $I(E) = -\log(1/2^n) = n$ bits. By the way, this result is valid whatever the considered sequence of n symbols. One sees that in this case the number of information bits corresponds to the number of "bits" (short for "binary digit"), the term used in information technology to designate a binary symbol. However if $p(0) \neq p(1)$, then $I(E) \neq n$ bits [in fact $I(E) < n$ bits].

Instead of speaking about the information delivered by the outcome of an event, one can speak about the information with respect to a random variable. Therefore we consider a random variable X which can take the values $\{a_1, \ldots, a_k, \ldots, a_m\}$ with respective probabilities $\{p_X(a_1), \ldots, p_X(a_k), \ldots, p_X(a_m)\}$. If X takes the value a_k, then the delivered information—the *(self-)information* of the event $x = a_k$—is:

$$I(a_k) = -\log p_X(a_k).$$

In short form one can write more conveniently:

$$I(x) = -\log p(x).$$

If one considers the mean value of $I(x)$, then one gets a number which is called the *entropy* H of the random variable X:

$$H(X) = -\sum_{k=1}^{m} p_X(a_k) \log p_X(a_k)$$

which in short form becomes:

$$H(x) = -\sum_{x} p(x) \log p(x)$$

Remark

After the introducing comments of this paragraph, it is clear that one can also say that $I(x)$ measures the *a priori uncertainty* of the fact that X can take on the value x. Once this value is known, this *uncertainty* becomes *information*.

B. Mutual and Conditional Information

One also introduces the notions of *mutual* and *conditional* information. Therefore one considers two random variables X and Y, which are *not* considered to be independent. These variables can take respectively the values $\{a_k, k = 1,2, \ldots\}$ and $\{b_j, j = 1,2, \ldots\}$. This pair of random variables (X,Y) is characterized by a joint probability distribution $p_{XY}(a_k,b_j)$.

Let us first consider the quantity of information delivered by the outcome of the event $y = b_j$, with respect to the probability of the event $x = a_k$. The original (*a priori*) uncertainty of the event $x = a_k$ (i.e., without having received the information that $y = b_j$), is equal to $-\log p_X(a_k)$, while the final (*a posteriori*) uncertainty (i.e., knowing that $y = b_j$) is equal to $-\log p_{X|Y}(a_k|b_j)$. The difference in uncertainty between the two described situations equals:

$$I_{XY}(a_k,b_j) = -\log p_X(a_k) - (-\log p_{X|Y}(a_k|b_j))$$

$$= \log \frac{p_{X|Y}(a_k|b_j)}{p_X(a_k)}$$

$I_{XY}(a_k,b_j)$ is thus the information delivered by the outcome of the event $y = b_j$ concerning the event $x = a_k$.

One easily shows that:

$$I_{YX}(b_j,a_k) = \log \frac{p_{Y|X}(b_j|a_k)}{p_Y(b_j)} = I_{XY}(a_k,b_j)$$

where $I_{XY}(a_k,b_j) = I_{YX}(b_j,a_k)$ is called the *mutual information* between the events $x = a_k$ and $y = b_j$.

In short form we write:

$$I(x,y) = \log \frac{p(x|y)}{p(x)}$$

If one takes the expected value of $I(x,y)$ for all possible values of the pair (x,y), then one gets the expected value of the mutual information between X and Y (sometimes called *transinformation*):

$$I(X,Y) = \sum_{x} \sum_{y} I(x,y)p(x,y)$$

$$= \sum_{x} \sum_{y} \log \frac{p(x|y)}{p(x)} p(x,y)$$

Note the difference in notation between $I(X,Y)$ and $I(x,y)$. The transinformation $I(X,Y)$ is a number that characterizes the pair of random variables (X,Y) together, while $I(x,y)$ characterizes a particular realization (x,y).

For two random variables X and Y one can also define the so-called *conditional* information of the event $x = a_k$, knowing that the event $y = b_j$ did occur:

$$I_{X|Y}(a_k,b_j) = \log \frac{1}{p_{X|Y}(a_k|b_j)}$$

or in short form:

$$I(x|y) = \log \frac{1}{p(x|y)}$$

It is the information that has to be delivered to an observer in order to determine the event $x = a_k$ com-

pletely, in the case that this observer knows already that $y = b_j$. In other words, it is the remaining uncertainty with respect to the event $x = a_k$, when it is known that $y = b_j$. Since this conditional information depends on the events $x = a_k$ and $y = b_j$, one can calculate its expected value by means of the probabilities $p_{XY}(a_k,b_j)$. In this manner one obtains the *conditional entropy* $H(X|Y)$, which is also called the *equivocation*:

$$H(X|Y) = \sum_{x,y} I(x|y)\,p(x,y)$$
$$= -\sum_{x,y} p(x,y)\,\log\,p(x|y)$$

One can easily check the following equality:

$$I(x,y) = I(x) - I(x|y)$$

The information delivered on x by the outcome of y is equal to the self-uncertainty of the outcome of x, reduced with the uncertainty of this outcome if one knows that y has already taken place. The equivalent of the last equation, after having calculated the respective expected values, is given by:

$$I(X,Y) = H(X) - H(X|Y).$$

The expected value of the information delivered on X by the outcome of Y is equal to the expected value of the *a priori* uncertainty on X, reduced with the remaining (*a posteriori*) uncertainty on X after Y has been observed. $I(X,Y)$ is *therefore* called the transinformation.

Theorem

$I(X,Y)$ is a non-negative quantity. This means that $H(X) \geq H(X|Y)$, with the equal sign if X and Y are two independent random variables.

C. Example of the Binary Symmetric Channel

The random variables X and Y can represent for instance, respectively, the emitted and received symbols on both sides of a noisy channel. The possible values of X and Y belong to the alphabet used. We shall suppose here that the channel is binary $A = \{0,1\}$. This channel has furthermore no memory and is characterized by the error probability ϵ (see Fig. 2):

$$p_{X|Y}(0|0) = p_{X|Y}(1|1) = 1 - \epsilon$$
$$p_{X|Y}(0|1) = p_{X|Y}(1|0) = \epsilon$$

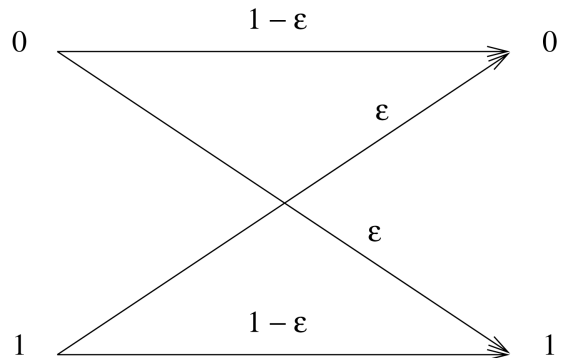


Figure 2 Transition probabilities of a binary symmetric channel.

Let us suppose, for instance, that the symbols at the entrance of the channel are equiprobable:

$$p_X(0) = p_X(1) = \frac{1}{2}$$

Then we obtain

$$p_{XY}(0,0) = p_{XY}(1,1) = \frac{(1-\epsilon)}{2}$$
$$p_{XY}(0,1) = p_{XY}(1,0) = \frac{\epsilon}{2}$$

It can easily be shown that the symbols at the output of the channel are also equiprobable:

$$p_Y(1) = p_X(1)(1-\epsilon) + p_X(0)\epsilon = \frac{1}{2}$$

This leads to:

$$p_Y(0) = \frac{1}{2}$$

The mutual information is then equal to:

$$I_{XY}(0,0) = I_{XY}(1,1) = \log \frac{(1-\epsilon)}{\frac{1}{2}} = \log 2(1-\epsilon)$$
$$I_{XY}(1,0) = I_{XY}(0,1) = \log 2\epsilon$$

The conditional information is in this case equal to:

$$I_{X|Y}(0|0) = I_{X|Y}(1|1) = -\log(1-\epsilon)$$
$$I_{X|Y}(1|0) = I_{X|Y}(0|1) = -\log\epsilon$$

Case 1: $\epsilon = 0$

The transmission channel is not noisy. The value of the symbol at the output allows us to determine without ambiguity the symbol at the input. Under these conditions we find for the events $x = 0 (x = 1)$ and $y = 0 (y = 1)$:

$$I_{XY}(0,0) = I_{XY}(1,1) = 1$$
$$I_{X|Y}(0|0) = I_{X|Y}(1|1) = 0.$$

One knows that $I_X(0) = I_X(1) = 1$ bit. The mutual information is in this case equal to the self-information. The uncertainty on X is zero as soon as Y is known. The events $x = 0$, $y = 1$ and $x = 1$, $y = 0$ are impossible [fortunately since $I_{XY}(0,1) = I_{XY}(1,0) = \infty$).

Case 2: $\epsilon = \frac{1}{2}$

The transmission channel is noisy and the value at the input does not influence the value at the output. X and Y are independent. Under these conditions one obtains for the events $x = 0 (x = 1)$ and $y = 0 (y = 1)$:

$$I_{XY}(0,0) = I_{XY}(1,1) = 0$$

$$I_{X|Y}(0|0) = I_{X|Y}(1|1) = 1$$

The value taken by Y (for instance, $y = 1$) does not remove any uncertainty of the event $x = 1$. For the events $x = 0$, $y = 1$ or $x = 1$, $y = 0$, the same results are found.

Case 3: $0 < \epsilon < \frac{1}{2}$

Let us take $\epsilon = 0.1$ as an example. We consider different subcases:

Event 1: $x = 0$, $y = 0$ or $x = 1$, $y = 1$

$$I_{XY}(0,0) = I_{XY}(1,1) = 0.85$$

$$I_{X|Y}(0|0) = I_{X|Y}(1|1) = 0.15$$

The information delivered by the outcome of $y = 1$, for instance (0.85 bit), ensures us that the uncertainty *a posteriori* of the event $x = 1$ is reduced to only 0.15 bit, while it started with an *a priori* uncertainty of 1 bit.

Event 2: $x = 0$, $y = 1$ or $x = 1$, $y = 0$

$$I_{XY}(1,0) = I_{XY}(0,1) = \log 0.2 = -2.3$$

$$I_{X|Y}(1|0) = I_{X|Y}(0|1) = -\log 0.1 = 3.33$$

This can be interpreted as follows: the channel is not very noisy and the fact that, for instance, $y = 1$, does make the event $x = 0$ less likely. The mutual information is negative, which means that the *a posteriori* uncertainty of the event $x = 0$ (in the case that $y = 1$) is bigger than the *a priori* uncertainty.

Case 4: $1 > \epsilon > \frac{1}{2}$

This case is treated in exactly the same way as case 3.

III. INFORMATION SOURCES AND SOURCE CODING

A. Introduction

To start off, we will only consider discrete sources, i.e., sources which use a finite alphabet. The source outputs thus a sequence of symbols which are chosen from this alphabet. This choice can happen according to a very complex statistic. We however are going to limit ourselves for most of the time to a specific case which is simple: the discrete memoryless source. With this type of source the probability of occurrence of a symbol does *not* depend on the preceding symbols.

B. Discrete Memoryless Sources

Let A be a finite source alphabet, built out of the symbols $\{a_j, j = 1, \ldots, m\}$. Let X_i be the random variable which represents the ith emitted symbol. The possible values for X_i are $\{a_j, j = 1, \ldots, m\}$.

The source is called discrete and memoryless if the source symbols are produced with a fixed set of probabilities, and if successive symbols are statistically independent.

C. Entropy of Discrete Memoryless Sources

The entropy H of a discrete memoryless source is the expected value of information delivered by the source which is characterized by the random variable X_i:

$$H(X) = \sum_{j=1}^{m} p_X(a_j) I_X(a_j) = -\sum_{j=1}^{m} p_X(a_j) \log p_X(a_j)$$

which in short form reduces to:

$$H(X) = -\sum_{x} p_X(\mathrm{x}) \log p_X(x)$$

Note that we are dealing here with the expected value of a set. It is the expected value of the information delivered by a countably infinite set of identical sources with the emission of the ith symbol. This expected value is by the way independent of i; that is why this index has been left out in the previous formula.

To understand the meaning of entropy, we will have a look at two properties of discrete memoryless sources.

D. Properties of Discrete Memoryless Sources

Property 1

One can show that the discrete memoryless source with *maximal* entropy is the one for which the m symbols are equiprobable:

$$p_X(x) = \frac{1}{m} \forall x$$

For the specific case of the *binary* memoryless source, we have that $A = \{0,1\}$ and we suppose that $p(0) = p$ and $p(1) = 1 - p$. This leads to the following entropy:

$$H = p \log \frac{1}{p} + (1 - p) \log \frac{1}{1 - p} = F(p)$$

This function $F(p)$ is ≤ 1, with $F(\frac{1}{2}) = 1$ (see Fig. 3).

Property 2

One can show that the expected value of the information delivered by a sequence of n consecutive emissions of a discrete memoryless source $H_n(\overline{X})$ equals n times the entropy of the source $H(X)$ (which is the expected value of the information delivered by one symbol):

$$H_n(\overline{X}) = nH(X) \qquad \text{with} \qquad \overline{X} = (X_1 X_2 \ldots X_n)$$

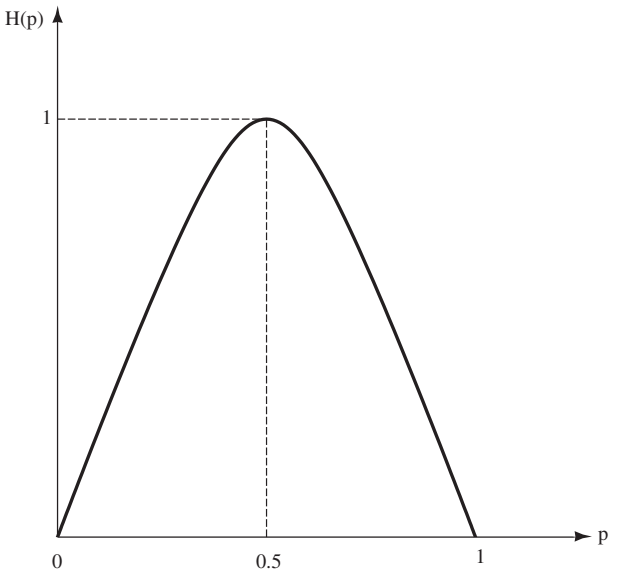


Figure 3 Entropy as a function of the probability of occurrence.

One can also show that if the n emissions are not independent (source with memory):

$$H_n(\overline{X}) < nH(X)$$

The case of the memoryless source gives an upper bound for the expected value of the information delivered by the source by the emission of a sequence of n symbols.

E. Information Rate and Redundancy of Information Sources

Sometimes it is useful to couple the information with time. In that case the *information rate* of the source is defined as the product of the entropy of the source (which is in fact the expected value of the information per source symbol) with the average number of symbols emitted by the source per second. If the average duration of a symbol is τ, then the expected value of the information rate (noted H_τ) of a source with entropy H is given (in bits/second) by:

$$H_\tau = \frac{H}{\tau}$$

The *redundancy* R_S of a source characterizes the deviation between the real entropy of that source and its maximal entropy (which is obtained when all the symbols are equiprobable):

$$R_S = H_{\max} - H$$

where $H_{\max} = \log m$, with m the number of symbols in the source alphabet.

F. Discrete Markov Sources

One can generalize the concept of the discrete memoryless information source by supposing that the emission of a symbol at each step i depends on the symbols emitted during the n previous steps. Such an information source is called a (discrete) *Markov source of order n*. The subsequent emissions of the source are no longer independent. An example of this kind of source is that of a language. In the English language for instance, a "Q" is often followed by a "U," while this is never true for an "X." This correlation between symbols is a measure for the redundancy of the language, which guarantees a certain "resistance" against transmission errors or incidental omissions of certain symbols. The notion of entropy which was introduced in the framework of the discrete memoryless source

can be extended to cover the case of the discrete Markov source. This lies however outside the scope of this article. We can remark that such a Markov source does have an entropy which is lower than the one of a memoryless source. This entropy reduction is of course due to the correlation between the consecutive symbols. Let us consider as an example the English language (26 letters + space = 27 symbols) in different cases.

Case 1

All letters are independent and equiprobable:

$$H = \log 27 = 4.75 \text{ bit}$$

Case 2

All letters are independent but appear with their real probability of emission:

$$H = 4.08 \text{ bit}$$

Case 3

The letters are no longer independent but different types of correlation are taken into account:

1. Correlation between pairs of letters: $H = 3.3$ bit
2. Correlation between groups of 3 letters: $H = 3.1$ bit
3. Taking into account all possible forms of correlation, Shannon estimated the real entropy of the English language on approximately 1 bit.

G. Discrete Time Analog Information Sources

If a source emits a sampled analog signal, then it is no longer characterized by a random series with a discrete alphabet, but by an alphabet that possesses an infinite number of values. We will only say here that in this case we can also define the different forms of entropy (self, mutual, conditional) that have been presented in the case of discrete sources.

H. Definition of Source Coding

Given a finite source alphabet A and a finite code alphabet B, then *source coding* can be defined as a rule that allows us to allocate to each source symbol a codeword, which consists of symbols from the code alphabet B. The set of used codewords constitutes the code. More formally a code can be defined as the set of codewords $\{K(a_1), \ldots, K(a_m)\}$, where:

- $A = \{a_1, \ldots, a_m\}$ is the source alphabet, which contains m symbols.
- K is the function which allocates a codeword to each source symbol.

In the case where the alphabet B is binary, one says that the code is binary.

One distinguishes between codes with codewords of variable length and codes with codewords of fixed length (block codes). The former type of code allows us to allocate the shortest codewords to the symbols of the source alphabet with the highest probability of occurrence. The Morse code for instance ($B = \{-,.,space\}$) allocates the codeword (.) to the letter e (which is the most frequently used letter in English), while the letter q is coded by (--.-). If the symbols of A are emitted at a fixed rate, then this also has to be the case for the corresponding codewords. This could lead to waiting times for the shorter codewords. This waiting problem does not exist for the fixed length codes. An example of a fixed length code is the ASCII code, in which $2^7 = 128$ source symbols (A, B,. . .) are coded in blocks of 8 binary code symbols (7 information symbols + 1 symbol for parity checking).

I. Fixed Length Source Codes

Let there be a discrete m-ary memoryless source with entropy H. We shall treat the general case in which the source words of length L are coded using a code alphabet consisting of q symbols. The codewords have a fixed length N. This means that there are q^N different codewords. To be able to allocate a codeword to each sequence of L source symbols, we need to have:

$$q^N \geq m^L$$

$$\frac{N}{L} \geq \frac{\log m}{\log q}$$

where N/L represents the number of code symbols per source symbol. This number must be at least equal to $\log m/\log q$. If one wishes to use less than $\log m/\log q$ code symbols per source symbol, then one has to accept that decoding the codeword will not always be possible. One can show that by choosing L such that it is sufficiently large, it is possible to make the probability of this impossibility of decoding arbitrarily small, and to bring the number of code symbols per

source symbol as close to $H/\log q$ as one wishes. This is based on a theorem of which the interpretation is the following one.

Since the codewords do form another representation of the source sequences, they need to have the same entropy as the source sequences. We already saw that $\log q$ is the maximal entropy per symbol which can be allocated to a sequence of symbols taken from a q-ary alphabet. This maximal entropy is reached when the symbols are independent and equiprobable. The considered theorem then simply says that it is possible to code the sequences of symbols of a discrete memoryless source in such a manner that the entropy per code symbol takes its maximal value.

Remark

The fact that coding very long sequences of source symbols is not very practical is not taken into account here.

J. Variable Length Source Codes

1. Generalities

Let $A = \{a_1, a_2, \ldots, a_m\}$ be a discrete memoryless source with probability distribution $\{p(a_1), p(a_2), \ldots, p(a_m)\}$. Let us suppose first that each element of the alphabet A has to be represented by a codeword composed of code symbols that belong to a q-ary alphabet $\{x_1, \ldots, x_q\}$. Let l_k be the number of symbols of the codeword associated with a_k. Later we shall consider the case in which we no longer code the *individual* source symbols, but *sequences* of sources symbols.

One wishes to minimize the expected value of the number of code symbols l_{av} per source symbol, with:

$$l_{av} = \sum_{k=1}^{m} p(a_k)\, l_k$$

Before determining a lower bound for l_{av}, let us look into some restrictions concerning the construction of codewords. First of all we will treat an example that highlights these constraints. Let us take a look at the codes represented in Table I. Codes I and II are not uniquely decodable:

$$\text{Code I: } 0 \rightarrow a_1 \text{ or } a_2$$

$$\text{Code II: } 00 \rightarrow a_1 a_1 \text{ or } a_3$$

2. Uniquely Decodable Source Codes

A code is uniquely decodable if, for each sequence of source symbols of finite length, the respective sequence of code symbols differs from each other code sequence. Although correct, this definition does not suggest immediately a means of verifying this property. Also, in the beginning one considers a more restrictive class of codes, which obviously satisfies the condition of unambiguous decoding. It concerns the *separable* codes, which fulfill the so called prefix condition.

Let $x = (x_1, \ldots, x_{lk})$ be a codeword. One calls the *prefix* of the codeword x the word $(x_1, \ldots, x_i)$, with $i \leq l_k$. A code is called separable (or a *prefix* code) if not one codeword is a prefix of another codeword. Codes I, II, and IV of Table I, do *not* satisfy this prefix condition. They are not prefix codes. The interesting property of prefix codes, besides the fact that they can be decoded without ambiguity, is that decoding can take place codeword per codeword, without taking into account the future codewords (*online* decoding or *instantaneous* decoding). So for example for code III:

$$0\ 1\ 1\ 1\ 1\ 0\ 0 \rightarrow a_1 a_4 a_2 a_1$$

Code IV however is, although not a prefix code, uniquely decodable. Indeed, the 0 indicates each time the beginning of a codeword (it serves, in a manner of speaking as a comma or a blank).

Code IV of Table I is that of a uniquely decodable *not* prefix code. The next example clarifies why the decoding procedure of a codeword cannot be executed without examining the next codewords. It is indeed sufficient to decode, for instance, 00010100 by decoding from the right to the left.

An elegant way of representing codes is the use of the so-called tree diagram. Figure 4 represents the tree

Table I Examples of Different Source Codes

Source symbols	$p(a_k)$	Code I	Code II	Code III	Code IV
a_1	0.500	0	0	0	0
a_2	0.250	0	1	10	01
a_3	0.125	1	00	110	011
a_4	0.125	10	11	111	0111

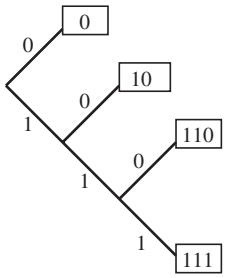


Figure 4 Tree diagram for code III.

diagram of code III of Table I. Each codeword corresponds to a path where each branch represents a symbol of the codeword. A codeword is found at the end of each path. For a prefix code all codewords are found at the end of a path. For a code that does not fulfill the prefix condition, some intermediate nodes do also form codewords.

3. Importance of Prefix Source Codes

It is clear that the uniquely decodable codes present a more difficult concept to handle than the prefix codes. We can limit our quest for codes with minimal length l_{av} to the prefix codes. To prove that this is true, two theorems are needed: the inequality of Kraft and the theorem of MacMillan. By combining these two theorems, it can be shown that there exists a uniquely decodable code for which the lengths of the codewords are given, if and only if there exists a prefix code with the same lengths for the codewords. This thus justifies the importance that has been given to prefix codes.

4. Fundamental Source Coding Theorem (1st Form)

Let there be a discrete memoryless source with entropy H. It is always possible to find a q-ary prefix code such that the expected value of the length l_{av} of the codewords which are allocated to the different source symbols satisfies the following condition:

$$l_{av} < \frac{H}{\log q} + 1$$

Furthermore, for each uniquely decodable code the following holds:

$$l_{av} \geq \frac{H}{\log q}$$

This theorem shows that the expected value of the length of the codewords l_{av} cannot be smaller than the entropy of the source divided by the maximal entropy of the code alphabet. This lower bound is reached when the symbols of the code alphabet are used in an equiprobable manner. In other words, the expected value of the information per symbol of the code alphabet $H(A)/l_{av}$ may not be greater than the maximal entropy of the code alphabet, namely, $\log q$:

$$\frac{H(A)}{l_{av}} \leq \log q$$

5. Capacity, Efficiency, and Redundancy of Source Codes—Optimal Codes

Let's start by briefly repeating the following notations: Source alphabet: $A = \{a_1, a_2, \ldots, a_m\}$ with probabilities $P = \{p(a_1), p(a_2), \ldots, p(a_m)\}$; q-ary code alphabet: $B = \{x_1, x_2, \ldots, x_q\}$.

a. Capacity C_S of a code

$$C_S \triangleq \max H(B) = \log q$$

b. Efficiency η of a code

$$\eta = \frac{(l_{av})_{\min}}{l_{av}}$$

where

$$(l_{av})_{\min} = \frac{H(A)}{\log q} = \frac{H(A)}{C_S} \qquad l_{av} = \frac{H(A)}{H(B)}$$

This leads to the following:

$$\eta = \frac{H(A)}{l_{av} \log q} = \frac{H(B)}{\log q}$$

c. Redundancy ρ of a code

$$\rho = 1 - \eta = \frac{l_{av} \log q - H(A)}{l_{av} \log q} = \frac{\log q - H(B)}{\log q}$$

Example

$$A = [a_1, a_2, a_3, a_4] \qquad P = \left[\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right]$$

from which we find $H(A) = 7/4$ bit per symbol.

Code 1. Alphabet $B = [0,1]$

$$a_1 \to 00 \qquad a_2 \to 01 \qquad a_3 \to 10 \qquad a_4 \to 11$$

from which we find that $l_{av} = 2$, $\eta = 0.875$, and $\rho = 0.125$.

Code 2: prefix code. Alphabet $B = [0,1]$

$$a_1 \to 0 \qquad a_2 \to 10 \qquad a_3 \to 110 \qquad a_4 \to 111$$

from which we find that $l_{av} = 1.75$, $\eta = 1$, and $\rho = 0$.

The second code is better than the first one, since the entropy of the code alphabet has been brought to its maximal value. One can indeed show that in the second code the symbols of the code alphabet 0 and 1 are equiprobable in the coded sequences.

6. Absolutely Optimal Codes

Absolutely optimal codes are codes for which

$$l_{av} = (l_{av})_{\min} = \frac{H(A)}{\log q}$$

This implies that the symbols of the code alphabet are being used with the same probability. From this one can of course see that their efficiency η is 100%. Since furthermore the subsequent symbols of the codewords are independent, we have:

$$p(a_i) = \left(\frac{1}{q}\right)^{l_i}$$

and since

$$\sum_{i=1}^{m} p(a_i) = 1$$

it follows that

$$\sum_{i=1}^{m} q^{-l_i} = 1$$

It is this last relation which has to exist between the lengths $l_1, \ldots, l_m$ of the codewords of an absolutely optimal q-ary code.

7. Fundamental Source Coding Theorem (2nd Form)

Even better results are obtained if, instead of coding the symbols emitted by the source individually, one associates codewords with sequences of L source symbols. In such a case we can formulate the second form of the source coding theorem as follows.

Let there be a discrete memoryless source with entropy H. It is always possible to find a q-ary prefix code such that the expected value of the length $(l_{av})_L$ (per source symbol) of the codewords allocated to the sequences of L source symbols satisfies:

$$\frac{H}{\log q} \leq (l_{av})_L < \frac{H}{\log q} + \frac{1}{L}$$

Furthermore the left inequality must be satisfied for each uniquely decodable code.

This theorem shows that by coding sequences of source symbols (instead of coding source symbols individually), the expected value of the information per symbol of the code alphabet $H(A)/(l_{av})_L$ can be brought as close as one wants to the capacity of the code $\log q$ (this means that the code is absolutely optimal).

This theorem, presented in the case of a discrete memoryless source, can be extended to the case of Markov sources of arbitrary order.

Although this theorem guarantees the existence of very efficient codes, it does not give a procedure to obtain those codes. Nevertheless it gives a very nice theoretical result, which allows us to judge the degree of efficiency of a given code.

In practice it is not conceivable that L (i.e., the number of source symbols grouped in the coding) could tend to infinity, and it is important to build codes with an efficiency that is as large as possible for a given L. In general such a code is then not absolutely optimal. One says that a code is optimal (not absolutely), when for a given $H(A)$ and a fixed L (often $L = 1$), the obtained efficiency is as large as possible. A method for constructing such optimal codes has been developed by Huffman.

IV. CHANNEL CODING

A. Goals

The previous section treated the problem of the representation of the emissions of a discrete source using codewords. This has led to a very striking interpretation of the notion of entropy in terms of the expected value of the length of the codewords.

Now we tackle the problem of the *transmission* of the codewords coming from the source, by means of a noisy transmission channel, and this from the viewpoint of information theory. Here we shall show the importance of the notion of mutual information. Furthermore we will find the ultimate theoretical bounds that can be achieved in realizing these transmissions.

To arrive at these results, it is good to first correctly describe the transmission channel.

B. Channel Models

A transmission channel can be described by giving the set of input signals that may be used at the entrance of the channel, the set of output signals, and for each input signal the conditional probability that it will be received as a specific output signal (and this for each possible output signal).

The simplest transmission channel is the one for which the input and output alphabets are finite and

for which the output at a certain moment in time depends statistically only on the corresponding moment in time at the input. This is the case of the discrete memoryless channel.

Another case is the one in which the input and output signals are discrete in time, but have continuous amplitudes. In other words, the input and output alphabet are infinite (analog signals at discrete moments in time).

This transmission channel is also memoryless if the statistic of the output symbols at a certain moment in time depends only on the input signal at the corresponding moment in time, and not on the preceding input signals.

Finally one has to consider the case in which the input and output signals are continuous both in time and in amplitude (analog signals in continuous time).

The most complex cases to deal with are those in which the transmission channels do have memory. Transmission channels that present intersymbol interference (ISI: as a consequence of limitations in bandwidth or of multipath propagation) or phenomena of signal extinction (fading) are examples of channels that do have memory.

We present here the case of the discrete memoryless channel.

C. Discrete Memoryless Channels

1. Generalities

A discrete memoryless transmission channel is characterized by an input alphabet X with K elements, and an output alphabet Y with J elements. Furthermore there is also a set of conditional probabilities $p(y|x)$, with $y \in Y$ and $x \in X$. From this it follows that for each given input sequence $\overline{x} = (x_1, \ldots, x_N)$, the conditional probability of an arbitrary output sequence $\overline{y} = (y_1, \ldots, y_N)$ is equal to:

$$p_N(\overline{y}|\overline{x}) = \prod_{n=1}^{N} p(y_n|x_n).$$

The most common case is that of the binary symmetric transmission channel, which was already presented in Section II.C. In that case we obtain $X = Y = \{0,1\}$ and the conditional probabilities are:

$$p(0|0) = p(1|1) = 1 - \epsilon$$
$$p(1|0) = p(0|1) = \epsilon$$

2. Mutual Information and Channel Capacity

Let us start by remarking that in the definition of a transmission channel, nothing is said about the probability distribution of the random variable X, which we will call here $q(x)$ instead of $p_X(a_k)$ as we did previously, for notational simplicity reasons. The mutual information between the events $Y = y$ and $X = x$ has been defined before as:

$$I(x,y) = \log \frac{p(y|x)}{p(y)}$$

with

$$p(y) = \sum_x q(x)\, p(y|x)$$

The expected value of the mutual information between X and Y (transinformation) is nothing other than:

$$I(X,Y) = \sum_y \sum_x p(y|x)\, q(x) \log \frac{p(y|x)}{\sum_{x'} q(x')\, p(y|x')}$$

This notion not only depends on the characteristics of the transmission channel (conditional probabilities), but also on the statistics of the source through $q(x)$. Let us remark that the relative frequencies of the channel input symbols can be adjusted by an encoder. To see if this is possible, it is sufficient to take a closer look at codes III and IV in Table I; the last line there shows that the probabilities of having a "0" or a "1" in the codeword are not the same! This shows that by changing the code we are also able to change $q(x)$. Bearing furthermore in mind that we want a quantity which only characterizes the transmission channel, we consider the maximal value of the transinformation $I(X,Y)$ that is calculated over all possible distributions $q(x)$. This maximal value is called the *capacity* of the transmission channel and it is noted traditionally as C:

$$C \triangleq \max_{q(x)} I(X,Y) \qquad \text{(bits/symbol)}$$

Let us denote with $q_C(x)$ the specific probability distribution that maximizes the transmitted transinformation. To obtain the capacity of the channel with a transmission, it is adequate to realize a statistical adaptation of the source to the transmission channel. This can be realized with a source code that transforms the primary random variable of the source into a secondary random variable, which is characterized by the optimal probability distribution $q_C(x)$. The capacity of a transmission channel can also be expressed in bits/second:

$$C_\tau \text{ (bits/second)} = \frac{C(\text{bits/symbol})}{\bar{\tau}(\text{seconds/symbol})}$$

where $\bar{\tau}$ is the average duration of a symbol in seconds.

The *redundancy* of a transmission channel is expressed, in analogy with the redundancy of a source, as

$$R_C \triangleq C - I(X,Y)$$

where $I(X,Y)$ is the actual realized transinformation on the transmission channel.

Some authors use the notion *relative redundancy* ρ_C:

$$\rho_C \triangleq 1 - \frac{I(X,Y)}{C}$$

The efficiency η_C of the use of the transmission channel is defined as

$$\eta_C \triangleq \frac{I(X,Y)}{C}$$

where, of course,

$$\eta_C = 1 - \rho_C$$

3. Calculation of Channel Capacity

The calculation of C boils down mathematically to the research of a bounded maximum of a function $I(X,Y)$ of the variables $q(x)$ with $x \in X$, subject to the constraint of the inequality $q(x) \geq 0$ and to the constraint of the equality $\Sigma_x\, q(x) = 1$. We will not treat this problem here, but we do point out that this problem is simplified by a fundamental property of $I(X,Y)$, namely, that this is a *convex* function.

Example: Binary Symmetric Transmission Channel

This channel is characterized by the transition probabilities presented in Section II.C.

We calculate $I(X,Y) = H(Y) - H(Y|X)$. Expressing $H(Y|X)$ gives:

$$H(Y|X) = -[\epsilon \log \epsilon + (1-\epsilon) \log (1-\epsilon)]$$

Note that $H(Y|X)$ is independent of the probability distribution $q(x)$ of X and it thus depends only on the transmission channel. One can then write:

$$C = \max_q [H(Y) + \epsilon \log \epsilon + (1-\epsilon) \log (1-\epsilon)]$$

Furthermore, $H(Y)$ is maximal (in this case 1) for $p(y_1) = p(y_2)$, which implies that $p(x_1) = p(x_2) = \frac{1}{2}$. The capacity of a binary symmetric transmission channel is thus

$$C = 1 + \epsilon \log \epsilon + (1-\epsilon) \log (1-\epsilon)$$

This equation is represented graphically in Fig. 5 for $0 \leq \epsilon \leq 0.5$.

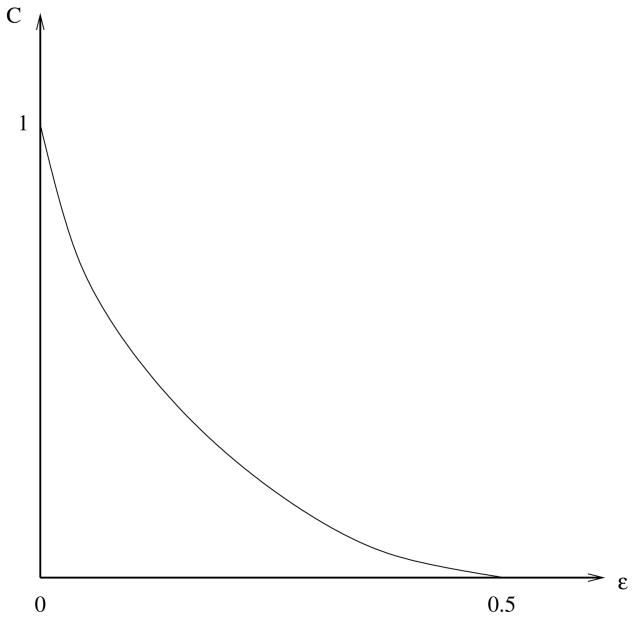


Figure 5 Capacity of a binary symmetric transmission channel.

In the absence of noise on the transmission channel we have that $\epsilon = 0$ and $C = 1$. When the channel is completely submerged in the noise, then we have that $\epsilon = \frac{1}{2}$ and $C = 0$. This derives from the fact that knowing the output does not modify our *a priori* knowledge of the input.

D. The Channel Coding Theorem

1. Introduction

Our goal is not to develop here the complete theory of information with respect to channel coding. We shall limit ourselves to the presentation of some results which will illustrate the notion of *capacity of a transmission channel*. One of the key notions in the theory of digital transmissions is the notion of error probability, i.e., the probability that a source symbol is wrongly reproduced at the destination. This error probability, which is a kind of quality criterion of the transmission, is the object of the most famous theorem of information theory, usually referred to as the *channel coding theorem*.

This theorem, which will further be formulated rigorously, says in essence that, for a very wide range of sources and transmission channels, and on condition that the entropy of the source per unit of time is smaller than the channel capacity per unit of time, the error probability can be reduced to an arbitrarily small level, providing we use a sufficiently complex code.

Before formulating this theorem, we will present the converse result: "If the entropy of the source is greater than the channel capacity, then it is impossible to reduce the error probability to an arbitrarily small number."

Note that the presented results are valid for a stationary discrete source (eventually with memory).

2. Definitions: Entropy per Symbol—Expected Value of the Error Probability for Symbol Sequences

Let there be a transmission system in which we consider a sequence of L symbols, emitted by a discrete source.

Let us denote by $\overline{X}$ the random vector variable that characterizes this transmission. The probability distribution of $\overline{X}$ will be denoted $p_L(\overline{x})$. The entropy per symbol is per definition:

$$H_L(\overline{X}) = \frac{H(\overline{X})}{L} = -\frac{1}{L}\sum_{\overline{x}} p_L(\overline{x}) \log \frac{1}{p_L(\overline{x})}$$

One can show that, for a stationary source, $H_L(\overline{X})$ is a not-increasing function of L, which tends to a limit $H_\infty(\overline{X})$ for L tending toward ∞. For a discrete memoryless source we find that $H_L(\overline{X}) = H(\overline{X})$, for $\forall L$.

If we denote by $\overline{y}$ the vector $(y_1, \ldots, y_L)$, which is delivered to the destination when the vector $\overline{x} = (x_1, \ldots, x_L)$ was sent, then the goal obviously is to have $\overline{y} = \overline{x}$.

If $x_l \neq y_l$, then there is an error in the lth transmitted symbol. If we denote by P_{el} the probability of such an error, then we define the probability $\overline{P}_e$ for a sequence of L symbols as:

$$\overline{P}_e = \frac{1}{L}\sum_{l=1}^{L} P_{el}$$

3. The Converse Channel Coding Theorem

If the entropy per source symbol for an infinitely long sequence is larger than the capacity of the transmission channel, then $\overline{P}_e$ will always be strictly positive.

4. The Channel Coding Theorem

a. Types of Transmission Channel Coding

Because we now want to consider the direct theorem instead of the converse, namely, that, under certain conditions, there exists *a* code which allows us to realize a transmission with an arbitrarily low error probability, we can now limit the type of codes that we are going to investigate. We first suppose that the source is coded with a source code that delivers a binary sequence at a rate of 1 binary symbol every τ_s seconds. The transmission channel coder realizes a *block code,* i.e., the system codes groups of L symbols with codewords of N symbols. Each symbol has a duration of τ_c, such that:

$$N\tau_c = L\tau_s$$

Let us denote by M the number of codewords needed.

If in this example $\tau_s/\tau_c = 5/2$, then there is synchronization between the flux of binary symbols which enter the coder and the flux of ternary signals which leave the coder.

Example

In Table II we present an example in which $L = 2$, $N = 5$ and where we suppose that the coder uses the alphabet {0,1,2}. The (discrete and memoryless) transmission channel is characterized by the transition diagram of Fig. 6.

One defines the binary rate R as:

$$R = \frac{\log M}{N} = \frac{L \log 2}{N}$$

This is expressed in bits (if the base 2 is used) and represents the number of binary symbols of the source per symbol of the transmission channel. If the binary symbols of the source are independent and equiprobable, then it is an information rate per symbol of the transmission channel. The binary rate per second is found by dividing R by τ_c. By means of the detected word at the output of the transmission channel, the decoder must estimate the word $\overline{x}$ which has been emitted by the source. Let $\overline{y}$ be this estimate. There is a decoding error in a word, when one or more errors did happen within this estimate, i.e., $\overline{x} \neq \overline{y}$. We shall denote by P_e the probability of error in decoding a word.

b. Types of Transmission Channel Decoding

In general a decoding law can be formally defined as a correspondence between the set Y^N of all possible sequences of N symbols at the output of the transmission channel, and the set of $M = 2^L$ source mes-

Table II Example of Channel Code

m	Binary source sequences	Codewords
1	00	00000
2	01	11110
3	10	22101
4	11	02222

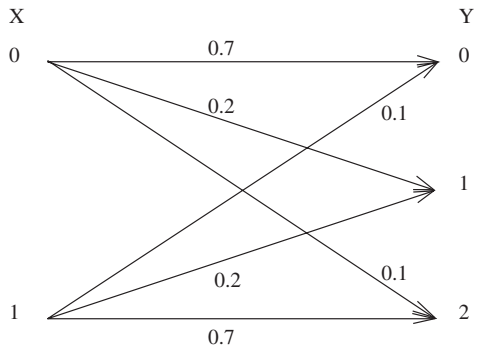


Figure 6 Transition probabilities of the channel of Table II.

Table III Example of a Channel Decoding Rule

	Received words $\bar{y}$	Decoded words $\bar{x}_m$
	000	000
$Y_1 \equiv Y_2^c$	001	000
	010	000
	100	000
	110	111
$Y_2 \equiv Y_1^c$	011	111
	101	111
	111	111

sages, composed of L binary symbols. One denotes Y_m the set of the received words which shall be decoded into the message m, and we denote by Y_m^c its complement. With every source message m, there corresponds a transmitted codeword $\bar{x}_m$. If the received word $\bar{y}$ belongs to Y_m^c, then there will be an error in the decoding. The probability of having a decoding error when the message m has been emitted is thus:

$$P_{em} = \sum_{y \in Y_m^c} p_N(\bar{y}|\bar{x}_m)$$

with $p_N(\bar{y}|\bar{x}_m)$ the probability that the vector $\bar{y}$ is received, when the vector $\bar{x}_m$ has been transmitted.

In the case of a memoryless transmission channel we have that:

$$p_N(\bar{y}|\bar{x}_m) = \prod_{i=1}^{N} p(y_i|x_{mi})$$

where x_{mi} and y_i are the respective components of $\bar{x}_m$ and $\bar{y}$. The (expected value of the) decoding error is thus:

$$P_e = \sum_{m=1}^{M} p(\bar{x}_m) P_{em}$$

Different decoding rules are possible. One of the most popular however is the maximum likelihood decoding. According to this principle, if y has been received, one chooses at the decoding the message m' such that:

$$p_N(\bar{y}|\bar{x}_{m'}) \geq p_N(\bar{y}|\bar{x}_m) \qquad \forall m \neq m'$$

As an example, one can show that in the case of the binary symmetric channel with an error probability of ϵ, the calculation of $p_3(\bar{y}|\bar{x}_1)$ for all four triplets $\bar{y}$ of Y_1^c gives:

$$P_{e,1} = 3(1 - \epsilon)\epsilon^2 + \epsilon^3$$

Example

Let us suppose that the codewords are $\bar{x}_1 = (000)$ and $\bar{x}_2 = (111)$. The (maximum likelihood based) decoding rule is shown in Table III.

c. The channel coding theorem

This theorem gives an upper bound for the error probability as a function of the binary rate R, of the length of the block N, and of the characteristics of the transmission channel. This bound has been found by analyzing the set of codes, rather than by the simple analysis of a good code. This approach is the result of the fact that for interesting values for R and N, until now there was no method for finding a code which minimizes the error probability. For the existing codes one can estimate this error probability, but the result of this estimation is much larger than the upper bound presented here. We denote by $\overline{P}_{em}$ the expected value of the error probability when the message m has been sent, an expected value which is calculated for the set of all block codes. If one finds an upper bound of $\overline{P}_{em}$, then at least one code of this set of codes must have an error probability which is smaller than $\overline{P}_{\text{em}}$, and thus smaller than the upper bound found.

A typical example of $E(R)$ is shown in Fig. 7. This theorem shows that, whatever the level of noise in the transmission channel, one can transmit messages on

Theorem

Let there be a discrete memoryless source with binary rate R (bits/second) and a transmission channel with capacity C. The expected value of the error probability $\overline{P}_{\text{em}}$ (calculated over the set of block codes of length N), when the word m was emitted by the source, satisfies:

$$\overline{P}_{em} \leq 2^{-NE(R)}$$

where $E(R)$ is a function which is called the *error exponent*, and has the following properties:

- $E(R) > 0 \qquad \forall R: 0 \leq R < C$
- $E(R)$ decreasing and concave to the top for $0 \leq R < C$.

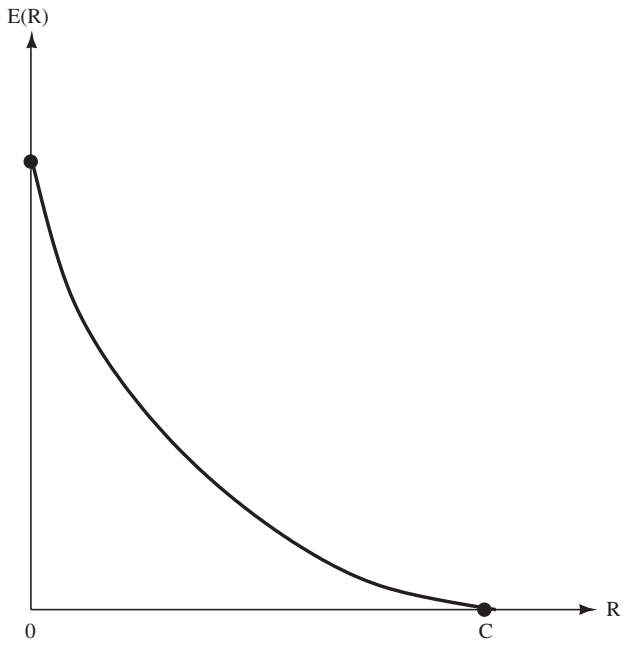


Figure 7 Typical example of an error exponent $E(R)$.

this channel with an arbitrarily low error probability. This is why this theorem is at the base of the enormous development of transmission channel coding.

The theory of Shannon however, does *not* specify how to realize these optimal codes. In practice, one can easily get very low error probabilities when $R < C/2$.

V. CONCLUSIONS

Information theory as introduced by Shannon in his famous papers on the mathematical theory of communications has the big advantage of showing the theoretical possibilities and limitations of both source and channel coding. However, the main drawback of this theory is that it does not show practical ways of designing optimal source or channel codes.

With respect to the practical development of source codes, we have already mentioned Huffman codes as an example of optimal source codes.

SEE ALSO THE FOLLOWING ARTICLES

Cybernetics • Data, Information, and Knowledge • Decision Theory • Error Detecting and Correcting Codes • Future of Information Systems • Information Measurement • Number Representation and Computer Arithmetic

BIBLIOGRAPHY

Anderson, J. B., and Mohan, S. (1991). *Source and channel coding—An algorithmic approach.* Amsterdam: Kluwer Academic Publishers.

Cover, T., and Thomas, J. (1991). *Elements of information theory.* New York: Wiley-Interscience.

Gallager, R. G. (1968). *Information theory and reliable communication.* New York: J. Wiley & Sons.

McEliece, R. J. (1977). *The theory of information and coding—A mathematical framework for communication.* Reading, MA: Addison-Wesley.

Pierce, J. R. (1980). *An introduction to information theory—Symbols, signals and noise,* 2nd ed. Dover Publications.

Shannon, C. E. (1948). A mathematical theory of communication, part I. *Bell Systems Technical Journal,* Vol. 27, 379–423.

Shannon, C. E. (1948). A mathematical theory of communication, Part II. *Bell Systems Technical Journal,* Vol. 27, 623–656.

Viterbi, A. J., and Omura, J. K. (1979). *Principles of digital communication and coding.* New York: McGraw-Hill.

Integrated Services Digital Network (Broadband and Narrowband ISDN)

Gene Mesher
California State University, Sacramento

GLOSSARY

asynchronous transfer mode (ATM) The data link layer protocol for Broadband-ISDN intended to deliver high-speed packet communications to end users, but now largely confined to wide area networking.

Intelligent Digital Network (IDN) The combination of digital trunk lines and digital switching that has turned the telephone network into a global digital network of specialized computers.

local loop or last mile The link between the telephone users and their nearest telephone switch; the last PSTN segment to remain analog.

public switched telephone network (PSTN) The telephone network as we know it today.

synchronous digital hierarchy (SDH) The highly scalable, physical layer digital communications protocol family now in worldwide use on trunk lines. SDH is capable of extremely high speeds currently in ranges tens to hundreds of gigabits per second. SONET, the synchonous Optical Network, is essentially the North American name for SDH.

Signaling System Seven (SS7) The operating and communications system used by intelligent telephone switches such as the 5ESS which allows telephone switches to exchange the details of calling information and line conditions.

wide area networking (WAN) Long distance communications requiring the use of transmission lines provided by or leased from telecommunications companies.

THE INTEGRATED SERVICES DIGITAL NETWORK (ISDN) concept was developed and presented as the new long-term strategic direction for the PSTN to take as part of the overall shift to digital technologies that began with the introduction of the first T-1 carriers in the early 1960s. This "ISDN vision" was based on the realization that over the long term, telecommunications would evolve from an analog to a digital form, that packet-switched services would replace circuit-switched communications forms, and that communications services that were previously transmitted through separate networks would eventually merge into a single high-speed, multimedia communications infrastructure. The ISDN "plan" was part of a three phase concept that was developed to enable this evolution in communications technology. First, with the digitalization of transmission and switching, an all-digital telecommunications infrastructure began to emerge in the 1970s, called the Intelligent Digital Network or IDN. In theory, the next phase was to add circuit-switched digital services to this network, integrating the local loop into the IDN in the form of narrowband ISDN. Once the N-ISDN became widespread, the final phase in the evolution of the telephone network was to be the implementation of broadband ISDN, intended as the vehicle for delivering high-speed packet-switched services to the world's telecommunications subscribers. B-ISDN combines the SONET and ATM protocols to enable the provision of high-speed digital services. While the ISDN plan has been influential in providing a vision of the future of telecommunications, narrowband-ISDN has

largely been a failure in the marketplace. The reason for this, in large part, has been due to the lack of market orientation on the part of the developers of ISDN. Instead of being the connection of choice for all telecommunications users, it has been beaten out of common use by cheaper, more rapidly evolving technologies and instead relegated to a few niche markets such as backup and recovery. B-ISDN is still in its early phases of development but would appear to be suffering the same fate. While the SDH protocol, has become the protocol of choice for physical layer WAN transport, ATM, B-ISDN's main choice for carrying real-time applications is not faring as well. While ATM has done reasonably well to date in WAN environments, it is now facing renewed competition from other protocols. It now appears that ISDN will be remembered for both its vision of the future of networking along with its lack of orientation towards the recently emerged competitive telecommunications marketplace where protocols must meet and adapt to market demand to succeed.

I. INTRODUCTION: THE ISDN VISION

> Vail accepted regulation in place of competition in order to build an efficient telephone network for a nation coming of age. . . I don't want to overstate the case. To put it graciously, I'd have to say we begrudgingly acceded to the public's desire for competition.
>
> Robert E. Allen, Chairman and CEO, AT&T, 1996

Although viewed today as just one of many access technologies in a competitive market for digital communications services, the Integrated Services Digital Network (ISDN) is unique in that it originated out of the need for the then monopoly-controlled telephone industry to develop a new, long-term strategic vision to reflect the changes in technology ushered in with the introduction of digital technologies in the early 1960s.

Thus, the origins of the idea for an ISDN began on the one hand, with the introduction of T-carriers that marked the beginning of the conversion of trunk lines from analog to digital transmission and, on the other hand, to the basic AT&T philosophy that telephone service was a natural monopoly. This latter idea dates back to Theodore Vail, AT&T's president at a pivotal period in the early 20th century when the company faced the most serious crisis in its history. By the early 1900s, Bell's original patents had expired. The company was facing bankruptcy along with an array of competitors that had grown into the thousands. Vail was able to reassert AT&T's dominance over the market by taking advantage of the company's newly gained patents over long distance telecommunications technologies to drive the company's competitors into submission.

The success of Vail's efforts hinged on getting government to take on the role of regulator and the public to accept the idea that telephone service was a "natural monopoly" and should be treated as a utility like water or electricity, i.e., best provisioned by a single large company that could best take advantage of economies of scale. Vail, himself a former postal administrator, further asserted the concept of Universal Service as a Bell system requisite, making telephones an essential service provided by a single carrier.

Although the concept of an ISDN did not arise for another half a century, when it did, it was very much the product of the monopoly environment born out of AT&T's early 20th century turnaround and the worldview that "The" phone company, and others like it around the world, had taken on. That worldview meant that monopoly telecommunications service providers were not only "price givers," they had also become service and technology givers. This monopolistic state of affairs was still in place during the 1960s, at the beginning of the move to convert national telephone networks to digital technologies. Thus, telephone organizations at the time sought to maintain their role as sole provider of end-to-end services as well as being responsible for any decisions as to which technologies and new services would be incorporated into the network.

A. ISDN and the Digitalization of the Telephone Network

The conversion of the telephone system from an analog to a digital network did not happen overnight. Indeed, while some elements of the Public Switched Telephone Network or PSTN, have become largely digital, others have yet to be converted. The telephone network is made up of four components (see Fig. 1):

1. **Customer premises equipment** (CPE), i.e., telephone sets and other equipment that directly connect to the phone network

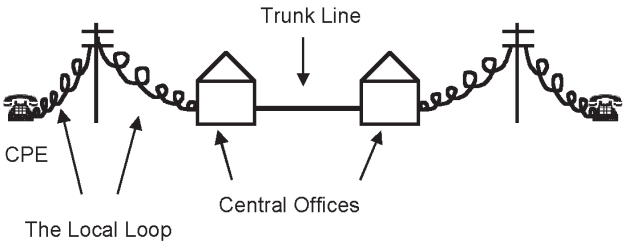


Figure 1 Basic components of the telephone network.

2. The **local loop** or "last mile," which connects the CPE to the nearest telephone switch
3. **Central offices,** also referred to as telephone exchanges or switches, whose function is to set up, maintain, and tear down the electrical circuits used to transmit a caller's voice during a phone call
4. **Trunk lines** that are the multiplexed lines that interconnect central offices

Historically, all four components of the network were analog. The term "Central Office," in fact, dates back to the time when the connections required as part of the telephone call set up process were taken care of by human operators. Although the process is far from complete, all four components are now in the process of being converted from analog to digital forms. At present, two of the components, namely the switches and trunk lines, have both been converted to digital technologies while the local loop still uses analog transmission. Customer premises equipment is still largely analog as well.

1. Digital Transmission

The first step towards converting the telephone network to an entirely digital form came with the introduction of digital transmission in the early 1960s when the first T-carrier trunk lines were installed. Although many benefits to digital transmission have since been recognized, initially, the focus was on improving signal quality. This is because digital transmission uses square waves that make possible the use of digital repeaters instead of analog amplifiers. The amplifiers that had been used in the telephone network to boost signal strength up until that time had a well-known flaw: along with increasing the strength of the signal being transmitted, amplifiers strengthen the background noise that has managed to creep into the signal. Repeaters, in contrast, reform incoming square waves enabling signals to be transmitted virtually noise free.

Instead of using frequency division multiplexing, which was previously the standard for multiplexing analog communications, digital transmission of telephone signals is multiplexed using time division multiplexing. The majority of trunk line transmissions in developed countries are now transmitted digitally, using T-carriers, E-carriers or SONET (see B-ISDN discussion later).

The conversion of the analog voice signal to a digital form is based on the pulse code modulation (PCM) encoding technique. In PCM, amplitude samples are taken of the analog signal generated by a telephone. These samples are then converted into an 8-bit code according to which of the 256 voltage levels the sample voltage corresponds. The sampling rate is determined by the Nyquist Theorem. This states that a signal needs to be sampled at twice the rate of the highest frequency if the signal is to be accurately reconstructed. Since the frequency range used to transmit voice is from 0 to 4000 Hz, 8000 samples need to be taken per second.

Since PCM uses 8 bits to encode each voice sample, the resulting digital signal transmits at a data rate of 64,000 bits per second. This is called a digital signal zero or DS-0. DS-0 signals are then multiplexed into either a T-carrier in North America or an E-carrier in most of the rest of the world. The lowest rung of the T-carrier hierarchy is the T-1 carrier that multiplexes 24 DS-0 signals at a combined data rate of 1.544 Mbps. The other commonly used T-carrier is the T-3 or DS-3, which is made up of 30 T-l's and transmits at an aggregate rate of about 45 Mbps.

Near the termination point of the transmission, the digital signal will be converted back into analog form. The same type of device generally does both and is called a codec, short for coder-decoder. A related device is the channel bank that converts 24 analog voice signals into PCM data streams and uses them to construct a T-1 frame.

In European countries and most of the rest of the world, a different digital transmission hierarchy is used called the E-carrier hierarchy. The E-carrier hierarchy also uses PCM and the DS-0 as its basic unit, but has a different frame structure. Instead of 24 voice channels, the E-1 carrier, which is the lowest level of the E-hierarchy, has an aggregate data rate of 2.048 Mbps and is composed of 32 channels. Thirty of these channels are DS-0 signals used to transmit voice signals while the other 2 channels are used to send control information. Unlike the T-carrier hierarchy in which every member of the hierarchy has a distinct frame structure, E-carriers use the idea of interleaved data to create higher level carriers whose data rates are exact multiples of the lower data rate carriers in the hierarchy. For example, an E-2 carrier has an aggregate data rate of exactly four times an E-1 carrier and an E-3 carrier four times that of an E-2. This principle was later used in the development of the SONET/SDH hierarchy discussed later in this article.

2. Digital Switching

The second step in the conversion of the telephone network to all digital technologies came with the introduction of digital switching, originally referred to as stored program control. Operational software first began to be used in the telephone network during

the late 1950s when the first digital exchanges were developed at Bell Labs and were first field tested in 1959. Based on this experience, the lESS (electronic switching system) digital switch was put into service in 1965 and later converted to an integrated circuit design during the 1970s. This second generation of switching hardware was the lA switch processor that was used in conjunction with the 2ESS. With it the operational software was also further developed.

Since that time a third generation of digital telephone switches has been introduced. These are the 5ESS switches that have been used in the field since the mid-1980s. This switch has many advanced features which make it much more flexible and powerful than earlier switches. It is the 5ESS that is capable of being programmed for delivering ISDN services to subscribers.

B. The Intelligent Digital Network (IDN)

Initially, digital transmission and digital switching were developed as separate features of the telephone network, but as more and more digital transmission lines were installed and digital switching techniques became more widespread, the idea was proposed that the two telephone network components be integrated into a single system in which digital switches also use digital transmission trunk lines to send control signals to each other. As the idea moved from concept to reality, the combination of digital switching and transmission became known as the Intelligent Digital Network (IDN) or simply the Intelligent Network.

1. Common Channel Signaling and SS7

A key step in the development of the IDN came during the 1960s with common channel signaling (CCS), also known as out-of-band signaling. Signaling in this context means sending control information through the telephone network. Control signals can be sent between telephone switches, such as routing information, or to and from telephone switches and terminal equipment (e.g., telephones), such as the off-hook and dial tone signals. Signals used by the telephone network can also be grouped into two categories: *supervisory signals,* which indicate the status of the circuits being used, and *information bearing signals.* The most familiar examples of supervisory signals are those related to call progress and include the off-hook signal, dial tone, call alerting (ringing), busy signal, and called party ringing or ringback. Information bearing signals include information related to the call such as the caller's number, calling party's number, and toll charges. While the previous signaling examples relate to call progress, other signals are sent between telephone switches to provide routing and flow control information, indicate equipment failures, and send test signals used for system maintenance.

Familiar signals such as a busy signal are called in-band signals because they are sent over the same channel as the voice signal. Out-of-band signals, however, are sent over separate lines from the voice network. Unlike the rest of the phone network, which is circuit-switched, out-of-band signals are sent over a separate, shared, packet-switched line, hence the term "common channel." In 1976, more than a decade after the first CCS circuits were installed, a second-generation common channel signaling protocol was introduced, known as Signaling System 7 (SS7), which effectively acts as a distributed operating system for the telephone network.

The use of CCS and SS7 has had many advantages for the telephone network. Call setup and teardown, for example, now occur at much faster rates of a second or less, while the use of a separate channel for control information completely eliminates the possibility of interference between control signals and voice traffic. SS7 also allows for much greater flexibility in network operations and consequently facilitates the introduction of new software into the telephone network.

C. A Broadband Vision of the Future

As the Intelligent Digital Network took shape, telecom engineers began to consider what the long-term implications might be from the digital technologies in the telephone network, especially with regards to users. The development of a broadband vision of the future telecommunications network was based primarily on the following trends:

- **High Speed—**Ever increasing transmission data rates used by both trunk lines and end users that promised to eventually provide end users with data rates in the multi-megabit range or even higher.
- **Packet-Switching—**The advantages of packet-switched over circuit-switched data communications techniques for transmitting variable bit rate data streams.
- **Digital Convergence—**The ability to combine digital information in a variety of formats into a single transmission that could accommodate a multimedia data stream.

Thus the vision was that of a future high-speed, packet-switched, multimedia network. This eventually became referred to as broadband ISDN or B-ISDN.

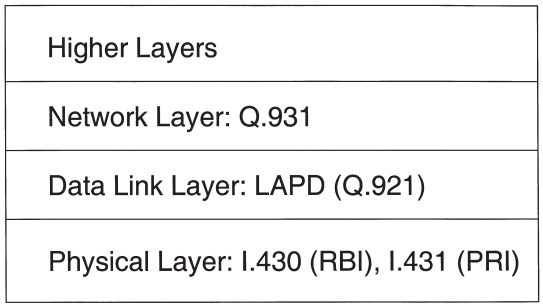


Figure 2 Narrowband ISDN network protocol model.

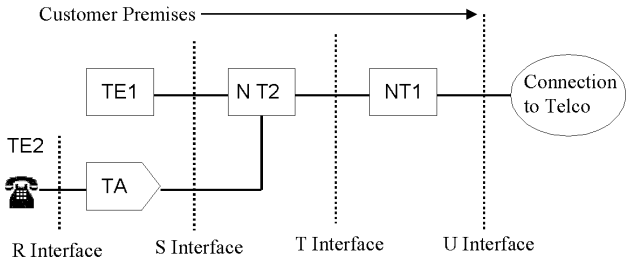


Figure 3 N-ISDN functional groups and reference points.

There were two major challenges to achieving this goal: first, defining the communication protocols and technology needed for the future broadband network and second, developing a bridging technology to provide the link between the analog infrastructure now in use in the local loop and the future high-speed broadband network that will come with the introduction of broadband services. Narrowband ISDN was intended as the technological bridge between the analog telephone network and the future high-speed packet switched, broadband ISDN network.

II. NARROWBAND ISDN

Although the ISDN concept was first proposed in the late 1960s, the actual work of standardizing ISDN did not begin until the late 1970s and the first standards were not published until 1984. ISDN standards were developed by study groups of the International Telephony and Telegraphy Consultative Committee (CCITT), which was the standardization group within the International Telecommunications Union until 1992. Figure 2 presents the N-ISDN protocol model.

A. ISDN Reference Points and Equipment Types

1. Equipment Types

At the user-network interface, N-ISDN uses the following functional groups (see Fig. 3):

- Network Terminal 1 (NT1)—Provides the T interface
- Network Terminal 2 (NT2)—Used for switching and multiplexing of the ISDN signal. (Note that NT2 equipment is not always included in the circuit. It would not be needed, for example, with a BRI connection connecting to a single device.)
- Terminal Equipment type 1 (TE1)—ISDN compatible equipment, such as an ISDN telephone
- Terminal Equipment type 2 (TE2)—Non-ISDN compatible equipment, such as an analog telephone
- Terminal Adapter (TA)—Used to connect TE2 equipment to the ISDN circuit

Note that these equipment types are meant to be flexible. Different types of equipment could be combined in any reasonable way. For example, combining the NT1 and NT2 interfaces or combining the NT2 interface with the terminal adapter (TA).

2. Reference Points

Three reference points, or interfaces, labeled S, T, and U, have also been defined for ISDN, which is implemented using 8-wire RJ-45 connectors. The S and T reference points provide a four-wire bus used by ISDN equipment to connect into the network (see Fig. 3). The U interface defines a two-wire connection to an open network. This was originally only important in the context of the North American markets, but has now become more important with the growth of open networks in Europe as well.

3. 2 Binary 1 Quaternary Transmission (2B1Q)

ISDN uses a digital transmission encoding scheme known as 2 binary 1 quaternary or 2B1Q. The 2B1Q means that there are four possible signal levels and that each signal level encodes two bits of information. In the case of ISDN, the four signal levels are +3 and −3 and +1 and −1 V (see Fig. 4). By encoding data at four different voltage levels, 2B1Q encoding is also able to provide a solution to an important technical challenge, namely, the problem of interfacing a four-wire network segment on the customer premises to a two-wire local loop in a relatively simple way with a minimum of cross-talk.

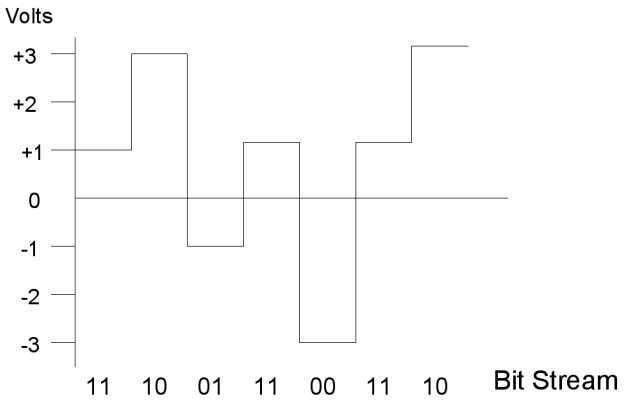


Figure 4 Example of 2B1Q digital encoding.

B. Basic and Primary Rate Interfaces

Just two access services have been defined for narrowband ISDN. The first is the relatively low speed basic rate interface or BRI that has an aggregate data rate of 144 kbps and roughly corresponds to a data rate equivalent to two DS-0s. The second offering is the relatively higher data rate primary rate interface or PRI that has an aggregate data rate of 1.544 Mbps, which is the equivalent of a T-1.

1. ISDN Channels

N-ISDN services include three types of channels:

- B channels, also called bearer channels, that transmit data at a rate of 64 kbps
- D channels or delta channels that transmit at different data rates of 16 to 64 kbps depending on the service being used
- H channels that are high bit rate channels that group a number of B channels together

B channels are so-called because they have the job of bearing data streams to and from users. When an N-ISDN data stream leaves the customer premises, the bearer channels flow into the switched digital network used to carry data and voice in the telephone network while the delta channels flow into the common channel signaling network used to carry control information.

D channels are primarily intended for transporting control information. When an N-ISDN data stream leaves the customer premises, the delta channels are sent over the common channel signaling network used to carry control information. The size of the Delta channel depends on which form of ISDN is being used. For the BRI form of ISDN the D channel data rate is 16 kbps while for the primary rate interface, which has a much larger number of channels requiring more signaling, the D channel data rate is 64 kbps.

H channels are high-speed channels available with PRI ISDN (see later) that have a combined data rate equal to several B channels. The main function of H channels has been as conduits for sending video transmissions. Three different H channels have been defined: H_0 which has a defined data rate of 384 kbps, equivalent to 6 B channels; H_{11} which has a data rate of 1472 kbps, equivalent to 23 B channels, and H_{12} which has a data rate of 1920 kbps, which corresponds to 30 B channels. The latter two H channels were designed for use with North American and European PRI ISDN services, respectively.

2. Basic Rate Interface: 2B + D

The basic rate interface was designed to become the digital access mode for the general public. The ISDN BRI of access is also referred to as 2B+D since it is made up of four channels, two 64-kbps Bearer channels, a 16-kbps Delta channel, and an overhead channel (see Fig 5.). The two bearer channels can be used separately, or they can be combined to provide a total of 128 kbps. The Delta channel is used for sending signaling information and can also be used for sending low-speed data services, such as telemetry, when not being used for signaling. A fourth channel also exists called an overhead or maintenance channel which is responsible for maintaining the framing and timing functions of the BRI connection between the telephone switch and the CPE. Thus, the aggregate data rate for a BRI access is 192 kbps.

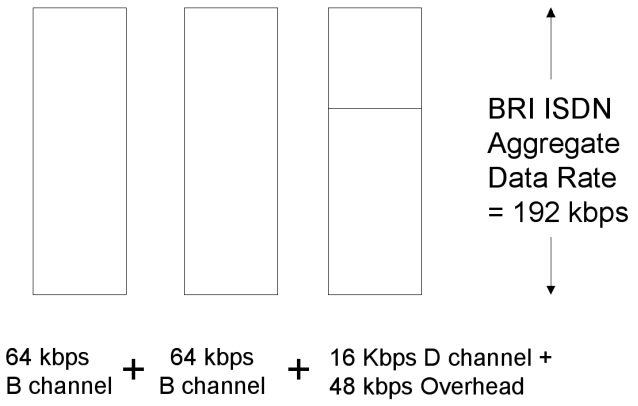


Figure 5 Basic rate interface at the S/T reference points.

3. Primary Rate Interface: 23B + D or 30B + D

The second form of ISDN access is the primary rate interface or PRI. PRI was designed as a product for business use rather than for home use; it has a much higher aggregate data rate than BRI, equivalent to a T-1 carrier. PRI is composed of 23 B channels of 64 kbps each along with a single D channel whose data rate is also 64 kbps making the aggregate PRI data rate 1.536 Mbps. In Europe and countries using European telecommunications standards the primary rate interface has a 30 B + D structure corresponding to the European E-1 carrier. The European PRI aggregate data rate is 2.048 Mbps.

C. ISDN Applications

Before discussing specific applications that have been used with ISDN it is important to note that the ISDN standard is essentially a standard for a digital "pipe" so that although it has certain specific characteristics that will make it more or less desirable for some applications, ISDN was not developed with any particular application in mind. ISDN applications can be viewed in two ways: suggested applications; i.e., the ways in which telecommunications providers implementing narrowband ISDN originally envisioned the technology would be used, and business applications, i.e., the ways in which ISDN actually has been used. The suggested applications include:

- Digital telephony
- Analog and digital fax
- Computer and videoconferencing
- Telemetry
- Packet switched data communications such as X.25 using the D channel

D. Business Use of ISDN

Business applications for ISDN have generally not been very successful, although there have been a few exceptions in which special characteristics of ISDN, especially its digital dial-up feature, have made specific uses attractive. Chief among these have been fast Internet access, backup and recovery, and videoconferencing.

1. Fast Internet Access

During the mid-1990s, before the introduction of more recent access technologies such as DSL and cable modem, ISDN was one of the first technologies to provide fast Internet access. This was also at a time when BRI-ISDN's 128-kbps aggregate bearer channel data rate was still attractive since modem speeds at the time were typically between 19.6 and 28.8 kbps.

2. Backup and Recovery

One of N-ISDN's strongest showings in the marketplace has been in the area of backup links. As a digital dial-up service, ISDN was well suited to a market niche for a service that is only used occasionally or for emergencies since the cost of the service is a combination of a relatively low monthly charge plus connect time. Thus, the service has been well positioned to compete against other digital services that charge a relatively high, fixed monthly fee.

3. Video Conferencing

A third important application for ISDN has been videoconferencing. For organizations for which cost is not critical, an H_0 channel can be set up on a PRI ISDN connection. Low data rate videoconferencing has been more popular. In this case, data rates that are intermediate between DS-0 and T-1 are used, so that the video signal adequately balances cost and image quality criteria. This can be done using a combination of ISDN and inverse multiplexing. Inverse multiplexing means taking a relatively high speed signal, breaking it up, and transmitting it at lower data rates over two or more transmission lines.

In the case of videoconferencing, a popular solution has been to use a 384-kbps transmission rate, transmitted over three ISDN BRI lines. Each BRI ISDN line will be separately connected and may, in fact, take very different routes between source and destination locations. In the case of an inverse multiplexed videoconferencing application, a technique called "scatter gather" is used in which video packets are sent alternately over each BRI ISDN connection.

E. ISDN's Failure in the Marketplace

While the ISDN vision appears to accurately reflect the long-term evolution of the global telecommunications network, the implementation of ISDN has, to say the least, met with only very limited success. In general, N-ISDN has been most successful in monopoly markets where competition was limited. In more competitive markets, the N-ISDN service found itself in competition with rapid increases in the data rates

achieved by analog modems at the low end of the market and with higher data rate offerings such as DSL and cable modems at the high end of the market. Thus, in competitive markets instead of becoming the dominant form of data access, ISDN has been relegated to a few niche markets where its digital dial-up feature is attractive, such as backup services and videoconferencing.

III. BROADBAND ISDN

Narrowband ISDN was never intended to be a long-term digital access solution, but rather a bridging technology enabling the entire telephone network to become completely digital as a step towards its further evolution towards a packet-switched, broadband endpoint. Unlike N-ISDN, which is an amalgam of circuit-switched and packet-switched signals, the vision of the future for the telephone network was (and remains) that of a high-speed packet-switched network. Packet networks may have more overhead, but they do make it possible for data to be sent at a variety of different data rates as well as allowing data of different types to be sent in a truly integrated fashion. In terms of the network protocols that will be used, broadband ISDN will likely be delivered over wide area networks to users as a stream of asynchronous transfer mode (ATM) cells sent using the synchronous digital hierarchy (see Fig. 6).

A. Synchronous Digital Hierarchy (SDH)

Although at the user network interface, broadband ISDN can be delivered in several ways, the preferred delivery method will most likely be "ATM over SONET." That is, to use asynchronous transfer mode (ATM) at the data link layer and the synchronous digital hierarchy (SDH) also known as the synchronous optical networking protocol or SONET at the physical layer.

Figure 6 Broadband ISDN network protocol model.

Figure 7 shows the structure of the basic STS-1 also known as the OC-1 (optical carrier) frame structure, the lowest layer of the SDH hierarchy. As with carriers in the T- and E-carrier hierarchies, an STS-1 frame is transmitted every 125 μs or 8000 times a second and has a 9 x 90 byte frame size. The STS-1 aggregate data rate is thus

$$9 * 90 \text{ octets/frame} * 8 \text{ bits/octet} * 8000 \text{ frames/s} = 51.84 \text{ Mbps}$$

The STS-1 frame is composed of several regions of data including three overhead sections known as the section, transport, and path overheads with the rest of the frame devoted to the delivery of data in a region known as the synchronous payload envelope (SPE). The frame design is very flexible and can carry nearly any type of data. Data in the SPE do not need to begin at the beginning of the SPE and frames that are too large or unfinished when the available capacity of a frame has been used up can be straddled between two SDH frames.

Transmission using synchronous digital hierarchy protocols is generally meant to be fiber-based and capable of very high speeds. Standards in the SDH carrier hierarchy are now available with data rates in excess of 100 Gbps. Unlike the T-carrier hierarchy in which each carrier is quite different from those at lower levels, the SDH hierarchy uses a regular pattern of interleaving bytes from lower level frames. Thus, the bit rate of an STS-3 or OC-3 carrier is three times that of the STS-1 or 155.52 Mbps and the bit rate of the STS-12/OC-12 carrier is 622.08 Mbps. Figure 8 provides a list of some of the more commonly used SDH data rates.

B. Asynchronous Transfer Mode (ATM)

The asynchronous transfer mode (ATM) protocol was designed as the means to deliver the high speed, "bursty" packet-switched real-time applications, such as video-telephony, that have been anticipated as *the* critical applications for the B-ISDN environment. Consequently, the ATM protocol includes features specifically intended to support real-time applications as part of mixed applications in a multimedia environment. These features include small fixed-sized frames, connection-oriented routing, and quality of service.

ATM frames, known as cells, are of fixed size and small, only 53 bytes in length with a 5-byte header field and a 48-byte information field (see Fig. 9). This is small compared to Ethernet and Frame Relay, both

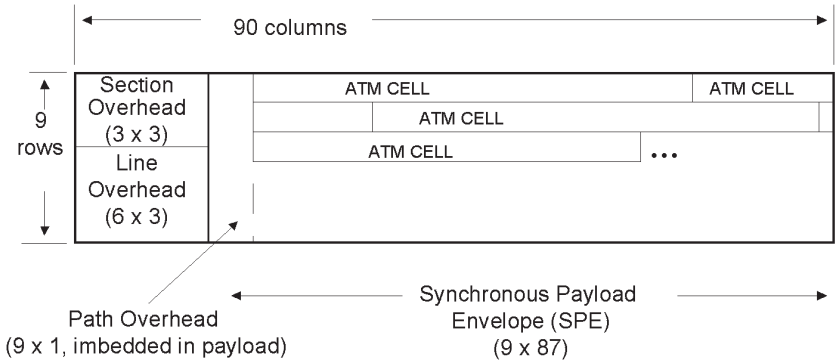


Figure 7 Basic SDH frame structure (STS-1/OC-1) with ATM cells shown in payload area.

of which have variable sized frames with information fields of up to 1500 and 8000 bytes, respectively. ATM's small cell size was chosen specifically to accommodate real-time applications that can tolerate neither long delivery delay or queuing delays due to large frames. Small, fixed-length frames help to minimize both of these concerns.

At the user-network interface, the 5-byte ATM header is composed of six fields. Perhaps the most important of which are the VPI and VCI, used for addressing and the one-bit cell loss priority (CLP) field, used to identify discard eligible cells. Addressing under the ATM protocol is handled using a pair of values: the one-byte virtual path identifier (VPI) and the two-byte long virtual connection identifier (VCI).

In order to convert packets of various types into ATM cells an additional protocol layer sits atop the ATM layer called the ATM adaptation layer (AAL). The AAL's primary responsibilities are segmentation and reassembly (SAR) and convergence (CS). SAR is the process of segmenting input data streams, which are generally organized into larger-sized data units into ATM's 53-byte cells and then reconstituting the data back into its original form at its destination point. The convergence sublayer (CS) of the ATM adaptation layer has the additional task of numbering cells so that they arrive in order and are reassembled in their original sequence.

Two competing standards have been proposed for implementing classes of service on ATM, one by the ITU-T and the other by the ATM Forum. In both cases, the idea is to provide a set of service classes that range from connection-oriented, constant bit rate services at one end of the spectrum for applications that are latency intolerant (such as telephony) to classes that are not intended for real time applications and for which delaying the delivery of cells does not critically effect the application (such as normal priority e-mail). Differences between service classes or categories are specified through a number of traffic-related parameters including cell transfer delay, cell delay variation, cell loss ratio, and minimum and peak cell rates.

The ATM Forum's classification scheme, for example, is made up of the following five service categories:

- Constant bit rate (CBR)
- Real-time variable bit rate (RT-VBR)
- Non-real-time variable bit rate (NRT-VBR)

Signal	Data Rate
STS-1/OC-1	51.84 Mbps
STS-3/OC-3	155.52 Mbps
STS-12/OC-12	622.08 Mbps
STS-48/OC-48	2.488 Gbps
STS-192/OC-192	9.953 Gbps

Figure 8 Some common SDH hierarchy carriers.

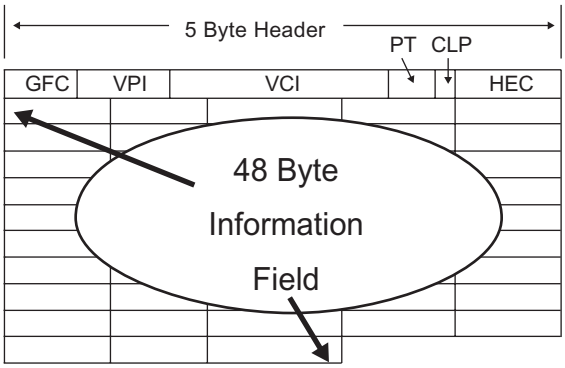


Figure 9 ATM cell structure.

- Available bit rate (ABR)
- Unspecified bit rate (UBR)

The constant bit rate (CBR) service category is a connection-oriented, constant rate service used by such applications as real-time voice and full-motion video in which a guaranteed transmission capacity is required and only very small differences in delivery time can be tolerated.

Variable bit rate—real time (VBR-RT) and variable bit rate—non-real time (VBR-NRT) are intermediate service categories. VBR refers to a data rate that can be specified in advance but is also allowed to vary between a minimum and a maximum bit rate. The VBR-RT service category was designed for real-time applications that can tolerate some latency, in which case a variable bit rate environment can provide the service, as in the case of some forms of videoconferencing. The VBR-NRT service category can be used in cases of data transmission in which the delivery of cells does not need to be under absolutely stringent control but for which some consideration still needs to be made regarding the timeliness of cell delivery, such as downloading a web page, in which the user is waiting for, say, a graphical image to be displayed by a web browser.

The lowest priority cells, such as normal priority e-mail, can be sent using one of the last two service categories: available bit rate (ABR) and unspecified bit rate (UBR). In these cases, in contrast to the VBR service categories, the bit rate of the service cannot be specified in advance. In the case of ABR, it is specified at the time of transmission and includes a minimum cell transmission rate. UBR, by contrast, is a best effort service that specifies neither a minimum bit rate nor provides any guarantee of cell delivery.

IV. CONCLUSION: THE ISDN VISION AND THE NEW AGE OF TELECOM

The Integrated Services Digital Network concept grew out of the vision of transforming the analog telephone network of the 1960s into an entirely digital telecommunications network. The vision also includes the idea that the ISDN would be the first step towards an even more advanced end point in which the digital network would eventually evolve into a high-speed, packet-switched multimedia network in which real-time applications, especially video-telephony, would be common place.

As the first stage in moving towards that vision, narrowband ISDN was later developed to provide a complete, end-to-end solution as the final stage in converting the analog telephone network to a circuit-switched, digital form on the one hand and as a intermediate, bridging technology to set the stage for the later implementation of a high-speed, packet-switched broadband network.

Narrowband ISDN was also developed in the monopoly-controlled industrial environment that had become the standard for the telecommunications market worldwide at the time. Just as ISDN entered the marketplace in the early 1980s, however, the telecommunications market itself, in the U.S. and later around the world, changed dramatically as the decades-old telecommunications monopolies began to face competition in a variety of market segments. The telephone companies that were responsible for providing the ISDN product suddenly found that they were competing with a variety of products in a variety of market segments.

The evolution of modem technology illustrates this. By the mid-1980s, data rates for voice modems had reached only about 1200–2400 bps. A decade later, however, modem data rates had increased by about a factor of ten and newer protocols such as V.34 also began to incorporate data compression techniques that multiplied the effective throughput by an additional factor of 2–4 times. The market for modems was also highly competitive driving down modem prices while delivering higher and higher speed modems into the marketplace.

ISDN had great difficulty competing in the modem market for three reasons. First, and perhaps most importantly, modems use voice grade lines, which were already ubiquitous. No special equipment beyond the modem and a PC or terminal was required to make the network connection. ISDN, by contrast, required that the lines sending ISDN signals be specially engineered to carry those signals, costs that were inevitably passed on to the customer. Second, the BRI ISDN offering, intended for the home or small business users, was not a scalable technology. Modems, by contrast, were effectively scalable since all that was required was for a consumer to replace the modem with a newer model. Third, the market for modems was extremely competitive, rapidly producing new products at increasingly lower prices. Under the circumstances of the ubiquity of the voice lines, ISDN's lack of scalability, and fierce price competition in the modem market, it is small wonder that ISDN, a product designed for a very different market, did poorly.

Rapid increases in the use of the Internet also provided a major source of competition for ISDN services. Although some authors have described ISDN as being "a solution looking for a problem," the same could easily be said of the Internet, which was developed as a general-purpose platform for networking.

Like the modem market, however, and partly driven by it, the Internet's high growth rates were sparked by low prices for Internet access, a very cheap access platform, that included the telephone network itself coupled with cheap modems and PCs, with the main applications being e-mail and the World Wide Web. ISDN, by contrast, was developed first and foremost, to solve the technological problem of providing digital telephony to telephone users. No part of this solution, however, was market-oriented and thus after solving the "problem" of digital telephony, it foundered because it failed to address user needs.

Although it is still too soon to make a final pronouncement on B-ISDN, that technology appears to be headed for a similar fate. While the SONET physical layer protocol has indeed achieved a global success and is clearly replacing the T and E carriers around the world, the combination of SONET and ATM has not become widespread.

In fact, it now appears that the ATM protocol may well face the same fate. Although the ATM protocol was developed with a number of features intended to make it the protocol of choice for real-time networking, such as high-speed, small, fixed packet size, quality of service, and end-to-end protocol implementation, ATM does a poor job of transporting what are now the most popular networking protocols, Ethernet and IP.

Because of the transport compatibility issue, ATM has not become popular in LAN environments, in spite of its high speed and other desirable features for real-time operations. More recently, Ethernet, which is now used on over 95% of LANs worldwide, has been introduced into the WAN and is being modified to include an additional tag field, that allows frame prioritization. While not initially suited to compete in the WAN environment or to transport real-time applications, Ethernet appears to be evolving to compete with ATM by adopting many of the features considered desirable in the ATM protocol. Thus, although ATM has done well in the WAN environments to date, it is now possible that the ATM protocol may be replaced by Ethernet over the next few years, and both ISDN and B-ISDN may well turn out to be remembered more for their failure to succeed in the globally competitive telecommunications marketplace than for their roles as first movers into the high-speed digital networking market.

SEE ALSO THE FOLLOWING ARTICLES

Frame Relay • Future of Information Systems • Mobile and Wireless Networks • Multiplexing • Standards and Protocols in Data Communications • Telecommunications Industry • Transmission Control Protocol/Internet Protocol (TCP/IP) • Voice Communications • Wide Area Networks

BIBLIOGRAPHY

Albuquerque, A. A., and Griffith, M. (1995). Overview of ISDN and B-ISDN: User activities and practices. Paper presented at the Asia-Pacific ISDN/B-ISDN Conference, Singapore, October. Available at http://media.it.kth.se/SONAH/ANALYSYS/race/pl4/dissemin/plpubs/isdn-cnf.htm.

Allen, R. E. (1996). Information unbound: Customer choice, technology and the communications revolution. Speech to the Midwest Research Institute. Available at http://www.att.com/speeches/item/0,1363,2007,00.html.

Arlandis, J. (1994). ISDN: A European prospect, in *Telecommunications in transition* (C. Steinfield, J. Bauer, and L. Caby, eds.), Chap. 12, pp. 223–235. Thousand Oaks, CA: Sage.

Azzam, A., and Ransom, N. (1999). *Broadband access technologies.* New York: McGraw–Hill.

Bates, R. J. (2000). *Broadband telecommunications handbook.* New York: McGraw–Hill.

Bellamy, J. (1991). *Digital telephony,* 2nd ed. New York: Wiley.

Inose, H. (1979). *An introduction to digital integrated communications systems.* Tokyo: Univ. of Tokyo Press.

Lai, V. S., and Guynes, J. L. (1993). ISDN: Adoption and diffusion issues. *Information systems management,* Vol. 10, No. 4, 46–52.

Lai, V. S., and Guynes Clark, J. L. (1998). Network evolution towards ISDN services: A management perspective. *Journal of Information Technology,* Vol. 13, 67–78.

Mayo, J. S. (1986). Evolution of the intelligent network, in *The information technology revolution* (T. Forester, Ed.), pp. 106–119. Cambridge, MA: MIT Press.

Miller, M. J., and Ahmed, S. V. (1987). *Digital transmission systems and networks. Vol. I. Principles.* New York: Comput. Sci. Press.

Mueller, M. (1997). *Universal service: Competition, interconnection and monopoly in the making of the American telephone system.* Cambridge, MA: MIT Press.

Noam, E. (1987). The public telecommunications network: A concept in transition. *Journal of Communication,* Vol. 37, 20–48.

Noam, E. (1989). International telecommunications in transition, in *Changing the rules: Technological change, international competition, and regulation in communications* (R. W. Crandall and K. Flamm, Eds.). Brookings Institute.

Ronayne, J. (1988). *The integrated services digital network: From concept to application.* New York: Wiley.

Srinivasan, S. (1995). Business applications of ISDN. *Journal of Information Technology,* Vol. 12, No. 3, 53–55.

Stallings, W. (1999). *ISDN and broadband ISDN with frame relay and ATM,* 4th ed. Englewood Cliffs,NJ: Prentice Hall.

Stewart, A. (1997). Avoiding the market mistakes of ISDN. America's Network, July 15. Available at http://www.americasnetwork.com/issues/97issues/970715/071597_isdnmistakes.html.

Xircom Corporation (2000). Technology background: ISDN Overview July 14. Available at http://www.xircom.com/cda/page/1,1298,0-0-1_1-333,00.html.

Intelligent Agents

Rahul Singh
University of North Carolina, Greensboro

GLOSSARY

agent One authorized to act for or in place of another, as a representative; something that produces or is capable of producing an active or efficient cause.

artificial intelligence A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes.

distributed systems Systems consisting of multiple computers or processing centers, each performing a part of the overall computing required to complete a task. In such systems, the processing and memory resources required for the computation are distributed across a network of computers to create a distributed computing environment or a distributed system.

e-commerce The exchange of goods and services over the Internet or the World Wide Web.

information systems A general term used for computers involved in processing the data and information resources for a business organization. The terms information systems, information technology, and management information systems are used interchangeably in the literature.

intelligent agents A computer system, situated in some environment, that is capable of flexible autonomous action in this environment in order to meet its design objectives

INTELLIGENT AGENTS and multiagent technologies are emergent technologies with potential for many applications in information technology. An intelligent agent is "a computer system situated in some environment and that is capable of flexible autonomous action in this environment in order to meet its design objectives" (Jennings and Wooldridge, 1998). Agent-based systems range from the simple, single agent system that performs tasks such as e-mail filtering to very complex, distributed systems of multiple agents each involved in individual and system-wide goal-oriented activities. With the recent growth of the Internet and Internet-based computing, and the explosion of commercial activity on the Internet in recent years, the application of agent-based systems is being explored in a wide variety of electronic commerce (e-commerce) applications, including on-line consumer purchasing, agent-based network management, supply chain systems, information retrieval, Internet-based auctions, and on-line negotiations. This chapter presents an overview of the current state of the art in intelligent agents, discusses their research foundations and their application in various areas, and explores some future directions in the application of intelligent agents and agent-based systems.

I. INTRODUCTION

Autonomous intelligent agents and multiagent technologies are emerging technologies with promise for a wide variety of applications in information technology. Research in intelligent agents, along with recent growth in the volume of e-commerce, has significant implications for many areas of application, including decision support, planning, and negotiation. However, there is little agreement in the fields of computer

sciences and information systems on the definition of an intelligent agent. One commonly accepted definition for an intelligent agent is given by Jennings and Wooldridge (1998) who define intelligent agents as *"a computer system situated in some environment and that is capable of flexible autonomous action in this environment in order to meet its design objectives."*

With foundations in artificial intelligence and distributed computing technologies, the study of intelligent agents has developed into its own area of research and application. Agents are applied in many application domains with roles that range from acting as personal digital assistants for end users to control agents in dynamic and complex time and mission critical process control systems. Multiple channels of e-commerce activities have seen successful applications of intelligent agent technology, including consumer-to-consumer, business-to-consumer, and business-to-business e-commerce.

The following section speaks to the foundations of intelligent agent research, which are grounded in the research in artificial intelligence and in the design and development of distributed computing systems. A number of definitions of intelligent agents and taxonomy of the various types of intelligent agents from the literature are reviewed and summarized. Agent-oriented systems in various areas of application are discussed with attention to security and applicability as major concerns in the successful deployment of agent-based applications. Agent-based systems deployed over the Internet and the multitude of their applications in e-commerce seem to offer promise for future applications of agent-based systems. This topic is presented in detail with reference to prevalent research. A summary is presented and directions for future research are discussed.

II. FOUNDATIONS

A. What Are Intelligent Agents?

The *Merriam-Webster's Collegiate Dictionary* defines an intelligent agent as

> something that produces or is capable of producing an effect: an active or efficient cause; a means or instrument by which a guiding intelligence achieves a result; one who is authorized to act for or in the place of another.

In the previous definition of an intelligent agent, autonomy of action refers to the ability to act without human intervention. The concept of agent-based systems was developed by McCarthy as systems that could carry out the details of the appropriate computer operations and could ask for and receive advice offered in human terms when required. An agent would be a "soft robot" living and doing its business in the computer's world. The terms agents, software agents, and intelligent agents are often used interchangeably in current information systems and computer sciences literature. However, it is noteworthy that all agents do not have to be intelligent. Jennings and Wooldridge (1998) observe that agents in agent-based systems are not necessarily intelligent and that the agent has to have *flexibility* in order to be considered intelligent. Flexibility in intelligent agent-based systems requires that the agents should have the following characteristics:

- Be cognizant of their environment and be responsive to changes therein
- Be reactive and proactive to opportunities in their environment
- Be autonomous in goal-directed behavior
- Be collaborative in their ability to interact with other agents in exhibiting the goal-oriented behavior
- Be adaptive in their ability to learn with experience.

Hess *et. al.* (2000) analyze many definitions from the literature and propose the following as a definition:

> An autonomous software agent is a software implementation of a task in a specified domain on behalf or in lieu of an individual or other agent. The implementation will contain homeostatic goals, persistence, and reactivity to the degree that the implementation (1) will persist long enough to carry out the goal(s), and (2) will react sufficiently within its domain to allow goal(s) to be met and to know that fact.

Table I summarizes the characteristics of intelligent agents that emerge as common aspects across many definitions found in the literature.

Agent-based systems may consist of a single agent engaged in autonomous, goal-oriented behavior or multiple agents that work together to exhibit granular as well as system-wide, goal-directed behavior. The general multiagent system is one in which the interoperations of separately developed and self-interested agents provide a service beyond the capability of any single agent-based system. Such mutli-agent systems provide a powerful abstraction to model systems where multiple entities, exhibiting self-directed behaviors, must coexist in an environment and achieve the system-wide objective of the environment. From a re-

Table I Summary of the Characteristics of Intelligent Agents

Characteristics of intelligent agents	
Responsive:	React to changes in the environment
Proactive:	Approach opportunities in the environment
Autonomous:	Goal-seeking behavior
Collaborative:	Able to interact with other agents in goal-oriented behavior
Adaptive:	Able to learn and improve with experience

view of the literature, it is clear that agents are powerful abstractions in the modeling and design of systems that require independent action at various levels of granularity of the system.

B. Foundations of Research in Intelligent Agents and Agent-Based Systems

Artificial intelligence (AI) is concerned with developing machines that can perform tasks believed to require human intelligence. The design of computer-based systems that can perform independent, intelligent actions that produce interesting and meaningful results is a clear objective of research in AI. AI inherits from many parent fields including mathematics, philosophy, psychology, computer engineering, and linguistics and has historically been concerned with symbolic representation and manipulation and theorem proving. This is demonstrated in Newell and Simon's research on logic theorist and the general problem solver (Newell *et al.,* 1963). Early work in AI by McCulloch and Pitts focused on modeling computational activity as a network of neurons. This work formed the basis for development of artificial neural networks. Much of the early research in AI was concerned with the creation of systems that could think like humans. Recent and more prevalent work in AI focuses on the problem-solving ability of systems and the development of systems that "can think like humans and not act like humans." Bellman (1978) defines AI as the automation of "activities that we associate with human thinking, activities such as decision-making, problem solving, learning." Winston (1992) defines AI as "the study of the computations that make it possible to perceive, reason and act." Schlakoff (1990) gives another definition of artificial intelligence: "Artificial intelligence is a field of study that seeks to explain and emulate intelligent behavior in terms of computational processes." Albus (Albus, 91) defines intelligence as "the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral sub-goals that support the system's ultimate goal." These definitions focus on the ability of intelligent entities to exhibit independent, directed, and goal-oriented behaviors while processing information in a rational manner in order to solve problems or make decisions. The study of intelligent agents focuses on the design of such independent entities that exhibit independent, goal-oriented behavior.

Conceptual foundations of research in intelligent agents are developed from research in distributed AI (Rich and Knight, 1991). Distributed AI grows from research in the fields of distributed computing and AI and attempts to develop ways to efficiently distribute computing across multiple, local computing sites while maintaining global, goal-oriented behavior. This research is concerned with the "analysis and development of intelligent communities of agents that comprise collections of interacting coordinated knowledge-based processes" (Ottaway and Burns, 1997). For distributed AI, it is observed that a system modeled as a collection of independently acting agents is more beneficial than a system modeled as a monolithic entity. Thus, a principal concern of distributed AI is the development of rules and architectures that govern the interaction between the multiple, independent, knowledge processes (Gasser, 1991). Distributed AI systems also afford the solution a degree of parallelism and make the resulting design more robust and scalable. The modularity of the models makes the programmers' tasks of implementation easier, since it is easier to conceptualize larger systems, and allows for an easier breakdown of the cognitive task load. The agent paradigm provides a powerful abstraction to model problem-solving efforts in this area.

A useful comparison from the systems development perspective is to analyze the contrast between the intelligent agents paradigm and agent orientation with object orientation. Object orientation and agent orientation represent powerful techniques to model modular systems. It should be clearly noted that the two are not the same. Jennings and Wooldridge (1998) provide some useful comparisons between the two powerful and popular modeling methods that help delineate this distinction. The key abstraction in an object-oriented system is an object that encapsulates information on the system state. Objects are able to communicate with other objects in the environment by passing messages to invoke encapsulated methods. When an object receives a valid message from another object, it performs the invoked method and may perform a transition of state. Agent-based systems use agents as the key

abstraction in modeling systems where the entities must execute independent, goal-oriented behaviors. In addition to encapsulating state, an agent also encapsulates behavior by maintaining control over the decision of execution of methods (Jennings and Wooldridge, 1998). Table II summarizes the comparison between object and agent orientations.

From its foundations in AI and distributed computing, research in intelligent agents has found favor in information systems and has seen an exponential jump since its demonstrated utility in Web-based applications for e-commerce and the need for management and control of remote systems over the Internet.

C. Types of Intelligent Agents

It is difficult to develop a complete taxonomy of a field that is so new in its research and application that novel classifications are being developed every day. Franklin and Graesser (1996) developed a taxonomy of intelligent agents that was inspired by the biological model for developing taxonomies. They also offer some suggestions on other possible means for the classification of intelligent agents, including those based on the technology used, central vs distributed systems, and learning vs nonlearning. Figure 1 shows the taxonomy developed by Franklin and Graesser.

An additional dimension for the classification of intelligent agents is based on their primary attributes. Nwana (1996) developed a set of primary attributes of intelligent agent-based systems and proposed a typology to classify agents on dimensions such as mobility; the presence of a symbolic reasoning model; and the exhibition of the primary attributes, such as autonomy, cooperation, and learning. From these dimensions, intelligent agents are classified as collaborative agents, interface agents, mobile agents, information/Internet agents, reactive agents, hybrid agents, and smart agents (Nwana, 1996; Bradshaw, 1997). Intelligent agents may be classified on these multiple dimensions.

The following is a nonexhaustive list of dimensions that may be used to classify intelligent agents and the classifications that may result.

1. Number of Agents Employed: Single Agent Systems and Multiple Agent Systems

In the simplest case, an agent-based system may consist of a single agent that is the locus of control for the system and is responsible for the goal-oriented behavior of the system. The single agent is responsible for receiving direction from the user and exhibiting autonomous, goal-seeking behavior to retrieve the desired results for the user.

Systems that employ multiple agents, each with behavior directed to the achievement of subgoals and working toward the attainment of a larger, system-wide objective, represent the multiple agent case. Such systems may take direction from single or multiple users and interact with each other to retrieve results for each user, the subgoals and fulfill the design objectives of the system.

2. Mobility: Static Agents and Mobile Agents

Static agents remain resident on a single computing location and do not change this location in the course of their autonomous, goal-oriented actions. A simple example of such agents is an e-mail filter agent that stays resident on a single machine and performs filtering actions on behalf of the user.

Mobile agents move from one computing location to another, carrying state and behavior information to

Table II Comparison of Object Orientation and Agent Orientation

Property	Object orientation	Agent orientation
Key abstraction	Objects as the modular components of the system	Agents as the modular components of the system
Autonomy	Limited autonomy; objects may exist independently, but they must execute valid methods upon invocation	Autonomous entities; agents decide whether to execute an invoked method
Encapsulation	An object encapsulates state information	An agent encapsulates state information and behavior
Proactiveness	An object is typically not proactive	An intelligent agent is proactive by design

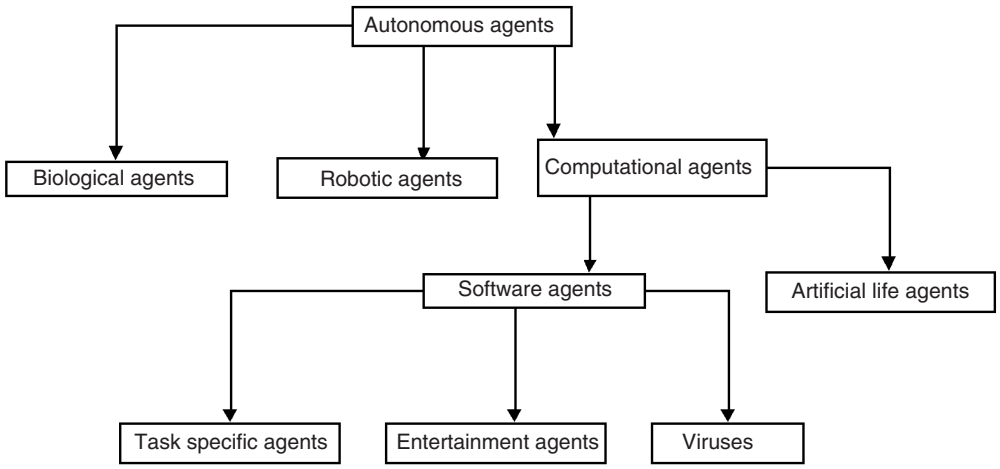


Figure 1 A taxonomy of intelligent agents. (Adapted from Franklin and Graesser, 1996.)

remote locations in working to achieve their objective. Examples of mobile agents include agents involved in negotiations between buyers and sellers for e-commerce and agents that help achieve load balancing in distributed computing environments.

3. Collaboration: Collaborative Agents and Competitive Agents

In multiagent systems, an agent will interact with many other agents that may exhibit collaborative or competitive behavior. Research has modeled the interaction between agents to be competitive or collaborative based on the modeled interaction of the system. A source of information about this behavior is "Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce" (Guttman *et al.,* 1998).

III. APPLICATION

A. Application of Intelligent Agents and Agent-Based Systems

Intelligent agents assist users by reducing the cognitive workload associated with difficult tasks, performing extensive searches of vast information spaces, and training users on how to perform complex tasks. Agents alter the state of the art in human computer interaction by allowing the computer to suggest tasks and alternatives for the user. The set of tasks that an intelligent agent can assist in seems to be limitless at first glance. A review of the recent literature reveals that agent technology holds much promise for the development of complex distributed systems.

Many technical and behavioral hurdles exist in the successful wide-scale deployment of intelligent agent technology. Some of these include:

- The development of appropriate models for the interaction between agents in multiagent systems
- Consideration of the security concerns that arise from allowing mobile and autonomous entities to have free rein of a computer network
- Consideration of privacy issues in ensuring that private and sensitive information is only exchanged between authorized parties

A common pitfall in the application of any emergent technology is over-application—the attempt to apply the technology in areas where its use is not appropriate. Intelligent agents offer promise in modeling a single or multiuser distributed environment where tasks are appropriately entrusted to software entities that perform autonomous behaviors on behalf of the user in reaching individual and system-wide goals. This is a broad definition and care needs to be taken to apply this in appropriate environments. Nwana (1996) offers perspectives on the applications of agent-based research and outlines pitfalls and shortcomings in the deployment of agent-based systems. Some areas of application that seem to hold promise include:

- E-commerce applications to provide value-added and personalized support and recommendation to on-line retail, supply chain integration, and personalized customer interaction
- Corporate intranets or outsourced solutions to provide value added and personalized searching and collation support

- Personal assistants to provide information filtering activities
- Resource allocation and management systems that achieve a system state through competitive and collaborative interactions among agents

B. Security and Intelligent Agents

Security in multiagent systems is a complex problem since agents are designed to exhibit autonomous behavior in a dynamic environment. This requires access to various resources on other computers sharing information with various other agents. Additionally, in the case of mutli-agent systems, multiple machines may be at risk since mobile agents are required to send messages to one another and may transport various state and behavior information to different machines. In such environments that are open by design, security takes on increased concern as important and even sensitive information may be exchanged between computing locations (Gray *et al.,* 1998). In mobile agent-based systems, the entities that need protection include the agents that interact with each other in execution of the goal-oriented behavior of the system and the resident computers that the agents reside on at various points in time. The systems' design must ensure that appropriate access rights are provided to agents and that the systems themselves maintain control of the actual data and processing of the information. Gray *et al.* (1998) provide a detailed discussion of the security implications of a mobile agent system.

Mobile agents contain processes and data that must be allowed to execute and reside on multiple hosts. A significant security concern with intelligent agents is the risk of virus infections that could be carried by the intelligent agents and may travel from one machine to another by attaching themselves to other agents that the carrier agent interacts with. Viruses are a major concern in the case of open systems, such as those that interact over the Web where servers must allow mobile code to execute within their domain. This concern intensifies in Internet agent systems where it is difficult to detect malicious activity before it becomes apparent through failure. Users of Web pages are typically unaware of background processing that occurs and may not be alert to malicious activities. Many servers may block access or otherwise control the activities of malicious agents; however, this may not be enough to prevent security threats to remote computers. Schoder (2000) lists security concerns as a significant and yet unsolved problem for mobile agent systems.

Table III Application Domains on Intelligent Agents

Industrial applications
Process control
Manufacturing
Air traffic control
Commercial applications
Information management
Electronic commerce
Business process management
Medical applications
Patient monitoring
Health care
Entertainment applications
Games
Interactive theater and cinema

C. Areas of Application of Intelligent Agents

Jennings and Wooldridge (1998) provide multiple application domains and examples of the application of intelligent agent technology and intelligent agent-based systems to real-world situations. They broadly classify intelligent agents application domains based on their area of application. Table III summarizes the various application domains.

IV. FUTURE DIRECTIONS: E-COMMERCE AND INTERNET AGENTS

In the past few years, the popularity and availability of the Internet and the World Wide Web have become major vehicles for growth in on-line commerce. The Census Bureau of the U.S. Department of Commerce estimated that U.S. retail e-commerce sales for the second quarter of 2000, not adjusted for seasonal, holiday, and trading-day differences, was $5.518 billion, an increase of 5.3% from the revised first quarter of 2000 level. The first quarter estimate was revised from $5.26 billion to $5.24 billion (U.S. Department of Commerce, 2000). The Department of Commerce measures e-commerce sales as the sales of goods and services over the Internet, an extranet, the electronic data interchange (EDI), or other on-line system, where payments may or may not be made on-line. A recent report on the digital economy attributes the recent rapid growth of the Internet to its strength as a medium of communication, education, and entertainment and as a tool for e-commerce. It is

clear from all sources that advances in information technology and e-commerce have been a significant contributor to the recent growths in the economy.

Bakos (1998) points out that markets match buyers and sellers, facilitate transactions between them, and provide an institutional infrastructure to support the transactions (Table IV). In the contemporary marketplace, the first two of these three functions are conducted with intermediaries. In the electronic marketplace, these functions may be facilitated using electronic intermediaries by leveraging the efficiencies afforded by information technologies (Bakos, 1998). Intelligent agent technologies hold great promise for fulfilling the role of intermediary in the electronic marketplace and for supporting, or conducting on behalf of the user, the processes involved in matching buyers and sellers and facilitating the transactions between them. The Internet is a large, distributed environment platform where multiple agencies conduct commercial activity. This activity involves the search for sellers with products to suit the buyers defined by price, quality, and other business considerations; the search for buyers who will buy the products of a seller; and the facilitation of such transactions. Intelligent agent technology has the ability to search a large information space for specific needs and to identify such sources. Intelligent agents can perform such searches within the parameters defined by the user and facilitate the transaction by bringing the resource to the user and acting on behalf of the user to facilitate transactions. Therefore, it is not surprising that significant attention by academic and business communities is paid to this technology to facilitate and empower e-commerce.

Table IV Functions of a Marketplace

Matching buyers and sellers
Determine product offerings
Product features offered by sellers
Aggregation of different products
Search (buyers for sellers and sellers for buyers)
Price and product information
Matching seller offerings with buyer preferences
Price discovery
Process and outcome in determination of process
Facilitating transactions
Logistics: delivery to buyer
Settlement: payment to seller
Trust: credit system, reputations, etc.
Institutional infrastructure
Legal
Regulatory

Adapted from Bakos, 1999.

Intelligent agents that are primarily directed at Internet and Web-based activities are commonly referred to as Internet agents. Many agent systems exist in the e-commerce area that perform limited functions to support users (examples include Andersen Consulting's BargainFinder that undertakes price comparison, as does Jango [see Doorenbos *et al.,* 1997] and AuctionBot [see Wurman *et al.,* 1998]; Kasbah [Chavez *et al.,* 1997] supports product transactions [Macredie, 2000]). The Association for Computing Machinery (ACM) dedicated a special issue of *Communications of the ACM* in March 1999 to intelligent agent technologies and focused on how "software agents independently and through their interaction in multi-agent systems are transforming the Internet's character. Agents and the business performance they deliver will be involved in up to $327 billion worth of Net-based commerce in five years according to Forrester Research" (Rosenbloom, 1999).

Intelligent agents carry out activities in the electronic marketplace on behalf of the human user. Maes *et al.* (1999) present a model for the behavior of intelligent agents over the Web through traditional marketing consumer buying behavior models. The consumer buying behavior model illustrates the actions and decisions involved in buying and using goods and services. This model is adapted in their research to consumer buying in the electronic marketplace and the use of intelligent agents in facilitating this activity. They present six states to categorize agent mediated e-commerce:

1. Need identification
2. Product brokering
3. Merchant brokering
4. Negotiation
5. Purchase and delivery
6. Service and evaluation

Intelligent agents may be used to facilitate a number of these stages in the consumer buying model. Table V presents a summary of the activities involved in each of these stages and provides suggestions for the facilitating role agents may play in each of these stages.

The model and the associated applications of intelligent agent technology shown in Table V provide a foundation for the analysis and development of intelligent agent-based systems for Internet-based application deployment. Individual components of the consumer behavior model's application to agent-assisted e-commerce may have greater significance than others based on the nature of application.

Table V The Consumer Buying Behavior Model and Intelligent Agent Support

Consumer buying behavior model stage	Activities	Intelligent agent facilitation
Need identification	Realization of unfulfilled needs by the consumer	Tools that alert the user to needs or opportunities based on knowledge of the user's preferences or business environment are useful in this regard
Product brokering	Refers to retrieval of information about products	The agent is primarily involved in search activities to determine products and services to suit the consumer's needs
Merchant brokering	This stage provides a filtering of the information retrieved by the agents on the various products based on the criteria of the buyer, resulting in the creation of a set of feasible sellers	This stage is analogous to many traditional decision support activities that require choice on the part of the user. Agents may facilitate the ranking of alternatives, thereby facilitating the generation of choice from the user
Negotiation	This stage determines the terms of the transaction. It varies in duration based on a number of factors including the complexity of the marketplace, the number of feasible alternatives, and the monetary and business value of the transaction	A dynamic stage where the buyer and seller(s) agents communicate preferences to find a mutually agreeable set of terms for the transaction. This activity may be facilitated by agents through communication abilities and matching of the needs of the buyer with the capabilities of the seller
Purchase and delivery	Upon completion of negotiations, the buyer makes the purchase and delivery of the goods or services occur based on the terms agreed upon in the negotiation stage	This is typically where the transaction moves from the electronic to the physical in the case of tangible goods or services, or comprises content delivery which simply requires a communication medium
Service and evaluation	After the purchase has been made, the buyer will engage in postpurchase evaluation to determine an overall satisfaction with the purchase decision	This is a subjective stage where the user decides the utility of the product or service delivered. This stage does form the input for developing a preference for one seller over another, which is useful input for the merchant brokering and negotiation stages. This information provides guidance and input for development of learning and the adaptive behavior of the intelligent agent

An analysis of the business and technology literatures provide indications of new and promising applications of intelligent agent-based technology every day. Noteworthy among them are applications in supply chain integration, business-to-business e-commerce, on-line negotiations, auctions, intelligent information retrieval, etc. It is clear that business and research communities share a zeal for the promise of intelligent agent-based technology and its potential for developing exciting applications over the Web. There are demonstrated successes in some areas, including information retrieval, personal assistants, negotiations, and auctions. The durability of these applications and designs is still an open question at this time and it will be interesting to see the long-term viability of these directions.

V. CONCLUSIONS

This article presents an overview of the current state of the art in intelligent agents. Intelligent agent technology holds a lot of promise for the development of complex and distributed systems over computer networks. The concept of autonomous digital emissaries that are capable of autonomous intelligent action is very appealing to information technology and the business computing communities at large. Many efforts are underway in industry and in research to realize the potential of intelligent agent technology.

Much of the recent work in intelligent agents is in the area of finding novel applications for this technology. Examples of such application efforts include supply chain systems, decision support systems devel-

opment, process control, automated parts procurement, and a wide variety of Internet-based applications. Preliminary results in these applications are certainly promising. As with many new and promising technologies, more research is needed in the development of models and tools for agent-oriented systems development. Research in the development of methodologies for agent-oriented systems development and in the appropriateness of the characteristics of a problem domain for agent-oriented development would help further the understanding of the domain and make for more robust systems development. As stated earlier, it is clear that agent orientation holds a lot of promise. More research is needed to provide guidance in the development of intelligent agent-based systems and in the selection of agent-based system as an appropriate methodology vis-à-vis other systems development methodology. Nevertheless, research in this field is certainly exciting and will make for some exciting application development and advances in a variety of areas.

SEE ALSO THE FOLLOWING ARTICLES

Distributed Databases • Electronic Commerce • Electronic Commerce, Infrastructure for • Expert Systems • Hybrid Systems • Machine Learning • Supply Chain Management

BIBLIOGRAPHY

Albus, J. S. (May/June 1991). Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. 21, No. 3.

Bakos, Y. (August 1998). The emerging role of electronic marketplaces on the internet. *Communications of the ACM* 41, 8, 35–42.

Bellman, R. E. (1978). *An introduction to artificial intelligence: Can computers think?* San Francisco: Boyd & Fraser Publishing Company.

Bradshaw, J. M., Ed. (1997). *Software agents.* Boston: MIT Press.

Chandra, J. (January 2000). Information systems frontiers. *Communications of the ACM,* Vol. 43, Iss. 1, 9, 71.

Chavez, A., Dreilinger, D., Guttman, R., and Maes, P. (April 1997). A real-life experiment in creating and agent marketplace. *Proceedings of the Second International Conference on the Practical Application of Agents and Multi-Agent Technology (PAAM '97),* London.

Doorenbos, R., Etzioni, O., and Weld, D. (February 1997). A scalable comparison-shopping agent for the world wide web. *Proceedings of the First International Conference on Autonomous Agents,* Marina del Rey, CA.

Franklin, S., and Graesser, A. (1996). Is it an agent or just a program? A taxonomy for autonomous agents. In *Proceedings of the third international workshop on agent theories, architectures, and languages.* New York: Springer-Verlag.

Gasser, L. (1991). Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence,* Vol. 47, 107–138.

Gray, R. S., Kotz, D., Cybenko, G. Rus, D. (1998). D'Agents: Security in a multiple-language, mobile-agent system. *Mobile agents and security* (G. Vigna, Ed.), 154–187. Springer-Verlag

Guttman, R. H., Maes, P. and Moukas, A. G. (May 1998). In Schmid, B. F., Selz, D., and Sing, R. *EM - Electronic Transactions. EM - Electronic Markets,* Vol. 8, No. 1.

Hess, T. J., Rees, L. P., and Rakes, T. R. (Winter 2000). Using autonomous software agents to create next generation of decision support systems. *Decision Sciences.*

Jennings, N. R., and Wooldridge, M. (1998). *Agent technology: Foundations, applications, and markets.* London: Springer-Verlag.

Kay, A. (1984). Computer software. *Scientific American,* Vol. 251, No. 3, 53–59.

Macredie, R. D. (October 1998). Mediating buyer-seller interactions: The role of agents in the Web commerce. In Schmid, B. F., Selz, D., and Sing, R., Eds. EM—Electronic Contracting. EM—Electronic Markets, Vol. 8, No. 3; URL. (2000). <http://www.electronicmarkets.org/netacademy/publications.nsf/all_pk/1081> [09/18/2000].

Maes, P., Guttman, R., and Moukas, A. (March 1999). Agents that buy and sell. *Communications of the ACM 42,* 3, 81–91.

Newell, A., Shaw, J. C., and Simon, H. A. (1963). Programming the logic theory machine. *Proceedings of the Western Joint Computer Conference,* 15: 218–239. Reprinted in Feigenbaum and Feldman.

Nwana, H. S., (1996). Software agents: An overview, *The Knowledge Engineering Review* 11 (3).

Ottaway, T. A., and Burns, J. R. (Summer 1997). Adaptive agile approaches to organizational architecture utilizing agent technology. *Decision Sciences,* Vol. 28, No. 3.

Rich, E., and Knight, K. (1991). *Artificial Intelligence,* 2nd ed. New York: McGraw-Hill.

Rosenbloom, A. (March 1999). Editorial pointers. *Communications of the ACM,* Vol. 42, No. 3.

Schalkoff, R. J. (1990). *Artificial Intelligence: An Engineering Approach.* Highstown, NJ.: McGraw-Hill.

Schoder, D. (June 2000). The real challenges of mobile agents. *Communications of the ACM,* Vol. 43.

Shoham, Y. (1997). An overview of agent-oriented programming. In *Software Agents* J. M. Bradshaw, Ed.). Menlo Park, CA: AAAI Press.

Winston, P. H. (1990). *Artificial Intelligence.* Reading, MA: Addison-Wesley.

Wurman, P., Wellman, M., and Walsh, W. (May 1998). The Michigan internet AuctionBot: A configurable auction server for human and software agents. *Proceedings of the Second International Conference on Autonomous Agents.* Minneapolis/St. Paul.

Internet Homepages

Michael L. Rodgers, William E. Snell, Jr., and David A. Starrett
Southeast Missouri State University

GLOSSARY

browser A program, such as Netscape Navigator or Internet Explorer, that requests and interprets hypertext markup language (HTML) documents from a server, rendering the documents viewable on the user's computer screen.

client An Internet-connected computer that requests and receives information from a server. A client typically uses a browser to communicate with servers.

firewall A system designed to prevent unauthorized access to or from a private network connected to the Internet. Implemented with hardware, software, or a combination of both.

home page A Web page; often intended to provide an introduction or an entry point to a Web site.

hypertext markup language (HTML) A document production tool that uses tags to specify the document's components or elements, primarily according to function. To make the document platform independent, HTML does not specify exact formatting. Formatting is applied to the document by the user's browser at the time of viewing.

Internet A globally accessible collection of computer networks linked by the Internet Protocol (IP).

intranet A private network that uses Internet protocols to connect a limited group of users. Intranets that are not completely separate from the Internet are linked to the Internet through a security system such as a firewall.

java A platform-independent, object-oriented programming language often used to add interactivity or special effects to Web pages.

JavaScript A scripting language developed to enable Web authors to design interactive sites. Developed independently of Java, JavaScript can interact with HTML source code, enabling Web authors to enhance their sites with dynamic content.

server An Internet-connected computer configured to deliver documents or services at the request of client computers.

server-side Referring to software that resides in and functions on a server, as opposed to the end user's (client's) computer.

uniform resource locator (URL) A file address that provides browsers with protocol information and the file's Internet location. The URL for a Web page specifies the hypertext transfer protocol (HTTP) protocol with the character string, "http:."

Web page An HTML document that is available to browsers connected to the World Wide Web.

Web site A collection of Web pages, usually thematically organized.

World Wide Web (WWW, W3, or "the Web") An Internet service based on HTTP and HTML. The Web is characterized by its multimedia capabilities and extensive use of hyperlinks to share information.

THE MILLIONS OF WEB PAGES available on the World Wide Web (the Web) have captured the interest and

financial resources of people throughout the world. Product information, entertainment, archived data, social encounters, and university educations are just a few of the things found packaged in Web pages available to anyone connected to the Internet. Indeed, the original intent for the Internet, to be a community-building tool that leverages resources and increases the speed of collaboration between users, is expressed again and again by the Web pages of today. Web pages are related to one another through logical connections of content that transcend time and distance, and they are shared by users connected to the Internet through a wide variety of hardware and software. While the technical expertise necessary to publish and maintain pages on the Web is by no means trivial, the procedures and infrastructure are well established and are accessible enough to allow even novices to have a meaningful presence on the Web. The simple fact that millions of people are putting pages on the Web suggests that Web pages and the technologies that support them are changing the way that people interact with one another.

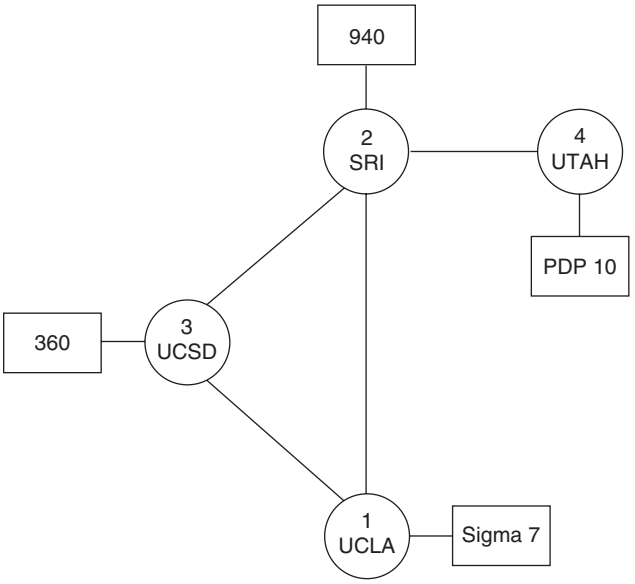


Figure 1 Early map of the ARPANET. (Courtesy of the Computer Museum History Center, Mountain View, California).

I. ORIGINS OF THE INTERNET, THE HOME OF THE WORLD WIDE WEB

The notion of a network of computers was developed about 15 years after the first digital computers were deployed in the late 1940s. Among a number of early efforts was a Rand Corporation study, written by Paul Baran who argued for a decentralized computer network that could continue to function after the loss of several of its members. The decentralized network was realized in 1969 with the development of the U.S. Department of Defense Advanced Research Project Agency's ARPANET, a collection of several computers that could share resources such as databases and software subroutines.

As Fig. 1 indicates, early maps of the ARPANET showed that different models of computers were connected, suggesting that hardware and software compatibility over multiple systems was a concern from the very beginning of computer networking. By the mid-1980s the compatibility issues as well as "infrastructure" issues, such as message routing and network loading, had been addressed well enough to allow many government agencies, defense contractors, and universities to establish ARPANET connections. The ARPANET's focus on community, especially as expressed through resource sharing, led to development of collaboration tools such as Telnet and e-mail, which ultimately powered the evolution of the Internet as a societal environment.

The success of the ARPANET as a shared-resource work platform led to a proliferation of academic, governmental, and business networks. The advent of the personal computer and high-speed modem made network connection cheaper and more widely available by the early 1980s. Perhaps inevitably, the value of collaboration between users in different networks was recognized, and efforts to reconcile the numerous custom-built networks to a more general standard began. In 1985, the U.S. National Science Foundation (NSF) assumed control of parts of the ARPANET with the intent to serve research and scholarly activity in U.S. colleges and universities. The NSF funded the connection of individual educational institutions to the Internet (known at the time as the NSFNET) with the requirement that access must be afforded to **all** qualified users at the institution. Clearly, the Internet was on a course to expand its user base and the variety of activities that it supported.

The Internet culture has been heavily influenced by its roots in academia. Free exchange of ideas, sharing of resources, and the requirement by many institutions that institutional resources (including computer networks) not be used to support for-profit ventures left the software and connection time largely free of charge. Even today, many in the general public expect browsers and other Internet-related software and content to be free and unregulated. Nevertheless, the Internet's potential as a globally accessible productivity tool and marketplace encouraged many

to seek commercial access. The Internet was opened for commercial use in 1991; NSF support was ended by 1995.

II. HYPERTEXT MARKUP LANGUAGE AND THE WORLD WIDE WEB

Much of the Internet's popularity is traced to its ability to create communities that are not linked by a common time and location. Collaborations among people that were formerly achieved by physically bringing a team together could be accomplished by e-mail or FTP. Indeed, the technical specifications for the Internet itself were developed with the help of a system called request for comments (RFC). From their beginnings as mailed paper notes, RFCs migrated to the Internet as text files, allowing easy access by anyone connected to the Internet. The effect of the move was to make participation in the RFC process open to anyone, not merely those who a moderator guessed might be interested in the project. The end users (**customers,** according to business models), and not a moderator, decided who had a voice in the continued development of Internet specifications. Thanks to the Internet, multiple-author comments were more likely to come from geographically diverse authors. The speed of collaboration increased greatly. Furthermore, the availability of actual working documents provided authentic, valuable examples to managers and students all over the world.

By the late 1980s, the Internet had proven itself as an infrastructure for information sharing. However, the information being shared was typically text based, and the information access tools (FTP, Gopher servers) were limited in their ability to meaningfully relate one document to another. In response to these limitations, physicist Tim Berners-Lee, working in Geneva, developed hypertext markup language (HTML), the document production tool that made the Web possible. Berners-Lee built HTML around several crucial innovations, most notably the uniform resource locator (URL) and hypertext transfer protocol (HTTP). The URL provided a convenient way for one document to refer to another, despite the two documents having different Internet locations. HTTP established a format for a client computer to exchange files with a server. With HTML in place, Berners-Lee was able to achieve his goal of a **web of documents,** each linked not by physical proximity, but by the authors' perceptions of the relationships between the documents in the web.

HTML was based on standard generalized markup language (SGML), a text formatting and processing tool. The term "markup" referred to SGML's ability to identify higher level concepts, such as chapters, lists, and paragraphs, in a document. Indeed, unlike other text processing languages, which specified placement of features and content in the document, SGML merely defined where a feature began and ended. The exact formatting of the feature was left up to the specific application. In HTML, Berners-Lee envisioned the **browser** as the application that expressed the format. Not only did this approach provide a way to make documents readable in a variety of platforms and operating systems, but the nesting of document features (divisions such as paragraphs, tables, sections, and chapters) emulated the hierarchical file system structure that computer users were accustomed to using in environments such as DOS and Microsoft Windows. Through adoption of the SGML philosophy, HTML promoted data sharing that was largely platform independent, yet organizationally familiar to users.

Built on the document-organization philosophy of SGML and supported by the innovations of the URL and HTTP, HTML was capable of creating both text-based and multimedia-enabled documents that could be shared over the Internet. With the introduction of interactive browsers, beginning with Mosaic (from the National Center for Supercomputing Applications in Illinois), Internet users could access multimedia documents with hypertext links to supporting documents at the click of a mouse. Unlike users of earlier Internet tools (Gopher, FTP, etc.), Web users did not need to know command prompt language, but could instead rely on intuitive, mouse-driven interactions on a graphical display screen. This advantage alone surely opened the Web to many new users. The first Web server and browser were made available to the public in 1991. By mid-1995, the Web rivaled e-mail and surpassed other Internet services (FTP, Gopher, news servers) in total bytes transmitted. The Web had established itself as a major new information medium, attracting the attention of businesses and venture capitalists, in addition to the more traditional constituencies in the education and research communities.

By the late 1990s, the enormous interest in the Web resulted in an immense flow of money and talent into the development of new infrastructure and tools to increase the amount and variety of content available on the Web. To oversee expansion of the Web on a rational basis, Berners-Lee founded the W3 Consortium (W3C). The W3C was charged with setting standards for HTML, HTTP, and document

addressing. Over the years, the W3C has produced several versions of HTML (HTML 2, HTML 4, etc.) through an open RFC process. Despite the W3C's oversight, competition among software companies in the browser market—particularly Microsoft Corp. and Netscape Communications—has produced numerous nonstandard extensions of HTML. The extensions or preliminary standards (i.e., standards still under discussion by the W3C) promoted by a company would often function properly only in that company's latest browsers. Even the confusion of nonstandard extensions and other new technologies did not slow the Web's development into a major medium for the dissemination of a truly diverse variety of information—text, images, moving pictures, and sound—doubtless because of the low cost to consumers (consistent with the notion of the Web as a "commodity" service), and the accessibility facilitated by the Internet itself.

III. WEB PAGE ORGANIZATION

Content on the Web is organized as sets of HTML documents stored on Web-enabled servers. Each HTML document is called a web page and consists of a simple text (ASCII) file organized according to rules within HTML. Each Web page is identified by a unique address (URL) that specifies the page's location on the Web.

A. External Organization

A Web page's URL normally consists of three parts:

- The **protocol,** "http://," tells the browser that the file is a Web page that follows the HTTP protocol.
- A **domain name** specifies the name of the server from which the file is served. Domain names typically include a two- or three-letter top-level domain (TLD) that specifies the branch of the Internet from which the Web page originates.
- The **path** consists of the HTML document's name and directory (folder) names.

A Web server requires a unique name, or address, to identify it on the Web. The physical entity (i.e. the server) requires an Internet protocol (IP) address, but it may have more than one address. The server uses the fixed address to identify itself on the Web. An IP address takes the form of a series of four numbers, each from 0–255, and each separated from the others by a period. For example, 1.255.10.240 could be an IP address. While a user could point a browser to the numerical IP address, names are preferable, since words are easier to remember than numerical sequences. To accommodate the natural preference for words, domain names are used to represent IP addresses. A domain name identifies one or more IP addresses. For example, the domain name "microsoft.com" represents about a dozen IP addresses. In the URL http://www.amazon.com/, the domain name is amazon.com. Every domain has a suffix that identifies its TLD. TLDs are separated from the rest of the domain by periods (the "dot" in the now-familiar term, "dot-com"). A list of TLDs, valid as of this writing (March 2001), appears in Table I.

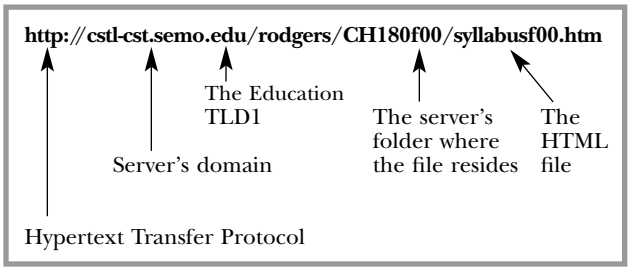
In addition, two-letter TLDs are used to indicate a particular country in which a Web site is hosted. For example, http://www.ibm.com.jp is the Web site for IBM in Japan. The country TLD for the United States, "us," is required in only a few U.S. addresses: nonfederal government sites use the .us domain rather than .gov (e.g., http://www.state.mo.us) and educational institutions not covered by .edu use the "us" TLD (e.g., http://www.lausd.k12.ca.us). TLDs are expected to proliferate in response to the growing pressures on the pool of available Web site names. Occasionally, a server will be configured to use a network port other

Table I Top-Level Domains

TLD	Meaning
gov	U.S. government agencies
edu	Educational institutions (colleges and universities)
org	Nonprofit organizations
mil	U.S. military
com	Commercial business
net	Network organizations
int	International treaty organizations
aero	Air transport industry
biz	Businesses
coop	Cooperatives
info	Unrestricted use
museum	Museum
name	Individuals
pro	Professionals (accountants, attorneys, physicians)

than port 80, the default HTTP port. In such cases, the network port number will follow the TLD, separated from the TLD by a colon (:).

The document name is usually accompanied by the "htm" or "html" file extension, separated from the file name by a period. A full URL might look like:



Relative (partial) URLs are used to refer to documents in the same directory to which the browser currently points. For example, a browser that was pointed to http://cstl-cst.semo.edu/rodgers/CH180f00/syllabusf00.htm could also reach the file "askeysf00.htm" in the same directory (http://cstl-cst.semo.edu/rodgers/CH180f00/) if the relative URL "askeysf00.htm" is used in the HTML. Relative URLs are favored by developers because they allow an entire Web site—a thematically organized collection of Web pages—to be produced locally and then moved to a server without losing the links between the pages within the Web site.

The file name "index.htm" is an important exception to the naming system described previously. Working on the assumption that directories (folders) are established to contain pages that are related on some rational basis (such as content), browsers are instructed by the server to look for a file named "index.htm" by default if the URL does not specify a file name [or, occasionally, index.html, homepage.htm(l), or default.htm(l) are used as specified by the server]. For example, http://cstl-cst.semo.edu/rodgers/CH180f00/ points the browser to the same page that the URL http://cstl-cst.semo.edu/rodgers/CH180f00/index.htm addresses. This exception makes the "index.htm" file the natural candidate for a Web site's home page. The absence of a file name also helps to focus user attention on the domain name, the part of the URL that often bespeaks an organization's identity. For example, who would doubt that http://www.microsoft.com/ is the home of Microsoft Corp.?

B. Internal Organization

The Web page itself is a simple text (ASCII) document, in which elements are denoted by HTML tags that surround content:

```
<tag>content</tag>
```

As was the case with its patriarch SGML, the HTML tags control the document's formatting. The tags also control hyperlinking. The tags do not constitute a programming language and do not function as a word processor. In general, spaces, indentation, carriage returns, and linefeeds are ignored. For example,

```
<p>Here is some text.</p>
  <p>Here is some additional
text.</p>
```

and

```
<p>Here is some text.</p><p>Here is
some additional text.</p>
```

are both rendered as

Here is some text.
Here is some additional text.

Every HTML document begins with the <HTML> tag and ends with the </HTML> tag. In between those tags are the document's **head** and **body,** also denoted by tags:

```
<html>
<head>
  <title>New Page 1</title>
</head>
<body>
  <p>Here is some text.</p>
  <p>Here is some additional
text.</p>
</body>
</html>
```

The page's content appears as tagged elements in the document's body section, between the <body> and </body> tags. The element tags may include **attributes** that specify the element's appearance, as the HTML in Fig. 2 shows. Tag ① specifies the image file ("collectm.jpg") used as the document's **background.** Tag ② establishes the borderless **image** "header.gif" at the top of the document. The Arial **font** face, size 4, is set for the text "Fall, 2000" by tag ③, and tag ④ sets

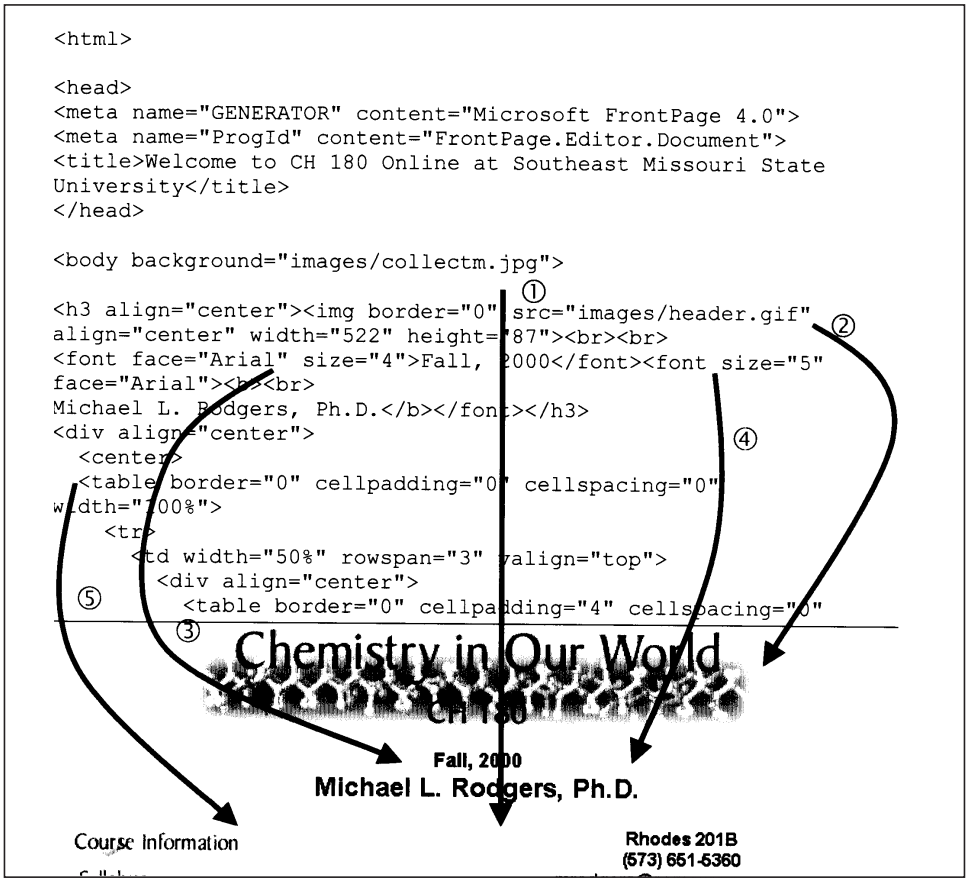


Figure 2 Tags and attributes guide page rendering by the browser.

the font size (size 5) and face (Arial) for the text "Michael L. Rodgers, Ph.D." Tag ⑤ places a borderless **table** in the document. As tag ⑤'s arrow suggests, the table does not explicitly appear on the page as rendered by the browser. Nevertheless, the table is a very useful addition to the page, as it controls positioning of the two columns of text and images appearing below "Michael L. Rodgers, Ph.D." on the rendered page.

When a Web page is ready for dissemination, it must be saved to a server. This task can be accomplished using FTP, the basic Internet file-moving service. Since the client's browser is responsible for rendering the document for display, HTML documents are truly portable: they can be viewed on any computer for which a browser is available. Indeed, while the usual delivery mode is an Internet download from a server, users can view HTML documents saved on a disk accessible to the user's local computer without involving the Internet at all.

Because Web pages consist only of text, they can be written in any text editor, such as Microsoft Notepad. However, developers who use a text editor have the burden of memorizing numerous tags and attributes. In addition, text editors generally cannot provide a view of the page as rendered by a browser, requiring the developer to work "blind" or to frequently save the HTML document and then switch between the text editor and a browser. To address these shortcomings, numerous WYSIWYG (**w**hat **y**ou **s**ee **i**s **w**hat **y**ou **g**et) HTML editors began to appear on the market soon after the launch of the Web in 1991. Products such as Claris HomePage, Adobe PageMill, Microsoft FrontPage, and Macromedia DreamWeaver follow a "word processor" model of authoring, in which formatting and attributes are selected through pull-down menus and clickable icons in a graphical interface. The editors convert the selections into tags and put the selection of graphics and sounds under the developer's control. The more sophisticated editors could establish hyperlinks and could even save HTML files directly to the server. The ability to save

files directly to the server has proven to be quite helpful to developers who produce and maintain entire Web sites.

IV. WEB PAGE ENRICHMENTS

Early Web pages usually consisted of text and some hyperlinks. Through the browser, the user would select pages to read, with little opportunity to interact with the people who provided the page's content. To more fully realize the Web's potential as a platform for social interactions, including commerce, numerous implicit and explicit approaches to two-way communication beyond the document request/response dialog between browser and server have been implemented. At about the same time, the dramatic rise in the Internet's ability to transmit information, coupled with improved software and hardware, led to the expansion of Web page content to include multimedia elements: images, video, and sounds. The juxtaposition of multimedia and interactivity powered the development of the Web as a virtual marketplace, workplace, and highly personalized entertainment source.

A. The Web Page as a Data Collection Tool

Much of the power of HTML lies in its ability to hyperlink documents residing anywhere on the Web. The anchor tags, `<a>` and `</a>`, define both the source and the destination of a hyperlink. The "href" attribute within the `<a>` tag defines the hyperlink as a **source,** meaning that the tag specifies the URL of a document that the browser will load when the user selects the hyperlink. Indeed, HTML's robust URL handling allows linking to any type of Internet document, including e-mail. With e-mail, asynchronous communication (i.e., communication that does not require the communicants to be simultaneously engaged in the act) between a Web page's viewer and other persons associated with the page is easily established. For example, at the bottom of Fig. 3 is an e-mail link in the form of a GIF image ("contact.gif"). The e-mail is linked via **mailto.** In this example, the link establishes communications between students and the instructor in an on-line course.

Asynchronous communication can also be supported by **bulletin board** software. A bulletin board is a server-side application, linked to a Web page, which organizes messages (often called "posts") both chronologically and by content. For synchronous communication, Web pages can link to electronic conferencing or **chat** software such as Internet Relay Chat (IRC). Electronic conferencing has become increasingly popular in the business world as a means to bring project teams, clients, and consultants into working environments without incurring the costs of travel. Like conferencing, chat can simultaneously bring to the Web page many individuals from any part of the Internet. More recently, Internet voice transmission technologies have also gained attention.

Along with e-mail, conferencing, and chat, Web pages permit data collection from users by way of **forms.** Managing and processing forms generally require a high degree of sophistication on the part of the Web page developer, but forms repay the investment by empowering the Web page owner to collect an immense range of valuable information quickly and at low cost. Forms processing is the foundation of electronic business ("e-business"), Web-based personal preference polling, and scientific data gathering, to name a few. Forms come under the general category of dynamic HTML (DHTML), the part of HTML that produces pages that respond to, or are customized by, information input by the user.

HTML forms require the developer to build a form

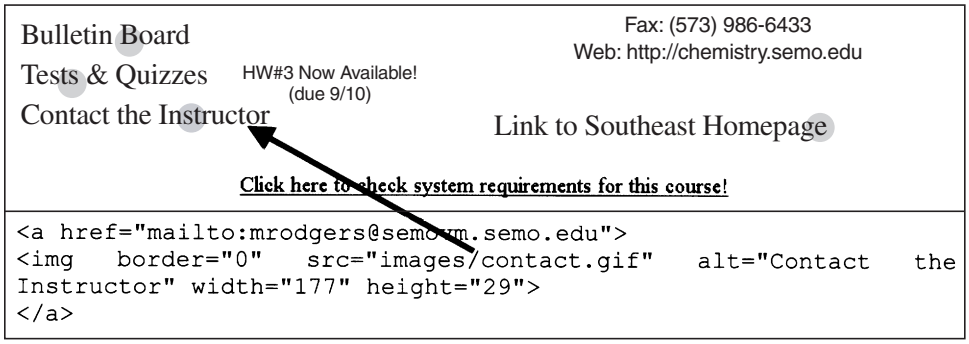
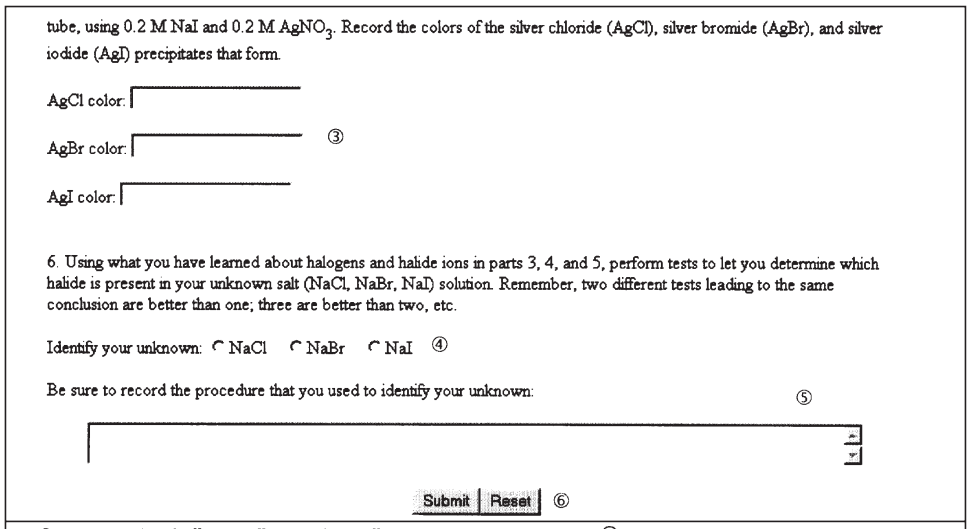


Figure 3 Establishing an e-mail link to an image on a page.

structure into the Web page that will display the form. In addition, the developer must implement a suitable script (often a Common Gateway Interface [CGI] program) or other server-side Internet application software to process the form. The script or application software assumes responsibility for saving the incoming data to a file or database on the server for future use by the Web page's owner(s). The scripts or software can pass data to many types of files. A **flat file** is often used when the amount and diversity of data is expected to be small, and **relational databases** are typically used for large applications. A flat file is a simple table in which all data follows one record description; a relational database allows for multiple tables, each of which can have a different record description. Flat files can be simple ASCII files, but relational databases are created and maintained with the help of **database management software** (DBMS). SQL, first marketed by Oracle Corporation in 1979, is a prominent example of DBMS.

The Web page's form structure is illustrated in Fig. 4. In the document's body is the `<form>` tag ①, which creates the form. The form ends at the `</form>` tag ②. The `<form>` tag's "method" attribute defines how




```
<form method="POST" action="--WEBBOT-SELF--">①
                                . . .
Record the colors of the silver chloride (AgCl), silver bromide
(AgBr), and silver iodide (AgI) precipitates that form.<br><br>
  AgCl color: <input type="text" name="agclcolor" size="20"
tabindex="32"></p>          ③
  <p align="left">AgBr color: <input type="text" name="agbrcolor"
size="20" tabindex="33"></p>
  <p align="left">AgI color: <input type="text" name="agicolor"
size="20" tabindex="34"></p><p align="left"><br>
6. Using what you have learned about halogens and halide ions in parts
3, 4, and 5, perform tests to let you determine which halide is
present in your unknown salt (NaCl, NaBr, NaI) solution. Remember, two
different tests leading to the same conclusion are better than one;
three are better than two, etc. </p><p align="left">
   Identify your unknown:
<input type="radio" name="unknownID" value="NaCl" tabindex="35">NaCl④
<input type="radio" name="unknownID" value="NaBr" tabindex="36">NaBr
<input type="radio" name="unknownID" value="NaI" tabindex="37">NaI</p>
  <p align="left">
  Be sure to record the procedure that you used to identify your
unknown: </p>       ⑤
  <p align="center"><textarea rows="2" name="unknownproc" cols="84"
tabindex="38"></textarea><br><br>
  <input type="submit" value="Submit" name="B1" tabindex="39"><input
type="reset" value="Reset" name="B2" tabindex="40"></p>⑥
</form>②
```

Figure 4 A form for collecting data from students in a Web-based course.

the browser sends the form's data to the server, and the "action" attribute specifies the script or application software that will process the form's data. One or more form elements are placed within the form; the form elements set the type of data to be entered by the user, and they organize the data into **name/value pairs.** A name/value pair is the identifier, or variable name of a data item, followed by the variable's value, as specified by the user submitting the form. The `<input>` tag specifies form elements. The "type" attribute determines the element's type. In Fig. 4, three elements are present:

- Tag ③—"text" elements, which accept one line of text, are typically used for data such as names and addresses.
- Tag ④—"radio" buttons follow a multiple choice format.
- Tag ⑤—A "textarea" element is a scrollable box intended to accept large amounts of text.

Not present in Fig. 4, but available to the Web developer, are several additional form elements: password boxes, check boxes, drop-down menus, hidden elements, and active images. Two special elements, "submit" and "reset," allow the user to send the form when finished or to start over if the user must erase form entries (tag ⑥).

The data collection technologies described above all work at the direction of the Web page user. However, more covert methods are also used in Web pages. For example, **hit counters** record the number of times users access a Web page. The hit counter performs its function without any direction on the part of the user. More generally, Web pages can send data files called **cookies** to the user's browser, requesting that the cookie be returned to the server after the browser has given the cookie information about the user's visit to the Web page. Cookies remain on the user's computer and can be accessed by the server at a particular Web site each time the user browses to that Web site. The process takes place outside of the user's control, and can be exploited to seamlessly provide customized content based on the user's demonstrated viewing preferences. However, cookies have come under some criticism as a threat to the user's privacy.

B. Multimedia in the Web Page

Multimedia is often employed to engage the user's attention more fully than is possible with text alone. Of course, some multimedia simply cannot be replaced by text. Unfortunately, multimedia elements are often very large files that require long download times on slower Internet connections. Moreover, developers considering the use of a multimedia element in a Web page are by no means assured that users will have browsers and plug-ins capable of properly rendering the element. Nevertheless, multimedia is now found on millions of Web pages.

From the many multimedia file types, proposed standards, and plug-ins developed or adapted for Web page use, some de facto standards have emerged. **Images** are typically made available in either the GIF or JPEG (JPG) format. GIFs and JPEGs use compression to achieve small file sizes, yet they are compressed in ways different enough to affect their performance in a Web page: GIFs generally work best for line drawings and other images with straight, distinct lines; JPEGs are preferred for photographs. Because multiple images (frames) can be encoded in GIF files, GIF files can show brief, relatively low-quality animations. Many of the moving banner ads in commercial Web pages are animated GIFs. **Sound** files are usually found in one of three formats: the au format is the most ubiquitous, but it has the disadvantage of being the lowest quality, allowing only 8-bit sampling. Macintosh computers established the AIFF format as a standard, and Windows-based systems likewise established the WAV format standard. Sound files tend to be quite large. However, the recent MP3 file standard manages to reduce file sizes to less than 10% of the original through a process that removes redundant sounds and parts of the sound signal outside the human audible range without compromising sound quality. The small size of MP3 files allows music to be readily distributed from Web pages; this practice has produced a spirited debate over the extent and enforcement of copyright protection law in the Internet environment. **Video** files, like sound files, tend to be so large that wise developers employ them only when other information media fail to adequately convey meaning. With the help of plug-in software, such as Apple's Quicktime, video files with .mov, .qt, or .mpg (or .mpeg) file extensions can be downloaded and viewed. Windows machines possess software to view AVI files. Recently, a new technology called **streaming media** has won widespread support as a way to deploy both video and audio over the Web in near real time, even over relatively slow Internet connections. Streaming technology greatly accelerates file downloading by passing the data directly to the computer's CPU for immediate processing, eliminating the need to first store the data in the computer's memory. While streaming technology does not allow the user to store a downloaded

file on the local computer, the technology has led to the delivery of radio programming, film clips, rock concerts, political speeches, college courses, and other media products to worldwide markets.

V. WEB PAGE DESIGN

Many early Web pages were painful testaments to the mistaken belief that when designing a page, "anything that could be done, should be done." As the Web began to mature in the late 1990s, developers paid greater attention to Web page design, recognizing that good design practices were of paramount importance to any page's success. Financial forces were so strong that many corporate Web site managers hired graphic designers and instructional design experts to complement their organizations' technical expertise in the hopes of launching a "killer" Web site.

The most important factor affecting Web page design is the nature of HTML itself. As a markup language, HTML provides only sketchy guidance to the browser about a document's formatting: details are left to each browser itself to decide. This approach facilitates HTML's platform independence, but it leaves the developer uncertain as to how the page will appear to users. Formatting options such as pixel-level positioning, which are easily executed in word processing and desktop publishing software, are quite difficult to realize in the HTML environment. HTML does offer a few formatting features, and good developers are generally those who know how to use HTML's limited formatting to produce pages that are **functional** in any browser, while retaining a sense of style and elegance.

True to its SGML roots, HTML identifies higher level elements such as paragraphs (`<p>`), ordered and unordered lists (`<ol>` and `<ul>`), and tables (`<table>`). Because they can be specified with no borders, tables have found great favor among developers as a means of aligning content horizontally and vertically on the page. Nesting of tables allows for a considerable degree of control, even in complex, information-rich pages. Further format control within a table is afforded by the **cellpadding** and **cellspacing** attributes, which control the number of pixels bordering data within a cell and the number of pixels between cells, respectively. The entire table's dimensions can be controlled by the **height** and **width** attributes in the `<table>` tag; the same attributes within a cell tag (`<th>` or `<td>`) control the dimensions of an individual cell. The **colspan** and **rowspan** attributes within a cell tag allow several cells to be condensed into a single cell over several rows or columns. Other attributes, notably **align** (horizontal alignment), **valign** (vertical alignment), and **nowrap** (control linebreaks within cells), offer additional control. Figure 5 provides a simple example of the use of a table to format a mathematical equation.

Good designers realize that the **human–computer interface** (the collection of ways that human beings interact with the computer's hardware and software) can affect the value and effectiveness of a Web page. Some basic considerations include eyeflow, horizontal and vertical scrolling, color, contrast, text size and style, and navigation aids for both the page and the site. Eyeflow recognizes the natural tendency of the eye to flow from the upper left to the lower right when reading. Scrolling has proved to be annoying to many Web page users; wise developers seek ways to reduce or avoid scrolling (especially horizontal scrolling). Color and contrast affect readability, and color can be used to assert the Web site sponsor's image. There is some *de facto* text color standardization on the Web: for example, blue text, especially if it is underlined, is commonly used to represent hyperlinks. As in printed text, text size is selected to signify titles and section headings. Text style can follow good practice for printed material: **sans serif fonts** for titles and headings and **serif fonts** for body text. However, many developers are finding that users interact with Web page text in a manner more consistent with heading than body text. Therefore, many of the best Web pages use sans serif text throughout, with special preference shown for the **Verdana font style.** Although browsers offer buttons to move forward and back through the queue of pages viewed by the user during a session, good Web pages will have a rational internal navigation system. Typically, the system takes the form of a horizontal or vertical list, or "bar," of hyperlinks that, at a minimum, will take the user to the Web site's home page and to the next higher and next lower levels of the site. Figure 6 provides examples of each list type. Note that the links can be either text or images.

Good page design also depends on numerous technical considerations: the client and server hardware CPU speeds, the connection mode and speed (T1, ISDN, 56k modem, etc.), and software (browsers, plugins) will all affect page content. As an extreme example, consider Arachne, a very good, fully graphical, DOS-based Web browser that has been successfully installed on many current and obsolete computers running DOS, including a 1980s vintage 640K IBM personal computer (PC). While the browser functioned properly on the PC, several **hours** were required to

```
<p><b><i><font color="#000000">Unit
conversions.</font></i></b><font color="#008000"> The diameter of a red
blood
cell is 0.00030 inch. Convert the measurement into the metric system
unit that you think scientists would MOST LIKELY use to clearly
describe the red blood cell's diameter.</font></p>
<p><b>A scientist would know that the human mind is limited in the
range of numbers it
can comprehend; a unit should be selected which would keep the value in
comprehensible range. See that <font face = "Symbol">m</font>m
works:</b></p>
  <table border="0" cellpadding="0" width="497">
    <tr>
      <td align="center" width="110"><b>0.00030 inch  </b></td>
      <td align="center" width="8"><b>x</b></td>
      <td align="center" width="76"><b><u>  2.54 cm
 </u></b></td>
      <td align="center" width="10"><b>x</b></td>
      <td align="center" width="73"><b> <u>   0.01 m
 </u></b></td>
      <td align="center" width="10"><b>x</b></td>
      <td align="center"
width="96"><u><b>     1 <font face =
"Symbol">m</font>m          
 </b></u></td>
      <td align="center" width="80">
        <p align="left"><b>= 7.6 <font face =
"Symbol">m</font></b></td>
    </tr>
    <tr>
      <td align="center" width="110"></td>
      <td align="center" width="8"></td>
      <td align="center" width="76"><b>1 inch  </b></td>
      <td align="center" width="10"></td>
      <td align="center" width="73"><b>1 cm  </b></td>
      <td align="center" width="10"></td>
      <td align="center" width="96"><b>0.000001 m</b></td>
      <td align="center" width="80"></td>
    </tr>
  </table>
```

Unit conversions. The diameter of a red blood cell is 0.00030 inch. Convert the measurement into the metric system unit that you think scientists would MOST LIKELY use to clearly describe the red blood cell's diameter.

A scientist would know that the human mind is limited in the range of numbers it can comprehend; a unit should be selected which would keep the value in comprehensible range. See that μm works:

$$0.00030 \text{ inch} \times \frac{2.54 \text{ cm}}{1 \text{ inch}} \times \frac{0.01 \text{ m}}{1 \text{ cm}} \times \frac{1\ \mu\text{m}}{0.000001 \text{ m}} = 7.6\ \mu\text{m}$$

Figure 5 Using a table to format a mathematical equation.

download and render simple GIF images. A developer hoping to reach users of such systems will certainly opt for text-only Web page designs. Occasionally, developers will produce parallel Web sites, where pages on each site are optimized for a single anticipated software environment. Parallel Internet Explorer/Netscape Navigator-specific sites have been tried, as have parallel sites for PCs and handheld wireless Web devices. The disadvantage to the multiple-site approach is obvious: each site requires its own labor.

Web pages are visited when they offer something of value to the user in a convenient, aesthetically pleasing package. Appropriate formatting, respect for the human–computer interface, and awareness of technical limitations all contribute to successful page design. In the end, however, a Web page is a delivery system for content, and good content can go a long way toward making good page design a reality. Many critics assert that developers fail if they merely reproduce on the Web page the same content that is available in other media. These critics argue that the Web page ought to contain **rich content:** content that takes advantage of the Web's unique information-sharing, interactive qualities. For example, a company seeking a

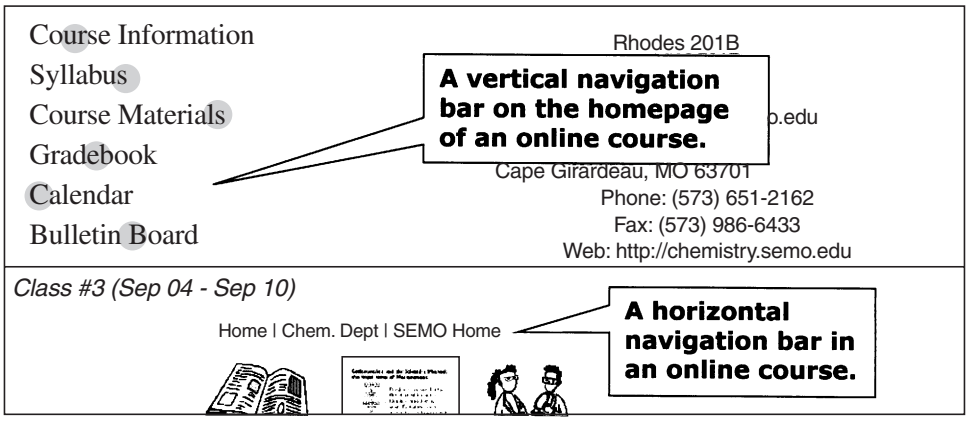


Figure 6 Vertical and horizontal navigation bars.

new logo might use the Web to combine the descriptive qualities of printed media (text and images) with the interactive qualities of a focus group: the Web page might display several proposed logos and have visitors vote on the best one, perhaps including an opportunity to comment on the pros and cons of each design.

Finally, good Web page design takes into account issues of accessibility by persons with disabilities. Compliance with the Americans With Disabilities Act (ADA) has been construed by some as necessary if the Web site's sponsor receives U.S. federal funding. Simple actions on the developer's part, such as using the alternate text attribute "alt" in the image tag, can greatly improve accessibility. The "alt" tag permits vision-impaired users who use text-only browsers to have a description of images appearing on the site. For example, the two rightmost images, ① and ②, in Fig. 7 are hyperlinks that have descriptive alternate text.

The W3C recommends the use of **cascading style sheets** (CSS) as a way to enhance accessibility by both persons with disabilities and international users. CSS, a component of DHTML, works by separating a page's content from its formatting information. The formatting information is kept in a simple text **.css** file, to which the Web page is linked by a URL in the document's `<head>` area. HTML tags within the `<body>` are still used to define the document's structure, but the tag attributes controlling the elements' appearance are removed to the style sheet. By this system,


```
<table border="0" cellpadding="3" cellspacing="0">
    <tr>
      <td>
        <p align="center"><img border="0" src="images/bd00163_.gif"
width="121" height="88"></p>
      </td>
      <td>
          <a href="ppt/lesson03.htm"><img border="2"                ①
src="images/lesson03.jpg" align="top" width="120" height="89" alt="PowerPoint
for Class #3">
          </a>
      </td>
      <td>
        <p align="center"><a href="labs/lab03.htm"><img border="0"   ②
src="images/bd10686_.gif" width="106" height="88" alt="Specific Heat and
Density Lab"></a></p>
      </td>
    </tr>
```

Figure 7 Using the alternate text attribute in image tags.

browsers for the visually impaired can ignore irrelevant formatting such as font styles, colors, etc. by ignoring the entire style sheet. Further information on ADA and accessibility guidelines can be found at the federal guidelines Web site (http://www.access-board.gov/news/508-final.htm), the federal IT accessibility initiative Web site (http://www.section508.gov/), and the World Wide Web consortium's Web accessibility initiative (http://www.w3.org/WAI/).

VI. MANAGING WEB PAGES AND WEB SITES

A successful Web site will entice a visitor to stay and look around. The longer the stay, the better the chance of conveying the information that the site was designed to deliver. This may lead to a product purchase, response to a request, or any of a large number of other intended uses. While the visitor should see a site that reflects simplicity, ease of use, content, and good aesthetics, a successful Web site is, in fact, a complex meld of such things as delivery type and style, security features, backup considerations, strategies in name use, timeliness, and database access. A successful Web site is backed by an appropriate mix of human expertise, software, and hardware.

A. Developing a Web Site

Competent personnel are perhaps the key requirement for successful Web site development and management. Managing a Web site is not a job for the novice, but it is not necessarily an elite job best filled by expensive outside consultants. Many companies do indeed hire help from the outside. This can range from contracting a large nation-wide comprehensive Web development, management and hosting firm to local high-school kids making extra money on the side. An alternate strategy is to establish positions within the organization to hire personnel devoted to Web site development and management. It may be possible to utilize personnel already in the organization to take on these tasks with varying requirements for retraining. Part of the basis for the decision depends on the scope of the organization's operations. A simple Web site designed to give an organization a presence on the Web with information about the organization and contact information may require only minimal investment in expertise, while a complex Web site requiring database access, timeliness, security, coordination with multiple locations, etc. will likely require a team of trained experts.

It is important to note that Web site management is really broken down into two main components: hardware and software. Hardware requires more technical expertise and includes such aspects as the server itself (i.e., the machine or CPU); the connection of the machine to the Internet; connection of any local workstations or desktop PCs to the machine; and any necessary peripherals such as printers, modems, tape drives, and UPS (uninterruptible power supply). Hardware needs also include memory, backup hardware, storage, and level of redundancy (RAID hard drives, for example). Software needs include choice of platform (e.g., Linux, Windows NT) and server software (e.g., Apache, IIS), as well as database servers, Web authoring software, Web/e-mail/FTP server software, network protocols, and other software designed for specific tasks. In addition, interactive Web sites may require CGI programming skills (e.g., Perl), JavaScript, Java, SQL, and/or other interpreted (script) or compiled languages. Also falling under the software category are security and reliability features such as encryption keys, firewalls, virus protection, and backup software.

From the software side, Web site management requires expertise in the use of security features. The level needed depends on the site content and complexity. Expertise is also needed in page development and maintenance and may include proficiency in Web authoring tools, programming, databases, etc. Again, depending on the Web site, these jobs may be filled by one person or may require a team with expertise in particular aspects of Web site maintenance and security. Decisions must be made on Web authoring software (if any), access to the server, server accounts, the level of interactivity required on the site, and many other aspects of Web site development and maintenance. A Web site manager may need to manage a team to control all aspects of the Web site. Web servers are often associated with e-mail programs and even database servers. As such, expertise required for Web site maintenance likely overlaps with the need for e-mail management, database development, etc., and personnel may, in fact, carry out a number of these duties.

One of the first decisions to be made is the name chosen to represent an organization's Web site. This is an often-overlooked aspect of development strategy. There is a definite strategy involved in choosing a name. The TLD used depends partly on the type of organization and will not necessarily be something that can be chosen. A domain name should identify the organization and be as logical as possible. Part of the reason for this is that a user attempting to find the organization

on the Web will usually start by typing the name directly into a browser's address window or into a Web search engine. The closer the domain name is to what they are looking for, the easier it will be for them to find the organization. At best, the organization name forms the domain name: Barnes & Noble uses www.barnesandnoble.com and can be easily found by prospective shoppers. Other possibilities include using acronyms; partial words; nicknames; associated names; and memorable, cute, or even tricky names. Note the infamous example of the Web site using the www.whitehouse Web address with a different TLD than .gov. Other examples of recognizable URLs include www.twa.com, www.toysrus.com, www.usps.com (note the lack of .gov for the privatized U.S. Postal Service), www.nfl.com, www.navy.mil, www.yellowpages.net, www.c-span.org, and www.harvard.edu.

Once a name has been chosen, it must be registered. This was formerly an easy process with only one organization having authority to register names. Now many companies or "registrars" have the authority to do so (for .com., .org., and .edu). InterNIC (www.internic.net) has been authorized by the U.S. Department of Commerce to provide information on Internet domain name registration services and links to domain name searchers and registrars. Domain names are registered for 1 year, with annual renewals available. Cost is generally $50–$100 for the first year and less for renewals. While the organization's type will restrict the TLD that can be used, more then one TLD can be registered to reduce the likelihood that a cybersquatter or copycat site becomes a problem (e.g., whitehouse.com vs whitehouse.gov). An organization can use all the TLDs it has registered and create automatic forwarding pages that will deliver visitors to the actual site. It may also be worth registering common spelling mistakes or homophones. This will help those visitors who might misspell the organization's name, as well as reduce the possibility that a cybersquatter diverts potential visitors. Other strategies include registering names in which a hyphen might be used, such as cyber-cafe.com and cybercafe.com, and registering available alternative names that potential competitors might find desirable.

B. Managing a Web Site

While it would be nice to think that a Web site can function without human intervention after the development and implementation of the site, this is probably not going to be a reality. In fact, successful sites need constant attention. By its nature, the Web is dynamic and has an inherent timeliness to it, allowing almost instant posting of up-to-date information. Many Web pages contain a "last-updated" stamp to inform visitors when the content was last updated. Many visitors will look for such a stamp in order to determine the timeliness of the page and perhaps imply its validity. As mentioned earlier, the type of site offered will to a degree determine the frequency and quantity of maintenance or changes. Certainly, a site containing information such as product prices or quantities must be updated frequently, if not instantaneously. Dates for sales, current promotions, messages, etc. might also require frequent updating. Even the site's overall look will likely change periodically as a new look is used to freshen up and increase interest in the site. Most successful large sites undergo periodic facelifts for aesthetics, content, and usability. Remember, visitors will come to a site and take a few seconds to look around. Users must be kept interested and involved so that they will explore further. A static, stale site will rapidly lose potential users. Keep the site current, interactive, and attention grabbing.

There are behind-the-scenes concerns in Web site management also. If the intent is to sell products on the site, the organization will need to consider what payment options to offer. Printable mail-in invoices are a possibility, but most customers prefer instant online payment options. This means establishing a mechanism for securely submitting credit card numbers. Data encryption protocols, such as the secure socket layer (SSL) protocol developed by Netscape for transmitting private documents, are commonly used for such purposes. SSL uses a private key to encrypt data transferred over the SSL connection. By convention, Web pages requiring an SSL connection have URLs beginning with https: instead of http:. SSL is considered an industry standard and is supported by both Netscape and Internet Explorer. Security entails not only secure transmission of information between site and customer, but also protection against unwanted access to the site. Web sites/servers must be guarded against viruses, denial of service attacks, and other types of hacker attacks. Virus protection software packages are widely available for reasonable costs from well-established and trusted companies. Firewalls are software-implemented gates that monitor and restrict access to the site. They are one of the main lines of defense against hackers. Firewalls should allow information to flow between an organization's server and legitimate visitors to the site, while at the same time minimizing access by hackers or others with unscrupulous intentions. Numerous firewall programs are also available at a reasonable cost.

Finally, good Web management practice includes maximization of reliability, availability, and data integrity. No computer is infallible. Computers can, and will, "crash." Having a server crash, taking all of your data, Web pages, etc. with it, can be a devastating experience. The wisely managed Web site exploits numerous features designed to minimize such losses in a crash. One mechanism for minimizing loss is the use of RAID or redundant array of independent (or inexpensive) disks; a category of disk drive that employs two or more drives in combination for performance. There are different levels of RAID, ranked for performance and reliability on a scale from 0 to 5. All utilize the same strategy of spreading or mirroring data across multiple disks to minimize the possibility that data will be irretrievably lost in the case of hard drive failure. A second approach is to regularly backup data to an external storage device. Typically, this is carried out by periodically writing data to a tape drive. This may mean writing the entire contents of a hard drive, or drives, to tape or merely writing data files or other dynamic content. Backups may be done monthly, weekly, daily, or even more frequently. Daily backups are common. Numerous tapes are used so that a Web site may be restored with data that may be many days or even weeks old. It might be necessary to go back to a point before a virus was introduced or data was corrupted. For instance, 14 tapes might be rotated with daily tape backups, ensuring that there are copies of data as it existed each day over the previous 2 weeks.

A third approach to ensure reliability is the use of a power supply backup. The most common method for doing this utilizes a UPS, a power supply that includes a battery to maintain power in the event of a power outage. Typically, a UPS keeps a computer running for several minutes after a power outage, enabling data to be saved and allowing the computer to be shut down properly. Many UPS devices offer a software component that enables automated backup and shut down procedures in case there is a power failure when the Web management team is not present. There are two basic types of UPS systems: standby power systems (SPSs), which monitor the power line and switch to battery power as soon as a problem is detected, and on-line UPS systems, which constantly provide power from built-in inverters, even when external power is present. In a SPS, the switch to battery can require several milliseconds, during which time the computer is not receiving any power. An on-line UPS avoids these momentary power lapses by always supplying power. These three approaches are not exclusive of each other, and, in fact, the highest level of reliability can be attained if all three approaches are used concurrently.

C. Successful Web Sites

Several well-known examples of successful Web sites include Amazon.com (http://www.amazon.com) and eBay (http://www.ebay.com). What has made these sites commonplace names in our culture? Both have addressed most of the considerations discussed above. A look at the Amazon.com site reveals an easy-to-use site. While originally an online bookstore, the site now offers numerous other products, such as video, software, toys, furniture, cars, art, electronics, music, and even an on-line auction. All of these products are grouped by category under a logically arranged directory found on the home page. Another feature that is almost a must for a successful large site is a search engine. Amazon.com utilizes a search engine found prominently displayed on the first page. A visitor can search the whole site or search just by product type (i.e., all books, all music, etc.). The site is kept up to date. A current bestseller list is displayed on the front page. After pulling up a specific book title, links to reviews of a particular book, customer reviews, other books bought by customers buying the featured book, excerpts from the book, and other related information are available. All of this is timely information and must be updated frequently. The Web site utilizes large databases that are queried with each book pulled up and which must yield all of the information described. Prices, promotions, etc. are pulled up with the book information.

Another feature found on successful retail sites is the **shopping basket** concept. The Amazon Web site has a link to the personal shopping basket of each customer. This requires that visitors register with the site and log in each time they come back to the site (the use of cookies likely minimizes the amount of information that is stored on the Amazon servers). Other important features on this site are the customer service links that allow review of return policies, privacy policies, order and account status, and an extensive help page. A security assurance page explains plainly to customers the security behind the exchange of credit card numbers and the like across the Web. Amazon utilizes SSL server software, the industry standard and among the best software utilities available today for secure commerce transactions. All personal information, including credit card number, name, and address, is encrypted so as to be unreadable as the information travels over the Internet.

Ebay.com uses many of the same strategies as Amazon.com. Users register and log in to the site, extensive help is available on the site, assurances of security in financial transactions are present, and a detailed

search engine allows visitors to quickly locate what they are looking for. eBay differs from Amazon in that the products available have been submitted to eBay from other customers. eBay serves as an auction site at which people can buy and sell products from each other, as well as from other companies that place items on the site. Since both sites are in retail, they minimize the presence of on-line advertising. Some sites do use on-line advertisers to reduce the cost of presenting materials on their Web site, much as commercial television makes programs "free" by allowing advertisements to be interspersed in the programming. While not apparent to visitors, both sites utilize complex systems to ensure site security and to assure reliability, availability, and data integrity. This likely includes firewalls, complex backup systems, and redundancies. It is important to remember some of the most important complaints of on-line shoppers include pages take too long to load, sites are confusing to navigate, products are out of stock, products arrive late or are never delivered, the site is "down" or crashes, orders cannot be tracked, and customer service is not easily accessible or waits are too long.

A successful Web site requires the combination of competent human resources and reliable hardware and software. There is an expense required in putting together these three components. However, the cost of purchasing good hardware and software and hiring a reliable Web site management team can be easily outweighed by the gains the organization will make in publicity, access, and delivery of services or products. A Web site can become the first and even the main line of presence an organization maintains with clients and customers. An organization may be judged by many based solely on its Web site. The proper development and management of an effective Web site cannot be minimized. The Web is a dynamic place with Web sites coming and going daily. Make sure an organization's site is available and maintains a presence to give that organization a good face on the Web.

VII. PROMISING TRENDS

Predicting the future is an inexact, if not dangerous, undertaking. This is particularly true in the field of computer and Internet technology where technological innovations and advancements are rushing forward at breakneck speed. At the turn of the new Millennium, Moore's law still holds true—hardware and infrastructure continue to improve at astonishing rates and new applications and uses of technology still catch us by surprise. Nonetheless, there are some current technologies just beginning to be realized and some extrapolation of their potential impact is appropriate.

The Web has been based on HTML since its inception. As the Web has exploded from a handful of users in 1995 to over 400 million users worldwide in 2000, the complexity and quantity of content delivered over the Web has likewise increased dramatically. HTML has undergone some changes over the years and has become more "dynamic." Features such as CSS make it possible to precisely position any item (text, graphic, etc.) on a page. Still, the increased interactivity, the need for international exposure, the need to include dynamic content such as changing information from a database, and the general increases in complexity and nature of material presented on the Web require additional methods of categorizing, retrieving, and presenting Web content.

In 1996, the W3C released its proposal for the eXtensible Markup Language (XML). XML uses tags to describe a piece of data, rather than merely its presentation. For instance, a price might be described with HTML as `<p><b>$179.99</b></p>`, while XML could simply describe it as `<price>179.99 </price>`. The HTML tag instructs the browser to display the price in boldface and in its own paragraph. XML instructs the browser to display the price according to parameters set for displaying a "price." XML can be integrated with HTML, and, in fact, XHTML is a reformulation of HTML 4.0 that allows XML and HTML to be used together on a page. Extensions are being added to XML to allow for more specialized functions such as Xlinks, which describe links to particular sections of a document, and XSL, which is a mechanism for describing style sheets (the XML version of CSS). A thorough discussion of XML would be lengthy and complex. The important feature to note is that XML allows for much easier retrieval and delivery of dynamic content. This translates into easier delivery of database-derived information such as prices, flight schedules, etc. and also allows for easier delivery of Web content in multiple languages. XML will also be used in other Internet-connected devices such as those described below. XML can be thought of as the next revolutionary step in information exchange, mirroring the explosion that HTML initiated in 1990.

With increased content and complexity of Web-based information, the speed at which this information is accessed becomes a major factor in the evolution of the Web. For the most part, the bottleneck in information flow is in the technology at which the

user accesses a Web site. That is to say that the speed of the modem connecting a computer to the Internet determines how fast one can download information from a Web site. One of the biggest increases in download time comes from the use of multimedia components on Web sites. Content in the form of streaming audio and video greatly enhances the quantity and complexity of information available. It also greatly increases the time needed to download information. Access through a standard telephone-connected modem is limited to speeds not much higher than 50 kbps. Numerous new technologies are emerging that will allow much faster transfer of data into our homes and onto our PCs.

Cable modems, ISDN, ADSL, DSL, satellite, and wireless technologies all promise to increase speed and access capabilities to the Internet. Cable modems rely on the same wiring (or at least the same type) as that used in cable television. Coaxial cable physically connects a house to a network of cabling which is connected to a cable service provider, which then in turn connects the user to the Internet. Cable data transfer rates can be as high as 10 mbps, though they are often more in the range of 5–8 mbps. They are therefore much faster than the standard telephone line modems that run at a maximum of 56 kbps, but functionally rarely go above 50 kbps. A downside to cable is that the user is sharing his/her connection with all his/her neighbors on the same backbone, which can mean reduced speeds and increased privacy and security concerns. An integrated services digital network (ISDN) utilizes a digital telephone line and functions much as a standard telephone connection. It is almost three times as fast as a standard telephone line modem. Digital subscriber lines (DSL) rely on a recent technology that allows more data to be sent over standard copper telephone lines. DSL is asymmetric, meaning that data move faster in one direction, typically data being downloaded or coming into your house or office is faster. DSL speeds range from 384 kbps to 1.5 mbps uploading and 1.5–10 mbps downloading. Satellite technology is also currently asymmetrical, with data being downloaded from a satellite at roughly 400 kbps, but uploaded through a standard 56 kbps telephone modem. Uploading through personal satellite broadcast dishes is not far off in the future and promises to increase uploading speed and therefore the overall attractiveness of satellite connections. One of the fastest growing areas is in wireless technology, which allows data to be beamed or broadcast much as cellular telephone calls are and at a rate in the range of 500 kbps to 10 mbps. As this technology continues to improve, wireless connections to the Internet and even between computers and other devices is expected to become much more popular.

The last decade of the 20th century saw portable computing become part of the mainstream. The distance between portable, or lap top, computers and desktop PCs narrowed immensely, and the practicality of using a lap top computer as an alternative to sitting at a fixed machine was realized. This trend continued as smaller and smaller versions of a PC showed up. Personal digital appliances (PDAs) have now become commonplace. It is possible to walk around and read e-mail, look at your calendar, play a game of solitaire, and much more from a palm-sized computer. Most televisions shipped now have Internet capabilities (often referred to as WebTVs). This trend will continue as other appliances start to utilize Internet connectivity. Refrigerators will be able to monitor supplies of groceries and e-mail a shopping list to a local grocery store, which may then ship it to the house. Microwaves and ovens will be able to download recipes from the Internet and have them available on a small computer screen. Automobiles will have wireless connections to the Internet, allowing for downloading of information about restaurants, hotels, maps, and much more. The two-way connection will also allow for uploading of information about car trouble to a dealer or automobile club, with possible solutions downloaded back to the car. In combination with a Global Positioning Satellite (GPS) receiver, a tow truck could even find exactly where a car broke down.

While it is impossible to accurately predict the future, many of the above-mentioned trends are becoming a reality even now. It is likely that many of these will become more commonplace in the future. Even more likely, some as yet unimagined application technology will catch us all by surprise!

Because a Web site may shut down, move to a new address, or radically modify its focus, a long list of Web sites valid at the time of this writing (March, 2001) might leave the reader frustrated. A few major (and, we think, relatively stable) Web sites worth visiting for more information are listed in the Bibliography, along with some books that address the history, philosophy, organization, and future of Web pages and the World Wide Web.

SEE ALSO THE FOLLOWING ARTICLES

Copyright Laws • Electronic Mail • End-User Computing Tools • Firewalls • Hyper-Media Databases • Integrated Ser-

vices Digital Network (Broadband and Narrowband ISDN) • Internet, Overview • Intranets • Java • Javascript • Multimedia • Search Engines • Word Processing • XML (Extensible Markup Language)

BIBLIOGRAPHY

Berners-Lee, T., and Fischetti, M. (1999). *Weaving the web,* 1st ed. New York: Harper Collins.

Corporate sites. Major corporations dedicated to providing hardware and software of use to Web site developers and users. http://www.microsoft.com/, http://www.adobe.com/, and http://www.macromedia.com/

Hafner, K., and Lyon, M. (1998). *Where wizards stay up late: The origins of the internet.* New York: Touchstone by Simon & Schuster.

InterNIC. Internet domain name registration services. http://www.internic.net.

Internet Society. A very well-done history of the Internet. http://www.isoc.org/internet/history/.

LanTimes. An example of a site supported by a professional publication dedicated to the computer industry. http://www.lantimes.com/.

Musciano, C., and Kennedy, B. (1996). *HTML the definitive guide.* Sebastopol, CA: O'Reilly & Associates.

World Wide Web Consortium. Sets and maintains Web standards. http://www.w3.org.

Internet, Overview

Raymond Greenlaw *Armstrong Atlantic State University*
Ellen M. Hepp *University of New Hampshire*

GLOSSARY

client/server model A scheme in which clients (or programs on a network) make requests to a small number of servers. The servers are other programs on a network that respond to client's requests.
cookie A piece of information that is saved by a user's web browser to a file on the user's disk. The information can be retrieved by a web server that the web browser accesses.
domain name system (DNS) A distributed naming scheme in which unique names are assigned to computers on the Internet.
e-mail Electronic mail or messages that are sent electronically over a network.
hypertext markup language (HTML) The programming language in which most web pages are written.
hypertext transfer protocol (HTTP) The rules that determine how hypertext is exchanged over the Internet.
Internet A global system of networked computers, including their users and data.
Internet address Numerical computer names that uniquely identify each computer on the Internet. It is also called an IP address and is represented by four decimal numbers concatenated by periods.
packet switching A method used to transmit data over the Internet that involves breaking a message into packets before sending.
router A special purpose computer that directs packets along a network.
server A computer that responds to requests for services.
transmission control protocol/Internet protocol (TCP/IP) The protocol suite that determines how computers connect, send, and receive information on the Internet. It is also called the "language of the Internet."
World Wide Web (WWW) An application that utilizes the Internet to send hypertext/multimedia documents.
World Wide Web consortium (W3C) A group that facilitates communication between groups and individuals about matters relating to the World Wide Web.

I. INTRODUCTION

We begin with a definition of the Internet as formulated by the *Federal Networking Council*.

The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term Internet. Internet refers to the global information system that:

1. Is logically linked together by a globally unique address space based on the Internet protocol (IP) or its subsequent extensions/follow-ons
2. Is able to support communications using the transmission control protocol/Internet protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols
3. Provides, uses or makes accessible, either publicly or privately, high-level services layered on the communications and related infrastructure described herein

Simplifying, we can condense this definition to: the Internet is a global system of networked computers together with their users and data. The system is global in the sense that people from all over the world can connect to it. The users of the Internet have developed their own culture and as such they are a defining factor of the Internet. Without the possibility of accessing data or personal information, very few would be excited about connecting to the Internet. The notions of being able to quickly and easily access information and communicate led to the vision of the Internet.

The Internet has also been referred to as the "Information Superhighway." Thirty years ago, information exchange and communication took place via the "back roads"—regular postal mail, a telephone call, a personal meeting, and so on. Today they take place over the Internet nearly instantaneously. The history section of this article describes the evolution of the vision of the Internet into today's Information Superhighway.

A. Information Superhighway

We expand on the superhighway metaphor. With cars there are levels of expertise—learning to drive is easy and knowing how to operate a vehicle is all you really need to know about cars in order to use them to get to where you are going. Driving is like learning to surf the Internet. In the course of driving you learn about highways and shortcuts, and so it is with the Internet. With practice, you will learn where and how to find information on the Internet.

In driving you can go a step further and learn how an engine works and how to do routine maintenance and repairs such as oil changes and tune-ups. With the Internet the equivalent is to learn how Web pages are created or how search engines find information.

A deeper level of involvement with cars is to learn how to do complex repairs, design them, and build them. Not many people demonstrate this level of interest in cars. On the Internet a similar level of interest involves writing software, either building *applets* in a language such as Java or developing more general purpose tools for others to use in navigating the Internet. Again, only a limited number of people aspire to this level of involvement.

Today the Information Superhighway is in place but the mysteries surrounding it for many people are where to go and how to travel. Like traveling a highway in a foreign country, unable to read the road signs, navigating the Information Superhighway can be frustrating and time-consuming without the right tools.

As far as "how to travel" the Information Superhighway, consider that there are many routes and many forms of transportation that we can take to get to where we want to go. We can follow sidewalks, roads, and freeways and we can take a bicycle, a bus, a car, or a pair of in-line skates. Similarly, there are many ways to use the Internet to send and retrieve information. These include (but are not limited to) e-mail, *file transfer, remote log-in,* and the World Wide Web. It is also very likely that new methods of traveling the Information Superhighway will be conceived and developed in the near future, and existing methods will be improved.

II. INTERNET HISTORY

We touch on the history of the Internet in this section with a focus on the 1990s. The references provided at the end of the article contain a great deal of additional information.

A. Internet in the 1960s

Essential to the early Internet concept was *packet switching,* where data to be transmitted is divided into small packets of information and labeled to identify the sender and recipient. These packets of information are sent over a network and then reassembled at their destination. If any packet does not arrive or is not intact, the original sender is requested to resend the packet. Prior to packet switching, the less efficient *circuit switching* method of data transmission was used. In the early 1960s several papers on packet switching theory were written that laid the groundwork for computer networking as it exists today. In 1969 Bolt, Beranek, and Newman, Inc. (BBN) designed a network called the Advanced Research Projects Agency Network (ARPANET) for the United States Department of Defense. The military had created ARPA to enable researchers to share "super-computing" power. It was rumored that the military developed the ARPANET in response to the threat of a nuclear attack destroying the country's communication system. Initially, only four nodes (or *hosts*) comprised the ARPANET. They were located at the University of California at Los Angeles, the University of California at Santa Barbara, the University of Utah, and the Stanford Research Institute. The ARPANET later became known as the Internet.

B. Internet in the 1970s

In this decade the ARPANET was used primarily by the military, some of the larger companies like IBM, and universities (for e-mail). People were not yet connected in their homes and very few people were online at work. The use of local area networks (LANs) became more prevalent during the 1970s. Also during the 1970s the idea of an *open architecture* was promoted—that networks making up the ARPANET could be of any design. In later years this concept had a tremendous impact on the growth of the ARPANET. In 1972 the ARPANET went international with nodes in Europe at the University College in London, England, and the Royal Radar Establishment in Norway. Ray Tomlinson, who worked at BBN, invented e-mail in 1972. In 1979 User Network (USENET) was started by using UUCP (UNIX to UNIX CoPy) to connect Duke University and the University of North Carolina at Chapel Hill. Newsgroups emerged from this early development.

C. Internet in the 1980s

Transmission control protocol/Internet protocol (TCP/IP), a set of rules governing how networks making up the ARPANET communicate, was established. In this decade, the term "Internet" was being used to describe the ARPANET for the first time. Security became a concern during the 1980s as viruses appeared and electronic break-ins occurred. The 1980s saw the Internet grow beyond being predominately research oriented to including business applications and supporting a wide range of users. As the Internet became larger and larger, the domain name system (DNS) was developed to allow the network to expand more easily by assigning names to host computers in a distributed fashion.

D. Internet in the 1990s

During the 1990s many commercial organizations started getting on-line. This stimulated the growth of the Internet like never before. Uniform resource locaters (URLs) appeared in television advertisements and for the first time young children went on-line in significant numbers. Graphical browsing tools were developed and the programming language HyperText Markup Language (HTML) allowed users all over the world to publish on the World Wide Web. Millions of people went on-line to work, shop, bank, and be entertained. The Internet played a much more significant role in society as many nontechnical users from all walks of life got involved with computers. Computer literacy and Internet courses sprang up all over the country.

- **Gopher, 1991—**Gopher was developed at the University of Minnesota, whose sports team's mascot is the the Golden Gopher. Gopher allowed you to "go for" or fetch files on the Internet using a menu-based system. Many gophers sprang up all over the country and all types of information could be located on gopher servers. Gopher is still available today and accessible through Web browsers, but its popularity has faded.
- **World Wide Web, 1991—**The World Wide Web was created by Tim Berners-Lee at CERN as a simple way to publish information and make it available on the Internet.
- **World Wide Web Available to Public, 1992—**The interesting nature of the Web caused it to spread and it became available to the public in 1992. Those who first used the system were immediately impressed.
- **Mosaic, 1993—**Mosaic, a graphical browser for the Web, was released by Marc Andreessen and several other graduate students at the University of Illinois, where one of the National Science Foundation's super-computing centers was located. Mosaic was first released under X Windows and graphical UNIX. It seems as though each person who used the system loved it and told five friends. Mosaic's use spread rapidly.
- **Netscape Communications, 1994—**Netscape Communications, formed by Marc Andreessen and Jim Clark, released Netscape Navigator. This Web browser captured the imagination of everyone who used it. The number of users of this piece of software grew at a phenomenal rate. Netscape made (and still makes) its money largely through displaying advertisements on its Web pages.
- **Yahoo!, 1994—**Stanford graduate students David Filo and Jerry Yang developed their now world famous Internet search engine and directory called Yahoo!.
- **Java, 1995—**The Internet programming environment, Java, was released by Sun Microsystems. This language, originally called Oak, allowed programmers to develop Web pages that were more interactive.
- **Microsoft and the Internet, 1995—**Microsoft became involved with the Internet by developing the Microsoft Internet Explorer (MIE) browser

and other Internet applications. The "browser wars" began as the various browsers competed for market share. Netscape Navigator and MIE are the last two big players standing.
- **Over 55 Million Nodes, 1999**—The number of Internet hosts grew to 55,000,000.
- **"I Love You Virus," 2000**—The "I Love You Virus" spread from the Philippines and infected millions of computers worldwide.
- **Google Indexes Over 1.3 Billion Web Pages, 2001**—The search engine Google claims a huge index of more than 1.3 billion Web pages.
- **Wireless Devices, 2001**—Many people now browse the Web and send e-mail over the Internet using wireless computer technology.

E. Internet Growth

The Internet is still growing at a rate of 100% each year; the number of computers linked to the Internet are now over 100 million and increasing. What permitted the technology to be adaptable enough to handle this amazing growth? Over the past three decades, the Internet has proven to be extremely flexible. Even though there were no personal computers, workstations, or LANs in existence when the early Internet came into being, the emerging Internet was versatile enough to be able to incorporate the new technologies. Obviously, the early researchers working on Internet technology had no inkling that what they were designing would accommodate the World Wide Web and other applications for millions and millions of users. The built-in flexibility has been a key to continuing growth.

In retrospect there were a number of key reasons for the Internet's great success:

1. Decisions were made on a technical rather than political basis, especially without the need for international political groups.
2. The Internet did not require a centralized structure that would not scale up; it was and is a distributed operation.
3. Due to the homogeneity of language and outlook, a sharp focus on the Internet itself could be maintained.
4. The Internet allowed people to do things of inherent interest, for example, send and receive e-mail.
5. The software involved was free or very low cost.

The Internet will continue to grow, change, and support new applications. Now, however, instead of only researchers initiating change and implementing new ideas, we also see entrepreneurs and politicians getting involved. Both small and large businesses will play an important role in setting new trends.

III. INTUITION FOR HOW THE INTERNET WORKS

We presented an intuitive idea of how the Internet works using the common analogy with the highway system in the United States. Additional technical details can be found in the references. To begin a discussion of the Internet, it is helpful to identify some of the benefits of networks.

A. Network Benefits

1. Provide convenience: Computers on a network can have their files backed-up over the network.
2. *Allow sharing:* Networked computers can share resources such as disks and printers.
3. *Facilitate communications:* Sending and receiving e-mail, transferring files, and videoconferencing are examples of how networks promote communication.
4. *Generate savings:* Networked computers can provide more computing power for less money. That is, several small computers connected on a network can provide as much as or more computing power than a single, large computer and still cost significantly less. Since resources can be shared, not everyone needs their own peripherals. This can result in a substantial cost savings.
5. *Provide reliability:* If one part of a network is down, it may still be possible to perform some useful work using a different network path.
6. *Simplify scalability:* It is relatively easy to add more computers to an existing network.

B. Interconnected Networks and Communication

The Internet is essentially a network of networks and its success depends upon "cooperation." Since no one person, organization, or government is responsible for the Internet, cooperation among the networks and computers that compose the Internet is paramount. The way that the computers and networks cooperate is by using a common set of protocols to communicate. The protocol that determines how computers connect, send, and receive information on

the Internet is called TCP/IP. In fact, TCP/IP consists of about 100 different protocols and new ones are regularly developed and added. Drawing on the freeway analogy, you can think of these protocols as forming the "rules of the road" ranging from who has the right of way to how you register your vehicle and get a driver's license.

TCP/IP has been described repeatedly as the "language of the Internet." In the same way that a common language allows people of diverse backgrounds to communicate, TCP/IP allows many different kinds of computers, from personal computers to mainframes, to exchange information. The two main protocols in the TCP/IP suite are TCP and IP. TCP allows for communication between the various computers on the Internet, while IP specifies how data is routed from computer to computer.

To illustrate how TCP/IP works, consider either sending an e-mail message to a friend or making a request for a Web page. In either case the information is "formatted" according to its specific application protocol—simple mail transfer protocol (SMTP) is used to format your e-mail message and hypertext transfer protocol (HTTP) is used to format your Web page request. Assuming that your computer has TCP/IP software installed, the information to be sent is split into *IP packets,* called *packets* for short, and transmitted over the Internet. There are several advantages to using packets to send the information:

- *Error recovery:* If a packet gets corrupted, only that (small) packet needs to be resent and not the entire message
- *Load distribution:* If one area of the network is congested, packets can be rerouted to less busy areas
- *Flexibility:* If the network experiences a failure or disruption in one locale, packets can be rerouted

TCP converts a message into a stream of packets. In addition to the message pieces, each packet of data also contains information about the computer that sent it, the computer it is being sent to, a *sequence number* indicating what part of the overall message the packet is, and some error checking information to ensure that the packet was not somehow corrupted while in transit. IP routes the packets. The packets are reassembled after being received at the destination computer. A message is sent from the destination computer to the sending computer to resend any missing or corrupted packets. Using this method, called packet switching, it is not necessary to send the packets of data in sequential order or even over the same network route. The sequence numbers can be used to reconstruct the original message if packets arrive out of order. After receiving the message, the destination computer will respond to the message in some appropriate way, for example, by delivering an e-mail message to the recipient's mailbox or by servicing a request for a Web page.

In the example just given suppose that a packet *did* get corrupted. The destination computer must send a message requesting that the packet be resent. What happens if the resent message gets corrupted or lost? We will not get into such details in this article other than to say that the protocols are complex enough to recover from all types of worst case error situations. Protocol design is a complicated process.

C. Physical Components

In addition to the various software protocols that make up the Internet, transmitting information over the Internet involves physical components as well. These components include servers, routers, and the networks themselves. *Servers* are computers that answer requests for services. A *router* is a special purpose computer that directs packets of data along the network. Routers can detect if part of the network is down or congested, and then reroute traffic. Networks provide the physical means to transport packets of information and the following mediums are employed:

1. Copper wires transmit messages as electrical impulses.
2. Fiber-optic cables use light waves to transmit messages.
3. Radio waves, microwaves, infrared light, and visible light carry messages through air.

D. Network Connections

Someone connecting to the Internet from home generally uses a *modem* and a regular telephone line (copper wire) to connect to an Internet service provider (ISP). A second modem completes the connection at the ISP's end and the slower of the two modem speeds determines the maximum connection speed—usually 56 kilobits per second (kbps). (A kilobit is 1000 bits.) It is worth noting that some important parts of the network are still audio-based, for example, the part of the phone system in your home or office. This means that modems need to convert from *analog* to digital and back again.

A business, organization, or school network typically uses *network interface cards* instead of modems to join the personal computers that are part of their LAN. Such an entity often has a higher speed connection, usually greater than 56 kbps, to connect to their ISP. These connections are usually leased from the telephone company.

Another option is integrated services digital network (ISDN) that provides connections with speeds up to five times faster than a traditional modem. By using regular telephone lines and replacing modems with special adaptors, ISDN offers a slightly more expensive but much faster alternative to a modem. Another possibility is a cable television connection.

If the connection to the ISP is a "driveway" in our highway analogy, then the backbones of the Internet are the "freeways." These freeways are run by network service providers (NSPs). Local ISPs connect to NSP networks. The connection between the ISPs and NSPs is usually over leased-lines from local telephone companies. These phone lines typically transmit data at a rate of 1.54 megabits per second (Mbps) or higher. A megabit is 1,000,000 bits (or 1000 kilobits).

The NSPs lease or buy lines consisting of copper wire, fiber-optic cable, or satellite communications from telecommunications companies. The NSP networks, like a freeway, can operate at very high speeds and transmit a lot of data over long distances.

E. Client/Server Model

The client/server model provides many of the network benefits described in this section. A client makes a request to the server and the server responds by satisfying the client's request. In the client/server model new clients and servers can be added incrementally as more users come on-line and the demand for services increases. That is to say, the client/server model is easily extensible and therefore scales well. Many clients can share the resources provided by a single server. This eliminates the need for each client to have their own "copy" of the resources. Each Internet service has its own associated set of clients and servers. For example, in the Web domain browsers are clients and Web servers are the servers.

F. IP Addresses

Each computer and router on the Internet must have a name so that it can be uniquely identified. After all, how can an e-mail message be delivered if there is any ambiguity in its destination? The DNS provides a computer-naming framework that is convenient for people since it uses symbolic names. However, computers themselves are better suited to manipulating numbers as opposed to symbolic names. *IP addresses* are numerical computer names that uniquely identify each computer on the Internet. Such numeric names are easy for computers to work with.

An IP address consists of 32 bits or equivalently four bytes. (A byte consists of 8 bits.) The largest possible number that 8 bits can represent is 255. Thus, one byte can represent a number from 0 (00000000) to 255 (11111111). Oversimplifying, each IP address consists of a *network* component and a *host* component. Each of the 4 bytes of an IP address can represent a natural number from 0–255. It is common to express IP addresses as four natural numbers separated by dots, for example, 132.88.5.111.

IP addresses play a vital role in the routing of packets over the Internet. A source and destination IP address is included in each packet to be routed. You can think of the addresses as providing directions on where the packet should go. How are IP addresses assigned? There must be a central authority to manage IP addresses because otherwise conflicts might arise. The Network Information Center (NIC) is in charge of assigning IP addresses but there is a plan to expand to multiple registries.

We need to know the relationship between IP addresses and domain names. IP addresses are 32-bit numbers, whereas domain names are easy to remember symbolic strings. When you type in an e-mail address, you always enter a symbolic string such as

```
killface@savannahcats.org
```

How does a computer make use of this since it needs to work with IP addresses? There is a program called a *resolver* that takes care of the translation. That is, the program converts a symbolic name into its corresponding IP address. You can think of a resolver as acting like telephone directory assistance. On some systems there is a program that allows you to enter an IP address and obtain its symbolic name back, and vice versa. For example, on UNIX-based systems the program nslookup does this. It is important to realize that for each symbolic name there is a unique IP address.

A permanently assigned IP address, one that is given to a computer or router connected to the Internet, is called a *static* IP address. If you connect to the Internet through an ISP, then typically each time you connect you will be assigned a different IP address, called a *dynamic* IP address, from the ISP's pool of IP addresses.

G. Internet Protocol Version 6 (IPv6)

IPv6 is the latest version of the IP routing protocols and was originally called IPng (IP next generation). The new protocol is necessary to accommodate the greater demands being placed on the Internet. The major changes in the new version are

1. *Expanded number of addresses:* This will be done by increasing the IP address size from 32 to 128 bits.
2. *Simplified IP headers:* The goal is to reduce the number of header fields needed in an IP packet.
3. *Added security features:* Greater support for privacy and security is planned.

IPv6 is being designed with many efficiency considerations in mind. You can find additional details about IPv6 on the Web.

IV. INTERNET CONGESTION

The Internet functions amazingly well for such an incredibly heavily used system. However, the number of users and their demands continue to grow almost without bound. In this section you will learn what researchers are trying to do to reduce congestion on the Internet. First though, we consider some of the limiting factors from a user's point of view.

Once you get a network connection and have a modem with a speed of 56 kbps or higher, the limiting factor of how quickly you can view Web pages often becomes the speed of your computer when rendering the Web pages. The computer speed in turn depends on a complex balance of CPU speed, bus speed, amount of memory, disk speed, and so on. In addition, there is a hierarchy of link speeds, with major backbones aiming for OC12 (622.1 Mbps) (OC stands for *optical carrier*), more and more regional and site-direct T3 (44.7 Mbps) links, and T1 (1.5 Mbps) links to institutions. The slowest link speed involved in a connection determines the level of performance you receive. The rate of growth of the Internet is so rapid that it is hard for technological improvements to keep up.

A. World Wide Wait Problem

Have you heard the phrase *World Wide Wait?* It has been around for a while, especially in overseas places where connections are notoriously slow. For example, in Spain Internet users say Espera en la Red Mundial. The literal translation of this phrase is "Wait in the World Network" and it refers to the ever increasing delays one experiences when trying to access information on the Internet.

With the advent of the World Wide Web and the development of graphical browsers came a surge of interest in the Internet. This increase in the number of Internet users coupled with the accompanying requests for Web pages containing elaborate in-line images, sounds, and video clips has degraded the speed of the Internet to the point where the Information Superhighway sometimes appears to have a traffic jam. Although new technologies are being employed to remedy the situation, the problem persists. We discuss what is currently being done to reduce the problem and what we can expect in the future.

B. Technical Solutions

Researchers working in conjunction with the World Wide Web Consortium (W3C) have been addressing the issue of network congestion. One of their stated goals has been to "save the Internet from the Web" by developing new technologies to help relieve the slowdown that has resulted from retrieving and displaying Web pages.

Some of the solutions offered involve HTTP and improving the way in which HTTP and TCP/IP interact. In particular, researchers have focused on the following issues:

- Introducing new techniques to expedite Web page requests
- Improving the process of connecting to a Web server
- Refining how a URL is resolved by introducing "persistent connections" that make it more efficient to retrieve pages from the same Web server.

W3C researchers have also put forth some suggestions for web page design. Since Web page content (that is, the graphics, sound, text, and/or video that makes up the page) dictates download time, the recommendation was made to avoid unnecessary graphics. Cascading Style Sheets (CSS), a Web page design tool, have the potential to improve download time as well. Lastly, the researchers recommend using the graphics format Portable Network Graphics (PNG) over Graphics Interchange Format (GIF) to represent images on Web pages since PNG images are generally smaller than GIF images plus they render more quickly.

Another active step being taken is to reduce the size of router tables by rearranging how blocks of addresses are identified. Routers face a formidable task when data flows at 44.7 Mbps or faster. They have to examine each packet to see where it is going, then look up that destination and send the packet on its way. They cannot fall behind because they would never catch up. Packets that cannot be resolved in the *threshold time* are thrown away and have to be retransmitted.

By developing technological improvements, researchers and the W3C are attempting to ease the congestion on the Information Superhighway created by the World Wide Web. It is believed that by using these new technologies Internet traffic due to the Web can be reduced by up to 50%. However, network traffic is increasing in many dimensions. At the same time that new users are being added, experienced users are requesting more information and spending more time on the Web. Thus, although suggestions like those just presented are worthwhile if many users treat them seriously, in reality they are expected to provide very little noticeable net relief.

C. Issues and Predictions

Recent technological advances have been significant but have certainly not yet managed to alleviate the World Wide Wait problem. Some users are attempting to deal with the slowdown by using the Internet during less busy periods. A number of businesses are bypassing the Internet completely and are creating isolated *intranets* for their companies. An intranet is a private network; such networks can have their own internal Web. Still others, frustrated by the delays and failures involved in transmitting and receiving information, may have severely limited their usage of the Internet. There has even been talk of creating the "Internet II" to be used exclusively by academia and researchers with no commercial traffic permitted, and to operate at much higher speeds.

Part of the reason the Internet has become so popular, especially in the United States, is that it is essentially free. Most ISPs offer a flat rate plan that allows for hours and hours of very inexpensive Web surfing and time on-line. From an economist's point of view, it may be time to start charging more for the use of the Internet in order to limit demand.

Originally, the government financed the Internet. Now users pay ISPs to connect them; ISPs pay NSPs; and NSPs in turn pay the long-haul carrier. The payments are not based on use but on how much capacity is provided. This contrasts with the telephone billing system currently in effect where there is a unit charge that is based on how far the call travels and/or how long it lasts. If the telephone system were run like the Internet (and were just as cheap), we can only imagine that demand for service would greatly increase and along with it, delays and more busy signals. However, since the telephone system *is* so expensive people have figured out ways of having "phone" conversations over the Internet. This technology has kept the telephone industry competitive.

In order to reflect the true costs involved in using the Internet, some economists have suggested prioritizing information and then charging more for high priority transmissions. Another idea is to charge for transmissions that occur when Internet traffic is heavy. How to meter usage once a billing method is selected is a topic currently being investigated. It seems clear that unless technology offers a viable solution to the World Wide Wait problem, our days of surfing the Web for free could be numbered.

V. INTERNET CULTURE

An entire culture has sprung up around the Internet. What began as an exclusive club for researchers and academics has now become open to the masses. Some of the original club members are still not that happy about this transition. The Internet has emerged from being a medium in which to exchange research ideas to one that includes advertising, commerce, and forums for exchanging ideas on a near infinite set of subjects. We describe the philosophy of this unique culture and some important issues to bear in mind while browsing.

A. Critical Evaluation of Information

Since the Internet is not regulated for content (and there are no immediate plans to regulate it), anything and everything can be found on the Web. The editorial control of traditional print media is missing. Being able to discern between inaccurate and accurate information is a necessary part of the Internet culture. Experienced Internet users know that not everything published on the Web is sound. They are cautious about believing anything they read and such users are always looking at information with a critical eye.

To find valuable information, you need to be able to sift through Web pages and separate the useless from the useful, and the invalid from the valid. This is especially true regarding medical information ob-

tained on the Internet or on any other topic that is going to affect someone's life in a significant way. Suspect information can surface on the Internet in any form: in an e-mail message, mailing list, newsgroup, or Web page. For example, in the 1996 United States presidential campaign bogus presidential Web pages were published.

What are some forms of information that are reliable? There are Web presentations that contain refereed and reviewed information, and some others that are monitored for accuracy. For example, there are electronic journals on the Web whose content is refereed. Such presentations are usually very reliable. Commercial presentations try to provide accurate and up-to-date information as well, since their reputation depends on presenting valid information. Some authors, by the very nature of who they are, can be trusted to display only accurate information. For example, if Lance Armstrong, three-time winner of the Tour de France, had a Web presentation about cycling, one would assume that the information about the Tour de France displayed on his Web pages would be accurate.

What are some methods you can use to critically evaluate information? Here are some questions to consider:

1. Are there errors in the content? For example, if you know the game of baseball requires nine players on a team and the document you are reading says it only requires six, you should be wary.
2. Who wrote the information? That is, was the person who wrote the material knowledgeable and careful? Was the author aware of what others have written? Does the author have a reputation to uphold? Can the author be trusted? What is the author's professional background? The answers to these questions will go a long way toward providing you a gauge of the accuracy of the information.
3. Is the writing quality high? A document riddled with typos is more likely to have inaccurate content than a carefully crafted exposition.
4. Is the document up to date? Try to determine whether the information contained in the document is current. When was it last updated? Does the document deal with up-to-date information?

B. Freedom of Expression

The lack of regulation that permits the proliferation of suspect information on the Internet actually facilitates the interchange of ideas. Anyone with an Internet connection can express their views globally and receive feedback. This allows a very small community to find itself. For example, there are people with very rare medical problems, largely ignored by the normal medical sources, who can offer each other support and exchange experience on coping, if not a cure. Many believe that this freedom of expression is the best feature as well as the most defining feature of the Internet. Some Web authors display a small blue ribbon graphic at the bottom of their pages in support of on-line "freedom of speech."

Related to the idea of personal expression is another aspect of Internet culture—not everyone agrees that everything and anything should be publishable. For example, some people find the availability of obscene or offensive material on the Internet unacceptable. Other people worry that small children may stumble across something they should not see or read. These concerns are definitely valid and several camps are busy discussing them. Regardless of the outcome of these discussions, most people would agree that the Internet provides the following benefits:

- The sharing of research ideas and information
- More educational opportunities for many children and adults
- The ability to communicate more readily with others all over the world
- The convenience of performing many functions such as banking and shopping on-line
- Opportunities for entertainment
- Rapid and global dissemination of important information
- Worldwide discussion forums to promote solutions to global problems

It is clear that to prohibit material on a specific topic from being published on-line diminishes someone else's freedom of expression. On the other hand, parents may feel their freedoms are violated if they cannot have the Internet and its benefits for themselves and their children without risking unintended exposure of their children to obscene material. Thus, the issue of *censorship* is a volatile one that has both supporters and opponents in the Internet community. Obviously, the issues involved in a worldwide attempt at censorship are very complex. No doubt this debate will continue uninterrupted for quite some time.

C. Communication Mechanisms

Another aspect of Internet culture is created by the communication channels that the Internet has

spawned. People from all over the world are able to exchange ideas via e-mail, Internet relay chat (IRC), instant messaging, mailing lists, newsgroups, and so on. Since there are no facial expressions, voice inflections, and/or body language available to convey or interpret a communication while typing, users must be careful to avoid ambiguity or misunderstanding by either spelling things out completely or by using *emoticons* to show emotion. Table I depicts a number of common emoticons. *Videoconferencing* is a way to include the otherwise missing audio and video.

To save time when typing messages, users sometimes employ a (friendly) shorthand for commonly used phrases.

- AFAIK—As far as I know
- IMHO—In my humble opinion (usually not so humble, of course)
- ROTFL—Rolling on the floor laughing
- YMMV—Your mileage may vary

On occasion people may misinterpret or take offense at an e-mail message or chat room conversation. Being rude or overly confrontational is called *flaming* and such messages are called *flames*. Some people find it easy to be rude when they do not have to confront a person face to face, while other users are just plain insensitive to how their messages come across. Flaming is not considered appropriate on the Internet and violates the commonly accepted guidelines of *netiquette*.

D. Advertising

Prior to 1995 there was very little advertising on the Internet. Along with the Web has come an avalanche of advertisements. Advertisements generate huge amounts of income for companies such as Netscape Communications, AltaVista, and Yahoo!. The Web pages of companies like these get many millions of hits per *day* so an advertisement placed on one of their Web pages has a very large audience. Naturally, marketing experts take advantage of this potential consumer base.

Most of the advertisements shown on Web pages are clickable images. On a standard screen, they are typically about one inch high and three to four inches wide. Many of the most popular Web pages have *revolving advertisements*. That is, each time you revisit the page or while you are visiting the page, you get a different ad. The advertisements usually consist of a carefully designed graphic with a catchy phrase superimposed over it. When you click on the graphic, the advertiser's Web page loads.

Many users manage to browse the Web without paying too much attention to the advertisements, other than noticing that the ads slow down the loading of pages. Obviously, the ads influence some people because companies continue to invest huge amounts of money in them. The marketing techniques for advertising on the Web are becoming more sophisticated all the time. An industry is developing whose function is to monitor who visits what sites so that ads can be targeted more specifically to certain users. Users who began surfing the Web in 1996 or later have always been accustomed to ads, but those who started earlier than this remember the days of very limited advertising. The style, form, and content of ads is becoming yet another part of the Internet culture that is rapidly emerging.

Table I Some Emoticons

Emoticon	Meaning
:-(	Frown
:-[	Grim
:-D	Laugh
:-)	Smile
;-)	Wink

E. Societal Impact

The Internet has had an enormous impact on society and no doubt its influence will continue. Nearly all facets of life have been affected. Many people now work in Internet-related jobs either building computer network components, writing software, creating Web pages, performing marketing research, designing graphics, or conducting business on the Web. Unless you are out hiking the Pacific Crest Trail, you will probably run across at least a few URLs *everyday* (even on the trail there are people carrying laptop computers). Many people obtain all their information and perform most of their communication using the Internet. Items like weather, news, stock prices, and travel information are accessed by hundreds of millions of users every day. It is difficult to think of *any* areas of society that have *not* been strongly impacted by the Internet.

VI. BUSINESS CULTURE AND THE INTERNET

Anyone who comes in contact with *any* form of media knows that many businesses are recognizing opportunities in on-line activities. In newspapers and magazines, we constantly see URLs advertised. On television we are flooded with URLs too. It was not that long ago when there was a big debate among advertisers about whether or not to include the ugly looking "`http://`" part of a URL in a television advertisement. Turn on the radio and you will hear announcers giving out URLs almost constantly. There are still some radio announcers who do not know how to read a URL, nevertheless they continue to try and most are getting the hang of it.

The question that many users and companies pose is: Is it safe to do business on the Internet? Several large computer companies have television ads out now trying to convince you that it is safe. Consumers are worried about whether businesses are going to find out everything about them. Some businesses are concerned because they are not knowledgeable enough about the Internet. Because business on the Web is still in its infancy, many questions remain unanswered. We explore some business issues in this section.

A. On-line Businesses

The Internet functions nicely as a means of facilitating business communications within a given company as well as between companies. Furthermore, the Internet is proving to be an excellent venue for advertising and conducting trade with consumers. It is currently possible to shop for goods and services through on-line catalogs, subscribe to on-line versions of magazines and newspapers, and purchase software to name just a few possible types of business transactions in the on-line marketplace.

In addition to lowering the costs of transacting business, the Internet is transforming the marketplace into a global environment where businesses and consumers are no longer restricted by their geographical locations. For companies this means more potential customers, and for consumers this means a greater selection of services and products. This revolution is literally changing the way a lot of companies do business.

1. On-Line Business Hurdles

Probably the most significant concerns that consumers have about doing business on-line are the issues of *privacy* and *security*. We touch on these items here. After disclosing personal information and revealing spending habits on-line, consumers want assurance that the information will go no further. It may be a bit unsettling to revisit a Web page and have them ask you if you would like to pick up where you left off. Or it could make some users nervous if their favorite on-line catalog remembers their hat sizes, shoe sizes, and credit card numbers. After all, what is to prevent this information from falling into the wrong hands? These snippets of personal information that businesses keep track of are actually stored as data on *your* hard disk in a file that is usually called `cookies`. They may also be stored in a `cookie` directory. For our discussion we assume the personal data is stored in the file called `cookies`. We refer to each entry in the file as a *cookie*.

B. Cookies

Information is sometimes collected about you when you visit a Web page. It might be your name, password (if you are registering for that page), preferences, your computer's name, flags that keep track of what you looked at, credit card number, phone number, address, or perhaps some other personal information. In fact we often volunteer this information by filling out a *form* on a Web page. Or it may be inherent in the transaction process, such as supplying a mailing address when ordering. Parts of this information may be *encrypted*. A Web server sends this information to your browser and the data is written to the `cookies` file that is stored on your disk. This process is known as *setting a cookie*. Some browsers allow you to select an option so that you are notified when a cookie is to be written. You have the option of not allowing the cookie to be written. Using the `cookies` file, a Web server can also keep track of which Web pages you visit. The next time you visit a particular Web page, the server will search the `cookies` file, retrieve the information that was stored there earlier, and use the information to customize its Web page to accommodate you.

In actuality, the amount of data that can be stored in a cookie is very limited. The most likely scenario is to store an `id` for you, fetch that `id` from the cookie, and then look up your `id` in a database on the server for your more detailed profile and history.

The purpose of putting information in the `cookies` file on *your* disk is to reduce the server's search time in locating a specific cookie, namely yours. Since the `cookies` file is limited in size, it typically contains up to about 300 cookies, locating a specific

cookie can be done fast. However, the size limitation means that after a period of time some cookies must be removed. The least recently used entries are deleted when space is needed. Cookies also specify an expiration date after which time they may be removed from the file. The name *persistent cookie* derives from the sometimes long periods of time that elapse before entries are deleted from the `cookies` file.

One concern about the `cookies` file is that information may be retrieved and used to determine an individual's personal habits. Aside from credit card account numbers, it is generally felt that the information that is recorded is fairly harmless. Credit card security is a valid concern but such numbers are *encrypted*. Many feel that the pluses of cookies outweigh the potential negatives. For instance, cookies are used to keep track of items on your shopping list as you go through an on-line catalog. When using the Netscape browser's preference options, your preference selections are saved as cookies on your disk so that your preferences do not have to be set every time you start the browser. These are but two interesting and harmless uses of cookies.

The cookie controversy is one that consumers will have to decide on their own. The general feeling is that the benefits and functionality provided by cookies are sufficient to justify their use. And, as with the use of credit card transactions, there is a tendency for journalism to stress the *possibility* of risk.

C. Is It Safe to Do Business on the Web?

Probably the biggest concern that consumers have about conducting business on-line is the issue of secure payment—is it safe to use your credit card on-line? The more relevant question may be whether or not it is as safe to do business on-line as it is to conduct business in other ordinary ways. Certainly, if it were true that conducting business on-line was no more vulnerable than conducting business over the telephone, then many users would feel very comfortable about on-line purchasing.

A level of concern and understanding of the transaction process is healthy, but there is a tendency in the reporting *habits* of the news media to greatly exaggerate the risks. In reality, our normal use of credit cards is not without similar risks. When ordering by credit card over the telephone, we trust the retailer to handle our information with care and confidentiality, or in a restaurant we might pay with a credit card that the waitperson takes from us and from our view.

Mechanisms to ensure secure payments are being refined. Secure Electronic Transactions (SET) is a technical standard to be implemented by Visa and MasterCard in order to make credit card payments over the Internet more secure. Other payment options are being developed as well including electronic money. The trend is that business transactions are becoming more widespread over the Internet and also more secure. It is only a matter of time before the relative level of security matches that of other transaction mechanisms or even exceeds it.

D. Legal Environment

While doubts about secure payments may scare away potential on-line consumers, uncertainty about the legal implication of doing business on-line has discouraged some companies from taking their business on-line. Consumers and businesses recognize that electronic commerce is emerging. Without a predictable legal structure and without a guarantee that governments will not suddenly impose taxes and tariffs on trade conducted over the Internet, there are a number of companies that find the risk too overwhelming. This number is probably small when compared to those businesses with pragmatic and legitimate concern that Web page visits will not translate into sales for their products or services.

E. United States Government's Commitment to Electronic Commerce

In July 1997, President Bill Clinton made a strong commitment to promoting global electronic commerce with the release of the report "A Framework for Global Electronic Commerce" or "Framework" for short. The Framework defines how policy about the Global Information Infrastructure (GII) should be developed in order to promote "the development of a free and open global electronic marketplace." The report is significant for what it plans to do as well as what it will not do. The underlying principles, summarized from the report, are as follows:

1. Governments should encourage self-regulation of the Internet and encourage the private sector to take the lead in organizing standards when needed.
2. Because technology is changing so quickly, governments should not attempt to regulate or restrict electronic commerce on the Internet since policies may be obsolete before they are enacted (except for the United States policy on export of encryption technology).

3. Governments should provide a legal environment to support electronic commerce and protect consumers when necessary.
4. Governments should acknowledge the uniqueness of the Internet by not trying to impose other regulatory structures on it such as those applying to the telecommunications industry, radio, and television.
5. Electronic commerce and the Internet should be promoted globally in a consistent manner regardless of where the buyer and seller reside.

The report addresses financial, legal, and market access issues and advocates a "hands-off" policy whenever possible. Recommendations are made about what government action may be necessary to ensure secure electronic payment systems and to safeguard personal privacy.

Currently, the Internet and electronic commerce seem poised for a free for all without any clearly defined boundaries. As governments formulate their policies regarding electronic commerce and the legal environment becomes better defined, we anticipate that both consumers and businesses will continue to feel more comfortable conducting business on-line.

VII. COLLABORATIVE COMPUTING AND THE INTERNET

Collaborative computing is currently generating great interest in many different areas of computing. Collaborative computing is defined by applications that allow the sharing of information and resources among two or more people. The World Wide Web, with its panoply of Web pages, is a collaborative computing platform that employs HTML and Web browsers. Lotus' Notes, Novell's Groupwise, and Microsoft's Exchange are examples of software supporting collaborative computing.

The need for collaborative computing is clear as businesses and individuals have to cope with more and more information, and the cost of travel for face-to-face meetings continues to escalate. Employees were spending too much time sorting through "data" that crossed their paths: e-mail, faxes, mail, memos, reports, and voice messages. This problem was compounded by the downsizing and restructuring going on in many companies that translated to fewer people doing more work. Organizing the information and correctly forwarding the information was time-consuming as well. To stay competitive, businesses and organizations are turning to collaborative computing to share knowledge and resources, and to move information along efficiently.

A networked computer system provides the basis for a collaborative computing infrastructure. The software that makes up the collaborative computing platform (sometimes referred to as *groupware*) allows users to schedule meetings, coordinate calendars, send e-mail, work jointly on a document, or confer without physically being in the same geographic location.

A. Applications of Collaborative Computing

From customer and account service to research and product development, collaborative computing is capable of enhancing many aspects of business. The most basic collaborative computing application is the one that has been around the longest, e-mail. In terms of office communication, e-mail has replaced the written memo in many organizations resulting in savings in both time (distributing the memo) and in money (paper costs).

Collaborative computing can also simplify the process of filling out an expense report. Using an "intelligent" form, an employee only has to enter expense amounts—the expense figures are then automatically calculated and the report is electronically submitted. After the form is automatically routed to the appropriate supervisor for review, it is electronically directed to the accounting department who disburses payment. At any point in the process the employee is able to track the report to determine its status. In a similar way, purchase orders can be filled in and dispatched. This model permits fast and easy distribution as well as convenient tracking.

Version control is another use of collaborative computing. Collaborative computing software helps make it possible for more than one person to work on a document at the same time by keeping track of the latest version of the document, and updating all other copies as needed. By not having to send hard copies of the current document back and forth (or even file versions as attachments to e-mail), a large time savings is realized. If the system works properly, there is little chance of users getting out of synchronization while working on a document.

One of the most exciting applications of collaborative computing allows for real-time interaction through *video teleconferencing* or simply *videoconferencing* (VC). Traditionally, business communication has involved the exchange of data and voice information, however, VC enables a real-time exchange of colorful video images and audio from one geographic location to

another. Uses for VC seem almost limitless but the most universal example of VC involves the business meeting. Businesses were the first to embrace VC technology despite its initial high cost; they could justify their investment in terms of travel costs and time savings. It is often the case that different groups of people in a single location are communicating with other groups somewhere else. Thus, multiple sets of VC equipment are necessary. Since the cost of good VC equipment is still fairly high, its use is not yet standard.

A less expensive technology for remote conferencing is called desktop videoconferencing (DTVC). This makes use of regular personal computers and provides interaction between groups of individuals, each situated at their own PC, rather than a group of people in a single location. Schools are one example of a group that could benefit from DTVC by connecting teachers and homebound students. A very simple DTVC setup might consist of a PC connected to a miniature video camera through a video card. A microphone could either be connected through a sound card to the PC or it might be part of the camera itself. More sophisticated DTVC systems contain the camera and microphone inside the monitor. Either way, a high-speed ISDN line should be used rather than a regular telephone or slower line to connect to the Internet since transferring audio and video data requires much more *bandwidth* than transferring just data. Bandwidth refers to the transmission capacity and is usually measured in bits per second.

It is worth emphasizing that a VC system may include any or all of the following aspects with varying technological costs. In order of increasing bandwidth requirements: real-time talk or chat, whiteboard graphics, audio, black and white video, and color video.

B. Impact of Collaborative Computing

The major benefits of collaborative computing are convenience and time savings; these amount to money. Employees can examine, organize, and route data efficiently while managers have easy access to data and are able to find information in a timely manner. Electronically forwarding and accessing information saves time since paper does not have to be physically distributed. By using audio, graphics, and video plus text in a collaborative computing environment, one has the means for clearer communications. This can result in fewer errors and misunderstandings. In addition, travel time and expenses can be significantly reduced by collaborative computing since being in the same geographical proximity is not a prerequisite for an exchange of ideas.

C. Future of Collaborative Computing

Collaborative computing may soon become a necessity for businesses that want to remain competitive. Unfortunately, a number of the commercial groupware products are quite expensive. There are, however, some applications that make use of Web technology and provide a cheaper alternative. For software that costs only a fraction of what the well-known groupware products cost, some companies are getting by utilizing a Web-based platform as their collaborative computing environment. In addition to being cheaper, many users find the Web technology easier to use. Both Microsoft and Netscape include groupware in their version 4.x or more recent browser suites.

The infrastructure for a Web-based collaborative computing platform is an intranet. Access within an intranet is limited to employees and business contacts by a security measure known as a *firewall*. Web software developers are busy developing more sophisticated security measures and are quickly producing workable solutions. Intranet-to-intranet communication across the Internet is possible if one uses a technology that does not require dedicated bandwidth.

In response, groupware providers are trying not to compete directly with the Web technology. Instead, they are attempting to make their products compatible with the Web by allowing various browsers to access their databases. How well groupware is able to meld with the Internet may determine its success. In the meantime, some companies are using a combination of Web technology and groupware. An internal Web page may exist to serve as a bulletin board or for displaying company manuals, while a product like Lotus Notes may be used for applications that require a measure of security.

Many businesses are already improving worker productivity through the use of collaborative computing. It seems clear that as developers overcome some of the current hurdles, collaborative computing will become even more prevalent.

ACKNOWLEDGMENT

The material in this article has been developed from Greenlaw and Hepp. The reader is referred there for more details.

SEE ALSO THE FOLLOWING ARTICLES

Computer History • Computer Viruses • Electronic Commerce • Intranets • Search Engines • Telecommunications Industry • Wide Area Networks

BIBLIOGRAPHY

About a framework for global electronic commerce. (1997). www.whitehouse.gov/WH/New/Commerce/about-plain.html.

A brief history of the Internet. (1997). www.isoc.org/internet-history/brief.html.

A framework for global electronic commerce executive summary. (1997). www.whitehouse.gov/WH/New/Commerce/summary-plain.html.

Comer, D. (1997). *The internet book. Upper Saddle River, NJ: Prentice-Hall.*

Cookie central. (2001). www.cookiecentral.com.

Cookies and privacy. (2001). www.epic.org/privacy/internet/cookies/.

CUSeeMe page. (2001). www.cuseeme.com/.

Cunningham, S. J. (June 1997). Teaching students to critically evaluate the quality of Internet research resources. *SIGCSE Bulletin,* 29(2), pp. 31–34.

Debunking myths about Internet commerce. (1997). commerce.ssb.rochester.edu/papers/comment.htm.

The Economist: Internet, too cheap to meter and the world wide wait. (1997). www-uvi.eunet.fr/hacking/nov13=17nov96-6.html.

FNC resolution: Definition of Internet. (1997). www.fnc.gov/Internet_res.html.

Greenlaw, R., and Hepp, E. (2002). *In-line/on-line: Fundamentals of the Internet and World Wide Web,* 2nd edition. New York: McGraw-Hill.

Hafner, K., and Lyon, M. (1996). *Where wizards stay up late: The origins of the Internet.* New York: Simon and Schuster.

Hahn, H. (1996). *The Internet: Complete reference.* New York: Osborne McGraw-Hill.

History of the Internet. (1997). www.davesite.com/webstation/net-history.shtml.

How the Internet works. (1997). www.iw.com/1996/howitworks.html.

Internet timeline by Hobbes. (2001). www.isoc.org/guest/zakon/Internet/History/HIT.html.

Internet engineering task force. (2001). www.ietf.org/.

Life on the Internet: Net timeline. (1997). www.pbs.org/internet/timeline.

Netscape cookie specification. (2001). www.netscape.com/newsref/std/cookie_spec.html.

Netscape world—Use cookies to analyze user activity and create custom web pages. (1997). www.netscapeworld.com/netscapeworld/nw-02-1997/nw-02-cookiehowto.html.

The truth about cookies—Christopher Barr. (1997). www.cnet.com/Content/Voices/Barr/042996.

Vint Cerf on the past, present and future of all things Internet. (1997). www.wiredguru.com/cd2.html.

W3C—Platform for Internet content selection. (2001). www.w3.org/PICS.

Intranets

Deborah Bayles Kalman
University of California, Irvine and Singapore Institute of Management

GLOSSARY

extranet An intranet that allows controlled access by authenticated inside and outside parties.
firewall A set of components that functions as a choke point, restricting access between a protected network (e.g., an intranet) and the Internet.
intranet An internal closed network based on Internet technology.
LDAP (light directory access protocol) A protocol enabling an enterprise to build a global directory that gets replicated to all the different services that are running in the enterprise. The directory lists all the directory entries and access control information for people in the enterprise.

AN INTRANET is internal closed network based on Internet technology designed to foster communication and collaboration within a single enterprise. In contrast, an *extranet* allows controlled access by authenticated outside parties. Typically an extranet links multiple intranets of distributed organizations for the purpose of conducting business. While an intranet and extranet are designed for collaboration and operate around limited access to information, there are some significant differences in each.

Table I shows the relationships and differences among an intranet, extranet, and the Internet. Note that the primary difference starts with the narrow target audience for an intranet, and that there is a subsequent building of processes to serve them slightly differently than with extranets or the Internet.

Successful process-oriented intranets look and work as differently as the processes they enable, but they share several common characteristics.

- They are built on smart information designed by and for employees.
- They focus on tasks, not documents, and aim to integrate those tasks into distinct processes.
- They encourage collaboration by creating shared and familiar spaces that reflect the personality of the company and create a common ground for all employees.
- They are built, deployed, and maintained around proven principles found in a software development project.
- They are an accurate reflection of the corporate culture, its values, and operating principles.

Just as physical workspaces rely on architectural plans to optimize efficiency, an intranet needs to be carefully designed to help those with access information collaborate effectively. Because the public does not see the intranet, information design for intranets often receives scant attention. Unlike customers, employees are assumed to be insiders, able to easily locate company information. So, while the company Web site usually has the input of the marketing department, design and structure of the intranet is often relegated to the IT department.

Some material in this article is reprinted by permission from Bayles, D. (1998). *Extranets: Building the Business-to-Business Web.* Upper Saddle River, NJ: Pearson Education, Inc.

Table I Brief Comparison of the Characteristics among Intranets, Extranets, and Internet Sites

	Intranet	Extranet	Internet
Target audiences	Employees	Suppliers, trading partners, customers	General public
Main business objective	Communication within company	Collaboration with select third parties	Achieve greater market awareness; branding
"Permeability"	Impermeable to outside	Semi-permeable; authentication required	Permeable; usually open to the general public
ROI goals	Cost savings through increased internal efficiency	Cost savings, greater profits through streamlined supply chains	Increased traffic, more effective marketing; new distribution channels
Use of graphics	Minimal	Minimal	Often extensive, including Flash technology
Version control	Important—Task made easier by single corporate policies	Very important—Extremely difficult because of different organizations and differing policies	Important—Task made easier by single corporate policies

A. Good Design Is Good Business

Irrespective of who designs and implements an intranet, its organization and design of information should map out the key business processes of a company and provide employees with access to the information and people necessary to carry out those processes.

The truly effective intranet creates new opportunities for communication that overcome inefficient organizational structures and foster new forms of efficient collaboration. It serves as a model for a company centered on processes rather than departments, collaboration rather than closed doors.

Building an effective intranet means thinking about how documents can be used to accomplish tasks, how tasks can be organized into processes, and how virtual work groups can carry out those processes collaboratively. The effective intranet is a tool; it is also a model for an efficient, process-centered enterprise. It is a machine for doing business.

I. CHARACTERISTICS OF AN INTRANET

In general, there are five characteristics to look for in identifying business processes that could be vastly improved by an intranet:

1. *Any business process that involves the production, requisition, distribution, and update of dynamic information traditionally published on paper.* Examples include employee directories, medical benefits descriptions, product specifications, user manuals, price lists, marketing collateral, financial reporting systems, policies, and procedures.
2. *Any business process that involves the consolidation of information from multiple data sources.* For example, a retail customer service representative must access and consolidate customer information, order history, and product information (description, pricing, availability) and enter sales order information—all while speaking to a customer on the telephone.
3. *Any business process that requires a high level of communication and collaboration between people, especially if they are separated geographically.* Today, for example, many engineering projects involve the coordination of multiple development groups scattered in several locations. Many companies have field sales offices that need constant, up-to-date access to company information as well as daily contact with the home office.
4. *Any business process that depends on people finding or requisitioning information or products.* Examples include reference manuals, internal requisition systems, channel distribution order systems, and fax-back systems.
5. *Any business process currently automated by a client-server or mainframe application.* This is particularly significant for companies with legacy systems that need to be brought up to date.

A recent survey by International Data Corp. (IDC) revealed that corporate intranets are expanding to include more and more applications. The survey revealed that there are five primary uses associated with intranets: information sharing, information publishing, communications through e-mail, document management, and data-conferencing. These and other applications are adding a sense of community and team building to the corporate information technology infrastructure by providing a synchronous component to communications.

Another significant change is in expanding the sources of information contained on corporate intranet sites. The intranet provides a platform for corporations to grow into and develop knowledge-management initiatives by adding knowledge contained in e-mail or in discussion databases so it can be accessed through the intranet, collected, and developed into a knowledge store. Once collected, the information can be used for its intended purpose or re-purposed and leveraged over different uses.

II. INTRANET ARCHITECTURE

Intranets should help employees collaborate on business processes such as product development or order fulfillment, which create value for a company and its customers. Specifically, intranets centralize the business process in an easily accessible, platform-independent virtual space. Successful intranets allow employees from a variety of departments to contribute their different skills to carry out a particular process. While each department of a company may have its own virtual space, intranets should be *organized primarily around the business processes* they help employees carry out, rather than the organizational chart of the company.

By default, an organizational chart of the company is often used to organize information on the intranet. While seemingly the obvious candidate for the structure of the intranet, an organization chart actually works against the collaboration the intranet is meant to foster. An organization chart cannot help employees from the marketing and legal departments collaborate on bringing a document through the approval process. It will not allow employees from marketing and research and development to work together to create a new product.

Focusing on processes rather than departments is now a widely hailed business trend. Recent shifts in corporate structure point to the emergence of "communities of process." To help companies move away from silos of vertical, hierarchical organizational lines management gurus are pushing enterprises towards horizontal, process-oriented groups that link cross-functional teams focused on the same set of business tasks. Process-based collaboration requires significant interaction between departments and functions, even those in other countries.

Thus the intranet is the ideal vehicle for creating and empowering process-based corporate communities.

A. Think about Tasks Rather Than Documents

Because the intranet is a tool, it is more than a collection of documents. While important, documents are usually a means to an end. People use documents to complete tasks. Tasks include fulfilling orders, looking up a customer's billing history, or collaborating on a research document. To complete these tasks, people need to have related documents and tools close at hand.

The principal of organizing by task can be illustrated by your working at a desk. When you sit down to begin a task (e.g., creating a budget), you have information and a variety of tools at hand. While a spreadsheet is a "calculation" tool, and last year's budget is an "internal document," they need to be next to each other to develop a new budget. Similarly, on the corporate intranet, the tasks of the users rather than the classification of documents or tools should dictate the organization of the intranet.

Designed effectively around dynamic tasks rather than static documents, intranets can contribute to dramatic increases in efficiency (as much as a 40% improvement in time spent processing documents, according to the GIGA Group). Organizing documents within the context of tasks also focuses employees on the function of the documents they are working with. For example, to save employee time while signing up for various benefits programs, information on various retirement plans (including links to financial Web sites) should be placed near the forms actually used to register for those plans.

B. Organize Tasks into Larger Processes

Isolated tasks are usually part of a larger process. Intranets should group together all the tasks that make up a business process. Processes can be relatively discrete, such as tracking deliveries or getting approval for documents, or they can be more complex, such as developing or selling products. The most important

processes in a company are those that create value for a customer and/or reduce process costs. These are the central processes that every intranet should help employees accomplish.

Even simple processes can become more efficient when incorporated into an intranet. For example, when Ford implemented an intranet, the company included an application to help geographically dispersed engineers get authorization for new projects. What was previously a time-consuming, expensive process, involving the potential for lost documents and delays, is now centralized in an efficient electronic process.

Also, more complex processes can be effectively integrated into an intranet. For example, Cadence Systems created an integrated section of an intranet for its entire sales process. Each phase of the sales process is represented on the intranet with relevant information and tools. So, the section covering an initial stage of the sales process includes links to customer presentations, sample letters, and internal forms. Organizing all steps of the sales process together also allows for easy tracking of each sales effort.

C. Create Virtual Workgroups Organized around Processes

Intranets can break though departmental walls to help accomplish business processes more efficiently. For example, a customer complaint might involve people and information from the accounting, sales, and marketing department. Even though the employees necessary to resolve the complaint work in different departments, they are all involved in the process of customer service. By creating spaces for cross-departmental collaboration, the intranet can help employees collaborate to carry out the central processes of the company efficiently and cut costs by avoiding in-person conferences and employee reallocations.

Intranets (and private extranets) can also bring together employees and partners who are geographically dispersed to work on common problems. Travel costs are eliminated, and employees can increase their productivity by sharing knowledge

The bulk of discussion about collaboration in and between companies centers around security, certainly an important issue to resolve. What receives less attention, but is central to the value of an intranet, is the design of virtual spaces that encourage new forms of collaboration. These, in turn, increase the efficiency of key business processes, such as product development, marketing, and customer service.

D. The Intranet Reflects the Company; the Company Reflects the Intranet

The corporate intranet can help a company organize around "communities of process" both on- and offline. Whether it precedes or follows the organizational shift, an intranet that encourages this type of collaborative work environment can provide a significant return-on-investment.

At the same time, using an intranet to shift the way work is done in an organization requires a cultural change within the organization. Unless there are clear commitments from senior management to have employees collaborate across departments to accomplish key business processes more efficiently, the intranet may have only limited application and benefit.

Even after the intranet is designed to encourage collaboration, marketing the intranet to employees remains essential. As the intranet creates new forms of collaboration, it challenges traditional ways of doing work and obtaining information. For the intranet to be successful, it must provide ways of empowering all employees, offering concrete incentives for employees to use it, and for them to encourage the use of the intranet. The process-oriented intranet, then, is "in sync" with the company it works for. This is where graphic design, tone, and standards emerge as vital to the intranet's success. Like it or not, intranets have personalities, which are amalgams of visual style, tone, and content. An intranet that reflects the culture of its company makes employees feel more at home, helps dispersed employees feel that they share the same space, and encourages collaboration and communication around the processes they support.

III. PLANNING ISSUES

Security, at the network and host levels, is critical within an intranet. Through the use of both physical and virtual firewalls, passwords, encryption, and various forms of user authentication, intranets must be able to manage security and accountability. Interaction and exchange of information throughout any participating organization must be protected from the public Internet as well as designated intranet members who should not be privy to certain information. The security model must be flexible in its architecture and should be able to provide access controls based on individual, group, organization, transmission type, or other business criteria.

A. Access Control

Access control is at the heart of every intranet. Managing access control can be a monumental task if the number of users is large. It involves defining user access privileges, managing passwords, user IDs, and authentication schemes, and maintaining user accounts.

One answer to this challenge is tools that employ LDAP (light directory access protocol) capabilities. With this tool, an enterprise can have a global directory that gets replicated to all the different services that are running. LDAP helps list all the directory entries and access control information for people in the enterprise. Users come into the network and log into the directory. They authenticate themselves to the enterprise's directory and have access only to the set of resources designated by the system administrator.

Suggested security levels are as follows:

1 = Public Access. No restrictions on viewing the information inside or outside of the company. This information would typically reside on an external Web site.

2 = Customer Access. Password required. Access to Level 1 information, plus information specifically for customer usage (i.e., customer support)

3 = Sales/Reseller/Distributor Access. Password required. Access to Levels 1 and 2, plus special pricing, technical, marketing, and sales information.

4 = Employee Access. Password and authentication required. Access to Levels 1–3, plus company confidential information (i.e., insurance benefits and enrollment, 401K plans, etc.).

5 = Administrator/Officer Access. Password and authentication required. Access to all levels, with special access to highly confidential company data. Full system privileges.

Table II displays these interrelationships among elements in the five levels.

It can be quite time-consuming to unearth and classify your company's information, commonly called corporate artifacts, but this task is necessary to structure the layout and password/authentication scheme for your intranet. It is also very important that consensus be reached on the security levels assigned to the content. Marketing may deem one document suitable for public consumption, while another department expects it to be highly confidential. The grid also helps clarify each user's "need to know"—in other words, access is granted depending on whether or not the data are necessary for the user to perform his or her job.

Contact management systems, e-mail gateways, pager systems, and collaborative groupware applications can be integrated into an intranet, enabling salespeople to proceed through the sales cycle without having to wait for traditional approvals, paperwork, or confirmations. A full-featured intranet can

Table II Interrelationships among the Five Levels of Access Control and Intranet Tasks

Security level	1	2	3	4	5
Human resources					
Employment opportunities	X				
Company event information			X		
Employee manual				X	
401K plan information/enrollment				X	
Sales					
Pricing manual			X		
Direct sales incentive program				X	
Product upgrade ordering		X			
Compensation plan					X
Reseller/partner program info			X		
Free evaluation software	X				
Customer support					
On-line help desk		X			
Product registration		X			
Support contracts					X
Software patches		X			
Bug tracking				X	

also maintain a constant stream of contact throughout the enterprise, so that important business deals are not compromised by missed calls or other frustrations. Customer relationship management, calendaring, and other applications can track a sales prospect through the sales cycle, instantly displaying where a prospect is in each stage of the sales process.

IV. VERSION CONTROL WITHIN A COLLABORATIVE INTRANET

Version control and configuration management are critical to a well-run intranet. Allowing participants to "check out" documents and files, make modifications, and comments on-line, and then check them back in has been expanded to enable concurrent application and site development.

A new generation of Web-based version control applications is enabling collaboration via "virtual teams" that come together for the purpose of a project, and then disband upon project completion. A geographically dispersed intranet community can jointly develop projects and not worry about overwriting each other's files or duplicating effort. Because geographical and time constraints are minimized with an intranet, entire virtual corporations are being built using an intranet as a backbone.

Management of distributed teams can be especially useful in projects involving multinational commerce, subcontractor relationships, worldwide associations or task forces, specialized development projects, and other activities that require interdepartmental cooperation and/or contribution from a geographically dispersed community.

A. Pitfalls

The Internet Web site and intranet of most companies start out as an "on-line brochure" that the Marketing Department has decided would be a great place to put all of the content that originally existed as printed collateral materials, manuals, and other documents. Similarly, there is also a good chance that all changes are being made on an ad hoc, chaotic basis. It probably continues to be managed as a "brochure project."

If all of these suppositions are correct, it is almost certain that the whole thing is being managed manually. That means there are many other serious pitfalls (described next) that can result in the loss of valuable time and data. Here are some of the intranet killers that often corrupt the very efficiency it is designed to correct:

Death by Change Requests—The Webmaster for a growing site becomes increasingly deluged with e-mails, voice mails, sticky notes, and memos, all demanding immediate changes to the Web site. Soon the Webmaster becomes consumed with administrative tasks, and the creation of new (and possibly revenue generating) site functionality is sacrificed.

Manual Merging—If multiple developers make different modifications to separate copies of the same file, then the Webmaster has to merge all of the changes manually from all of the different copies into one version of the file.

Accidental Overwrites—If a developer makes changes to a file without communicating that an updated version of the file exists on the network, other developers may accidentally overwrite the first developer's changes.

File Lockouts—A well-meaning Webmaster, in an effort to avoid overwrites, may try to set up an environment in which only one developer can work on a file at a given time. This locked-file approach impedes the work of other developers who may need to make edits to a file while a developer is working on it.

Multiple File Formats—Web sites, like software applications, include more than source code. Graphics files, analysis and design diagrams, user requirements, marketing materials, and supporting documentation are all part of a typical application development project. When dealing with Web sites, you are also dealing with a wide variety of file formats, such as:

- Mark-up language files (HTML, SGML, VRML, XML)
- Image files (GIF, JPG)
- Common Gateway Interface scripts (CGI)
- Perl scripts (PL)
- Java source code (JAVA)
- Java object code (CLASS)

Keeping track of even a small project involves manually coordinating hundreds of changes and keeping all related documentation and types of files in sync with those changes.

B. A Virtual Team Is the Answer

The only way for a Webmaster to survive, and a corporate intranet to succeed, is to make the whole thing a managed, collaborative effort much *like a software development project.* This means involvement at all levels of the business process by authors/developers who have banded together as a "virtual team." The team

approach has been common in software application development for years. Thus, there is every good reason to view the development of an intranet as an extremely robust, business-critical application development project.

The key to making virtual teams work is to enable true collaboration with an automated version control and configuration management solution. There will always be the complications of differing work methodologies, company priorities, and individual agendas when a company allows outside third parties into its intranet, but solid version control and configuration management measures give a company a better chance at true collaboration.

C. Automated Version Control and Software Configuration Management

The primary role of version control and software configuration management software is to enable team collaboration by providing:

- Developers with team productivity tools that unobtrusively extend existing development environments
- Project managers with better and more concise project information
- Everyone with the ability to contribute comments, suggestions, problem reports, and project documents
- A project-oriented repository for the collection, organization, and distribution of project materials

1. The Principles of Software Configuration Management

Software configuration management (SCM), or software change management, as it is sometimes called, consists of four major activities:

- **Configuration Identification—**This is the process of identifying all of the components of a project and ensuring that these components can be found quickly throughout the project life cycle. As previously mentioned, a typical intranet project is like a software development project and is comprised of much more than source code or HTML. Configuration identification breaks a project into smaller, more manageable subprojects, such as design documents, special graphic files, and so forth. A good automated SCM package supports mapping of a project tree, indicating the logical configuration hierarchy, as well as the directory structure, or physical configuration hierarchy. The version control and SCM product must be able to cross all departmental boundaries to include a wide variety of project participants.
- **Configuration Change Control—**This important activity coordinates access to project components among team members so that data do not "fall through the cracks," become lost, or that unauthorized changes are made. To provide protection from lost changes, most SCM systems offer a check-in/check-out process that allows write-access to a single user for a project file. Current and previous versions of a file are identified and tracked, with the ability for a user to request a copy of a previous version of a file at any time.
- **Configuration Auditing—**Configuration auditing is a process that confirms that a software or intranet project is on track and that the developers are building what is actually required. By developing a series of checklists that specify what components are in a given baseline, a company can audit the degree to which a project or intranet is complete.
- **Configuration Status Accounting—**The goal of configuration status accounting is to record why, when, and by whom a particular change is made to the source code of a project. In the past, developers would manually keep notebooks and insert comments into the code, but good SCM systems keep automated histories of all changes and generate reports that describe the changes over a period of time.

a. INTEGRATION

A good version control and configuration management product should provide a set of integrated tools that address the following three functional areas:

i. Functional Integration

- Project organization and navigation—Easy-to-use visual project trees to quickly organize files and navigate through projects, with version histories to trace the evolution of each document/subsite.
- Version control—This function focuses on tracking and coordinating changes to documents and includes facilities to organize files and manage storage, library check-in/check-out, and file locking to control access to shared documents, file comparison, and differencing utilities to pinpoint discrepancies between different versions of documents and to resolve conflicts, branching

and merging mechanisms to manage parallel development.
- Defect tracking—Online defect submission, severity rating, and tracking through the defect resolution cycle.
- Threaded conversations—Online discussion facilities that allow multiple authors to discuss changes and enhancements relevant to each version.
- Build and milestone management—Facilities to identify milestones and group associated documents/files with configuration management that stores information relative to each build, for build re-creation.
- Auditing and reporting—Complete audit logs to track processing of changes, and flexible reporting to see what has changed at a glance.

ii. Departmental/Partner Integration The objective of any good version control product is to provide a collaborative environment for a number of internal departments and external teams. If appropriate, customers and end users may also want to provide feedback or participate in the site development process in a controlled fashion. This level of integration provides benefits to:

- Sales and Marketing—Become major sources of intranet content who can provide valuable feedback as the "gateway" groups between the intranet and Internet audiences.
- Development—Has access to the contributions of all departments working on the project, with a low time investment.
- Quality Assurance (QA)—Gains greater visibility into project component changes and their role in future builds with a good version control system.
- Technical Support—Gains an interactive forum for discussing components of the intranet via threaded discussions.
- Executive Management—Gains a "big-picture" view of the entire development process, with the option of receiving more detailed reports.
- Strategic Partners, Distributors, Customers and End Users—Can engage in documented discussions with developers about the intranet's components and can report defects and enter change requests without intermediate paperwork.

iii. Geographic Integration

- Extranet Support—Using TCP/IP, users can connect over the Internet, LAN, and/or Wide Area Network (WAN), forming virtual teams without regard to physical location.
- Web Browser Support—On an intranet, the Web browser is the universal client. Any Software Configuration Management (SCM) package must enable local and remote users to participate in intranet development using only a browser.

2. The Benefits of Automated Version Control and Configuration Management

Other than avoiding mass chaos, there are so many additional benefits to implementing automated version control and configuration management that many companies want to invest the additional time, effort, and relatively minor upfront costs to implement it. Version control and configuration management provides the following benefits:

i. Improves Communication among Intranet Partners/Content Developers By automating the communication process, a version control system enables the Webmaster to establish a single, consistent channel for communicating and processing change requests, ensuring that none falls through the cracks. Employing a consistent communication mechanism also ensures that threats to quality and schedules are discovered, communication bottlenecks are eliminated, and development and test time is saved. Most programs can be configured to notify users automatically that their requests were received, and the team can be confident that all requests are reliably stored and easily accessible.

ii. Protects Shared Web Source Files under Rapid Development A version control system helps you store and track changes to Web source files. A good system can accommodate Web sites containing as little as a few pages to sites with thousands of pages and multiple combination of file formats. It should be flexible enough to enable you to customize the program to accommodate any Web directory structure. Version control systems use a check-in/check-out process to protect shared files from being accidentally overwritten in a team environment. To edit a project file, a developer checks it out of the archive and puts a lock into effect. While the file is locked, no other developer can modify the file until the first developer checks it back in. Most systems also enable you to allow multiple developers the ability to edit copies of the same file in parallel. Later, the version control system automatically merges the changes into a single version.

iii. Enhances Development Workflow Another benefit of implementing version control is that it encourages establishment of good workflow practices. A good version control system automates development workflow by enabling the Webmaster to quickly prioritize

and assign Web content requests, run reports to determine the status of any request, determine whether project files are still checked out, or view a summary of the modifications made to project files. Report summaries should be available that show the classification of job priorities, workload assignments, and job progress updates. In addition, managerial reports that illustrate trends, number of requests, project closure rates, requests by originator categories, and department and resource allocation should be available.

iv. Saves Time Enhancements, new features, and content can be added with an integrated system much more quickly and at less expense. The resulting information, products, and services provided by the intranet team can reach users, prospects, and customers faster. This translates into the potential for increased revenues.

v. Reduces the Number of Defects Introduced into the System Many of the most common defects that are introduced during the development process can be eliminated with automated version control and configuration management. Defects caused by accidental overwrites, lack of communication, and manual merging of changes can be prevented by a good version control system.

vi. Reduces the Costs and Time to Find Defects That Are Introduced Most version control systems feature a severity rating system that enables team members to specify the priority level of their change requests. A list of requests sorted by severity rating can then be generated so that the most important defects can be addressed immediately, resulting in the rapid resolution of the most severe and revenue-critical defects.

vii. Reduces Maintenance Costs An important part of an automated version control and configuration management product is its ability to re-create an earlier revision, or build, of the system. The software maintains a cumulative history of the changes made to each source file, including what has been changed, when, and by whom. It then becomes easy to restore an earlier version of a file, reducing maintenance costs. With the rapid application development cycles involved in intranet maintenance, it is often necessary to restore an earlier version of a file as a basis for a new Web page or image.

viii. Improves Productivity of the Development Team When communication is streamlined and everyone has visibility into all aspects of a project, true team collaboration is possible, and productivity skyrockets.

ix. Reduces the Costs of Content and Application Development by Eliminating:

- Unproductive meeting time and redundant e-mails
- Rework and unnecessary changes
- Time spent preparing manual reports

x. Improves the Quality of Intranet Applications by:

- Ensuring that outstanding issues get resolved
- Enabling early and ongoing participation by nontechnical staff
- Encouraging software component reuse

a. Secondary Benefits of Version Control and Configuration Management

- Better corporate image
- Improved team morale when the intranet team feels that their efforts are being supported
- Less overtime and weekends required on the part of the development staff
- Increased respect for the intranet development team from organizations external to the effort
- A more competitive stance in the marketplace
- Increased customer satisfaction
- Improved communications among all staff at all levels and between levels

3. Managing the Life Cycle of Your Intranet

a. Putting Together an Intranet Control Board

To make the virtual team development process more effective, it is very important to put together an Intranet Control Board (ICB). It is a common practice to run Change Control Boards within software development departments, and the same concept is useful for managing your intranet. It is best to enlist top-level delegates from a number of departments and partners to achieve buy-in and sustained support throughout the life of the intranet.

One of a company's main tasks is to develop a set of clearly defined roles and responsibilities for all parties involved with its intranet in order to know how they interact with each other and within the parameters of the design documentation in use. Also, the ICB is a forum in which to develop intranet policies and procedures, define style guides, and so on. One member of the ICB may perform one or more of these roles in an intranet project.

Here are some suggested roles and responsibilities:

- **Intranet Board Member—**The Intranet Control Board is comprised of members who are responsible for all decisions regarding major additions, deletions, or other changes to the intranet's content. A major change would be one that substantially alters the functionality, intent, or

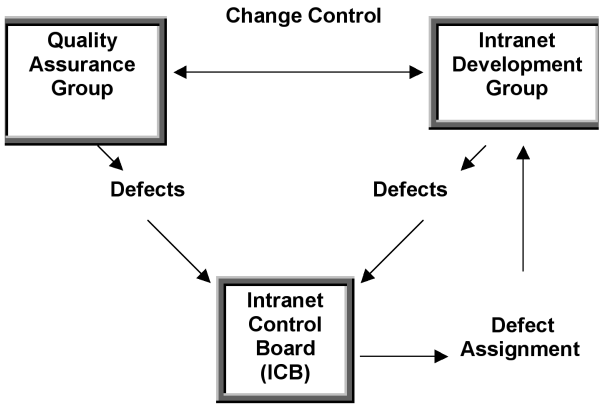


Figure 1 The interaction of the Intranet Control Board (ICB).

implementation of the intranet. These decisions are based on requirements documents, design documents, and other strategic plans.

- **Webmaster/Team Leader—**A large or far-flung intranet may have multiple Webmasters or team leaders. They are responsible for prioritizing and assigning work to the developers based on projects approved by the Intranet Board.
- **Developer—**Analysts, designers, database administrators, language translators, writers, programmers, and other people who have a direct role in the design and implementation of the intranet fall into this category.
- **Quality Assurance (QA) Manager—**The QA Manager, a crucial member of the ICB, has the responsibility of determining whether the requirements specified in the design documentation can be demonstrated in the components of the intranet.
- **Tester/Reviewer—**These members of the ICB include those who have enough knowledge of the intranet's components and applications to be able to give useful feedback as to their functionality and performance. Often selected customers or third parties perform this function as participants in a beta test program.
- **Product Manager—**The Product Manager ensures that the particular content area or application on the intranet fulfills the needs of the target audience. For example, a company may have one manager dedicated to implementing electronic commerce for its intranet, and therefore they would conduct market and competitive research, poll prospective users, and then develop application content criteria based on the findings.

With a well-run ICB, a company can gain senior-level support across all of the organizations, achieve early buy-in and involvement by the different business entities, and help establish a level of ownership that sustains the project.

The ICB should be directly involved in the development process. Figure 1 shows a simplified example of the interaction of the ICB through the assignment of "defects." Defects in this case cover all types of change requests.

SEE ALSO THE FOLLOWING ARTICLES

Electronic Mail • Extranets • Firewalls • Internet, Overview • Local Area Networks • Mobile and Wireless Networks • Security Issues and Measures • Telecommuting • Virtual Organizations • Wide Area Networks

BIBLIOGRAPHY

Bayles, D. L. (1998). *Extranets: Building the business-to-business web,* 1st ed. Upper Saddle River, NJ: Prentice-Hall.

Bayles, D. L. (2001). *E-commerce logistics and fulfillment: Delivering the goods,* 1st ed. Upper Saddle River, NJ: Prentice-Hall.

Bidgoli, H. (2002). *Electronic commerce: Principles and practice.* San Diego: Academic Press.

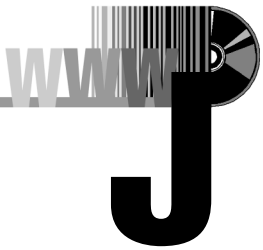
McDyson, D. E. (2000). *VPN applications guide: Real solutions for enterprise networks,* 1st ed. New York: Wiley.

Norris, M. and Pretty, S. (2000). *Designing the total area network: Extranets, VPNs and enterprise networks explained,* 1st ed. New York: Wiley.

Phifer, L. (2000). Tunneling at layer two. ISP-Planet, internet.com Corporation.

Szuprowicz, B. O. (2000). *Implementing enterprise portals: Integration strategies for intranet, extranet, and Internet resources,* 1st ed. Charleston, SC: Computer Technology Research Corporation.

Wilson, C., and Doak, P. (2000). *Creating and implementing virtual private networks,* 1st ed. Scottsdale, AZ: The Coriolis Group.



Java

Andrew F. Seila, John A. Miller, and Senthilanand Chandrasekaran
University of Georgia

GLOSSARY

applet A Java program that is exectued on a client in a browser.
CORBA (Common Object Request Broker Architecture) A standard architecture and infrastructure for storing and retrieving objects in a distributed environment, specified by the Object Management Group.
EJB (Enterprise JavaBeans) A collection of standards and packages that support distributed object management and object-relational mapping with databases.
J2EE (Java 2 Enterprise Edition) A collection of APIs and classes that supports multi-tier applications involving clients, Web servers, and database servers.
JavaBean A standard for declaring Java classes that allows other Java programs to query objects to discover their properties and capabilities and enable easy program design in GUI programming environments.
JDBC (Java database connectivity) A standard API for accessing data in relational databases from Java programs.
JINI A technology that supports communication, discovery of capabilities, and interaction between independent objects in a network environment.
JSP (Java Server Pages) A convenient way to define and create servlets using a special set of tags and a JSP processor on the Web server.
JVM (Java Virtual Machine) A software component that executes the bytecode in a Java class file on a microprocessor.
RMI (Remote Method Invocation) A distributed object model in which the methods of remote objects written in Java can be invoked from other JVMs, possibly on other hosts.
servlet A Java program that is executed on a Web server by having its calling Web page loaded by a Web browser.
socket One end of a bidirectional communication link between two Java programs.
thread A section of code that may execute concurrently with other threads.

JAVA is an object-oriented general purpose programming language that includes features to support programming in an Internet environment. The language was developed by researchers at Sun Microsystems to program consumer devices, but with the emergence of the Internet, found its place as a general purpose programming language for networked environments. Platform independence is achieved by providing a Java Virtual Machine, or JVM, for each architecture to be employed. The Java compiler outputs Java bytecode, which is machine independent, and the JVM executes the bytecode. Java is strongly typed and strongly object-oriented, and has a syntax very similar to C++. Native support for concurrent programming using threads is provided. Some of the facilities provided by Java for network programming include applets, sockets, remote Method Invocation, and servlets. Applets are Java programs that execute in a Web browser. Sockets provide a primitive method to communicate between two computers on the Internet. Remote Method invocation allows a Java program on one computer to execute a method located on another computer.

Servlets are Java programs that are executed on a Web server. Servlets provide a powerful set of tools for server-side programming in a Web environment. Java Server Pages (JSP) offer a simpler means to create and run servlets. Java 2 Enterprise Edition (J2EE) is an addition to the basic Java software development kit that supports multitier enterprise applications and provides facilities for distributed object management and object-relational mapping with databases.

I. INTRODUCTION

Java is an object-oriented general-purpose programming language that provides special features to support programming in an Internet environment. Before discussing the specific features and capabilities of Java, it is interesting to review the history of this language.

A. A Short History of Java

In 1991, Sun Microsystems created a team to explore and develop technology for emerging trends in computing. This group, dubbed the "Green Team," anticipated the convergence of computing and consumer devices and believed that much of the programming on these devices would be transmitted to the device, rather than being stored on the device in read-only memory. This approach would provide much more flexibility and indeed let the device's behavior be determined by the content of the media. It was in this context that Java technology was created in 1991 by Patrick Naughton, Mike Sheridan, and James Gosling.

A program that is sent to another device for execution cannot anticipate the hardware capabilities of the device. Thus, it is not feasible to compile the program into machine code before making it available to the device, since any deviation from the assumed hardware architecture would render the code useless. To overcome this problem, the Green Team proposed using a solution that involves compiling the source code into machine code for a "virtual machine," then using an interpreter on the device that executes this code on the physical hardware. This method had been used earlier in the UCSD Pascal compiler, among other places. The output of the Java compiler is called Java byte code, and the software that runs on the device to interpret Java byte code is called the Java Virtual Machine (JVM). With this approach, the program, after having been compiled into Java byte code, can run on any computer or device that has a Java Virtual Machine installed.

Java, which was first named "Oak" after the tree outside Gosling's window, was originally demonstrated on a hand held wireless device called "*7" that would communicate with a TV set-top box to provide interactivity and additional functionality. The market for such products was still immature, so the Green Team, which had been renamed "First Person," started looking for a new market for this technology. They decided that Java technology was a perfect fit for the Internet, which was in its infancy at this time. Thanks to the NCSA Mosaic browser, which was published in 1993, the Internet was becoming a popular medium for text, graphics, and video using a network of heterogeneous devices. This was a perfect environment for Java since Java had been designed not only to move media from place to place but also to provide the logic to control the devices in a way that HTML could not do. The phrase that has been used to describe the philosophy of the Java platform is, "Write once, run anywhere."

The development team renamed this new technology Java in honor of the many cups of coffee they consumed during the long hours of development and offered it to developers on the Internet for beta testing. Downloads of the product soared and the rest, as they say, is history. Java has been the fastest growing new software technology in the history of computing and today is the primary development language for Internet-based applications.

B. What Is Java?

Java is a general-purpose programming language that can be used to write general-purpose programs. These programs can run as stand-alone applications on the computer or other digital device, or they can be written to run within a Web browser. A Java program that runs within a Web browser is called a Java applet. In either case, the program is written using a text editor, then compiled into Java byte code using the Java compiler and finally run by either commanding the Java runtime interpreter to execute the program or by creating a file that, when loaded into a Web browser, calls the java runtime interpreter and executes the program within the browser.

The are roughly five paradigms for general-purpose programming languages: very low-level languages (e.g., Assembly Language), imperative languages (e.g., Fortran), structured imperative languages (e.g., Pascal, C), object-oriented languages (e.g., Smalltalk, C++), and functional languages (e.g., ML, Haskell). Java is a synthesis of two existing object-

oriented languages: C++ and Smalltalk. Java's syntax is very similar to C++, but is tilted in the direction of Smalltalk: it is more object-oriented in that everything is embedded within objects/classes; it has a comprehensive class library; methods are polymorphic by default; and it is cleaner, smaller, safer, and slightly higher level than C++. All these characteristics make programming easier by sacrificing a bit of execution efficiency. Java programmers are less burdened with the details of the machine their program is running on, and the Java platform allows a program to run on computers running different operating systems such as Windows, Linux, or UNIX as well as different hardware, from cell phones to appliances to workstations, servers, and large mainframes. This makes Java the current leader in universal portability ("Write Once, Run Anywhere").

While a text editor can always be used to create Java programs, the process is frequently made easier with the use of a Java IDE (Integrated Development Environment). Java comes with many packages to make such things as GUI creation, communication over the Internet using standard protocols, and XML usage easy and reliable, but it is difficult for the developer to recall the names and usage details for these packages. An IDE automates the tedious coding required for these tasks and allows the developer to concentrate on the core logic of the application. An IDE also integrates code development, testing, and debugging to easily locate the parts of the code that are misbehaving and determine the specific problems. Some Java IDEs that are currently available are JBuilder by Borland, Forte for Java by Sun Microsystems, and VisualAge for Java by IBM.

II. JAVA BASICS

A. Obtaining, Installing, and Using Java

The primary repository for Java compilers and JVMs is Sun's Java homepage at http://java.sun.com. The software development kits can be downloaded from here and installation instructions can be found at this site. The Java Software Development Kit (SDK) consists of the Java compiler (`javac`), the runtime interpreter (`java`), and many packages for such things as GUI development, network communication, and database connectivity. After following the installation instructions, the two commands `javac` and `java` will be available to compile and run a Java program.

A short Java program that simply prints the text "Hello" is shown in Fig. 1.

After entering this program using a text editor and saving it in a file named `Hello.java`, one can compile it using the command

```
> javac Hello.java
```

Note that the filename before the `.java` extension is the same as the name of the class that constitutes the main program. Java will return an error if the name of the file and the name of the program do not match exactly. After compilation, a new file, `Hello.class`, will exist in the current directory; the Java interpreter (`java`) can be invoked to run the program using the command

```
> java Hello
```

Note that the parameter to the java command does not include the `.class` extension. This executes the main method of `Hello` and produces the output

```
Hello
```

B. Objects and Classes

Because of Java's strong object-orientation, it is appropriate to introduce the language by defining objects. An *object* holds data and exhibits behavior. At any point in time, the current data content constitutes the *state* of the object. The data is stored in the *attributes* (or fields) of the object. Behavior is prescribed through the object's constructors and methods. A *constructor* is code that is used to initially create an object, while *methods* are code that is used to query or update the object's state. Query methods return

```
public class Hello
{
    public static void main (String [] args) {
        System.out.println ("Hello");
    }; // main
}; // Hello class
```

Figure 1 A simple Java program.

values computed from the object's state, but do not modify the object's state. Updating methods do not have this restriction and actually change the object's state. In a program, many instances of objects of a given type can be created and exist simultaneously. Constructors and methods are shared by all objects of the same type, while state information is unique to each object.

The type of an object is specified in a *class definition.* The class definition gives the name of the class as well as its fields, constructors, and methods. The example given in Fig. 2 for a `Matrix` class has three fields, two constructors, and four methods. The last method, `main`, is special in two ways: First, it is *static,* meaning that an object instance is not used to invoke main—only the existence of the class is required. Second, because of its special signature, `main (String [] args)`, it can be invoked from the command-line; i.e., it defines the main program that is executed by the Java Virtual Machine.

The modifiers public, protected, and private control access to the fields, constructors, and methods in an object. Briefly, *public* means that any other object may access it. The *private* restricts access to other objects in the same class, while *protected* relaxes private

```
public class Matrix
{
    public final int rows;
    public final int cols;
    private final double [][] value;

    public Matrix (int _rows, _int cols) {
         rows = _rows;
         cols = _cols;
         value = new double [rows][cols];
    }; // Matrix

    public Matrix (double [][] array) {
         rows = array.length;
         cols = array [0].length;
         value = array;
    }; // Matrix

    public Matrix add (Matrix other) {
         if (rows != other.rows || cols != other.cols) {
              System.err.println ("Matrix.add: failed due to incompatible sizes");
         }; // if
         Matrix result = new Matrix (rows,cols);
         for (int i = 0; i < rows; i++) {
              for (int j = 0; j < cols; j++) {
                   result.value [i][j] = value [i][j] + other.value [i][j];
              }; // for
         }; // for
         return result;
    }; // add

    public Matrix mul (Matrix other) {
         if (cols != other.rows) {
              System.err.println ("Matrix. mul: failed due to incompatible sizes");
         }; // if
         Matrix result = new Matrix (rows,other.cols);
         for (int i = 0; i < rows; i++) {
              for (int j = 0; j < other.cols; j++) {
                   double sum = 0.0;
```

Figure 2 A Java `Matrix` class.

```
                for (int k = 0; k < cols; k++) {
                    sum += value [i][k] * other.value [k][j];
                }; // for
                result.value [i][j] = sum;
            }; // for
        }; // for
        return result;
    }; // mul

    public void print (String name) {
        System.out.print (name + ":");
        for (int i = 0 < rows; i++) {
            System.out.print ("\t[");
            for (int j = 0; j < cols; j++) {
                System.out.print (value [i][j] + " ");
            }; // for
            System.out.println (" ]");
        }; // for
    }; // print

    public static void main (String [] args) {
        Matrix a = new Matrix (new double [][] {{ 1, 2, 3}, {4, 5, 6}, {7, 8, 9}} );
        Matrix b = new Matrix (new double [][] {{ 9, 8, 7}, {6, 5, 4}, {3, 2, 1}} );
        Matrix c = a.add (b)
        Matrix d = a.mul (b);
        a.print ("a"); b.print ("b"); c.print ("c"); d.print ("d");
    }; // main

}; // Matrix class
```

Figure 2 Continued

by also granting access to objects in any extended or child class. An *extended* or *child* class is any class whose objects inherit properties and methods from another class known as the parent class. If no access modifier is given, the default is that any object from the same package is granted access.

1. Java Statements

A method or constructor body is made up of statements. From the Matrix example it is apparent that the statement syntax is almost the same as C or C++. Java provides the following statement types: expression, assignment, block, if, switch, while, do-while, for, break, continue, and try-catch. All statements in Java must be terminated by a semicolon.

Expression Statement. The simplest statement is an expression. An expression applies operators to variables and constants to compute values. For example, an expression can increment the value of a variable. An expression may include a call to a method, and normally, when an expression is used as a statement, it is a call to a method to execute the statements in that method. If the method returns a value, it can be chained into a complex expression; otherwise, a (void) method may be called by itself.

```
<expression> ;
```

Assignment Statement. Often, a value computed by an expression is saved for later use in a variable using the assignment operator (=).

```
<variable> = <expression> ;
```

Block Statement. A sequence of statements can be grouped together by enclosing them in braces to form a block statement. A block statement can be used anywhere that a statement can be used.

```
{ <statement> ... <statement> }
```

If Statement. Branching statements allow conditional execution of statements, and thus can change the order in which statements are executed. The most commonly used branching statement is the if statement, which

comes in two forms as shown below. If the boolean-expression evaluates to true, the statement is executed. In the second form, the statement after else is executed when the boolean-expression evaluates to false.

```
if (<boolean-expression>) <statement>

if (<boolean-expression>) <statement>
  else <statement>
```

Switch Statement. A branching statement that facilitates multiway branching is the switch statement. A value is first computed for the integer-expression. This value is then used to match a case, causing that case's statement to be executed.

```
switch (<integer-expression>)
{ case <integer-value> : <statement>
  case <integer-value> : <statement> ...
  default <statement>
}
```

While Statement. Java provides looping capabilities with three types of statements. The while loop will cause the statement(s) in the loop to be executed repeatedly, as long as the boolean-expression evaluates to true. This is a pretest loop because the boolean-expression is evaluated before the statement in the loop is executed.

```
while (<boolean-expression>)
  <statement>
```

Do-While Statement. The do-while statement implements a post-test loop by executing the statement in the loop before evaluating the boolean expression.

```
do <statement>
while (<boolean-expression>)
```

For Statement. Java provides the same general-purpose for loop as C. In the for statement, the initializer statements initialize one or more variables that are used to control the loop execution. The boolean expression tests for the end of the loop. The loop will be executed until the boolean-expression evaluates to false. The incrementer specifies how the loop control variable(s) will be incremented after each iteration of the loop.

```
for (<initializer> ;
<boolean-expression> ; <incrementer>)
  <statement>
```

Break Statement. This statement is used to break out of a loop from its middle.

```
break;
```

Continue Statement. This less commonly used statement is like a break statement, but rather than exiting the loop, it starts the next iteration.

```
continue;
```

Try-Catch Statement. Java provides facilities for catching errors that may happen during program execution and gracefully recovering from them. When an error has occurred, an exception is said to have been thrown. The try-catch statement provides a way to separate the regular code from the code provided to handle the errors. In this statement, exception is the name of a specific exception that can be thrown by the code within the try block. There can be multiple catch phrases in this statement.

```
try <statement>
    catch (<exception> <variable> )
    <statement>
```

C. Packages: Collections of Classes

Nontrivial Java programs use many classes. Some of these are included in the Java source. Others are provided with the Java runtime environment or by another provider of code. It is handy to have a way of organizing classes that avoids the difficulties of having to uniquely name each class. Java provides this capability in packages. A *package* is a collection of classes that normally have related functionality. For example, a package might be a collection of classes for representing and manipulating lists.

Although it is possible to write a class that does not use or interact with other classes, it would be of little use. In the `Matrix` example above, the class called `System` was used to print the results. `System` is part of the standard library which is in the most commonly used package (`java.lang`). If a package is required, it must be explicitly imported using the `import` directive. Since `System` is used so frequently, it is automatically imported.

To illustrate the use of packages and how objects interact we will extend the previous example. The `Matrix` class can be put into a package called `uga.linalgebra` by adding the line

```
package uga.linalgebra;
```

Now we will create a second package containing a second class. Ordinarily, several classes would be grouped into one package based on related functionality. The `MoveablePoint` class in Fig. 3 will use the `Matrix` class to move points around in a coordinate space. The import statement is required for this class to use the `Matrix` class.

```
package uga.animation;

import uga.linalgebra.*;
public class MoveablePoint
{
    private Matrix loc;

public MoveablePoint (Matrix _loc) {
    if (loc.rows != 2 || loc.cols != 1) {
        System.err.println ("MoveablePoint.MoveablePoint: failed due to incompatible sizes");
    }; // if
    loc = _loc;
    print ("loc");
}; // MoveablePoint

public void move (Matrix translate) {
    if (translate.rows !=2 || translate.cols != 2) {
        System.err.println ("MoveablePoint.move: failed due to incompatible size");
    }: // if
    loc = translate.mul (loc);
}; // add

public void print (String name) {
    loc.print (name);
}; // print

public static void main (String [] args) {
    MoveablePoint p = new MovablePoint (new Matrix ( new double [][] {{10}, {100}} ) );
    Matrix        m = new Matrix ( new double [][] {{1.1, 0}, {0,0.9}} );
    m.print ("m");
    for (int i = 0; i < 12; i++) {
        p.move (m);
        p.print ("p" + i);
    }; // for
    }; // main
}; // MoveablePoint class
```

Figure 3 Moveable point class.

The relationship between classes can take on several forms:

1. Generalization/Specialization (is-a relationship). A class extends another class. In this case, objects in the new class possess all of the properties and behavior of the original class, plus additional properties and methods.
2. Aggregation (has-a relationship). A class has a field whose type is that of another class. Here, one or more properties of a new class are defined by another class.
3. Usage (uses relationship). A class may access objects from another class via instance variables or parameters. One class manipulates or otherwise uses objects from another class.

In the case of the MoveablePoint class, has-a and uses relationships are represented. One could create a class that extends MoveablePoint, for example, by allowing the color of the point to be changed as

```
public class FlexPoint extends
  MoveablePoint { . . . }
```

III. CONCURRENT PROGRAMMING WITH JAVA

Unlike most general-purpose programming languages, Java has built-in support for concurrent programming through constructs in the language and through the standard library. Java allows a program to be divided into multiple threads that run at the same

time. Each thread can be assigned to a separate processor so they can truly execute in parallel. If there are more threads than processors, threads can share a processor via interleaved execution.

A. Threads

For simple cases, all that is required to make a program concurrent is to use the `Thread` class found in the `java.lang` package. We will illustrate this by writing a new class called `MovingDot` that creates a thread for each moveable point. This class is shown in Fig. 4. Over time, each moving dot will adjust its coordinates to indicate motion.

These moving dots (or balls) can be displayed in a graphical user interface (GUI) by using packages built into Java. Java provides two packages for this purpose: `java.awt` and `javax.swing`. The `javax.swing` package augments the `java.awt` package with additional capabilities. The `Animator` class shown in Fig. 5 will pop up a window as a swing `JFrame` and show the dots as they move through the frame from left to right.

Since the `Animator` class implements the Runnable interface, its run method can be executed by the `displayer` thread. This thread will (1) adjust the coordinates of the ball (in red), (2) sleep for 500 ms, and (3) paint the frame again (`repaint` tells the system to call `paint`). The dot in blue is able to move itself since it has its own thread.

```
package uga.animation;

import java.awt.*;
import java.util.*;
import uga.linalgebra.*;

public class MovingDot extends Thread
{
     private static final Random rn = new Random ();
     private final MoveablePoint dot;
     private final String   nam;
     private final Matrix   map;
     public MovingDot (MoveablePoint_dot, String_nam, Matrix_map) {
          dot = _dot;
          nam = _nam;
          map = _map;
     }; // MovingDot

     public void run () {
          for (int i = 0; i < 12; i++) {
               dot.move (map);
               dot.print (nam);
               try {
                    sleep ( (int) (100 * (1 + rn.nextDouble ())) );
               } catch (Exception ex) {
                    System.err.println ("MovingDot.run: sleep failed");
               }; // try
          }; // for
     }; // run

     public static void main (String [] args) {
          Matrix map   = new Matrix (new double [] [] {{1.1, 0}, {0, 0.9}} );
          MovingDot d1 = new MovingDot (new MoveablePoint (new Matrix ( new double [] [] {{10}, {90}}
                         new String ("dot1"), map);
          MovingDot d2 = new MovingDot (new MoveablePoint (new Matrix (new double [] [] {{25}, {75}}
                         new String ("dot2), map);
          MovingDot d3 = new MovingDot (new MoveablePoint (new Matrix (new double [] [] {{50}, {50}}
                         new String ("dot3"), map);

          d1.start (); d2.start (); d3.start ();
     }; // main

}; // MovingDot class
```

Figure 4 Movingdot class.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Animator extends Jframe implements Runnable
{
    static final Dimension dim = new Dimension (600, 350);
    final Thread displayer
    final Point ball;
    final MovingDot dot;

    public Animator (String title) {
       super (title);
       addWindowListener (new WindowAdapter () {
           public void windowClosing (WindowEvent e) {System.exit (0); }
       });
       ball = new Point (10, 10);
       dot  = new MovingDot (new MoveablePoint (new Matrix (new double [][] {{30}, {300}} )),
                             new String ("dot"),
                             new Matrix (new double [][] {{1.15, 0}, {0, 0.95}} ) );
       setLocation (100, 100)
       setSize (dim);
       setVisible (true);
       dot.start ();
       displayer = new Thread (this,"displayer");
       displayer.start ();
    }; // Animator

    public void paint (Graphics gr) {
       gr.setColor (Color.red);
       gr.fillOval (ball.x, ball.y, 10, 10);
       gr.setColor (Color.blue);
       Point dotp = dot.getPos ();
       gr.fillOval (dotp.x, dotp.y, 10, 10);
    }; // paint

    public void run () {
       for ( ; ball.x < 500; ball.x += 10) {
           ball.y = 50 + (int) ((ball.x - 250) * (ball.x - 250)) / 250;
           //System.out.println (" (" + ball.x + " , " + ball.y + ") ");
           try {
               displayer.sleep (500);
           } catch (InterruptedException ex) {
               System.err.println ("Animator.run: sleep failed");
           }; // try
           repaint ();
       }; // for

    };// run

    public static void main (String [] args) {
       Animator frame = new Animator ("Animator");
    }; // main
}; // Animator class
```

Figure 5 Animator class.

Multithreaded programming is necessary anytime one creates a program for displaying animations. More importantly, multithreading may provide for more efficient program execution. Some computations can be sped up executing parts of them in parallel. This is particularly the case for servers which provide services to multiple clients. Typically, a server will spawn a thread for each incoming request from a client. In this way, one client need not wait for another client's request to complete.

B. Synchronization

Multithreading also introduces a fair amount of complexity. If two threads modify a shared variable, unless the access is controlled or synchronized the final value of the variable is indeterminate. The simplest way to prevent this problem is to make the methods that access shared variables synchronized.

```
synchronized void increment (int delta)
synchronized void decrement (int delta)
```

Now, for a given object, only one of these methods may be run at a time. Additional facilities for synchronizing multiple threads include synchronized statements, the complementary `wait`, and `notifyall/notify` methods, as well as the `yield`, `join`, and `interrupt` methods.

IV. NETWORK PROGRAMMING WITH JAVA

Although Java makes an excellent conventional programming language and even adds built-in high-level features for concurrent programming, its real forte is network programming. Java provides extensive libraries devoted to network programming. For instance, of the 44 packages that make up the current (JDK 1.4) standard `java.*` core library, 17 packages are for network programming (`java.applet.*`, `java.net.*`, `java.nio.*`, `java.rmi.*`, `java.security.*`). In the subsections below, three of these packages will be discussed, as well as a related one from `javax`.

1. Applets (`java.applet`) allow small Java applications to be downloaded from the Web and executed on a Web client.
2. Sockets (`java.net`) provide the foundational networking capabilities for Java.
3. Distributed objects (`java.rmi`) can communicate through remote method invocations which allow development of distributed applications at a high level.
4. Servlets (`javax.servlet`) provide a simple way for executing Web services by executing Java on a Web server and delivering output to a browser.

A. Web Clients: Applets

The simplest form of distributed programming in Java involves running a "small" application or applet (Fig. 6). The applet is typically executed by the JVM embedded in a Web browser (e.g., Netscape Navigator or Microsoft Internet Explorer).

Writing code for a Java applet is very simple. Minimally, all that is required is to create a class that extends Applet and implements a method called `start`. The `AniApplet` class is a simple example that invokes `Animator.main`.

```
import java.applet.*;

public class AniApplet extends Applet
{

    public void start () {
      System.out.println
      ("Start animation from applet");
      Animator.main (null);
      super.start ();
    }; // start

}; // AniApplet class
```

The code for the applet may be located anywhere on the Web and is found by giving the URL of a Web page referencing the applet.

For example, the following Web page references the AniApplet applet.

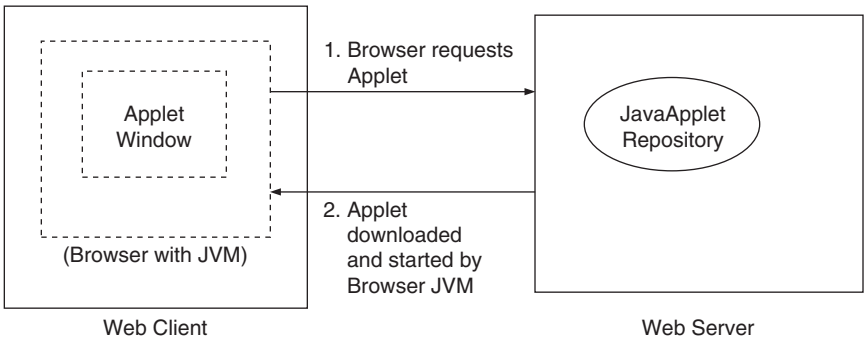


Figure 6 Java applets.

```
http://chief.cs.uga.edu/~jam/jsim/jch
  apter/aniapplet.html
```

Pointing the browser at this page will display the contents of this page as well as trigger the execution of AniApplet. This Web page is just like any other HTML page, except that it contains an <applet> tag. The applet tag indicates which Java .class file to load and execute. It may also indicate dimensions for the applet window as well as other information.

```
<HTML>
<BODY>
    <H1> Animation Applet </H1>
    <APPLET CODE = "AniApplet.class"
      HEIGHT = 50 WIDTH = 150>
    </APPLET>
</BODY>
</HTML>
```

In more complex situations, one would typically also implement an `init` and `stop` method. The `init` method initializes fields in the applet and is invoked when the applet is loaded into the JVM for execution. The `stop` method is invoked whenever the Web browser moves on to another page. Conversely, the `start` method is invoked whenever the browser goes to or returns to the triggering page.

B. Network Connections: Sockets

In Java, sockets provide low-level communication capabilities which enable Java programs to talk to each other. A socket is one endpoint of a bidirectional communication link between one Java program and another program. The program that initiates a communication is typically referred to as a client program, while the program that waits for incoming communication requests is typically referred to as a server program. Sockets are an innovation of Berkeley UNIX that allow a programmer to treat a network connection as just another stream into which bytes can be written and from which bytes can be read, and shield the programmer from the details of and variances between networking technologies.

The `java.net` package provides two important classes for network communication—`Socket` and `ServerSocket`. The description of these two types of sockets and how they are used is described below in the following scenario where many clients connect to a single server.

The server creates a `ServerSocket` object for a particular port of the machine it is running on and waits on that port for client connections. Invocation of the `accept` method will cause the server to wait for incoming requests from clients on the given port.

A client which needs to connect to the server creates a `Socket` object with the `host name` and `port number` on which the Server is listening, as parameters. The successful creation of this `Socket` object indicates the establishment of the connecion between a client and the server. From this point onwards the client can use this `Socket` object to send/receive messages to/from the server.

As far as the server is concerned, once it gets a connection request from a client, the accept method on which it was waiting returns a normal `Socket` object. This Socket object is used by the Server to send/receive message to/from that specific client. Thus, the server maintains a `Socket` object for each client that connects to it. This helps it to maintain sessions and aids record keeping.

Figures 7 and 8 provide the server and client portions of an example that illustrates how a client-server application is implemented in Java. It has two programs: `EchoServer` and `EchoClient`. This example shows how to establish connection between two java progams through a network. Here `EchoServer` accepts strings from a single client and echos the string back to the client. This example could be extended to handle multiple simultaneous clients using the multithreading paradigm discussed earlier.

The EchoServer program creates a ServerSocket at port number 4444 and calls the `accept` method on this ServerSocket. It then waits until it gets a connection request from a client.

The EchoClient program requests a connection to the server via the `Socket` constructor call, passing "chief" and 4444 as parameters. Once the client connects to the server, the server initializes the `InputStream (BufferedReader)` and `OutputStream (PrintWriter)` objects for that client from the `Socket` object that was returned from the `accept` method. Then, the server enters a while loop in which it reads each line sent by the client echoing the line back to the client. The client's `OutputStream` and `DataInputStream` objects are used to send and receive messages from the server. The server comes out of its loop when the client enters the string "Bye.". Then it closes both the sockets.

C. Distributed Objects: Remote Method Invocation

Although low-level communication with sockets works fine, it is easier to develop distributed applications in a programming style that is closer to conventional programming. In object-oriented programming, coding often involves having one object invoke a method on

```
import java.net.*;
import java.io.*;

public class EchoServer
{
   public static void main (String [] args) throws IOException {
      String       inputLine;
      String       outputLine;
      ServerSocket serverSocket  = null;
      Socket       responseSocket = null;
      try {
           serverSocket = new ServerSocket (4444);
           System.out.println ("Server is up and running on 4444 for Client");
      } catch (IOException ex) {
           System.err.println ("EchoServer.main: unable to listen on port 4444. " + ex);
           System.exit (-1);
      }; // try
      try {
           reponseSocket = serverSocket.accept();
      } catch (IOException ex) {
           System.err.println ("EchoServer.main: accept failed. " + ex);
           System.exit (-2);
      }; // try
      PrintWriter out = new PrintWriter (reponseSocket.getOutputStream (), true);
      BufferedReader in = new BufferedReader (new InputStreamReader (
                                                reponseSocket.getInputStream ()));
      while ((inputLine = in.readLine ()) != null) {
           out.println ("Echoed: " + inputLine);
           if (inputLine.equals ("Bye.")) break;
      }; // while
      out.close ();
      in.close ();
      reponseSocket.close ();
      serverSocket.close ();
   }; // main
}; // EchoServer class
```

Figure 7 Server portion of socket example.

another object. A straightforward extension of this is to allow an object to invoke a method on a remote object. This capability is provided by Java using Remote Method Invocation (RMI) (Fig. 9).

A degree of independence between clients and servers is established by having them interact through an interface. The client creates a proxy object that allows methods in the interface to be invoked. The server must provide full implementations of the methods in the interface. If the client is local, the proxy can be the server object itself. If this is the case, then the client can directly invoke the server's method.

Figure 10 illustrates this: The interface called `ISearch` specifies a method to minimize a univariate function (i.e, find the value of x that minimizes $f(x)$). This interface has an associated abstract class called `Function` which allows clients to specify functional forms by extending their own classes from `Function`.

The code for the `Function` class appears in Fig. 11. This class must be distributed with the `ISearch` interface since the minimize function takes an argument of type `Function`.

Now, there are several algorithms for minimizing a univariate function, so that having the option of using one of several possible candidate servers would be useful. One such algorithm is the golden section search shown in Fig. 12.

The reader may be wondering why an example of a local client is given. The answer is that this example can be turned into a remote example by just making two single-line changes to the code (shown within /** . . . **/).

```
import java.io.*;
import java.net.*;

public class EchoClient
{
   public static void main (String [] args) {
      try {
         int     c;
         String  responseLine;
         Socket  echoSocket = new Socket ("chief", 4444);
         OutputStream     os = new OutputStream (echoSocket.getOutputStream ());
         DataInputSteam   is = new DataInputStream (echoSocket.getInputStream ());
         System.out.println ("Enter the string to be sent (Bye. for termination):");
         while ((c = System.in.read ()) != -1) {
            os.write ( (byte) c);
            if (c == '\n') {
               os.flush ();
               responseLine = is.readLine();
               System.out.println (responseLine);
               System.out.println ("Enter the string to be sent (Bye. for termination) :");
            }; // if
         }; // while
         os.close ();
         is.close ();
         echoSocket.close ();
      } catch (Exception ex) {
         System.err.println ("Exception: " + ex);
      }; // try
   }; // main
}; // EchoClient class
```

Figure 8 Client portion of socket example.

First, if the server is remote, the client must not construct the server object with `new GSection ()`. Indeed, the client will likely not even have access to the server code which is on another machine. Rather, it must find a server on another machine using a naming service (`Naming.lookup ("//chief.gssection")`). This will create a proxy that is capable of communicating with the remote server in order to execute the method. The client invokes the method on the proxy (`optimizer.minimize (f)`) which will facilitate the execution of the method within the remote server, obtain results back, and give them to the client. The methods in the proxy can be viewed as stubs, while the full implementations are in the server. The code (class definition) is automatically generated by executing the `rmic GSection` command which creates the following file: `GSection_Stub`.

Second, the server class must extend a class (e.g., `UnicastRemoteObject`) that allows it to act as a remote server. Also, the main method of the server must (i) replace the default security manager with an RMI security manager (`new RMISecurityManager ()`) and (ii) construct the server object (`new GSection ()`) which invokes a parent (`Unicast RemoteObject`) constructor.

On the issue of who is talking to whom, the client indicates the name of the server object (e.g., `gssection`) and identifies the machine it is running on (e.g., `chief` or `chief.cs.uga.edu` if not running on the same network). The server object names itself when registering with the RMI registry (`Naming.rebind ("//chief/gsserver", server)`). As indicated above, the client object uses the server object's name and and host machine's name to lookup the server object.

`(Naming.lookup ("//chief. gssection")).`

Codewise, this is all that is required. In order for the client to run, the server must already be running, unless the Java Activation Framework (JAF) is used. Before running the server, the RMI registry must be started. The following sequence of commands is therefore needed.

```
> rmiregistry &
> java -Djava.security.policy
```

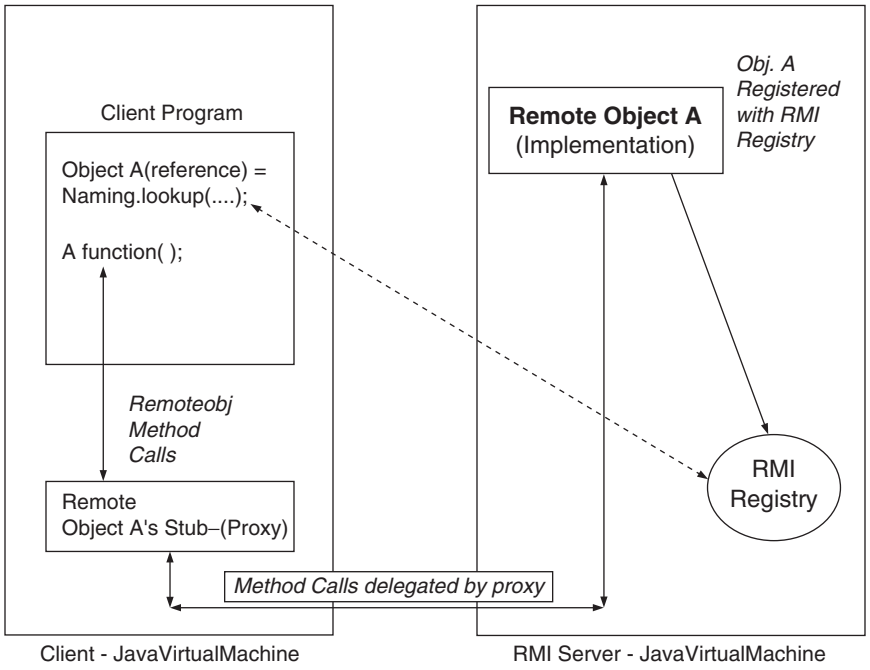


Figure 9 Java RMI architecture.

```
=java.policy GSection &
> java Client
```

The java command is passed an option to define the security policy. If this is left out, the security manager will not allow a connection to be made.

```
grant {
    permission java.net.Socket
      Permission "*:1024-65535",
      "connect,accept":
};
```

D. Web Services and Java Server Pages

While applets allow Java code to be executed on the client side, servlets allow Java code to be executed on the server side. Moreover as a servlet is coded as a Java class, it has full access to Java's advanced features like database connectivity, network awareness, object orientation, and support for multithreaded programming. Java Server Pages (JSPs) provide a convenient way to execute servlets.

1. Servlets

A servlet is executed as follows: A Web client such as a Web browser invokes a servlet when the user clicks a submit button on a Web page which includes a request to launch the servlet.

An example of an HTML page that launches a servlet is shown in Fig. 13.

In this example, the form tag specifies the action to be `/servlet/Authenicator` and the form inputs will be passed to the servlet as inputs. In this example, the contents of the username and password

```
import java.rmi.*;

public interface Isearch extends Remote
{
    double minimize (Function f) throws RemoteException;

}; // Isearch interface
```

Figure 10 Interface Isearch.

```
 import java.io.*;

public abstract class Function implements Serializable
{
   public final double xMin;
   public final double xMax;

   public abstract double eval (double x);

}; // Function class
```

The example code below represents a local client (i.e., the client invokes the server in the same JVM).

```
import java.io.*;
import java.rmi.*;

public class Client
{
   public static class Quad extends Function implements Serializable
   {
      public Quad (double x1, double x2) { xMin = x1; xMax = x2; };

      public double eval (double x) {
         return 5 * x * x - 20 * x + 40;
      }; // eval

   }; // Quad nested class

   public static void main (String [] args) {
      System.out.println ("Client started");
      Function f = new Quad (0, 100);
      try {
          ISearch optimizer = new Gsection ();
          /** = (ISearch) Naming.lookup ("//chief/gsserver"); **/
          System.out.println ("minimum found at " + optimizer.minimize (f));
      } catch (Exception ex) {
          System.err.println ("Client.main: " + ex);
      }; // try
   }; // main

}; // Client class
```

Figure 11 Function class code for RHI example.

textboxes on the HTML form are passed as input to the servlet `Authenticator`. The Web server invokes the servlet by passing the request to a servlet engine, which is normally installed on the same machine as the Web server. Some of the popular servlet engines are Apache Tomcat, Macromedia JRun, New Atlanta ServletExec, and Caucho Resin. Some Web servers have servlet engines built-in and therefore are able to run Java servlets out of the box. Others need a separate Java servlet engine to be installed. Apache JServ is a popular servlet engine that can be integrated into the Apache Web server. A JServ installation must include the JDK (Java Development Kit) and the JSDK2.3 (Java Servlet Development Kit) in addition to Apache Web server on the server machine. Once installed the JServ servlet engine will start automatically when the Apache Web Server is started and it will handle requests made to standard paths like `/servlet`.

Java servlets are written in Java and compiled to a Java class file before being installed on the server, and therefore must be executed within the JVM contained in the servlet engine. The job of a servlet engine is to intercept incoming HTTP servlet requests, execute the appropriate servlet bytecode in a JVM on the server, and send the output of the servlet back to the requesting Web client.

```
import java.rmi.*;
import java.rmi.server.*;

public class Gsection /** extends UnicastRemoteObject **/ implements ISearch
{
   private static final double RATIO    = 0.382;
   private static final double EPSILON = 0.001;

   public GSection () throws RemoteException {
      super ();
   }; // GAection

   public double minimize (Function f) throws RemoteException {
      double x1 = f.xMin;
      double x2 = f.xMax;
      double y1 = f.eval (x1);
      double y2 = f.eval (x2);
      double xa;
      double xb;
      double ya;
      double yb;
      double delta;
      while ( (delta = (x2 - x1) * RATIO) > EPSILON) {
          System.out.println (" x1 = " + x1 + " y1 = " + y1
                              + " x2 = " + x2 + " y2 = " + y2);
          xa = x1 + delta;
          xb = x2 - delta;
          ya = f.eval (xa);
          yb = f.eval (xb);
          if (y1 + yb < ya + y2) {
              x2 = xb; y2 = f.eval (x2);
          } else {
              x1 = xa; y1 = f.eval (x1);
          }; // if
      }; // while
      return (x1 + x2) / 2.0;
   }; // minimize

   public static void main (String [] args) {
      System.setSecurityManager (new RMISecurityManager ());
      try {
          ISearch server = new GSection ();
          Naming.rebind ("//chief/gsserver",server);
          System.out.println ("gsserver registered and waiting for requests");
      } catch (Exception ex) {
          System.err.println ("GSection.main: " + ex);
      }; // try

   }; // main

}; // GSection class
```

Figure 12 Golden section search.

```
<HTML><TITLE> Login </TITLE>
<BODY>
    <H3> Welcome, please enter your name and password to login </H3>
    <FORM ACTION = "/servlet/Authenticator" METHOD = get>
    <CENTER>
        Name: <INPUT TYPE = text NAME = "name" > <BR>
        Password: <INPUT TYPE = password NAME = "password" > <BR>
        <INPUT TYPE = submit VALUE = "enter" > <BR>
    </CENTER>
    </FORM>
</BODY>
</HTML>
```

Figure 13 HTML page to launch a servlet.

From the application creator's perspective, all that needs to be done to use a servlet is to create an HTML page (or other means) to invoke the servlet residing in the Web server at a particular location.

Figure 14 illustrates the main components involved in executing servlets.

2. Implementation Details of a Servlet

Two packages make up the Application Programming Interface (API) for servlets: `javax.servlet.*` and `javax.servlet.http.*`. The `javax.servlet` package contains classes that support generic, protocol-independent servlets. These classes are extended by the classes in the `javax.servlet.http` package to add HTTP specific functionality. Every Servlet should implement the `javax.servlet.Servlet` interface. Most servlets implement it by extending one of the two special classes: `javax.servlet.GenericServlet` or `javax.servlet.http.HttpServlet`. An HTTP servlet should subclass `javax.servlet.http.HttpServlet` in order to provide HTTP-specific functionality.

Unlike a regular Java Program, but like an applet, a servlet does not have a `main()` method. Instead, certain methods in the servlet are invoked by the servlet engine in the process of handling requests. For generic servlets, the servlet engine invokes the servlet's `service()` method, which should be overridden and include code to carry out the actions of the servlet. The `service` method accepts two parameters: a request object and a response object. The request object provides the servlet with input data for the request, while the response object is used to return a response from the servlet.

HTTP servlets usually do not override the `service()` method. Instead, they override the `doGet()` method to handle HTTP GET requests and the `doPost()` method to handle HTTP POST requests. The `service()` method of the `javax.servlet.http.HttpServlet` class handles the setup and invokes the appropriate `doGet()` and/or `doPost()` methods. Figure 15 illustrates how an HTTP servlet handles GET and POST requests.

Coding an HttpServlet class involves coding three types of methods, corresponding to the servlet's lifecycle:

1. Each servlet class can define an `init` method. The `init` method can be called when the server starts, when the servlet is first requested, or when the server administrator requests. It is used to

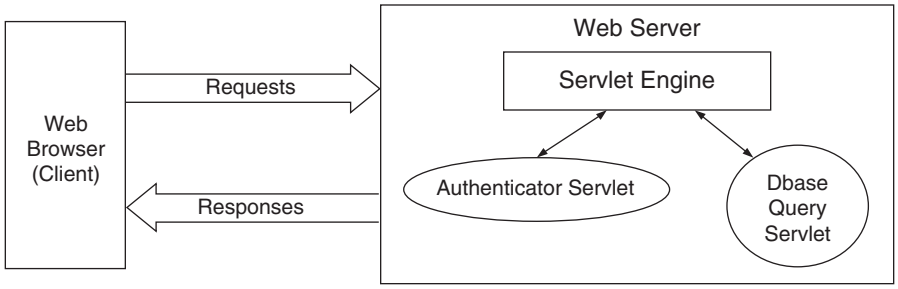


Figure 14 Java servlets managed by a servlet engine.

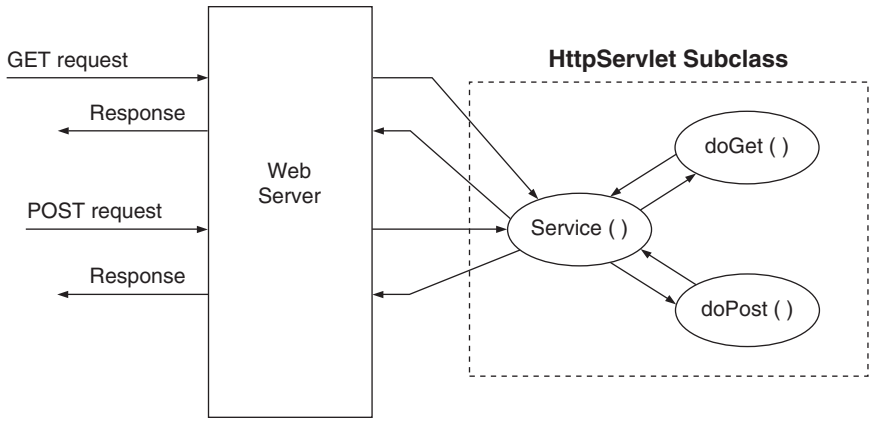


Figure 15 An Http servlet handling GET and POST requests.

perform servlet initialization, like creating or loading objects, that might be required by the servlet to handle requests. The `ServletConfig` object supplies the servlet with information about the parameters to `init`.

2. Each `HttpServlet` class normally overrides the default `doGet()` and `doPost()` methods to make the servlet perform the user specified tasks for GET and POST requests, respectively. Other methods which may be overridden in an HttpServlet are `doDelete()`, `doOptions()`, `doPut()`, and `doTrace()` to handle DELETE, OPTIONS, PUT, and TRACE Http requests, respectively.
3. For cleanup purposes, a servlet may have a `destroy` method. This method is called by the servlet engine just before the servlet is unloaded to clean up and release any resources allocated in the `init()` method.

The Java code for the Authenticator servlet is given in Fig. 16. This servlet gets the username and password from the Web client through the HTML form and uses it to authenticate the user.

3. Java Server Pages

JSPs provide a more convenient way to access and execute servlets. Servlets can be tedious to use because much of the code in servlets is used to output static HTML. Much like PHP and ASP, JSPs allow the developer to mix Java with HTML by putting Java into special tags `<%` and `%>`.

The JSP translator turns the Java Server Page into a servlet automatically as the page is processed and executes the servlet in a servlet container, returning the output to the Web server. The important advantage of JSPs is that the JSP processor handles the tedious work of keeping track of the bytecode and associating it with the page. Since the code must be compiled the first time the translator sees it, the page will experience a delay the first time it is requested. However, subsequent requests will display very quickly since the servlet does not need to be recreated and the Java code does not need to be recompiled.

Figure 17 is a Java Server Page that performs a similar function to the Authenticator servlet just presented.

This application assumes that a JavaBean `AuthenticatorBean` in a package `auth` is in another file and has been compiled and provides method `resetAuth` and property `authCount`. The important thing to note about this example is that one does not need to write output statements to create the final HTML that will be displayed. Instead, the Java in the JSP tags will select the portions of the HTML to be displayed.

V. ENTERPRISE APPLICATIONS WITH JAVA

Java was originally created to communicate with a diverse group of remote devices. However, with the development of the Internet, Java developers realized that it is also useful for distributed enterprise applications. As a result, the language has been extended through APIs and additional packages to support multitier distributed applications. The two most important extensions are the Java 2 Platform, Enterprise Edition (J2EE) and JINI.

J2EE is a collection of APIs designed to let developers create multitier applications that support many users in an environment that has diverse clients on one end (the client tier), one or more databases on the backend, and application servers between them (the middle

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.http.*;

public class Authenticator extends HttpServlet
{
    private HashMap users = new HashMap ();

    public void init (ServletConfig conf) throws ServletException {
        super.init (conf);
        users.put ("Smith","teetoteler");
        users.put ("Mark","pilot");
        users.put ("Waugh","soccer");
    }; // init

    public void doGet (HttpServletRequest req, HttpServletResponse res)
                       throws ServletException, IOException {
        res.setContentType ("text/html");
        PrintWriter out = res.getWriter ();
        String name     = req.getParameter ("name");
        String password = req.getParameter ("password");

        if ( ! userCheck (name, password)) {
            out.println ("<HTML><HEAD><TITLE> Permission Denied </TITLE></HEAD>\n"
                       + "<BODY><H3> YOU HAVE ENTERED AN INVALID LOGIN OR PASSWORD </H3>\n"
                       + "Try reentering <A HREF=\"/login.html\"> Try again </A>\n"
                       + "</BODY></HTML>");
        } else {
            out.println ("<HTML><HEAD><TITLE> Access Granted </TITLE></HEAD>\n"
                   + "<BODY><H3> Here is the link to your destination </H3>\n"
                   + "</H4><A HREF=\"/destination.html\"> Destination Page Link </A>\n"
                         + "</BODY></HTML>");
        }; // if
    }; // doGet

    public boolean userCheck (String name, String password) {
        return password.equals ((String) users.get (name));
    }; // userCheck

}; // Authenticator class
```

Figure 16 Authenticator Servlet code.

tier). The design goals of J2EE are to create applications that are robust, secure, and scalable and that consist of reusable components. This section will discuss the components of J2EE and explore how these components work together to create a platform to realize these goals.

An example will illustrate the type of enterprise application Java is designed to support. Consider a health care provider that operates hospitals and medical clinics. When a patient enters the emergency department, a clerk gets the patient's name and uses a browser-based client to enter this data to search the provider's database for an entry. If none is found, the same client displays a form for the clerk to enter the patient's name, address, insurance, and other information. If the patient has a record, the data are displayed in order to be verified and updated. Using the same enterprise application, the triage nurse can view the patient's previous medical history and enter the results of the current examination. This information is also available to the emergency department physician, and a subset is available to the technicians in imaging and the laboratory. The patient's personal physician can access the patient's medical records online from his or her office, and the clinic's billing

```
<%@ page import = "auth.AuthenticatorBean" %>

<jsp:useBean id="authenticator"class="AuthenticatorBean" scope="session" />
<jsp:setProperty name="authenticator" property="*" />

<html>
<head>
   <title> Authentication </title>
</head>
<body>

<% if (authenticator.getSuccess()) {

   authenticator.resetAuth(); %>

   <h2>Access Granted.</h2>
   You may <a href="login.html"> login </a>.

<% } else if (authenticator.authCount == 0) { %>

   <h2> Authentication Page: Please enter your username and password. </h2>
   <form method=get>
   Username: <input type=text name=username><br>
   Password: <input type=text name=password><br>
   <input type="submit" value="Submit" >
   </form>
<% } else { %>

   <h3>YOU HAVE ENTERED AN INVALID USERNAME OR PASSWORD</h3>
   <center> Please try again. </center>

   <form method=get>
   Username: <input type=text name=username><br>
   Password: <input type=text name=password><br>
   <input type="submit" value="Submit">
   </form>
<% } %>

</body>
</html>
```

Figure 17 Java Server Page for Authenticator Servlet.

department can create an insurance claim and submit it online using the same application. Within the clinic, various wireless devices can be used to provide data to and display data from the patient's records, including graphics such as MRI images.

This example illustrates a system where there are multiple users with differing roles who will access the data in the provider's information system using Web-based clients, usually browsers. All data are stored in an enterprise information system (EIS). Access to the data is restricted by the user's authentication on the services available and the presentation of data depends upon the user's role. For example, a clerk in the billing department can only view the patient's name, address, dates of service, and insurance information. These data are displayed in a format appropriate to create a bill for services. On the other hand, a physician can view the entire medical history and the data are presented in a form that color-codes the various items. Additional services are available to physicians to look up drug data in a drug database and compute dosages. Since the system is available over the provider's intranet and extranet, it is available to users regardless of location.

Applications like this can be created using many current platforms. J2EE provides a selection of tech-

nologies that have implemented the features that are common to all enterprise applications, such as roles and authentication, HTML and XML processing, and database connectivity. Thus, the developer can implement and customize these services. J2EE also includes facilities for managing objects so the user can build or purchase reusable objects. The goal is to minimize the amount of code that must be written and maximize productivity. This is achieved in part by separating the business logic from the presentation, thus letting the developers concentrate on coding just the business logic and designers focus on designing the user interface.

The core components used in J2EE are:

1. JDBC
2. JavaBeans
3. Enterprise JavaBeans (EJB)
4. Servlets
5. JavaServer Pages (JSP)

Servlets and JavaServer Pages were discussed in the previous section. JDBC, JavaBeans, and Enterprise JavaBeans will be discussed here.

A. JDBC

JDBC has been a component of the Java platform since very early in its development and is not, strictly speaking, a component of J2EE. However, JDBC is a necessary part of any enterprise application. JDBC is an API that provides a standard for connecting to databases from Java applications. Using JDBC, the developer can access data from any supported database by calling standard methods to connect to the database server, authenticate to the desired database and query or update the database tables. Since the API is standard, a change of the database can be effected by simply changing the JDBC driver, without changing the Java code.

B. JavaBeans

One of the design concepts of Java from the beginning has been that the code be componentized and reusable. Objects allow the code to be componentized, but more is needed for reusable components to be developed. In particular, an application must be able to query an object to determine what that object's capabilities are, i.e., to discover the object's properties and methods. JavaBeans provides a set of standards that allow Java objects to query one another and perform this process of discovery. Once objects have been written as JavaBeans, they can be managed using visual development tools such as the Bean Development Kit (BDK) from Sun or similar features in the various Java IDEs from Borland, IBM, and Sun. A JavaBean can be used simply by dragging and dropping it onto the application. The application then can change the bean's properties and invoke its methods without having to write explicit code.

C. Enterprise JavaBeans

It is unfortunate that Enterprise JavaBeans (EJBs) are called JavaBeans because they actually have little in common with (regular) JavaBeans. That is, Enterprise JavaBeans are not an extension or specialized version of JavaBeans. Rather, EJBs provide a means to distribute components of an enterprise application to several computers and provide an object-relational mapping between Java objects and rows of a database. Enterprise Java Beans (EJB) represent the heart of the Java 2 Enterprise Edition (J2EE). Its principal use is in developing three-tier information technology applications in which the top tier is usually a graphical user interface (GUI), while the bottom tier is usually a data source such as a database system. J2EE provides an application server which forms the middle tier which makes it easier for GUI clients to access and integrate multiple data sources.

EJBs can be stored in and retrieved from a repository. This allows the developer to create general-purpose objects and use them as plug-in components in an application. The objects can be stored on multiple servers, and the enterprise application can query the servers to discover the location of the ojects.

The EJB container also has the ability to handle the mapping between objects and rows in database tables. Thus, the programmer defines the database tables and the objects that correspond to them, but the EJB takes care of retrieving and storing the data. EJBs can also give objects transactional characteristics, allowing multiple actions to be performed as a group.

Part of the design philosophy behind Java has been to provide a standard platform for software design. The APIs are open and standard, so all developers can provide components that will work across a broad range of platforms. The EJB APIs are standard across application servers, so a developer can migrate an application from one application server to another without having to modify the code.

D. JINI

JINI is a technology that uses the Java platform to create applications that can communicate with one another to set up services on the Internet or an organization's intranet. The core idea in JINI is that of a *federation*. A federation is a collection of objects that work together to achieve a defined goal. More information about JINI can be found at http://java. sun.com.

E. Package Information

Currently `java.applet`, `java.net`, `java.rmi` and `org.omg` are distributed with Java 2 Platform, Standard Edition, v. 1.4 (J2SE 1.4) SDK, while `javax.servlet` and `javax.ebj` are distributed with Java 2 Platform, Enterprise Edition, v. 1.3 (J2EE 1.3) SDK. These Software Development Kits (SDK) are available for download from http://java.sun.com/products. The Jini Technology Starter Kit (starter kit) and the Jini Technology Core Platform Compatibility Kit (TCK) are available for download from http://www.sun.com/jini.

SEE ALSO THE FOLLOWING ARTICLES

Internet Homepages • Internet, Overview • Javascript • Linux Operating System • Object-Oriented Databases • Object-Oriented Programming • Operating Systems • Programming Languages Classification • Unix Operating System • XML (Extensible Mark-up Language)

BIBLIOGRAPHY

Arnold, K., Gosling, J., and Holmes, D. (2000). *The Java programming language,* 3rd ed. Reading, MA: Addison–Wesley.

Bergsten, H. (2001). *JavaServer pages.* Sebastopol, CA: O'Reilly.

Campione, M., Walrath, K., and Huml, A. (2001) *The Java tutorial,* 3rd ed. Reading, MA: Addison–Wesley.

Flanagan, D. (1999). *Java foundation classes in a nutshell: A desktop quick reference.* Sebastapol, CA: O'Reilly.

Gittleman, A. (2002). *Advanced Java: Internet applications,* 2nd ed. El Granada, CA: Scott Jones.

Java. http://java.sun.com. Primary repository of Java development kits, Java runtime environments, advanced Java technologies, and documentation.

Reese, G. (2000). *Database programming with JDBC and Java,* 2nd ed. Sebastopol, CA: O'Reilly.

Shannon, B., *et. al.* (2000). *Java 2 platform, enterprise edition: Platform and component specifications.* Reading, MA: Addison–Wesley.

JavaScript

Blaine T. Garfolo
San Francisco State University

GLOSSARY

alert A message box presented by the browser to display information.
browser The software used to access the Internet. The most common browsers are Netscape Navigator and Microsoft Internet Explorer.
document The document object represents the Hypertext Markup Language (HTML) page. Document objects can be contained in windows, frames, or layers.
platform The operating system installed on a computer. The most common platforms on the Internet can be classified as flavors Windows, Unix, or MacOS.
prompt A message box presented by the browser that prompts the user for his or her input.
window The window object represents the browser window and is the highest level object available.

I. INTRODUCTION TO JAVASCRIPT

A. What Is JavaScript?

JavaScript is an object-based scripting language developed by Netscape (primarily Brendan Eich) for client and server applications. It was introduced with Netscape 2.0 and was intended to be a cross-platorm, client-side scripting language to be embedded directly into HTML documents. The core of the JavaScript language centers around the scripting language standardized by the European Computer Manufactures Association, ECMA-262.

There has been considerable confusion as to what JavaScript's relation is to the Java programming language. So before we go any further, let us clear that up right now. JavaScript is not a simplified version of Java and was never intended to be. In fact, it was originally called "LiveScript" and was part of Netscape's "LiveWire" Common Gateway Interface (CGI) alternative. The similarities in the name are pure marketing, as it was intended to capalitize on the popularity of the object-oriented programming language Java.

On the surface, the languages appear to be related. This is because both languages have adopted the majority of the C/C++ programming languages syntax and can both be employed to add executable content to a Web page. However, JavaScript is not a typed language. That is, a variable in JavaScript can hold a value of any data type (string, number, etc.). JavaScript's use of variables is situation dependent, converting a variable automatically from one type to another depending on the context of how it is used.

Fundamentally, JavaScript and Java are quite different in their execution approach. JavaScript is an interpreted language. When the browser loads a JavaScript page, the original source code is present on the page. The browser translates each line of code into machine language as it is loaded and stores it in the browser's memory. This translation is accomplished with the JavaScript interpreter located in the

user's browser. When a new page is needed, the old page is discarded and the browser reads and translates the next page into its memory. This would seem to be fairly straightforward and, in fact, is except for one small wrinkle: Microscoft has added a JavaScript interpreter to their Internet Explorer browser that is not quite 100% compatible with either Netscapes JavaScript or the ECMA standards. So, there is a good chance that the simple script written for one manufacturer's browser will not run on the others.

Java, by contrast, is a full-featured general programming language with extensive library support, graphics/display, and networking capabilities. Java is a compiled language (into what is called bytecodes) and the compiled form is only machine readable. Additionally, a Java program can read and write files in the local computer's file system, unlike JavaScript which has no access to local files. Mini programs written in Java, called "applets," can be executed inside of a Web browser. However, though they appear to be integrated, they actually are independent programs which are not a part of the HTML Web page and, in fact, have no access to the "surrounding" Web page. Finally, unlike JavaScript whose code can be viewed, because Java is a compiled language, all of the programmer's logic, algorithms, etc, are not viewable by anyone. Being compiled into machine-readable bytecodes protects Java. Now, as we can see, other than a clever marketing scheme, there are few real similarities between Java and JavaScript.

B. Client-Side/Server-Side JavaScript

Both client-side and server-side JavaScript share the same common core language as specified in the ECMA-262 document by the European standards body. However, additions to the core language have been added to enable the language to best function in its predetermined environment (client/server). Core JavaScript contains features such as variables, functions, core objects such as DATE and ARRAY, LiveConnect, statements, operators, and expressions. Client-side JavaScript then encompasses the core language and any additional predefined objects/functions necessary for running JavaScript in a browser.

Server-side JavaScript also has at its heart the core JavaScript lanugage. Additionally, it includes those predefined objects/functions (primarialy server and database object/functions) necessary for its running on a server. Like client-side JavaScript, server-side JavaScript is also embedded in HTML pages. However, unlike client-side JavaScript, server-side JavaScript is compiled into bytecodes. Server-side JavaScript is allowed to interact with relational databases; to access the servers file system; and, using LiveConnect and Java, to communicate with other applications. Server-side JavaScript provides an alternative to CGI programming.

C. JavaScript Security

From all of the evidence, it would appear that when JavaScript was conceived, security did not play a big role in its overall design. Early JavaScript security was really left to the user's browser and the absence of certain disk access functions. If the user's browser had a design flaw, then an individual could exploit it to perform unauthorized access of an individuals computer. Some authors have stated that JavaScript implements a sandbox security (as Java does); however, this is untrue as JavaScript can send e-mail, upload files, and write to local files via a dialog box. These features, coupled with early browser design limitations, allowed JavaScript to read a user's history, read the URL cache, list the files on a local disk, forge e-mail, access a user's cookies list, invoke the browser file upload command, etc. As you can see, security, or its lack, plagued JavaScript and Netscape in its infancy.

Since Navigator 2.0, Netscape has tried to plug the security holes that had plagued earlier releases. Currently, there are two security policies instituted in JavaScript: same origin policy and signed script policy. The oldest of the two, same origin policy, dates from Navigator 2.0 and is currently the default. In simplest terms, the same origin policy states that when loading a document from origin A, a script from another origin point, say origin B, may not read/modify certain predefined properties (image, layer, location, window, document) of an object in a window or frame unless it has the same host origin/port/protocol. That is, a script from origin B may not read or modify the properties of windows or documents unless they originate from origin B. Clearly, without this restriction, windows could spy on other windows and gather a host of inappropriate data.

The signed script policy is new to Communicator 4.0 and is based upon the current Java security model, object signing. A signed script requests expanded privileges and, if granted, is allowed to gain access to restricted information. This request is made by using LiveConnect and the new Java Capabilities Application Programmer Interface (API) classes. You can sign any inline script that occurs within the SCRIPT tag in an HTML document, JavaScript entitie, event han-

dler, and separate JavaScript files. Once the script has been written, you sign it using Netscape's Page Signer tool which associates a digital signature with the scripts on an HTML page. This digital signature is tied to a security certificate granted by one of the governing bodies of the Internet. The certificate is proof that the owner of the certificate has been validated to be who they say they are. The user can then decide, based on the request and certificate validity, whether to grant the expanded privileges or not.

II. JAVASCRIPT BASICS—LEXICAL STRUCTURE

A. General Information

When we look at or use a programming language, we need to understand the basic rules which determine how a program is written or used. This is referred to as the lexical structure or syntax of the language. In general, a JavaScript program can be subdivided into five major categories:

- Variables: These are the identifiers; references to the data.
- Expressions: Evaluations/manipulations that affect variables. Simply, it is a condition which evaluates to some value. An expression operates by taking the values on the right-hand side of the equal sign and performing some arithmetic operation, the result of which is stored in the variable on the left-hand side of the equal sign. Expressions can be any valid set of variables, literals, operators, and expressions which evaluates to a single value.
- Control structures: Statements which control the flow of program execution.
- Functions: Convenient reuseable blocks of program statements.
- Objects/arrays: A convenient way of packaging related data together.

B. Variables

Variables or identifiers are the names of items (things) you wish to create and use in JavaScript, i.e., Salary, VacationPay, etc. In simple terms, they are containers for values that can be used by the program. They allow us to easily associate (assign) data to an identifier for the purpose of its manipulation.

Variable names in JavaScript must adhere to the following rules:

1. Variable names may contain any letter of the alphabet or the digits 0–9 and the "underscore" character ('_').
2. The first character of a variable name cannot be numeric.
3. Variable names are case sensitive, e.g., SALARY, Salary, and SaLaRy are treated as different variable names.
4. Variable names may not contain spaces or any punctuation characters (e.g., comma, semicolon, etc.).
5. There is no official limit to the length of a variable name. However, it must fit on one line.
6. Variable names must not be keywords (reserved words) used by JavaScript.

1. Variable Scope

All variables, regardless of the programming language, support the notion of scope. That is, they have a specific range (sphere of influence) in which they are valid over. For instance, a variable declared within a function would be local to that function and said to have a local scope. We refer to these variables as local variables, and they generally only work within the function in which they are declared: they have a local scope. Any change made to a local variable is not reflected outside the area of its scope.

Conversely, any variable declared outside of a function has a global scope. That is, it is not localized to a specific block of program code. We refer to this type of variable as a global variable. Global variables can be accessed from anywhere in the program.

2. Variable Declaration and Type Declarations

In order to use a variable in JavaScript, it must be "declared" to the program. The syntax for variable declaration in JavaScript takes the following form:

```
var variable_name;
```

where var is a JavaScript keyword (reserved word). It is through the use of keywords that a programmer communicates program intent. There are 60 reserved words in JavaScript (see Table 1).

3. Type

Unlike most other programming languages, JavaScript does not have explicit data types. That is, there is no way to specify that a variable is, for example, a character, integer, or string. Formally, we say that JavaScript

Table I JavaScript Reserved Words

abstract	double	instanceof	super
boolean	else	int	switch
break	enum	interface	synchronized
byte	export	label	this
case	extends	long	throw
catch	final	native	throws
char	finally	new	transient
class	float	null	try
comment	for	package	typeof
const	function	private	var
continue	goto	protected	void
debugger	if	public	while
default	implements	return	with
delete	import	short	FALSE
do	in	static	TRUE

variables are untyped—they can contain any type of data (an integer, real, string). A JavaScript variable can potentially be any data type depending on its use (contextually). So if a variable is used in an integer context, then it is an integer. This leads to confusion for a novice reading a JavaScript program, as the variable could be one type in context A but then be interpreted as a totally different type in context B. In JavaScript then, the variables type depends on the type of data assigned to it.

4. Operators

In simplest terms, an operator is something that "operates" on a value. JavaScript supports the standard C/C++ compliment of operators.

Arithmetic operators perform their actions on numbers.

Operator	Type	Example
+	Addition	7 + 3 = 10
−	Subtraction	7 − 3 = 4
*	Multiplication	7 * 3 = 21
/	Division	9/3 = 3
%	Modulus	10%3 = 1
		for x = 10;
++	Increment	x++ = 11
−−	Decrement	x−− = 9
−	Unary negation	−10

Assignment operators assign values to variables.

Operator	Example	What it means
+=	a+=b	a = a + b
−=	a−=b	a = a − b
*=	a*=b	a = a * b
/=	a/=b	a = a/b

Logical operators compare two values and, based on whether the comparison is true (or false), return either a "true" or "false."

Operator	What it means	Example	Results
		a = 10; b = 7;	
&&	AND	(a = 10) && (b > 3)	TRUE (both sides of the operator have to be true)
||	OR	(a = 10) || (b > 13)	TRUE (only one side of the operator has to be true)
!	NOT	a ! = b	TRUE (negation, in this case, not equal)

Bitwise operators work on the binary representation of the number (a bit pattern is made up of 0s and 1s) and operate individually upon each bit in the operand with every 0 or 1 undergoing the operations individually.

Operator	What it means	
&	Bitwise AND	7 & 3
|	Bitwise OR	7 | 3
^	Bitwise NOT	7 ^ 3
<<	Left shift	7 << 3
>>	Right shift	7 >> 3

Comparison operators function by comparing two values and then returning true or false depending on the outcome of the comparison.

Operator	What it means	Example
==	Equal to	a == b
!=	Not equal to	a != b
>	Greater than	a > b
>=	Greater than or equal to	a >= b
<	Less than	a < b
<=	Less than or equal to	a <= b

Like C/C++, JavaScript also contains a comparison operator. A conditional operator is a convenient shorthand for an If-Else statement (covered in next section). A conditional operator looks at the value of the expression and, if true, executes one condition or, if false, executes another condition. The conditional operator takes the following form:

```
variable = (evaluated condition) ?
  value1 : value2
```

where value1 is done if the evaluated condition is true, otherwise value2 is done when the evaluated condition proves false. The conditional operator helps to make a program more concise and possibly more efficient.

The scope of a string operator is limited to segments of text. Operations are limited as strings have no numeric foundations. Consequently, mathematical operations such as addition, subtraction, multiplication, division, etc. are not allowed. In general, the string operator allows us to concatenate strings. For example,

```
a = "Hello"
b = "Mom"
c = a + b
```

The variable c now has as its value the string "HelloMom." Note that there is no space between Hello and Mom. This is because there was no space at the end of either Hello or Mom. The string operator simply puts together exactly what it was given. If you need a space between the two words, you can include it as a double quoted string within the operation. That is,

```
c = a + " " + b;
```

We can now put this all together and create a precedence table which shows the order of JavaScript operators (see Table 2).

Arrays are an indexed collection of items all of which have the same underlying type. A construct supported by almost every computer programming language is that of the array. An array consists of a contiguous block of memory. That is, it consists of consecutive memory storage locations in main memory. Into this block of memory, you may store multiple data values of the same type. Each value is individually addressable and can be accessed (examined) directly by providing the appropriate array location. An array provides an excellent way of organizing and accessing data. Conceptually, we view an array as:

	index 0	index 1	index 2
scores	73	79	84

Table II JavaScript Operator Precedence

Precedence	Operator	Notes
1	( )	From innermost to outermost
	[ ]	Array index value
	function ( )	Any remote function call
2	!	Boolean Not
	~	Bitwise Not
	-	Negation
	++	Increment
	- -	Decrement
	type of	
	void	
	delete	Delete array or object entry
3	*	Multiplication
	/	Division
	%	Modulo
4	+	Addition
	-	Subtraction
5	<<	Bitwise shifts
	>	
	>>	
6	<	Comparison operators
	<=	
	>	
	>=	
7	= =	Equality
	!=	
8	&	Bitwise And
9	^	Bitwise XOR
10	|	Bitwise Or
11	&&	Boolean And
12	||	Boolean Or
13	?	Conditional expression
14	=	Assignment operators
	+=	
	-=	
	*=	
	/=	
	%=	
	<<=	
	>=	
	>>=	
	&=	
	^=	
	|=	
15	,	Comma (parameter delimiter)

The scores array consists of three integers, each addressable individually. To create this array, our JavaScript program would have made the following declaration:

```
var scores = new array(3);
```

This tells the system to reserve memory for an array of three elements. C, C++, Java, and JavaScript are zero-based indexing languages, which means that the index numbering begins with the number 0. So, for example, the first score, value 73, is available by accessing the scores array as scores[0] (array indexes are accessed using the square brackets). This would give us access to the first score that we have put into our array.

Arrays may store any type of information. For example, to store character strings of data, say the names of the 12 months of the year, we could declare and initialize the array as follows:

```
month = new Array(12);
```

Next, we would set a numeric value for each month) in the array. For example,

```
month[0] = "January";
month[1] = "February";
:
:
month[11] = "December";
```

Alternatively, we could have declared and initialized the array as:

```
month = new Array("January",
  "February", "March", "April",
  "May", "June", "July", "August",
  "September", "October",
  "November", "December")
```

Here, JavaScript computes the size of the array from the values it was given to initialize the array.

III. STATEMENTS AND FUNCTIONS

Through the use of control structures, a program's purpose or expressive power begins to take shape. The Dutch computer scientist Edsger Dijkstra has shown that for a language to be productive, it must have three essential control structures:

1. Sequence (stepping)
2. Iteration (looping)
3. Decision (branching)

JavaScript provides all three control structures.

A. Sequence

A serial execution of statements that involves no looping or branching is called a sequence. A sequence is the most basic of control structures. Consider, for example, the following code fragment:

```
.
.
.
Var firstnumber,
    Secondnumber,
    Total;

Firstnumber = 7;
Secondnumber = 3;
Total = Firstnumber + Secondnumber;
Document.writeln("The sum is " +
  Total);
```

This is executed sequentially; that is, one statement followed by another with no branching or calls to other code segments. Sequence is the simplest of control constructs.

B. Iteration

There are three basic iteration (looping) constructs in JavaScript:

1. while <expression> <statement>
2. do <statement> while <expression>
3. for <expression> <statement>

The While statement is a looping statement, repeating all of the statements that fall within its scope of execution. This can be a single statement or a group of statements within the block of control (open and closed braces, i.e., {}). The format of a While statement provides for a conditional test of success at the top of the loop. The loop will continue to execute all statements within its scope until the test condition proves false. As the test is performed at the top of the loop, it is possible that the condition will initially prove false and no statements within the While loop will be executed. An example of a While loop is:

```
:
:
while (i < 1 0) {
        document.write("Hello
          Mom!");
        i++;
}
```

The While loop tests for the condition i < 10. While this is true, the loop will continue to display "Hello Mom." As soon as the loop test condition proves false, the loop

is exited and execution continues from the first statement following the closing '}' of the While loop.

Like the While statement, the Do-While statement also provides for a repeating loop with a test condition. However, unlike the While construct which tests for the truth of a condition at the top of the loop, the Do-While tests for this condition at the bottom of the loop. The result is that all statements within the scope of the Do-While will be executed at least once. We see the above example rewritten for a Do-While loop:

```
.
.
.
do {
      document.write("Hello Mom!");
      i++;
} while (i <1 0);
```

The For loop also provides for a test. Additionally, it provides the ability for a variable to be initialized and a statement to be executed. The general format is:

```
for (<initialize> ; <test> ;
  <update>) {
        <statement>}
```

The For loop provides us with the ability to execute a series of statements for a measured number of times. The typical sequence is <initialize a variable><test the truth of a condition><execute a statement(s)<update the variable><test the truth of the condition again>, etc. An example of the for loop is as follows:

```
for(i=1; i<10; i++)
        document.write("Hello
          Mom!");
```

This loop will continue to display, for 10 iterations, "Hello Mom!".

C. Decision

Decision making (branching) in JavaScript is accomplished by three basic constructs: the If statement, the If-Else statement, and the Switch statement.

1. The simplest decision statement is the If statement. It is a simple test of truth followed by the execution of a statement(s). The general form of an if statement is:

```
if (conditional expression) {
    statement(s);
}
```

For example,

```
if (exam_score >= 90){
        document.write("Your exam
          score is an A");
}
```

Alternatively, the above could have been written as:

```
if (exam_score >= 90)
        document.write("Your exam
          score is an A");
```

Remember, if your control construct (simple decision, loop, etc.) contains only one executable statement, then the need for open and closed braces to control its scope of action is not necessary, as the normal scope of effect is to only control and execute the single statement following the construct. Placing braces around the statement is not required, but does help to show the reader what action is being controlled.

2. An If-Else branch gives us the opportunity to execute an alternative choice should the first prove false. That is, if the first condition tested for proves false, then execute the statement in the else portion of the branch. The general form is:

```
if (conditional expression) {
    statement(s);
}
else {
  statement(s);
}
```

An example of this is:

```
if (exam_score >= 70)
        document.write("I am happy
          to say you have passed
          the exam!");
else
        document.write("I am sorry
          to inform you that you
          have not passed the
          exam.");
```

The branching If-Else additionally gives us the capability of allowing another test condition in the else. For example,

```
if (conditional expression) {
    statement(s);
}
else (conditional expression) {
 statement(s);
}
```

In this instance, if the initial condition tests true, test for another condition to see if there is a possible action to execute. An example of this would be:

```
if (exam_score >= 70)
        document.write("I am happy
          to say you have passed
          the exam!");
else (if exam_score >= 65)
        document.write("You have
          received a conditional
          pass. More work is
          needed.");
        else
          document.write("I am
            sorry to inform you
            that you have not
            passed the exam.");
```

3. The Switch statement gives us the ability to choose from several different possibilities in a much more refined way than multiple If-Else statements. The general form is:

```
switch (expression) {
   case label :
      statement(s);
      break;
   case label :
      statement(s);
      break;
   ...
   default :
      statement(s);
}
```

Here, the Switch keyword causes JavaScript to evaluate the expression that it was passed. Based on its evaluation, if it matches a case label, then execution continues at the first statement following the label. If, after evaluation, no label matches the expression, the flow of control is passed to the default statement and execution begins there. Execution of statements in a switch is terminated with the keyword "break." Without a break statement, execution is sequential and will continue until the closing brace of the Switch statement. The following is an example of a Switch statement:

```
switch (letter) {
   case 'A' :
   case 'E' :
   case 'I':
   case 'O':
   case 'U'
         document.write("You have
           typed in a vowel!");
         break;
...
default :
   document.write("I am sorry, but
     that was not a vowel!");
   break;
}
```

D. Introduction to Functions

As we create larger and larger programs, we often find that we write and rewrite similar sets of programming instructions. JavaScript provides the ability to functionally group together these statements and refer to them in order to perform work as often as we like. This process is called function creation.

A function in JavaScript is a reusable block of program code that has a finite existence. That is, its life is from the opening brace, which begins the enclosure of a series of JavaScript statements, until the closing brace. The purpose of a function is to efficiently package together a number of statements that perform some action and that can be conveniently invoked (called). The programming statements in the function are not executed until the function's name has been called by the program. JavaScript functions can be passed values from either the main program or another function to act upon, and additionally, a JavaScript function can return a value as well.

E. Define and Invoke a Function

Generally, a JavaScript function declaration takes the following form:

```
function name (arg1, arg2, ...
  argN) {
     body
}

where:
   name is the name of the function.
   arg1 ... argN are the names of
     arguments (or parameter values
     passed)
       to the function.
```

Arguments to the function are optional; however, the syntax dictates that the open and closed parentheses are to be used even if no arguments are passed. The body of the function consists of a series of statements and even calls to other functions. The duration or scope of effect of a function is from opening brace to closing brace, even if only a single statement comprises the function.

For example, let us define a function that determines the largest of two numbers passed to it. The definition of the function would appear as follows:

```
function getLarge(number1, number2){
       if (number1 < number2)
                return number2;
         else
                  return number1;
}
```

We have now defined the function to the program. To make a call to a function, that is, to be able to use the function, we simply use its name followed by a parenthesized parameter expression, i.e.,

```
name (expr1, expr2, ... exprN);
```

So, from the example above, a call to use the function and the value it returns would look something like the following:

```
Largest = getLarge(number1,
  number2);
```

with the variable Largest holding the result from the function call by means of a simple assignment statement.

F. Some Built-In Functions

JavaScript provides a very small number of built-in functions. Table 3 lists these functions. The function eval evaluates a string as a JavaScript expression or statement. It will either execute the statement or return a value. For example,

```
number = eval("getLarge(x,y);");
```

The function ParseInt determines if there is an integer value at the beginning of a string. If so, it returns the number; otherwise, the return is "NaN" (not a number). The function parseFloat determines if there is a floating-point number at the beginning of a string. Like ParseInt, if the search is successful, it returns the number; otherwise, the return is NaN. The IsNaN() function returns a value of true for a string not evaluated to be a number. Currently, this function only works on UNIX platforms. The escape() function works by converting a string to an URL-encoded (escaped) format. All nonalphanumeric characters are converted to %. Additionally, their ASCII value is represented in hexadecimal. The unescape() function is the inverse of the escape() function. It works by converting an escaped URL-encoded string into normal text. The purpose of the taint() function is to mark a variable or property with the current script's taint code. You can taint data elements (properties, variables, functions, objects) in your scripts. The purpose is to prevent the returned values from being used in an inappropriate way by other scripts. The data-tainting security model was implemented as "experimental" in Netscape Navigator 3 to prevent private data from being accessed on the Web. It proved to be unsuccessful and is now [along with the untaint() function] deprecated. For the untaint() function, see above.

Table III JavaScript Built-In Functions

Function	Description
eval(string)	Evaluates a string 'string' as a JavaScript expression
escape(string)	Converts strings to HTML special characters
IsNan(string)	Tests whether a string represents a number
unescape(string)	Inverse of escape()
parseFloat(string, radix)	Converts a string to a floating-point number (if it can)
ParseInt(String)	Converts a string to an integer number (if it can)
taint(string)	
untaint(string)	

IV. DOCUMENT OBJECT MODEL—WEB PAGE ORGANIZATION

The concept of the Document Object Model (DOM) is an interesting one. In simplest terms, the DOM is a set of interfaces and objects which give the Web page developer the ability to manage the appearance and functionality of a Web page. The JavaScript language specification ECMA-262 does not actually specify the details of the DOM. In fact, the specification of the

DOM could actually be implemented by other scripting languages. This is at the core of the incompatibility problem with respect to creating Web pages for Netscape Navigator and Microsoft Internet Explorer. They have each developed their own DOM and they are slightly different from each other. Now, having said this, let us give a word of support to Netscape and Microsoft in that they have both remained fairly true to the core concept of the DOM. Both have left the basic syntax of JavaScript alone. In fact, the point of contention really lies in how they have each come up with their own similar (and slightly dissimilar) naming conventions for the elements that make us a Web page. In our discussion, we will try to steer clear of the differences and focus more on the dynamics of creating and managing a Web page, which is what the DOM is really all about.

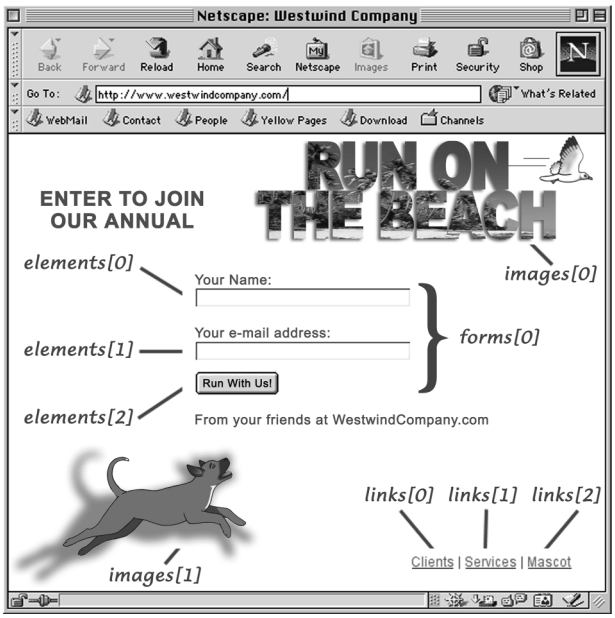


Figure 1 Web page layout.

A. The Document Object

The DOM is a fairly platform/language neutral API which allows the programmer to dynamically access a document and express a measure of control over its content presentation and behavior. This allows any browser to visually present its structure and content without concern for the particular operating system, hardware, or, for that matter, the scripting language being used.

Netscape and Microsoft have both held to the basic concept that a Web page can be broken down into four fundamental component types: objects, properties, methods, and events. The DOM describes each Web page as a document and the elements of the page as objects. The DOM then manages each page as a related set of objects. For example, a page may have on it images, text, forms, links, etc. (see Fig. 1).

The DOM is responsible for the management of each of these objects. It is convenient then to think of the DOM as a high-class object manager, allowing the programmer to control their placement, visual appearance, functionality, and associated actions. We stated that the DOM breaks the Web page down into four fundamental component types. The key to understanding the DOM is to understand the purpose of each of the four types.

1. Object

In concept, it is useful to think of an object as a container which represents some aspect (element) of a Web page. The object then expresses the characteristics of the element through the **properties** of the element and executes/performs any required actions through its associated **methods.** For instance, the *reset* object would contain the properties and methods necessary to reset a form back to its default values. An objects name is typically a noun. For example, image, window, and document are JavaScript object names. Table 4 shows a list of JavaScript objects.

2. Property

In general, characteristics (properties) of an object are represented by a single word attribute. For example, bgcolor, fgcolor, linkcolor, and title are all prop-

Table IV JavaScript Objects

Anchor	Frame	Location	RegExp
Applet	Function	Math	Reset
Array	Hidden	MimeType	screen
Boolean	History	Navigator	Select
Button	Image	netscape	String
Checkbox	Java	Number	Style
Date	JavaArray	Object	Submit
document	JavaClass	Option	Sun
event	JavaObject	Packages	Text
FileUpload	JavaPackage	Password	Textarea
form	Layer	Plugin	window

erties of the document object. Each property has a data value associated with it. For example, if a Web page developer wanted his or her Web page to have a particular background color, say blue, this could be expressed by assigning the value "blue" to the bgcolor property of the document object (bgcolor="blue"). By modifying an object's properties, the developer can change the fundamental visual presentation of a Web page. Currently, there is no limit to the number of properties/methods/objects that can be contained in an object. Table 5 lists some of the most common object properties of JavaScript.

3. Method

The action that a particular object takes is implemented by a method. A method then is basically a function that is designed to perform a specific task on an object. JavaScript comes with many predefined methods. To access one of these methods, the user needs to know the method's name, the type of data that needs to be sent to it, and, additionally, the type of data it will return. Keep in mind that not all methods require that the user send it data, and conversely, not all methods will return data to the user. In a nutshell then, methods simply act on an object and can set or return information about that object. Table 6 lists methods currently included in the Javascript language. As a method is action oriented, the user will note that method names tend to be represented as verbs. The text "Pure Javascript" by Wyke, Gilliam, and Ting (1999) provides excellent coverage as to the function of each of the methods in Table 6.

4. Event

The way in which JavaScript works is to respond to actions (events) caused by a user accessing a Web page by clicking a button or activating a link. That is, JavaScript functions by trapping the action and then

Table V JavaScript Properties

Type	Item	Description
Property	alinkColor	Refers to the color of an activated link
	all	Refers to an array of all of the HTML tags in a document
	anchors	Refers to an array of anchor objects
	applets	Refers to an array of applet objects
	bgcolor	Refers to the background color of a document
	classes	Refers to the style sheet classes array
	cookie	Refers to the cookie associated with a document
	domain	Refers to the domain of a document
	embeds	Refers to an array of embedded objects
	fgcolor	Refers to the text color in a document
	forms	Refers to an array of form objects
	ids	Refers to the style sheet IDs array
	images	Refers to the array of image objects
	lastModified	Refers to the date in which the document was last modified
	layers	Refers to the array of layer objects
	linkColor	Refers to the color of the links
	links	Refers to the array of link objects
	plugins	Refers to the array of embedded objects
	referrer	Refers to the URL of the document to which the current document is linked
	tags	Refers to the style sheet tags array
	title	Refers to the title of the document
	URL	Refers to the current documents URL
	vlinkColor	Refers to the color of the visited links

Table VI JavaScript Methods

abs	clearTimeout	getHours	open	small
acos	click	getMinutes	parse	sqrt
alert	close	getMonth	pow	strike
anchor	confirm	getSeconds	prompt	sub
asin	cos	getTime	random	submit
assign	eval	getTimeZoneoffset	round	substring
atan	exp	getYear	select	sup
back	fixed	go	setDate	tan
big	floor	indexOf	setHours	toGMTString
blink	focus	italics	setMinutes	toLocaleString
blur	fontcolor	lastindexOf	setMonth	toLowerCase
bold	fontsize	link	setSeconds	toString
ceil	forward	log	setTimeout	toUpperCase
charAt	getDate	max	setTime	UTC
clear	getDay	min	setYear	write

responding to it. How does JavaScript respond to an event? Java-Script uses an event handler. By convention, the name of an event handler is usually the name of the event preceded by "on." For instance, if the event is *submit* for submitting a form, then the event handler for this event would be called *onsubmit*. In this way, it is clear which handler is associated with which event. In reality, event handlers are just built-in JavaScript methods that allow users to control the response to an event. The nice thing about JavaScript is that if users come up with a new user interaction, they can write their own event handler to process it. Events are transparent to the user. No one is really aware that they are clicking a mouse button. However, the programmer can choose to trap (intercept) this action and trigger a specific piece of code to respond to the action. This provides a great measure of control over what happens on a Web page. Table 7 lists common JavaScript event handlers.

V. THE OBJECT HIERARCHY

Now, how does all of the above fit together? The browser window is associated with a hierarchical collection of objects. The main object of interest for us would be the *document* object. Figure 2 illustrates the hierarchical organization of the DOM.

At the top of the hierarchy is the window object. Each browser window that is open has associated with it a window object. As the DOM is a hierarchical structure, all objects associated with the window object are its child objects. A basic Web page has the following objects associated with it: *window, navigator, location, history,* and *document.*

- Window object: This is the root (highest) level object in the DOM. All other objects are termed child objects, as they are decended from the main parent object, the window. Each Web page (document) is associated with the window object with its own URL, which is stored in the window's location object. As the window visits many different URL documents over the course of browsing the net, these pages are referenced in the properties of the window's history object.
- Navigator object: The navigator object contains the properties necessary to determine the version of the Navigator in use. Additionally, it maintains properties to determine the MIME types supported by the client and all plug-ins the client currently has installed.
- Location object: The location object maintains properties which describe the location (URL) of a particular document. Included in the determination of where a document is located are properties related to the *hostname* of the system hosting the document, its *pathname,* and its *port* number.
- History object: The history object maintains properties used to refer to the list of previously visited URLs. The main property of the history

Table VII JavaScript Event Handlers

onAbort	User depresses the stop button to abort the loading of an image
onBlur	A form element loses focus
onChange	The text, select, or textarea field loses focus; its value is modified
onClick	A form object is clicked
onDblClick	User double clicks a form element or a link
onDragDrop	User drags (drops) an object (e.g., file) onto the browser window
onError	An error happened during loading of a document or image
onFocus	The window, frame, frameset, or form element receives focus
onKeyDown	User depresses a key
onKeyPress	User presses (holds down) a key
onKeyUp	User releases a key
onLoad	The browser finishes loading a window or all frames within a frameset
onMouseDown	User depresses a mouse button
onMouseMove	User moves the mouse cursor
onMouseOut	Mouse leaves an area or link
onMouseOver	Mouse hovers over an object or area
onMouseUp	User releases a mouse button
onMove	User/script moves a window or frame
onReset	User resets a form by depressing (clicking) reset button
onResize	User/script resizes a window or frame
onSelect	User selects some of the text within a text or textarea field
onSubmit	User submits a form by clicking (depressing) submit button
onUnload	User exits a document (leaves a Web page)

object is *length,* which is a numeric counter of the number of URLs the window has been to.

- Document object: Every window has an associated document object. The document object maintains information about its objects (forms, links, images, etc.) and their placement on the page, as well as information about the subelements of the page (a forms buttons, text, etc.). In order to refer to the methods and properties of an object, the user has to understand its naming convention. There are three simple rules to follow when referring to an object:

1. The name of an object contains the names of all objects that comprise it. That is, the name is ordered by its appearance (lineage) in the DOM hierarchy starting with the object's highest name, *window.* This is referred to as the object's "pathname." Additonally, each object is separated from the other by the use of a "dot" character. For example, suppose that I wanted to refer to a radio button on a particular document's form. To access this element, I would specify its pathname: window.document.forms[0].elements[1]. Finally, JavaScript allows us to omit the window reference as all objects officially start from there. Our pathname now to the radio button example would be document.forms[0].elements[1].
2. You may have noticed that in the pathname example both forms and elements are specified as arrays. In the DOM hierarchy presented in Fig. 2, notice that a document can be made up of multiple forms, each of which may have multiple elements associated with it. In point of fact, a document may have multiple instances of any type of object associated with it (images, forms, links, etc.). To manage multiple instances of an object, we use an array. We can see immediately that we have an array of objects as we use the open and close bracket ([]) with its index number to indicate which instance we are addressing. Additionally, the object's name is made plural to show that more than one of this type of object is present.

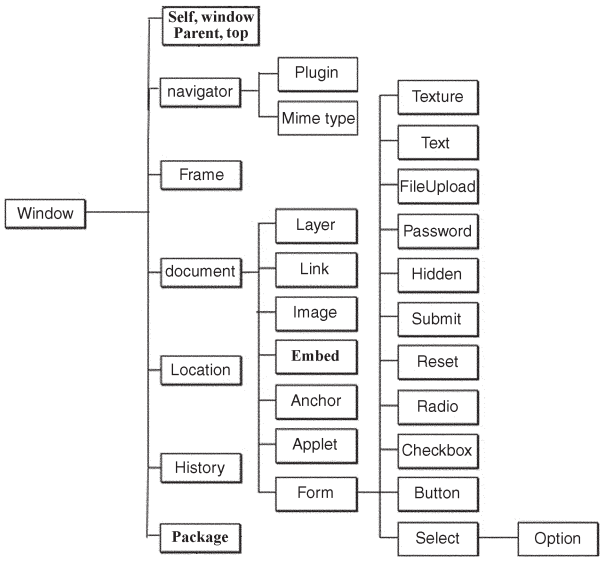


Figure 2 DOM hierarchy.

3. In the DOM hierarchy chart presented in Fig. 2, we see that a form can be made up of many different object types. Collectively, we refer to these object types as elements of the form. They are stored in the elements array associated with the form and can be accessed by indexing into the array (elements[8]). Carrying on with our example, document.forms[0].elements[1] would give us access to the second element of the first form associated with our document.

VI. FORMS AND FORM ELEMENTS

A convenient way of obtaining information from a user or, for that matter, of interacting with a user in general is through the use of an HTML form. An HTML form, however, is not very practical from a processing standpoint. A standard HTML form can do very little with respect to validating data input. An HTML form simply accepts user input and passes it on to the server for processing. This is really a complete waste of server-side resources and Internet bandwidth. Why ask the server to do something for you when the client can easily accomplish it more efficiently. So how can the client perform this task? Why, through JavaScript of course!

A. The Form Object

JavaScript extends the functionality of an HTML form. To JavaScript, a form is simply an object that contains one to many different elements, each of which can be accessed and manipulated by JavaScript. The most effective way of doing this is through two concepts introduced earlier: JavaScript *function* and *event handler.*

A function, as you recall, allows us to package together a series of program statements that will accomplish a particular task for us. A function is called by the JavaScript program in response to a particular user event, such as a cursor movement event, pressing a submit button, etc. We then have an event handler which recognizes the event that has occurred and calls the associated function that was written to handle the user-initiated event.

Let us, for example, create a very simple form. This is an HTML form that does absolutely nothing much except ask the user to push a button.

```
<HTML>
<HEAD>
<TITLE>Forms: The Official JavaScript
   Button Form</TITLE>
</HEAD>

<BODY>

<FORM>
        <INPUT TYPE=button
        VALUE="Click Me"
        onClick="alert('You are a
          FIRST CLASS button
          pusher!.');">
</FORM>

</BODY>
</HTML>
```

As you can see by the above example, this is a simple use of an HTML form. It is embedded in the BODY of an HTML document, and the form contains only a single element, that being a button. Elements are specified by the INPUT TYPE keyword. The event handler that we have specified to be triggered when the button is pressed is the *onClick* event handler. In the above example, the event handler causes an alert window to be displayed with our message in it. This form requires no remote processing. It is a *local* form with all processing handled by the client.

There are times, however, when you need to be online to be able to process data on a remote server. For example, suppose that I want to allow certain individuals to access my course notes. I would create a form and require that the individuals identify themselves with their first name, last name, and social security number. After gathering the data on the client form, it is then submitted to the server where a server-side

program would compare their name with an approved list of individuals that can access the course material. This requires the form to communicate with the server by calling a stored program (script) on the server. This is accomplished through the use of the ACTION keyword in the HTML form. The ACTION keyword contains the file name of the script responsible for processing the form. When we are ready to initiate the action, we use the "submit" button to send the data to the server for processing.

As you can see from the following example, our simple form has three form elements (fName, lName and idNumber) which are text fields. These elements are managed by the form object, which is managed in turn by the document object.

```
<HTML>
<HEAD>
<TITLE>Forms: Course Access</TITLE>
</HEAD>
<BODY>
<H3>Please Identify Yourself To
  Access Course Notes</H3>
<FORM Action="http://www.bgarfolo/
  cgi-bin/login.cgi"
      Name="login">

<PRE>
      First Name: <INPUT Type="TEXT"
        Name="fName">
      Last Name: <INPUT Type="TEXT"
        Name="lName">
      Student ID: <INPUT Type="TEXT"
        Name="idNumber">
      <INPUT Type="Submit">
</PRE>
</FORM>
</BODY>
</HTML>
```

Both of the above examples demonstrate a single aspect of a JavaScript form: either it operated primarily on the client (local) or on the server (remote). So where is all of the strength JavaScript is suppose to provide? The strength comes from being able to preprocess the form to determine if it is appropriate to send the form to the server for processing.

```
<HTML>
<HEAD>
<TITLE>Forms: Course Access</TITLE>
<SCRIPT Language="JavaScript">
<!—

       function validSID(sidNumber){
       var index
       var character
       var error_message = "Please
         enter your nine digit
         Student ID number."

       if (sidNumber.length != 9){
                alert(error_message)
                return false
       }

       for (index=0; index
         <sidNumber.length;
         index++){
                character =
                  sidNumber.charAt
                  (index)
                if (character > "9"
                  || character <
                  "0"){
                          alert
                            (error_
                            message)
                          return
                            false
                }
       }

       return true
}

//—>
</SCRIPT>

</HEAD>
<BODY>
<H3>Please Identify Yourself To
  Access Course Notes</H3>
<FORM Action="http://www.bgarfolo/
  cgi-bin/login.cgi"
      Name="login"
      onSubmit="return
        validSID(this.sid
        Number.value)">

<PRE>
      First Name: <INPUT Type="TEXT"
        Name="fName">
      Last Name: <INPUT Type="TEXT"
        Name="lName">
      Student Number: <INPUT
        Type="TEXT" Name="sid
        Number">
      <INPUT Type="Submit"> <INPUT
        TYPE="RESET">
```

```
</PRE>
</FORM>
</BODY>
</HTML>
```

In this example, we use some local processing in the form of a JavaScript function to determine if an individual has entered a reasonable student ID number (a series of nine numeric digits). This is handled by the function validSID. When the user presses the submit button, the event handler *onSubmit* handles the event by calling the validSID function. This function will return a value of "True" if they have entered a proper student ID number. If they have entered an improper student ID number (to short, to long, alphabetic characters, etc.) the function displays an error message to the user and returns a value of "False." Only when the function returns a value of True will the form be submitted to the server for processing. This is accomplished by the inclusion of the "*return*" keyword in the *onSubmit* clause. The *return* sends the value back to the event handler which determines whether further processing is warranted. If the *return* is not included in the *onSubmit* clause, the form will always be sent to the server regardless of whether the student ID is valid or not.

Creating the form in this way is very efficient, as it is the client that performs the student ID validity, leaving the server to determine if it is a valid student for the course. Finally, for good measure, we also include a call to the form *reset method* to allow the user to clear out the form and start over again when an error has occurred.

VII. FRAMES

In early Web browsing a severe limitation existed in that a user could view only one Web document page at a time. Netscape created the ability for a browser to display and access multiple Web pages simultaneously by extending the HTML language to include the concept of a **frame**. The frame, in essence, subdivides the main browser window into some number of subwindows, each displaying a Web page. Each window has available to it all of the properties, methods, etc. that a standard window has as specified by the DOM. As the frame concept is now part of the DOM, this means that each subwindow can be further subdivided into a number of subwindows (will this ever stop?). Today, all major browsers support frames.

A. Relationship between Frames

As we recall from our earlier discussion on the DOM, JavaScript views everything from a hierarchical standpoint. Let us say that we want to split our main browser window into two separately addressable windows. The main window is now considered to be the parent of the two child windows as seen in Fig. 3.

To accomplish Fig. 3, our start-up HTML window document must describe the subwindows and their organization. The code for Fig. 3 would be as follows:

```
<HTML>
<FRAMESET ROWS="55%,*">
<FRAME SRC="frame1.html"
   name="left">
<FRAME SRC="frame2.html"
   name="right">
</FRAMESET>
</HTML>
```

The HTML FRAMESET tag is the key to dividing up the main browser window into multiple windows. Actually, the above code is what is loaded into the main document window, or parent window in this case. The user does not see this code. It controls the action of the browser and instructs it to display two windows, side by side (ROWS). The first window (frame1) will take up 55% of the overall browser window space. The second window (frame2) will get the remaining (referred to by the *) 45% of the window. As previously stated, each window must have a unique name, so we have named the frame1 window *left* and the frame2 window *right* to help us visualize how the system functions. Based on the above frame definition, Fig. 4 illustrates how the invisible controller would look.

For us to be able to access the child windows (frames), each must have a unique name. We will name our frames left and right with the *name* attribute. We refer to each frame by its frame name (left and right in this example), which in turn references an HTML file. What is happening is that we have now subdivided our browser window into two separately addressable windows, each of which has its own HTML controller files. These files can control the windows as separate, noninteracting windows or, through the use of JavaScript, data from one window can be sent to another window for processing.

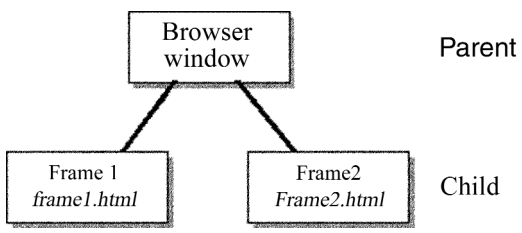


Figure 3 Window–subwindow relationship.

Figure 4 Sample frame layout.

B. Interaction between Frames

One of the most annoying limitations of a standard HTML-based frame setup is that you cannot change the content of the page once you have loaded it into the browser window. True, you can enter data into input boxes, but selection features (check boxes, buttons) or data transfer from another window is not available without some very complicated multiform magic which will probably involve reloading pages. This is all very disruptive to the flow and concentration of the user interacting with the Web site. We get around all of this by turning to JavaScript.

Using the above frame definition, let us consider two frame interactions: the first where we print a simple message from the left frame into the right frame, and the second where we actually transfer data from the left frame into the right frame for processing. In our examples, frame2 really is only a receiver of data. It will do no work by itself. For this, we need only a primitive Web page to hold a clean spot in the frame for printing. The code example for frame2.html would appear as follows:

```
<HTML>
<HEAD>
<TITLE>Place Holder....Gives a clean
   frame to write on </TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Both of our examples require the use of a form. Primitive frame interaction, that being frame A displays a Web page in frame B, by the activation of a link is really nothing more than having the output of some activated link target its output to another frame. Additionally, having a JavaScript print statement send its output to another frame is still just a trivial lesson in printing as seen in the following code example for frame1.html:

```
<html>
<head>
<title> A very very simple
   JavaScript Frame Printing
   Example </title>
<body>
<script>
        parent.right.document.write
          ("<h3>HELLO</h3>");
</script>
</body>
</html>
```

Real work is done when we interact with the user. To actually have frame A write something to frame B based on user input will require the use of a form, as it is our prime conduit for data movement in JavaScript.

C. Example 1: Printing from One Frame into Another

As you recall, in order to access any variable in a multiframe environment, we need to refer to the variable's hierarchical path. Printing in JavaScript is accomplished through the use of the *write* method. The general format for writing is **document.write**(some text or data). In a multi-frame environment, we would prepend the hierarchical path to the frame in which we wanted to participate (parent.the_frame_name. document.write(A whole lotta something). Remember, the "dot" in the path is critical for specifying the complete path. For instance, if we wanted the first frame to write to the second frame, our statement in the first frame would be parent.second.document. write(Hello. Isn't JavaScript Fun!). The following is the new code for frame1 layout.

```
<html>
<head>
<title> A simple JavaScript Frame
   Printing Example </title>
<script>
        function a_greeting
          (simple_form)
        {
                  parent.right.
                    document. write
                    ("<h3>" +
                    simple_form.
                    a_message.value +
                    "</h3>");
        }
</script>
```

```
</head>
<body>
        Please enter a message
          below for remote printing.
<P><P>
<form name="simple_form">
<input type="text" name="a_
  message"><p>

Now, press this button to see it
  print:
<input type="button"
  name="distant_print" value="push
    to print"
         onclick="a_greeting
           (simple_form);">
<P>
</form>
</body>
</html>
```

Now, let us look at something slightly more complicated. Here, we will take some user-supplied input data from the first frame and move it into the second frame. We will use the frame controller we have and only change the names of the HTML files it is controlling, in this case, frame1b and frame2b. Our simple receiver frame (frame1) is easily transformed into frame2b by adding a form object that will receive the user data from frame2a. The following is the new frame2:

```
<html>
<head>
<title> A simple JavaScript Frame
  Data Transfer Example </title>
</head>
<body>
<h1>Data Output Frame</h1>
<P>
<form name="outform">
<input type="text"
  name="outmessage">
</form>
</body>
</html>
```

Now the new frame1 code for gathering the user input data is:

```
<html>
<head>
<title> A simple JavaScript Frame
  Data Transfer Example </title>
</head>
<body>
<h1>Data Input Frame</h1>
<P>
<form name="inform">
Please enter a message for data
  transfer:
<input type="text" name="inmessage">
<P>
Now press this button to transfer
  the message to the output
  frame:<br>
<input type="button" name="transfer"
  value="transfer to output frame"
onclick="parent.right.document.
  outform.outmessage.
  value=parent.left.document.
  inform.inmessage.value;">
<P>
</form>
</body>
</html>
```

Now the temptation is to say, "How is this really different from the previous example?" On the surface, it does look quite similar in that a message from frame1 is printed onto frame2. However, this example differs in one major way: we are now able to use the message in frame2 as it has been assigned to an object element of frame2. In the first example, the *onClick* event handler invoked our Javascript function to display a message in frame2. In this example, the *onClick* event handler assigns the value of the user-supplied data in frame1, inmessage, to the form element outmessage in frame2. We now have access to this data and can manipulate it. Here, we convert the user-supplied data to all upper case by calling the *toUpperCase* method to create a new frame2.

```
<html>
<head>
<title> A simple JavaScript Frame
  Data Transfer Example </title>
<script>
function bigstuff()
{
  document.outform.outmessage.value
    = document.outform.
    outmessage.value.toUpperCase();
}
</script>
</head>
<body>
<h1>Data Output Frame</h1>
<P>
```

```
<form name="outform">
<input type="text"
  name="outmessage">
<p>
Now lets manipulate this new data
  by converting it to upper case:
<input type=button value="Make it
  BIG" onClick="bigstuff()">
</form>
</body>
</html>
```

VIII. FINAL THOUGHTS

Javascript is a very powerful scripting language that has the ability to give a Web page "life." It gives the designer the ability to go beyond the simple two-dimensional page layout and allows the user to truly interact with the Web page. It is through the use of this powerful programming language that the Internet truly transcends the sterile information page and becomes a tool for e-business.

IX. APPENDIX

Putting it all together, the following table is a listing of the most common objects, properties, methods, and event handlers. This table is organized alphabetically by object with its corresponding properties, methods, and event handlers.

Object	Property	Method	Event handler
Anchor	None	None	None
Anchors array	length	None	None
Applet	None	All public methods of applet	None
Applets array	length	None	None
Area	hash host hostname href pathname port protocol search target	None	onMouseOut onMouseOver
Array	length prototype	toS reverse sort	None
Button	form name	blur click	onBlur onClick
	type value	focus	onFocus
Checkbox	checked defaultChecked form name type value	blur click focus	onBlur onClick onFocus
Date	prototype	getDate getDay getHours getMinutes getMonth getSeconds getTime getTimezoneOffset getYear parse setDate setHours setMinutes setMonth setSeconds setTime setYear toGMTString toLocaleString toString UTC valueOf	None
document	alinkColor Anchor anchors Applet applets Area bgColor cookie domain embeds fgColor Form forms Image images lastModified linkColor Link location links referrer title URL vlinkColor	close open write writeln	None
FileUpload	form name	blur focus	onBlur onChange

(continues)

Appendix *(continued)*

Object	Property	Method	Event handler
	type		onFocus
	value		
Form	action	reset	onReset
	Button	submit	onSubmit
	Checkbox		
	elements		
	encoding		
	FileUpload		
	Hidden		
	length		
	method		
	name		
	Password		
	Radio		
	Reset		
	Select		
	Submit		
	target		
	Text		
	Textarea		
Forms array	length	None	None
Frame	defaultStatus	blur	onBlur
	frames	clearTimeout	onFocus
	length	focus	
	name	setTimeout	
	opener		
	parent		
	scroll		
	self		
	status		
	top		
	window		
Frames array	length	None	None
Hidden	name	None	None
	type		
	value		
History	current	back	None
	length	forward	
	next	go	
	previous		
History array	length	None	None
Image	border	None	onAbort
	complete		onError
	height		onLoad
	hspace		
	lowsrc		
	name		
	prototype		
	src		
	vspace		
	width		
Images array	length	None	None
Link and Area	hash	None	onClick
	host		onMouseOut
	hostname		onMouseOver
	href		
	pathname		
	port		
	protocol		
	search		
	target		
Links array	length	None	None
Location	hash	reload	None
	host	replace	
	hostname		
	href		
	pathname		
	port		
	protocol		
	search		
Math	E	abs	None
	LN2	acos	
	LN10	asin	
	LOG2E	atan	
	LOG10E	atan2	
	PI	ceil	
	SQRT1_2	cos	
	SQRT2	exp	
		floor	
		log	
		max	
		min	
		pow	
		random	
		round	
		sin	
		sqrt	
		tan	
MimeType	description	None	None
	enabledPlugin		
	suffixes		
	type		
MimeTypes array	length	None	None
Navigator	appCodeName	javaEnabled	None
	appName	taintEnabled	
	appVersion		
	mimeTypes		
	plugins		
	userAgent		
Options array	length	None	None
Options array elements	defaultSelected	None	None
	index		
	length		
	selected		
	selectedIndex		
	text		
	value		
Password	defaultValue	blur	onBlur

Object	Property	Method	Event handler
	form	focus	onFocus
	name	select	
	type		
	value		
Plugin	description	None	None
	filename		
	length		
	name		
Plugins array	length	refresh	None
Radio	checked	blur	onBlur
	defaultChecked	click	onClick
	form	focus	onFocus
	length		
	name		
	type		
	value		
Reset	form	blur	onBlur
	name	click	onClick
	type	focus	onFocus
	value		
Select	form	blur	onBlur
	length	focus	onChange
	name		onFocus
	options		
	selectedIndex		
	text		
	type		
String	length	anchor	None
	prototype	big	
		blink	
		bold	
		charAt	
		fixed	
		fontcolor	
		fontsize	
		indexOf	
		italics	
		lastIndexOf	
		link	
		small	
		split	
		strike	
		sub	
		substring	
		sup	
		toLowerCase	
		toUpperCase	
Submit	form	blur	onBlur
	name	click	onClick
	type	focus	onFocus
	value		
Text	defaultValue	blur	onBlur
	form	focus	onChange
	name	select	onFocus
	type		onSelect
	value		
Textarea	defaultValue	blur	onBlur
	form	focus	onChange
	name	select	onFocus
	type		onSelect
	value		
Window	closed	alert	onBlur
	defaultStatus	blur	onError
	document	clear TimeOut	onFocus
			onLoad
	Frame	close	onUnload
	frames	confirm	
	history	focus	
	length	open	
	location	prompt	
	name	setTimeOut	
	opener		
	parent		
	scroll		
	self		
	status		
	top		
	window		

SEE ALSO THE FOLLOWING ARTICLES

C and C++ • Internet Homepages • Internet, Overview • Java • Linux • Object-Oriented Programming • Operating Systems • Programming Languages Classification • Search Engines • Unix Operating System • XML (Extensible Markup Language)

BIBLIOGRAPHY

Flanagan, D. (1997). *Javascript: The Definitive Guide*. Sebastopol, CA: O'Reilly & Associates.
Gosselin, D. (2000). Javascript. Course Technology.
Negrino, T., and Smith, D. (1999). *Javascript for the World Wide Web*. Peachpit Press.
Winsor, J., and Freeman, B. (1997). *Jumping Javascript*. Englewood Cliffs, NJ: Prentice-Hall.
Wyke, G., and Ting. (1999). *Pure JavaScript*. SAMS.

Web Sites

http://developer.netscape.com/docs/manuals/communicator/jsguide4/index.htm.
http://developer.netscape.com/tech/javascript/index.html
http://msdn.microsoft.com