CHAPTER 12

Artificial Intelligence and Expert Systems

LEARNING OBJECTIVES

1 Understand the concept and evolution of artificial intelligence
2 Understand the importance of knowledge in decision support
3 Describe the concept and evolution of rule-based expert systems (ES)
4 Understand the architecture of rule-based ES
5 Learn the knowledge engineering process used to build ES
6 Explain the benefits and limitations of rule-based systems for decision support
7 Identify proper applications of ES
8 Learn about tools and technologies for developing rule-based DSS

In addition to the use of data and mathematical models, some managerial decisions require qualitative information and the judgmental knowledge that resides in the minds of human experts. Therefore, it is necessary to find effective ways to incorporate such information and knowledge into decision support systems (DSS). A system that integrates knowledge from experts is commonly called a knowledge-based decision support system (KBDSS) or an intelligent decision support system (IDSS). A KBDSS can enhance the capabilities of decision support not only by supplying a tool that directly supports a decision maker, but also by enhancing various computerized DSS environments. The foundation for building such systems is the techniques and tools that have been developed in the area of artificial intelligence—rule-based expert systems being the primary one. This chapter introduces the essentials of artificial intelligence and provides a detailed description of expert systems.

12.1 Opening Vignette: A Web-Based Expert System for Wine Selection
12.2 Concepts and Definitions of Artificial Intelligence
12.3 The Artificial Intelligence Field
12.4 Basic Concepts of Expert Systems
12.5 Applications of Expert Systems
12.6 Structure of Expert Systems

12.7 Knowledge Engineering
12.8 Problem Areas Suitable for Expert Systems
12.9 Development of Expert Systems
12.10 Benefits, Limitations, and Critical Success Factors of Expert Systems
12.11 Expert Systems on the Web

12.1 OPENING VIGNETTE: A WEB-BASED EXPERT SYSTEM FOR WINE SELECTION

MenuVino, Inc., a Web-based wine retailer, has developed several online knowledge automation expert systems to provide expert advice on wine selection. The systems analyze a Web site visitor's individual flavor preferences to develop a personal taste profile in order to recommend wines that the customer is more likely to enjoy. The systems also match wines to particular meals as well as to food-related details such as ingredients, preparation methods, sauces, and so on. The expert knowledge embedded into the system allows gastronomic specification along with the combination of ingredients to be used collectively in determining very detailed matching of specific wines to specific meals. The advising expert systems are part of a commercial site aimed at matching users with their ideal wines.

Problem

Selecting the right wine for a given situation requires significant amount of expertise. People often buy a wine without knowing its taste, whether it suits their taste preferences, or if it is in harmony with the intended meal or occasion. Most nonexperts pick a wine based on price (higher is better!) and high-level classification (red for meat and white for fish, etc.). Beyond this simple classification schema, most people are clueless what it is that they are buying. Everyone agrees that it is preferable to savor the wines we genuinely like and appreciate. However, it is difficult to form an opinion unless you have tasted them all, which is an impossible proposition if you are buying them over the Web. Tastes are very personal to individuals, and to discover yours may be difficult even if you have the opportunity to try them all. Even then, they tend to change based on other factors such as the occasion, meal, mood, and so on.

Solution

Using the knowledge of many wine experts, MenuVino developed an expert system that mimics the advising one would receive from a guru. In fact, since it encompasses the knowledge of many experts, it may provide even better recommendations than a single human expert. The MenuVino Web-based expert system was developed with Exsys Knowledge Automation systems, which capture "deep" expert knowledge in a very complex area. The system uses Corvid's MetaBlock approach for probabilistic product selection. The user interface is run with the Corvid Servlet Runtime, which builds graphical and attractive HTML screens to ask questions and interact with the users. The system runs in both French and English.

MenuVino's expert advisor has two main functions: taste profiling and pairing of wines with food. The taste profile subsystems emulate the conversation a person would have with a wine expert (or sommelier). The taste-profiler portion of the expert system aims to identify the user's personal preferences. Using the interactive features of the Corvid expert system shell, it asks expert questions to reveal particular characteristics of the user. Once the user profile is established, the system recommends appropriate wines in different price ranges. The system also allows price limitation and feedback opportunities.

The pairing subsystem recommends the best wines for different flavor combinations. Finding a wine that will complement a meal is difficult unless you are a professional cook or a wine expert. The pairing subsystem incorporates many different types of food, their ingredients, and cooking methods. Hundreds of ingredients, condiments, and styles of preparation cover most of occidental cuisine. Want to know the ideal wine to go with braised kangaroo marinated in Bourgogne mustard and rice wine vinegar—you'll find it here (Domaine André, Mireille et Stéphane Tissot En Barberon 2004, Red). Maybe you prefer broiled sea bream in lime with coriander salt and grey pepper (Pétale de Rose côtes-de-provence rosé 2005, Rosé). This level of detail and granularity can recommend the ideal wines for most any type of meal.

Results

MenuVino provides visitors with a Web-based expert system where the knowledge and experience of wine experts is embedded into an interactive information system. It built a Web-based expert system that not only advises the user on wine selection but that also acts as an educational tool capable of democratizing and facilitating the discovery of wine. As MenuVino says, "Here, you are home. Take a seat at our table. You may be surprised, even amazed. Do not hesitate to participate. We appreciate the interaction, and all your requests will be taken in consideration . . . MenuVino—Wine has never been so simple." As good as it sounds, do not take our word for it! Try it yourself. Go to **menuvino.tv**, register as a new user for free, run the system, and get expert wine advice.

Questions for the Opening Vignette

1. Describe MenuVino's motivation for developing the Web-based expert system.
2. Do you think that expert systems are a good fit for this application domain? Explain. What other application domains may have similar needs?
3. What major difficulties might emerge in the process of developing such an expert system? How can they be overcome?
4. How is the system described in this case different from traditional DSS tools and techniques described in previous chapters?
5. Can you think of alternative tools, technologies, or solutions to the problem described in the case? How do they compare to the presented expert system solution?

What We Can Learn from This Vignette

With the help of artificial intelligence (especially with expert systems), many different types of specialized knowledge and experience can be extracted and represented in a computer. When the experts and expertise are hard to find, such an automated system can be very useful. This case presented a typical application of an expert system where the knowledge of wine experts is embedded into a Web-based information system so that it can be readily used by the non-experts. Making such specialized knowledge accessible to many users in an automated and interactive environment has great potential to boost the utility and profitability of many business applications.

Sources: Exsys, "MenuVino—Wine Advisor," **www.exsys.com/winkPDFs/CommercialOnline WineAdvisors.pdf** (accessed June 2009); and MenuVino, Inc., **menuvino.tv/gouts.php** (accessed June 2009).

12.2 CONCEPTS AND DEFINITIONS OF ARTIFICIAL INTELLIGENCE

The opening vignette illustrates that in some situations the support that can be offered by data and data-driven models may be insufficient. With regards to wine selection, support was provided by rule-based expert systems to substitute for human expertise by providing the necessary knowledge in the form of an automated and interactive information system. In addition to rule-based expert systems, several other technologies can be used to support decision situations where expertise is required. Most of these technologies use qualitative (or symbolic) knowledge rather than numeric and/or mathematical models to provide the needed support; hence, they are referred to as **knowledge-based systems (KBS)**. The overarching field of study that encompasses these technologies and underlying applications is called *artificial intelligence.*

Artificial Intelligence (AI) Definitions

Artificial intelligence (AI) is an area of computer science. Even though the term has many different definitions, most experts agree that AI is concerned with two basic ideas: (1) the study of human thought processes (to understand what intelligence is) and

(2) the representation and duplication of those thought processes in machines (e.g., computers, robots).

One well-publicized, classic definition of AI is "behavior by a machine that, if performed by a human being, would be called intelligent." Rich and Knight (1991) provided a thought-provoking definition: "Artificial intelligence is the study of how to make computers do things at which, at the moment, people are better."

A well-known application of artificial intelligence is the chess program called Deep Blue, developed by a research team at IBM (see Application Case 12.1). The system beat the famous world champion, Grand Master Garry Kasparov, in a game that usually only highly intelligent people can win.

APPLICATION CASE 12.1

Intelligent System Beats the Chess Grand Master

Games are a classic application domain for intelligent systems. An extremely successful system is the computer chess program developed by IBM. In 1997, Deep Blue, a computer system armed with artificial intelligence, beat 34-year-old Russian World Chess Champion Garry Kasparov in a six-game match. This was the first time that a computer demonstrated intelligence in an area that requires human intelligence.

The system ran on an IBM RS/6000 SP machine that was capable of examining 200 million moves per second—or 50 billion positions—in the 3 minutes allocated for a single move in a chess game. A six-person design team led by Chung-Jen Tan designed a hybrid heuristic and brute-force search model to assess the values of different moves.

In February 2003, another human–computer match between Garry Kasparov and Deep Junior, a three-time computer chess champion programmed by Air Ban and Shay Bushinsky in Israel, ended in a 3–3 tie. This further confirmed that the knowledge captured in the computer chess program could be as powerful as that of the best human player. In 2004, a computer–computer match between Deep Junior and Fritz ended in a draw. Fritz is an intelligent system designed by German scientists that can execute 350,000 operations per second.

Although these computer victories do not imply that computer intelligence will prevail, they do indicate the potential of artificial intelligence, particularly in the area of intelligent decision support. Computers armed with intelligent reasoning capabilities could help managers minimize risks and maximize performance.

Sources: IBM Research, "Deep Blue Overview," **research.ibm.com/deepblue/** (accessed June 2009); and *ChessBase News*, "Kasparov vs. Deep Junior Ends in 3-3," February 8, 2003, **chessbase.com/newsdetail.asp?newsid=782** (accessed June 2009).

To understand what artificial intelligence is, we need to examine those abilities that are considered to be signs of intelligence:

- Learning or understanding from experience
- Making sense out of ambiguous or contradictory messages
- Responding quickly and successfully to a new situation (i.e., different responses, flexibility)
- Using reasoning in solving problems and directing conduct effectively
- Dealing with perplexing situations
- Understanding and inferring in a rational way
- Applying knowledge to manipulate the environment
- Thinking and reasoning
- Recognizing and judging the relative importance of different elements in a situation

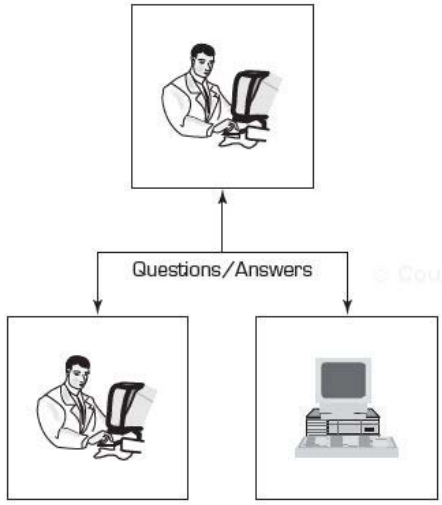

FIGURE 12.1 A Pictorial Representation of the Turing Test

Alan Turing designed an interesting test to determine whether a computer exhibits intelligent behavior; the test is called the **Turing test**. According to this test, a computer can be considered smart only when a human interviewer cannot identify the computer while conversing with both an unseen human being and an unseen computer (see Figure 12.1).

In the next section, we discuss the major characteristics of artificial intelligence.

Characteristics of Artificial Intelligence

Although the ultimate goal of AI is to build machines that mimic human intelligence, the capabilities of current commercial AI technologies are far from exhibiting any significant success in reaching this goal. Nevertheless, AI programs are continuously improving, and they increase productivity and quality by automating many tasks that require human intelligence. Artificial intelligence techniques usually have the features described in the following sections.

SYMBOLIC PROCESSING Symbolic processing is an essential characteristic of AI, as reflected in the following definition: **Artificial intelligence (AI)** is the branch of computer science that deals primarily with symbolic, non-algorithmic methods of problem solving. This definition focuses on two characteristics:

- ***Numeric versus symbolic.*** Computers were originally designed specifically to process numbers (i.e., numeric processing). However, people tend to think symbolically; our intelligence is based, in part, on our mental ability to manipulate symbols rather than just numbers. Although symbolic processing is at the core of AI, this does not mean that AI cannot use numbers and math but rather that the emphasis in AI is on the manipulation of symbols.
- ***Algorithmic versus heuristic.*** An algorithm is a step-by-step procedure that has well-defined starting and ending points and is guaranteed to find the same solution to a specific problem. Most computer architectures readily lend themselves to this type of step-by-step approach. Many human reasoning processes however tend to be non-algorithmic; in other words, our mental activities consist of more than just following logical, step-by-step procedures. Rather, human thinking relies more on rules, opinions, and gut feelings, learned from previous experiences.

HEURISTICS **Heuristics** are intuitive knowledge, or rules of thumb, learned from experience. AI deals with ways of representing knowledge using symbols with heuristics methods for processing information. By using heuristics, we do not have to rethink completely what to do every time we encounter a similar problem. For example, when a salesperson plans to visit clients in different cities, a popular heuristic is to visit the next nearest one (i.e., the nearest-neighbor heuristic). Many AI methods use heuristics to reduce the complexity of problem solving.

INFERENCING As an alternative to merely using individual heuristics, AI also includes reasoning (or inferencing) capabilities that can build higher-level knowledge using existing knowledge represented as heuristics in the form of rules. *Inference* is the process of deriving a logical outcome from a given set of facts and rules.

MACHINE LEARNING Learning is an important capability for human beings; it is one of the features that separate humans from other creatures. AI systems do not have the same learning capabilities that humans have; rather, they have simplistic learning capabilities (modeled after the human learning methods) called machine learning. Machine learning allows computer systems to monitor and sense their environmental factors and adjust their behavior to react to changes. Technically speaking, machine learning is a scientific discipline that is concerned with the design and development of algorithms that allow computers to learn based on data coming from sensors or databases. Many machine-learning techniques exist for developing intelligent information systems, some of the most popular ones are described in this part.

Section 12.2 Review Questions

1. What is artificial intelligence?
2. What are the major capabilities of artificial intelligence?
3. What are the major characteristics of artificial intelligence?
4. What are heuristics? Give an example.

12.3 THE ARTIFICIAL INTELLIGENCE FIELD

The field of AI is quite broad. In this section, we introduce its evolution, compare artificial intelligence with natural intelligence, and provide an overview of several major applications.

Evolution of Artificial Intelligence

The evolution of artificial intelligence includes five major stages. Figure 12.2 shows the evolution from 1960 onward until now.

The major event that triggered the wave of artificial intelligence is believed to be the Dartmouth Meeting. A group of computer scientists gathered at Dartmouth College to discuss the great potential of computer applications in 1956. They were confident that, given their enormous computing power, computers would be able to solve many complex problems and outperform human beings in many areas. At that time, scientists had little understanding of the complexity of human intelligence and were overly optimistic about what computers could achieve. Many solutions created at that time were primitive, and hence the stage is called the naïve solutions stage.

After several years of trial and error, scientists started focusing on developing more effective problem-solving methods, such as knowledge representation schemes, reasoning strategies, and effective search heuristics. Because the primary feature of this

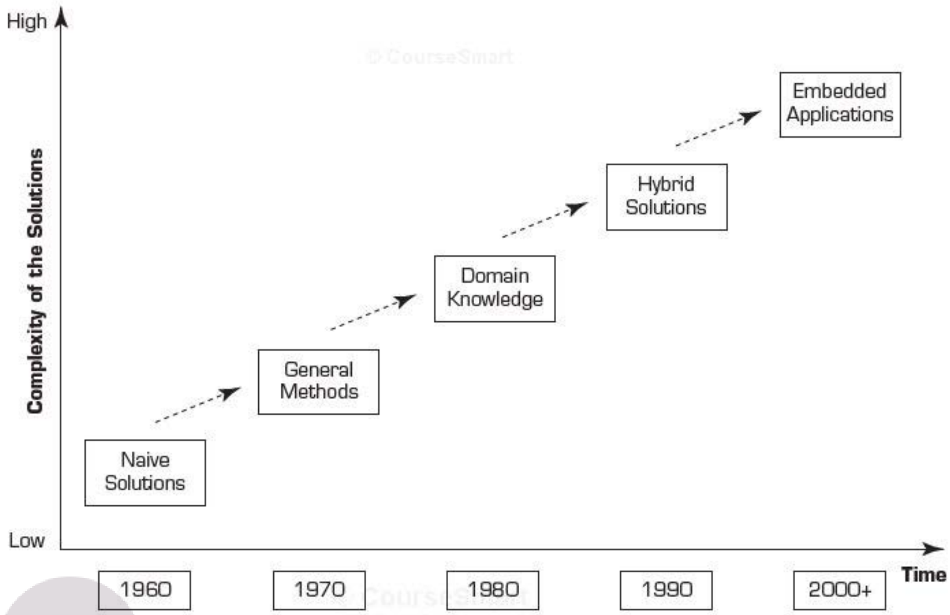

FIGURE 12.2 Stages of AI Evolution

stage was the development of general-purpose methods, it is called the general methods stage.

After building enough general-purpose methods, people started applying them to real-world applications. The applications at this stage were different from the first stage in that people already knew that it was difficult to program how to solve commonsense problems into a computer. Therefore, most applications were targeted at a narrowly defined domain with specialized knowledge. These kinds of systems are called expert or knowledge-based systems. Acquisition of expert knowledge played a key role in the development of such systems. We call this the domain knowledge stage.

After 1990, more advanced problem-solving methods were developed, and there was a strong need to integrate multiple techniques and solve problems in multiple domains. Hybrid systems, such as integrations of rule-based and case-based systems and integrations of ANN and genetic algorithms become necessary. We call this the multiple integration stage.

Since 2000, the trend has been to embed various intelligent components into popular applications. Intelligent systems and robotics continue to spread into everyday use, from video games and business rules to homeland security. The systems we are dealing with today are much smarter than earlier systems. The applications of embedded AI systems are the primary features of the embedded applications stage.

The use of artificial intelligence in BI and DSS has advantages and limitations. See Technology Insights 12.1 for a brief comparison of artificial and natural intelligence.

Applications of Artificial Intelligence

Artificial intelligence is a collection of concepts and ideas that are related to the development of intelligent systems. These concepts and ideas may be developed in different areas and be applied to different domains. In order to understand the scope of AI, therefore, we need to

TECHNOLOGY INSIGHTS 12.1 Artificial Intelligence Versus Natural Intelligence

The potential value of artificial intelligence can be better understood by contrasting it with natural, or human, intelligence. AI has several important advantages over natural intelligence:

- AI is more permanent. Natural intelligence is perishable from a commercial standpoint, in that workers can change their place of employment or forget information. However, AI is permanent as long as the computer systems and programs remain unchanged.
- AI offers ease of duplication and dissemination. Transferring a body of knowledge from one person to another usually requires a lengthy process of apprenticeship; even so, expertise can seldom be duplicated completely. However, when knowledge is embedded in a computer system, it can easily be transferred from that computer to any other computer on the Internet or on an intranet.
- AI can be less expensive than natural intelligence. There are many circumstances in which buying computer services costs less than having corresponding human power carry out the same tasks. This is especially true when knowledge is disseminated over the Web.
- AI, being a computer technology, is consistent and thorough. Natural intelligence is erratic because people are erratic; they do not always perform consistently.
- AI can be documented. Decisions made by a computer can be easily documented by tracing the activities of the system. Natural intelligence is difficult to document. For example, a person may reach a conclusion but at some later date may be unable to re-create the reasoning process that led to that conclusion or to even recall the assumptions that were part of the decision.
- AI can execute certain tasks much faster than a human can.
- AI can perform certain tasks better than many or even most people.

Natural intelligence does have some advantages over AI, such as the following:

- Natural intelligence is truly creative, whereas AI is uninspired. The ability to acquire knowledge is inherent in human beings, but with AI knowledge must be built into a carefully constructed system constrained by a large number of assumptions.
- Natural intelligence enables people to benefit from and use sensory experience directly in a synergistic way, whereas most AI systems must work with numeric and/or symbolic inputs in a sequential manner with predetermined representational forms.

see a group of areas that may be called the AI family. Figure 12.3 shows the major branches of AI applications. These applications are built on the foundation of many disciplines and technologies, including computer science, philosophy, electrical engineering, management science, psychology, and linguistics. A sample of representative application areas of AI is briefly described next.

EXPERT SYSTEMS The term *expert system* was derived from the term *knowledge-based expert system*. An **expert system (ES)** is an information system that uses human knowledge captured in a computer to solve problems that ordinarily require human expertise and reasoning. Later sections of this chapter provide more details on ES.

NATURAL LANGUAGE PROCESSING **Natural language processing (NLP)** (as described in detail in Chapter 7) is a collection of technologies aimed to provide necessary mechanisms to enable computers and computer users to communicate with each other using native human language. These technologies aim to provide a conversational type of interface between the human and the machine, in contrast to the traditional interfaces where a

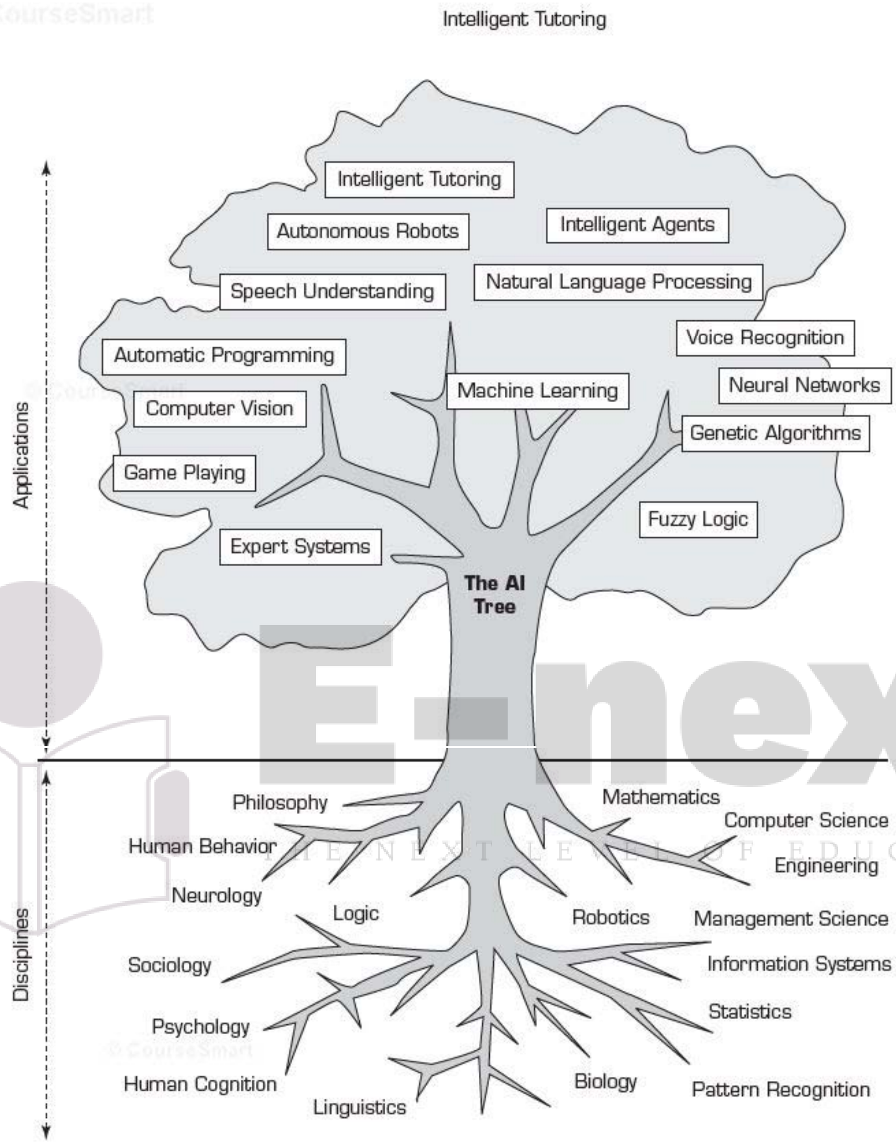

FIGURE 12.3 The Disciplines and Applications of AI

programming language that consists of computer jargon, syntax, and commands is used. NLP includes two main subfields:

- *Natural language understanding* involves the use of technology to enable computers to comprehend (i.e., "understand") human language (syntax as well as semantics) much the same way that a human being does. Here, the ultimate goal is to have computers truly understand human language.
- *Natural language generation* involves the use of technology to enable computers to produce and express ordinary human language. Here, the ultimate goal is to have people understand computers by having them converse in human languages.

Much of the success of NLP is exemplified in text-mining systems, where unstructured text documents are successfully processed (recognized, understood, and interpreted) for acquisition of new knowledge (see Chapter 7 for more details).

SPEECH (VOICE) UNDERSTANDING **Speech (voice) understanding** is the recognition and understanding of spoken language by a computer. Applications of this technology have become increasingly popular. For instance, many companies have now adopted this technology in their call centers (see Application Case 12.2). An interesting application of speech understanding is the Literacy Innovation that Speech Technology Enables (LISTEN), developed at Carnegie Mellon University, which provides a pleasant and automated environment to improve literacy by listening to children read aloud (try it at **cs.cmu.edu/~listen**).

APPLICATION CASE 12.2

Automatic Speech Recognition in Call Centers

More and more companies are using automated speech recognition technology for interacting with customers. This is particularly popular in call centers. In the United States alone, more than 50,000 call centers spend more than $90 billion dealing with customers' requests. With human operators costing around $1 per minute to maintain, speech recognition offers an enormous opportunity to companies to lower their costs and become more competitive. If done properly (by using smart technologies such as expert systems), such an automated process can also improve customer satisfaction by providing answers to their questions in a timely manner.

Charles Schwab, a U.S. discount stockbroker, introduced the first speech system for retail brokering in 1996. That year, the number of new accounts with the company increased by 41 percent, and the call centers took 97 million calls. An automated attendant in this system can understand 15,000 names of individual equities and funds, takes up to 100,000 calls per day, and is 93 percent accurate in identifying queries the first time they are made. Costs have been cut to $1 per call, down from $4 to $5 per call.

Hotel chain Wyndham International has used speech automation technology to develop intelligent agents with conversational skills to handle advanced customer service calls at Wyndham's call center. Having fast access to all database records, such an intelligent agent (enhanced with conversational skills) is often capable of resolving customer issues more accurately in a fraction of time, compared to its human counterpart.

In 2004, U.S. companies spent $480 million on speech-enabled self-service technology (SESST). By the end of this decade, over 1 billion speech-enabled service centers are expected to be in operation, replacing functions currently taking place in call centers in India and some other developing nations around the world. A call serviced through speech automation costs approximately 15 to 25 percent of the cost of a call handled by an agent in India. Voice technology is destined to have a significant impact on the outsourcing of call centers.

Sources: Adapted from "Just Talk to Me," *The Economist Technology Quarterly,* December 8, 2001; and P. Gooptu, "Threat to Call Centers: Voice Automation!" November 23, 2004, **rediff.com/money/2004/nov/23bpo.htm** (accessed June 2009).

ROBOTICS AND SENSORY SYSTEMS Sensory systems, such as vision systems, tactile systems, and signal-processing systems, when combined with AI, define a broad category of systems generally called *robots*. A **robot** is an electromechanical device that can be programmed to perform manual tasks. The Robotics Institute of America formally defines a *robot* as "a reprogrammable multifunctional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks" (Currie, 1999).

An "intelligent" robot has some kind of sensory apparatus, such as a camera, that collects information about the robot's surroundings and its operations. The intelligent part of the robot allows it to interpret the collected information and to respond and adapt to changes in its environment rather than just follow prespecified instructions. A recent report described the humanoid HRP-2 robots, named Promets, developed by the National Institute of Advanced Industrial Science and Technology in Japan, an excellent example of today's robotic technology. These robots are designed to respond to verbal instructions to perform certain tasks, such as moving chairs or turning on a TV, and they are capable of capturing three-dimensional images of objects and locating them through an infrared sensor (see Yamaguchi, 2006). A Reuters report indicated that Toyota intends to sell service robots by 2010 (Reuters, 2005).

COMPUTER VISION AND SCENE RECOGNITION **Visual recognition** is a form of computer intelligence where the digitized representation of visual information (received from one or more sensors, such as a camera, radar, an infrared or ultrasound machine, etc.) is used to accurately recognize the underlying object. The output information (recognition of the object) is then used to perform operations such as robotic movement, conveyor speed, and rerouting of out-of-spec products passing through a production line.

The basic objective of computer vision is to interpret scenarios rather than to identify individual pictures. Interpreting scenarios is defined in different ways, depending on the application and the context of the scenario. For example, in interpreting pictures taken by a satellite, it may be sufficient to identify regions of crop damage. In another example, a robot vision system can be designed to identify assembly components so that the robot can correctly affix them to the item being assembled.

INTELLIGENT COMPUTER-AIDED INSTRUCTION **Intelligent computer-aided instruction (ICAI)** refers to machines that can tutor humans. To a certain extent, such a machine can be viewed as an expert system enhanced with the knowledge of a human expert. However, the major objective of an ES is to render advice, whereas the purpose of an ICAI is to teach. Therefore, the structure of an ordinary ES needs to be tweaked to make it behave like a teacher as opposed to an advisor.

Computer-assisted instruction, which has been in use for many years, brings the power of a computer to bear on the educational process. Now AI methods are being applied to the development of ICAI systems in an attempt to create computerized tutors that shape their teaching techniques to fit the learning patterns of individual students. These are known as **intelligent tutoring systems (ITS)**, many of which have been implemented on the Web. An application can be found in Lopez et al. (2003).

AUTOMATIC PROGRAMMING Writing computer programs has always been a tedious and error-prone task. Automatic programming allows computers to automatically generate computer programs, usually based on specifications that are higher level and that are easier for humans to specify than ordinary programming languages. Automatic programming occurs with the help of AI techniques embedded within integrated development environments (IDE).

NEURAL COMPUTING **Neural computing** (or neural networks) describes a set of mathematical models that simulate the way a human brain functions. Such models have been implemented in flexible, easy-to-use software packages such as NeuroSolutions (**nd.com**), BrainMaker (**calsci.com**), and NeuroShell (**wardsystems.com**). The applications of neural networks in business are abundant. We discussed neural computing and its wide range of applications in Chapter 6.

GAME PLAYING Game playing was one of the very early application areas that AI researchers studied. It is an excellent area for investigating new AI strategies and heuristics because the outcomes are rather easy to demonstrate and measure. The success of Deep Blue (described in Application Case 12.1) is a good example of successful AI-based game development.

LANGUAGE TRANSLATION Automated translation uses computer programs to translate words and sentences from one language to another without much interpretation by humans. For example, you can use Babel Fish Translation, available at **world.altavista.com**, to try more than 20 different combinations of language translation. Similarly, you can also use Google's free tool to translate between 41 different languages (**translate.google.com**).

FUZZY LOGIC **Fuzzy logic** is a technique for processing imprecise linguistic terms. It extends the notions of logic beyond simple true/false statements to allow for partial (or even continuous) truths. Inexact knowledge and imprecise reasoning are important aspects of expertise in applying common sense to decision-making situations. In fuzzy logic, the value of true or false is replaced by the degree of set membership. For example, in the traditional Boolean logic, a person's credit record is either good or bad. In fuzzy logic, the credit record may be assessed as both good and bad, but with different degrees. See Chapter 13 and Tanaka and Niimuara (2007) for more details.

GENETIC ALGORITHMS Genetic algorithms are among the advanced search techniques that resemble the natural process of evolution (i.e., the survival of the fittest). For a specific problem, the solution template is formulated as a "chromosome" structure that contains groups, or genes (often represented with sequences of 0s and 1s), representing the values of decision variables. The genetic algorithm starts with a randomly generated population of solutions (a collection of chromosomes) and then by identifying and using the best solutions (based on a fitness function) it reproduces future generations using genetic operators (e.g., mutation and crossover). The recursive process of "evolution" continues until a satisfactory solution or some other stopping criterion is reached. See Goldberg (1994) for an excellent introduction to genetic algorithms. Ghanea-Hercock and Ghanea-Hercock (2003) discussed Java implementation of the algorithm, and Chapter 13 provides a more descriptive explanation of genetic algorithms.

INTELLIGENT AGENTS **Intelligent agents (IA)** are relatively small programs that reside (and run continuously) in a computer environment to perform certain tasks automatically and autonomously. An intelligent agent runs in the background of a computer, monitors the environment, and reacts to certain triggering conditions based on the knowledge embedded into it. A good example of an intelligent software agent is a virus detection program. It resides on your computer, scans all incoming data, and removes found viruses automatically while learning new virus types and detection methods. Intelligent agents are applied in personal digital assistants (PDAs), e-mail servers, news filtering and distribution, appointment handling, e-commerce, and automated information gathering. See Application Case 12.3 for an example and Chapter 13 for more details on this rapidly popularized AI application area.

APPLICATION CASE 12.3

Agents for Travel Planning at USC

Planning business trips is a tedious task that includes selecting a flight, reserving a hotel, and possibly reserving a car. When a schedule is set, many other decisions must be made, based on past experiences, such as whether driving to the airport or taking a taxi. The time and effort required to make more informed decisions usually outweigh the cost. Schedules can change, prices can decrease after purchasing a ticket, flight delays can result in missed connections, and hotel rooms and rental cars can be given away because of late arrivals. All these contingencies add stress for the traveler.

To address these issues, the University of Southern California (USC) developed an integrated travel planning and monitoring system called Travel Assistant. The system provides the user with the information necessary to make an informed travel plan. It uses information agents to provide information for planning and monitoring agents to trace any changes in the original plan. An information agent takes a particular information request, navigates to the appropriate Web site, extracts information from the Web site, and then returns the information as an XML document for processing.

Monitoring agents are built on top of the information agents and keep track of the status of the schedule. If any information (e.g., cancellation or delay of the flight) that might cause schedule changes is found, the agents send a message to the user. These agents perform their tasks at regular intervals. These agents can send messages about flight delays, flight cancellations, reductions in airfares, and about availability of earlier flights. They can also send faxes to a hotel.

An intelligent agent project that supports travel and tourism planning was also conducted at RMIT University in Australia. Their system was said to support mobile applications running on PDAs, cell phones, and other personal devices.

Sources: Adapted from C. Knoblock, "Agents for Gathering, Integrating, and Monitoring Information for Travel Planning," *IEEE Intelligent Systems,* Vol. 17, No. 6, 2003; and L. Mathies, L. Padgham, and B. Q. Vo, "Agent Based Travel and Tourism Planning," *Proceedings of the Autonomous Agents and Multiagent Systems Conference,* Utrecht University, the Netherlands, July 25–29, 2005.

Section 12.3 Review Questions

1. What are the major advantages of artificial intelligence over natural intelligence?
2. What are the major disadvantages of artificial intelligence compared to natural intelligence?
3. What are the major characteristics of artificial intelligence?
4. Describe an artificial intelligence application.
5. What technology can help move call centers that have been outsourced to other countries back to the United States?
6. Define *natural language processing* and describe an NLP application.
7. Define *speech recognition* and discuss one application of the technology.
8. Define *intelligent agent* and describe one application of the technology.

12.4 BASIC CONCEPTS OF EXPERT SYSTEMS

Expert systems (ES) are computer-based information systems that use expert knowledge to attain high-level decision performance in a narrowly defined problem domain. MYCIN, developed at Stanford University in the early 1980s for medical diagnosis, is the most well-known ES application. ES has also been used in taxation, credit analysis, equipment maintenance, help desk automation, environmental monitoring, and fault diagnosis. ES has been popular in large and medium-sized organizations as a sophisticated tool for improving productivity and quality (see Nedovic and Devedzic, 2002; and Nurminen et al., 2003).

The basic concepts of ES include how to determine who experts are, the definition of expertise, how expertise can be extracted and transferred from a person to a computer, and how the expert system should mimic the reasoning process of human experts. We describe these concepts in the following sections.

Experts

An **expert** is a person who has the special knowledge, judgment, experience, and skills to put his or her knowledge in action to provide sound advice and to solve complex problems in a narrowly defined area. It is an expert's job to provide knowledge about how he or she performs a task that a KBS will perform. An expert knows which facts are important and also understands and explains the dependency relationships among those facts. In diagnosing a problem with an automobile's electrical system, for example, an expert mechanic knows that a broken fan belt can be the cause for the battery to discharge.

There is no standard definition of *expert*, but decision performance and the level of knowledge a person has are typical criteria used to determine whether a particular person is an expert. Typically, experts must be able to solve a problem and achieve a performance level that is significantly better than average. In addition, experts are relative (and not absolute). An expert at a time or in a region may not be an expert in another time or region. For example, an attorney in New York may not be a legal expert in Beijing, China. A medical student may be an expert compared to the general public but may not be considered an expert in brain surgery. Experts have expertise that can help solve problems and explain certain obscure phenomena within a specific problem domain. Typically, human experts are capable of doing the following:

- Recognizing and formulating a problem
- Solving a problem quickly and correctly
- Explaining a solution
- Learning from experience
- Restructuring knowledge
- Breaking rules (i.e., going outside the general norms), if necessary
- Determining relevance and associations
- Declining gracefully (i.e., being aware of one's limitations)

Expertise

Expertise is the extensive, task-specific knowledge that experts possess. The level of expertise determines the performance of a decision. Expertise is often acquired through training, reading, and experience in practice. It includes explicit knowledge, such as theories learned from a textbook or in a classroom, and implicit knowledge, gained from experience. The following is a list of possible knowledge types:

- Theories about the problem domain
- Rules and procedures regarding the general problem domain
- Heuristics about what to do in a given problem situation
- Global strategies for solving these types of problems
- Metaknowledge (i.e., knowledge about knowledge)
- Facts about the problem area

These types of knowledge enable experts to make better and faster decisions than non-experts when solving complex problems.

Expertise often includes the following characteristics:

- Expertise is usually associated with a high degree of intelligence, but it is not always associated with the smartest person.
- Expertise is usually associated with a vast quantity of knowledge.
- Expertise is based on learning from past successes and mistakes.
- Expertise is based on knowledge that is well stored, organized, and quickly retrievable from an expert who has excellent recall of patterns from previous experiences.

Features of ES

ES must have the following features:

- ***Expertise.*** As described in the previous section, experts differ in their level of expertise. An ES must possess expertise that enables it to make expert-level decisions. The system must exhibit expert performance with adequate robustness.
- ***Symbolic reasoning.*** The basic rationale of artificial intelligence is to use symbolic reasoning rather than mathematical calculation. This is also true for ES. That is,

knowledge must be represented symbolically, and the primary reasoning mechanism must be symbolic. Typical symbolic reasoning mechanisms include backward chaining and forward chaining, which are described later in this chapter.

- ***Deep knowledge.*** Deep knowledge concerns the level of expertise in a knowledge base. The knowledge base must contain complex knowledge not easily found among nonexperts.
- ***Self-knowledge.*** ES must be able to examine their own reasoning and provide proper explanations as to why a particular conclusion was reached. Most experts have very strong learning capabilities to update their knowledge constantly. ES also need to be able to learn from their successes and failures as well as from other knowledge sources.

The development of ES is divided into two generations. Most first-generation ES use if-then rules to represent and store their knowledge. The second-generation ES are more flexible in adopting multiple knowledge representation and reasoning methods. They may integrate fuzzy logic, neural networks, or genetic algorithms with rule-based inference to achieve a higher level of decision performance. A comparison between conventional systems and ES is given in Table 12.1. A simple scenario of ES usage is given in Technology Insights 12.2.

TABLE 12.1 Comparison of Conventional Systems and Expert Systems

Conventional Systems	Expert Systems
Information and its processing are usually combined in one sequential program.	The knowledge base is clearly separated from the processing (inference) mechanism (i.e., knowledge rules are separated from the control).
The program does not make mistakes (programmers or users do).	The program may make mistakes.
Conventional systems do not (usually) explain why input data are needed or how conclusions are drawn.	Explanation is a part of most ES.
Conventional systems require all input data. They may not function properly with missing data unless planned for.	ES do not require all initial facts. ES can typically arrive at reasonable conclusions with missing facts.
Changes in the program are tedious (except in DSS).	Changes in the rules are easy to make.
The system operates only when it is completed.	The system can operate with only a few rules (as the first prototype).
Execution is done on a step-by-step (algorithmic) basis.	Execution is done by using heuristics and logic.
Large databases can be effectively manipulated.	Large knowledge bases can be effectively manipulated.
Conventional systems represent and use data.	ES represent and use knowledge.
Efficiency is usually a major goal.	
Effectiveness is important only for DSS.	Effectiveness is the major goal.
Conventional systems easily deal with quantitative data.	ES easily deal with qualitative data.
Conventional systems use numeric data representations.	ES use symbolic and numeric knowledge representations.
Conventional systems capture, magnify, and distribute access to numeric data or information.	ES capture, magnify, and distribute access to judgment and knowledge.

TECHNOLOGY INSIGHTS 12.2 Sample Session of a Rule-Based ES

A rule-based ES contains rules in its knowledge base. The rules are used to generate questions for the user. The user's responses allow the system to provide recommendations. Suppose that you have an ES that recommends notebook computers based on a customer's needs. The following is a possible consultation session:

1. What is your primary task to be performed on the notebook computer?
- ☐ Word processing
- ☐ Communications
- ☐ Multimedia applications

Answer: 1 (click the first checkbox)

2. Where are you going to use the notebook most often?
- ☐ In the office
- ☐ While traveling

Answer: 2 (click the second checkbox)

3. What is your budget range?
- ☐ Below $1K
- ☐ Between $1K and $2K
- ☐ Above $2K

Answer: 2 (click the second checkbox)

System recommendation: You should consider buying a Dell Latitude X1.

Reasons: It is light (2.5 pounds) and more suitable for your word processing and travel needs; it is also priced at $1,950, which fits your budget.

Of course, ES are not exact replicas of the real experts. Compared to real experts, they have advantages and shortcomings. Table 12.2 shows a simple comparison between human experts and ES on several key features.

TABLE 12.2 Differences Between Human Experts and Expert Systems

Features	Human Experts	Expert Systems
Mortality	Yes	No
Knowledge transfer	Difficult	Easy
Knowledge documentation	Difficult	Easy
Decision consistency	Low	High
Unit usage cost	High	Low
Creativity	High	Low
Adaptability	High	Medium
Knowledge scope	Broad	Narrow
Knowledge type	Common sense and technical	Technical
Knowledge content	Experience	Rules and symbolic models

APPLICATION CASE 12.4

Expert System Helps in Identifying Sport Talents

In the world of sports, recruiters are constantly looking for new talent and parents want to identify the sport that is the most appropriate for their child. Identifying the most plausible match between a person (characterized by a large number of unique qualities and limitations) and a specific sport is anything but a trivial task. Such a matching process requires adequate information about the specific person (i.e., values of certain characteristics), as well as the deep knowledge of what this information should include (i.e., the types of characteristics). In other words, expert knowledge is what is needed in order to accurately predict the right sport (with the highest success possibility) for a specific individual.

It is very hard (if not impossible) to find the true experts for this difficult matchmaking problem. Because the domain of the specific knowledge is divided into various types of sports, the experts have in-depth knowledge of the relevant factors only for a specific sport (that they are an expert of), and beyond the limits of that sport they are not any better than an average spectator. In an ideal case, you would need experts from a wide range of sports brought together into a single room to collectively create a matchmaking decision. Because such a setting is not feasible in the real world, one might consider creating it in the computer world using expert systems. Because expert systems are known to incorporate knowledge from multiple experts, this situation seems to fits well with an expert system type solution.

In a recent publication Papic et al. (2009) reported on an expert system application for the identification of sports talents. Tapping into the knowledge of a large number of sports experts, they have built a knowledge base of a comprehensive set of rules that maps the expert-driven factors (e.g., physical and cardiovascular measurement, performance test, skill assessments) to different sports. Taking advantage of the inexact representation capabilities of fuzzy logic, they managed to incorporate the exact natural reasoning of the expert knowledge into their advising system.

The system was built as a Web-based DSS using the ASP.NET development platform. Once the system development was completed, it was tested for verification and validation purposes. The system's prediction results were evaluated by experts using real cases collected from the past several years. Comparison was done between the sport proposed by the expert system and the actual outcome of the person's sports career. Additionally, the expert system output and the human expert suggestions were compared using a large number of test cases. All tests showed high reliability and accuracy of the developed system.

Sources: V. Papic, N. Rogulj, and V. Pletina, "Identification of Sport Talents Using a Web-oriented Expert System with a Fuzzy Module," *Expert Systems with Applications,* Vol. 36, 2009, pp. 8830–8838; and N. Rogulj, V. Papic, and V. Pletina, "Development of the Expert System for Sport Talents Detection," *WSEAS Transactions on Information Science and Applications,* Vol. 9, No. 3, 2006, pp. 1752–1755.

Section 12.4 Review Questions

1. What is an ES?
2. Explain why we need ES.
3. What are the major features of ES?
4. What is expertise? Provide an example.
5. Define *deep knowledge* and give an example of it.

12.5 APPLICATIONS OF EXPERT SYSTEMS

ES have been applied to many business and technological areas to support decision making. Application Case 12.5 shows a few real-world applications of ES. Table 12.3 shows some representative ES and their application domains.

APPLICATION CASE 12.5

Sample Applications of ES

ES can have many applications. The following are a few examples.

Customer Support at Logitech

Logitech is one of the largest vendors of mouse devices and Web cameras in the world. Because the company offers many different models of these tools, customer support is a major challenge. To take advantage of the Internet and technologies in intelligent systems, Logitech deploys an interactive knowledge portal to provide Web-based self-help customer support to its QuickCam customers in North America. The noHold Knowledge Platform emulates the way a human would interact with a customer, allows the user to ask questions or describe problems in natural language, and carries on an intelligent conversation with the user until it has enough information to provide an accurate answer.

China's Freight Train System

An ES was developed in China to allocate freight cars and determine what and how much to load on each car. The ES is integrated with the existing MIS and is distributed to many users.

Electricity Market Forecaster

EnvaPower developed an electricity market forecasting system, called MarketMonitor, that uses AI techniques to gather, synthesize, and analyze a large number of factors that may affect the consumption of electricity.

Rule-Based Engine for Mobile Games

In reaction to the rapid growth in mobile devices and entertainment needs, a group of researchers in the United Kingdom is creating a rule-based AI engine that can support the development of games on mobile devices. The system allows downloadable games to have AI components so that they can become more intelligent.

SEI Investment's Financial Diagnosis System

SEI Investment uses business rules management technologies to create an enabling platform for delivering "financial wellness" solutions to its clients. The system includes rules for regulatory and application checks, transaction management governance, and automation of transactions without human interruption.

Sources: "Logitech Deploys Online Customer Support," *Expert Systems,* November 2001; G. Geng, B. Zhang, J. Zhu, and C. H. Zhong, "Applying AI to Railway Freight Loading," *Expert Systems with Applications,* January 1999; and L. Hall, A. Gordon, R. James, and L. Newell, "A Lightweight Rule-Based AI Engine for Mobile Games," *ACM SIG International Conference on Advances in Computer Entertainment Technologies,* 2004.

TABLE 12.3 Sample Applications of Expert Systems

Expert System	Organization	Application Domain
Classical Applications		
MYCIN	Stanford University	Medical diagnosis
XCON	DEC	System configuration
Expert Tax	Coopers & Lybrand	Tax planning
Loan Probe	Peat Marwick	Loan evaluation
La-Courtier	Cognitive Systems	Financial planning
LMOS	Pacific Bell	Network management
PROSPECTOR	Stanford Research Institute	Discovery of new mineral deposits
New Applications		
Fish-Expert	North China	Disease diagnosis in fish
HelpDeskIQ	BMC Remedy	Help desk management
Authorete	Haley	Business rule automation
eCare	CIGNA	Insurance claims
SONAR	NSAD	Stock market monitoring

Classical Applications of ES

Early ES applications, such as DENDRAL for molecular structure identification and MYCIN for medical diagnosis, were primarily in the science domain. XCON for configuration of the VAX computer system at Digital Equipment Corp. (a major producer of minicomputers around 1990 that was later taken over by Compaq) was a successful example in business.

DENDRAL The DENDRAL project was initiated by Edward Feigenbaum in 1965. It used a set of knowledge- or rule-based reasoning commands to deduce the likely molecular structure of organic chemical compounds from known chemical analyses and mass spectrometry data.

DENDRAL proved to be fundamentally important in demonstrating how rule-based reasoning could be developed into powerful knowledge engineering tools and led to the development of other rule-based reasoning programs at the Stanford Artificial Intelligence Laboratory (SAIL). The most important of those programs was MYCIN.

MYCIN MYCIN is a rule-based ES that diagnoses bacterial infections of the blood. It was developed by a group of researchers at Stanford University in the 1970s. By asking questions and backward chaining through a rule base of about 500 rules, MYCIN can recognize approximately 100 causes of bacterial infections, which allows the system to recommend effective drug prescriptions. In a controlled test, its performance was rated to be equal that of human specialists. The reasoning and uncertainty processing methods used in MYCIN are pioneers in the area and have generated long-term impact in ES development.

XCON XCON, a rule-based system developed at Digital Equipment Corp., used rules to help determine the optimal system configuration that fit customer requirements. The system was able to handle a customer request within 1 minute that typically took the sales team 20 to 30 minutes. With the ES, service accuracy increased to 98 percent, from a manual approach with an accuracy of 65 percent, saving millions of dollars every year.

Newer Applications of ES

More recent applications of ES include risk management, pension fund advising, business rule automation, automated market surveillance, and homeland security.

CREDIT ANALYSIS SYSTEMS ES have been developed to support the needs of commercial lending institutions. ES can help a lender analyze a customer's credit record and determine a proper credit line. Rules in the knowledge base can also help assess risk and risk-management policies. These kinds of systems are used in over one-third of the top 100 commercial banks in the United States and Canada.

PENSION FUND ADVISORS Nestlé Foods Corporation has developed an ES that provides information on an employee's pension fund status. The system maintains an up-to-date knowledge base to give participants advice concerning the impact of regulation changes and conformance with new standards. A system offered on the Internet at the Pingtung Teacher's College in Taiwan has functions that allow participants to plan their retirement through a what-if analysis that calculates their pension benefits under different scenarios.

AUTOMATED HELP DESKS BMC Remedy (**remedy.com**) offers HelpDeskIQ, a rule-based help desk solution for small businesses. This browser-based tool enables small businesses to deal with customer requests more efficiently. Incoming e-mails automatically pass into HelpDeskIQ's business rule engine. The messages are sent to the proper technician, based on defined priority and status. The solution assists help desk technicians in resolving problems and tracking issues more effectively.

HOMELAND SECURITY PortBlue Corp. (**portblue.com/pub/solutions-homeland-security**) has developed an ES for homeland security. It is designed to assess terrorist threats and provide (1) an assessment of vulnerability to terrorist attack, (2) indicators of terrorist surveillance activity, and (3) guidance for managing interactions with potential terrorists. Similarly, the U.S. Internal Revenue Service uses intelligent systems to detect irregular international financial information and to block possible money laundering and terrorist financing.

MARKET SURVEILLANCE SYSTEMS The National Association of Security Dealers (NASD) has developed an intelligent surveillance system called Securities Observation, New Analysis, and Regulations (SONAR) that uses data mining, rule-based inference, knowledge-based data representation, and NLP to monitor the stock markets and futures markets for suspicious patterns. The system generates 50 to 60 alerts per day for review by several groups of regulatory analysts and investigators (Goldberg et al., 2003).

BUSINESS PROCESS REENGINEERING SYSTEMS Reengineering involves the exploitation of information technology to improve business processes. KBS are used in analyzing the workflow for business process reengineering. For example, Gensym's System Performance Analysis Using Real-Time Knowledge-based Simulation (SPARKS) can help model the formal and informal knowledge, skills, and competencies that must be embedded in a reengineered system. SPARKS has three components: a process flow model, a resource model, and work volumes and descriptions.

Areas for ES Applications

As indicated in the preceding examples, ES have been applied commercially in a number of areas, including the following:

- ***Finance.*** Finance ES include insurance evaluation, credit analysis, tax planning, fraud prevention, financial report analysis, financial planning, and performance evaluation.
- ***Data processing.*** Data processing ES include system planning, equipment selection, equipment maintenance, vendor evaluation, and network management.
- ***Marketing.*** Marketing ES include customer relationship management, market analysis, product planning, and market planning.
- ***Human resources.*** Examples of human resources ES are human resources planning, performance evaluation, staff scheduling, pension management, and legal advising.
- ***Manufacturing.*** Manufacturing ES include production planning, quality management, product design, plant site selection, and equipment maintenance and repair.
- ***Homeland security.*** Homeland security ES include terrorist threat assessment and terrorist finance detection.
- ***Business process automation.*** ES have been developed for help desk automation, call center management, and regulation enforcement.
- ***Health care management.*** ES have been developed for bioinformatics and other health care management issues.

Now that you are familiar with a variety of different ES applications, it is time to look at the internal structure of an ES and how the goals of the ES are achieved.

Section 12.5 Review Questions

1. What is MYCIN's problem domain?
2. Name two applications of ES in finance and describe their benefits.
3. Name two applications of ES in marketing and describe their benefits.
4. Name two applications of ES in homeland security and describe their benefits.

12.6 STRUCTURE OF EXPERT SYSTEMS

ES can be viewed as having two environments: the development environment and the consultation environment (see Figure 12.4). An ES builder uses the **development environment** to build the necessary components of the ES and to populate the knowledge base with appropriate representation of the expert knowledge. A nonexpert uses the **consultation environment** to obtain advice and to solve problems using the expert knowledge embedded into the system. These two environments can be separated at the end of the system development process.

The three major components that appear in virtually every ES are the knowledge base, the inference engine, and the user interface. In general, though, an ES that interacts with the user can contain the following additional components:

- Knowledge acquisition subsystem
- Blackboard (workplace)
- Explanation subsystem (justifier)
- Knowledge-refining system

Currently, most ES do not contain the knowledge refinement component. A brief description of each of these components follows.

Knowledge Acquisition Subsystem

Knowledge acquisition is the accumulation, transfer, and transformation of problem-solving expertise from experts or documented knowledge sources to a computer program

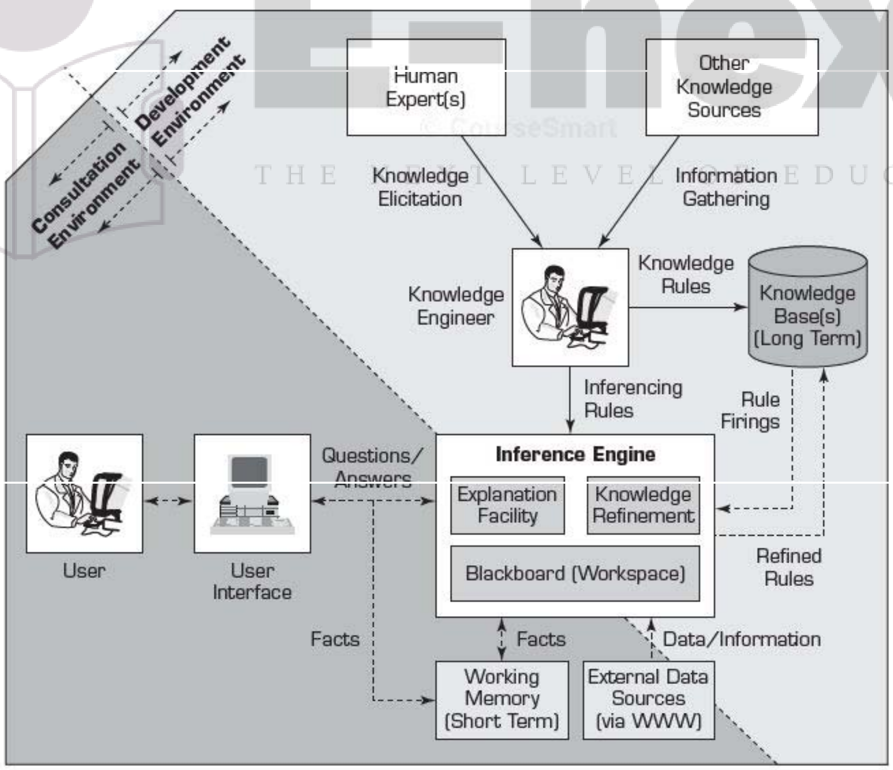

FIGURE 12.4 Structure/Architecture of an Expert System

for constructing or expanding the knowledge base. Potential sources of knowledge include human experts, textbooks, multimedia documents, databases (public and private), special research reports, and information available on the Web.

Currently, most organizations have collected a large volume of data, but the organization and management of organizational knowledge are limited. Knowledge acquisition deals with issues such as making tacit knowledge explicit and integrating knowledge from multiple sources.

Acquiring knowledge from experts is a complex task that often creates a bottleneck in ES construction. In building large systems, a knowledge engineer, or knowledge elicitation expert, needs to interact with one or more human experts in building the knowledge base. Typically, the **knowledge engineer** helps the expert structure the problem area by interpreting and integrating human answers to questions, drawing analogies, posing counterexamples, and bringing conceptual difficulties to light.

Knowledge Base

The **knowledge base** is the foundation of an ES. It contains the relevant knowledge necessary for understanding, formulating, and solving problems. A typical knowledge base may include two basic elements: (1) facts that describe the characteristics of a specific problem situation (or *fact base*) and the theory of the problem area and (2) special heuristics or rules (or *knowledge nuggets*) that represent the deep expert knowledge to solve specific problems in a particular domain. Additionally, the inference engine can include general-purpose problem-solving and decision-making rules (or *meta-rules*—rules about how to process production rules).

It is important to differentiate between the knowledge base of an ES and the knowledge base of an organization. The knowledge stored in the knowledge base of an ES is often represented in a special format so that it can be used by a software program (i.e., an expert system shell) to help users solve a particular problem. The organizational knowledge base, however, contains various kinds of knowledge in different formats (most of which is represented in a way that it can be consumed by people) and may be stored in different places. The knowledge base of an ES is a special case and only a very small subset of an organization's knowledge base.

Inference Engine

The "brain" of an ES is the inference engine, also known as the *control structure* or the *rule interpreter* (in rule-based ES). This component is essentially a computer program that provides a methodology for reasoning about information in the knowledge base and on the blackboard to formulate appropriate conclusions. The inference engine 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 a consultation takes place. It is further discussed in Section 12.7.

User Interface

An ES contains a language processor for friendly, problem-oriented communication between the user and the computer, known as the **user interface**. This communication can best be carried out in a natural language. Due to technological constraints, most existing systems use the graphical or textual question-and-answer approach to interact with the user.

Blackboard (Workplace)

The **blackboard** is an area of working memory set aside as a database for description of the current problem, as characterized by the input data. It is also used for recording intermediate results, hypotheses, and decisions. Three types of decisions can be recorded on the blackboard: a plan (i.e., how to attack the problem), an agenda (i.e., potential actions awaiting execution), and a solution (i.e., candidate hypotheses and alternative courses of action that the system has generated thus far).

Consider this example. When your car fails to start, you can enter the symptoms of the failure into a computer for storage in the blackboard. As the result of an intermediate hypothesis developed in the blackboard, the computer may then suggest that you do some additional checks (e.g., see whether your battery is connected properly) and ask you to report the results. This information is also recorded in the blackboard. Such an iterative process of populating blackboard with values of hypotheses and facts continues until the reason of the failure is identified.

Explanation Subsystem (Justifier)

The ability to trace responsibility for conclusions to their sources is crucial both in the transfer of expertise and in problem solving. The **explanation subsystem** can trace such responsibility and explain the ES behavior by interactively answering questions such as these:

- Why was a certain question asked by the ES?
- How was a certain conclusion reached?
- Why was a certain alternative rejected?
- What is the complete plan of decisions to be made in reaching the conclusion? For example, what remains to be known before a final diagnosis can be determined?

In most ES, the first two questions (why and how) are answered by showing the rule that required asking a specific question and showing the sequence of rules that were used (fired) to derive the specific recommendations, respectively.

Knowledge-Refining System

Human experts have a **knowledge-refining system**; that is, they can analyze their own knowledge and its effectiveness, learn from it, and improve on it for future consultations. Similarly, such evaluation is necessary in expert systems so that a program can analyze the reasons for its success or failure, which could lead to improvements resulting in a more accurate knowledge base and more effective reasoning.

The critical component of a knowledge refinement system is the self-learning mechanism that allows it to adjust its knowledge base and its processing of knowledge based on the evaluation of its recent past performances. Such an intelligent component is not yet mature enough to appear in many commercial ES tools but is being developed in experimental ES at several universities and research institutions.

APPLICATION CASE 12.6

A Fashion Mix-and-Match Expert System

In today's highly competitive marketplace, fashion clothing retail businesses are looking for ways to meet and exceed customers' needs and wants. One strategy that is becoming a routine part of fashion retail businesses is providing customers with mix-and-match expert advice. Customers who like a particular article of clothing are often interested in other items that would fashionably go with it. Such advice enhances customer satisfaction (and loyalty) and also improves sales.

The mix-and-match recommendations are usually provided by individual sales personnel who have the necessary knowledge and experience in the latest fashion trends as well as on the available clothing items in the inventory. Good advisors are either hard to find or are prohibitively expensive for many smaller retailers. The characteristics of this problem resemble those addressed by many successful ES applications. By eliciting and representing the mix-and-match knowledge in the form of production rules, a capable expert system can be created.

Wong et al. (2009) developed an interesting ES to automate fashion mix-and-match advising. The fashion mix-and-match ES was developed to provide customers with professional and systematic mix-and-match recommendations automatically. Based on knowledge acquired from fashion designers, the system emulates the advising decisions on apparel coordination. A set of attributes of apparel coordination were identified and formulated, and their corresponding importance was defined according to the designers' opinions. Wong and colleagues devised a *fashion coordination satisfaction index* that uses a fuzzy screening approach to represent the degree of coordination between pairs of apparel articles.

In order to automatically recognize clothing items, they proposed an item-level RFID system. When a customer picks a clothing item, the automated mix-and-match ES detects it via RFID and accesses the store database to query the characteristics that it needs to know about the item. The inference mechanism then takes over to advise on clothing items that match with the selection and the customer's taste. Their experimental results demonstrated that the proposed system can generate effective mix-and-match recommendations. The system has been integrated with a smart dressing system used in a fashion store in Hong Kong.

Source: W. K. Wong, X. H. Zeng, W. M. R. Au, P. Y. Moka, and S. Y. S. Leung, "A Fashion Mix-and-Match Expert System for Fashion Retailers Using a Fuzzy Screening Approach," *Expert Systems with Applications,* Vol. 36, 2009, pp. 1750–1764.

Section 12.6 Review Questions

1. Describe the ES development environment.
2. List and define the major components of an ES.
3. What are the major activities performed in the ES blackboard (workplace)?
4. What are the major roles of the explanation subsystem?
5. Describe the difference between a knowledge base of an ES and an organizational knowledge base.

12.7 KNOWLEDGE ENGINEERING

The collection of intensive activities encompassing the acquisition of knowledge from human experts (and other information sources) and conversion of this knowledge into a repository (commonly called a *knowledge base*) are called **knowledge engineering**. The term *knowledge engineering* was first defined in the pioneering work of Feigenbaum and McCorduck (1983) as the art of bringing the principles and tools of artificial intelligence research to bear on difficult application problems requiring the knowledge of experts for their solutions. Knowledge engineering requires cooperation and close communication between the human experts and the knowledge engineer to successfully codify and explicitly represent the rules (or other knowledge-based procedures) that a human expert uses to solve problems within a specific application domain. The knowledge possessed by human experts is often unstructured and not explicitly expressed. A major goal of knowledge engineering is to help experts articulate *how they do what they do* and to document this knowledge in a reusable form.

Knowledge engineering can be viewed from two perspectives: narrow and broad. According to the narrow perspective, knowledge engineering deals with the steps necessary to build expert systems (i.e., knowledge acquisition, knowledge representation, knowledge validation, inferencing, and explanation/justification). Alternatively, according to the broad perspective, the term describes the entire process of developing and maintaining any intelligent systems. In this book, we use the narrow definition. Following are the five major activities in knowledge engineering:

- ***Knowledge acquisition.*** Knowledge acquisition involves the acquisition of knowledge from human experts, books, documents, sensors, or computer files. The knowledge may be specific to the problem domain or to the problem-solving procedures, it may be general knowledge (e.g., knowledge about business), or it may be metaknowledge (knowledge about knowledge). (By *metaknowledge,* we mean information about how experts use their knowledge to solve problems and about problem-solving procedures in general.) Byrd (1995) formally verified that knowledge acquisition is the bottleneck in ES development today; thus, much theoretical and applied research is still being conducted in this area. An analysis of more than 90 ES applications and their knowledge acquisition techniques and methods is available in Wagner et al. (2003).
- ***Knowledge representation.*** Acquired knowledge is organized so that it will be ready for use, in an activity called knowledge representation. This activity involves preparation of a knowledge map and encoding of the knowledge in the knowledge base.
- ***Knowledge validation.*** Knowledge validation (or verification) involves validating and verifying the knowledge (e.g., by using test cases) until its quality is acceptable. Test results are usually shown to a domain expert to verify the accuracy of the ES.
- ***Inferencing.*** This activity involves the design of software to enable the computer to make inferences based on the stored knowledge and the specifics of a problem. The system can then provide advice to nonexpert users.
- ***Explanation and justification.*** This step involves the design and programming of an explanation capability (e.g., programming the ability to answer questions such as why a specific piece of information is needed by the computer or how a certain conclusion was derived by the computer).

Figure 12.5 shows the process of knowledge engineering and the relationships among the knowledge engineering activities. Knowledge engineers interact with human experts or collect documented knowledge from other sources in the knowledge acquisition stage. The acquired knowledge is then coded into a representation scheme to create a knowledge base. The knowledge engineer can collaborate with human experts or use test cases to verify and validate the knowledge base. The validated knowledge can be used in a knowledge-based system to solve new problems via machine inference and to explain the generated recommendation. Details of these activities are discussed in the following sections.

Knowledge Acquisition

Knowledge is a collection of specialized facts, procedures, and judgment usually expressed as rules. Knowledge can come from one or from many sources, such as books, films, computer databases, pictures, maps, stories, news articles, and sensors, as well as from human experts. Acquisition of knowledge from human experts (often called *knowledge elicitation*) is arguably the most valuable and most challenging task in knowledge acquisition. The classical knowledge elicitation methods, which are also called *manual methods*, include interviewing (i.e., structured, semistructured, unstructured), tracking the reasoning process, and observing. Because these manual methods are slow, expensive, and sometimes inaccurate, the ES community has been developing semi-automated and fully automated means to acquire knowledge. These

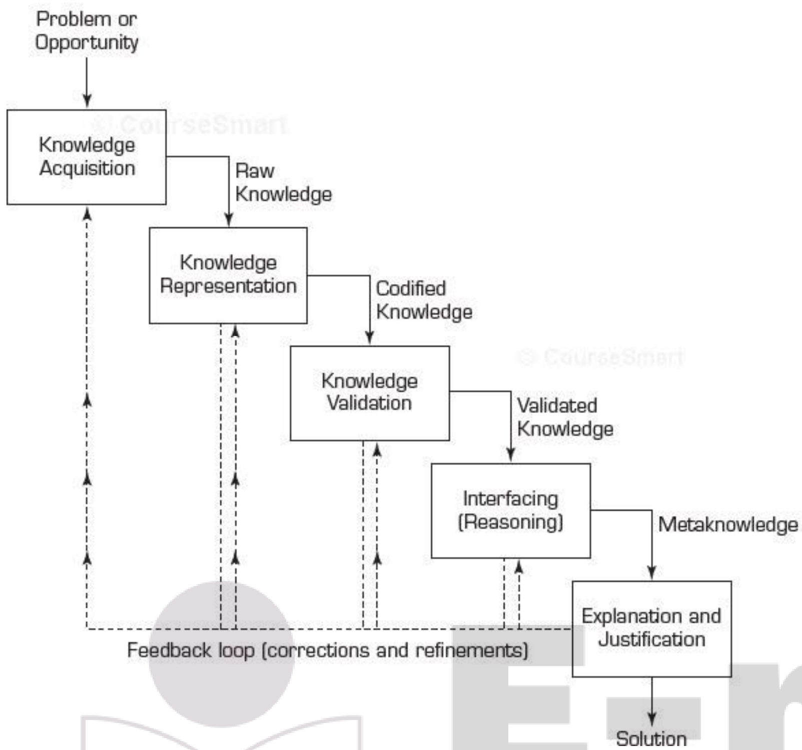

FIGURE 12.5 The Process of Knowledge Engineering

techniques, which rely on computers and AI techniques, aim to minimize the involvement of the knowledge engineer and the human experts in the process. Despite its disadvantages, in real-world ES projects the traditional knowledge elicitation techniques still dominate.

TECHNOLOGY INSIGHTS 12.3 Difficulties in Knowledge Acquisition

Acquiring knowledge from experts is not an easy task. The following are some factors that add to the complexity of knowledge acquisition from experts and its transfer to a computer:

- Experts may not know how to articulate their knowledge or may be unable to do so.
- Experts may lack time or may be unwilling to cooperate.
- Testing and refining knowledge are complicated.
- Methods for knowledge elicitation may be poorly defined.
- System builders tend to collect knowledge from one source, but the relevant knowledge may be scattered across several sources.
- System builders may attempt to collect documented knowledge rather than use experts. The knowledge collected may be incomplete.
- It is difficult to recognize specific knowledge when it is mixed up with irrelevant data.
- Experts may change their behavior when they are observed or interviewed.
- Problematic interpersonal communication factors may affect the knowledge engineer and the expert.

TABLE 12.4 Advantages and Shortcomings of Using Multiple Experts

Advantages	Shortcomings
• On average, multiple experts make fewer mistakes than a single expert.	• Fear on the part of some domain experts of senior experts or a supervisor (i.e., lack of confidentiality).
• Elimination of the need for finding and using the world-class expert (who is difficult to identify and acquire).	• Compromising solutions generated by a group with conflicting opinions.
• Wider domain than a single expert.	• Groupthink.
• Synthesis of expertise.	• Dominating experts (i.e., controlling, not letting others speak).
• Enhanced quality due to synergy among experts.	• Wasted time in group meetings and scheduling difficulties.

A critical element in the development of an ES is the identification of experts. The usual approach to mitigate this problem is to build ES for a very narrow application domain in which expertise is more clearly defined. Even then, there is a very good chance that one might find more than one expert with different (sometime conflicting) expertise. In such situations, one might choose to use multiple experts in the knowledge elicitation process. Some of the advantages and shortcomings of using multiple experts are listed in Table 12.4.

Four possible scenarios, or configurations, can be used when working with multiple experts (see O'Leary, 1993; and Rayham and Fairhurst, 1999): individual experts, primary and secondary experts, small groups, and panels. Each is described in the following list:

- ***Individual experts.*** In this case, several experts contribute knowledge individually. Using multiple experts in this manner relieves the knowledge engineer of the stress associated with multiple expert teams. However, this approach requires that the knowledge engineer have a means of resolving conflicts and handling multiple lines of reasoning.
- ***Primary and secondary experts.*** A primary expert may be responsible for validating information retrieved from other domain experts. Knowledge engineers may initially consult the primary expert for guidance in domain familiarization, refinement of knowledge acquisition plans, and identification of potential secondary experts.
- ***Small groups.*** Several experts may be consulted together and asked to provide agreed-upon information. Working with small groups of experts allows the knowledge engineer to observe alternative approaches to the solution of a problem and the key points made in solution-oriented discussions among experts.
- ***Panels.*** To meet goals for verification and validation of ongoing development efforts, a program may establish a council of experts. The members of the council typically meet together at times scheduled by the developer for the purpose of reviewing knowledge base development efforts, content, and plans. In many cases, the functionality of the ES is tested against the expertise of such a panel.

Knowledge Verification and Validation

Knowledge acquired from experts needs to be evaluated for quality, including evaluation, validation, and verification. These terms are often used interchangeably. We use the definitions provided by O'Keefe et al. (1987):

- *Evaluation* is a broad concept. Its objective is to assess an ES's overall value. In addition to assessing acceptable performance levels, it analyzes whether the system would be usable, efficient, and cost-effective.

- *Validation* is the part of evaluation that deals with the performance of the system (e.g., as it compares to the expert's). Simply stated, validation is building the right system (i.e., substantiating that a system performs with an acceptable level of accuracy).
- *Verification* is building the system right or substantiating that the system is correctly implemented to its specifications.

In the realm of ES, these activities are dynamic because they must be repeated each time the prototype is changed. In terms of the knowledge base, it is necessary to ensure that the right knowledge base (i.e., that the knowledge is valid) is used. It is also essential to ensure that the knowledge base has been constructed properly (i.e., verification).

Knowledge Representation

Once validated, the knowledge acquired from experts or induced from a set of data must be represented in a format that is both understandable by humans and executable on computers. A variety of knowledge representation methods is available: production rules, semantic networks, frames, objects, decision tables, decision trees, and predicate logic. Next we explain the most popular method—production rules.

PRODUCTION RULES **Production rules** are the most popular form of knowledge representation for expert systems. Knowledge is represented in the form of condition/action pairs: IF this condition (or premise or antecedent) occurs, THEN some action (or result or conclusion or consequence) will (or should) occur. Consider the following two examples:

- If the stop light is red AND you have stopped, THEN a right turn is okay.
- If the client uses purchase requisition forms AND the purchase orders are approved and purchasing is separate from receiving AND accounts payable AND inventory records, THEN there is strongly suggestive evidence (90 percent probability) that controls to prevent unauthorized purchases are adequate. (This example from an internal control procedure includes a probability.)

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, yielding better results than the sum of the results of the individual rules. In some sense, rules can be viewed as a simulation of the cognitive behavior of human experts. According to this view, rules are not just a neat formalism to represent knowledge in a computer; rather, they represent a model of actual human behavior.

KNOWLEDGE AND INFERENCE RULES Two types of rules are common in artificial intelligence: knowledge and inference. **Knowledge rules**, or *declarative rules,* state all the facts and relationships about a problem. **Inference rules**, or *procedural rules,* offer advice on how to solve a problem, given that certain facts are known. The knowledge engineer separates the two types of rules: Knowledge rules go to the knowledge base, whereas inference rules become part of the inference engine. For example, assume that you are in the business of buying and selling gold. The knowledge rules might look like this:

Rule 1: IF an international conflict begins, THEN the price of gold goes up.

Rule 2: IF the inflation rate declines, THEN the price of gold goes down.

Rule 3: IF the international conflict lasts more than 7 days and IF it is in the Middle East, THEN buy gold.

Inference rules contain rules about rules and thus are also called meta-rules. They pertain to other rules (or even to themselves). Inference (procedural) rules may look like this:

Rule 1: IF the data needed are not in the system, THEN request them from the user.

Rule 2: IF more than one rule applies, THEN deactivate any rules that add no new data.

Inferencing

Inferencing (or reasoning) is the process of using the rules in the knowledge base along with the known facts to draw conclusions. Inferencing requires some logic embedded in a computer program to access and manipulate the stored knowledge. This program is an algorithm that, with the guidance of the inferencing rules, controls the reasoning process and is usually called the **inference engine**. In rule-based systems, it is also called the *rule interpreter*.

The inference engine directs the search through the collection of rules in the knowledge base, a process commonly called *pattern matching*. In inferencing, when all of the hypotheses (the "IF" parts) of a rule are satisfied, the rule is said to be fired. Once a rule is fired, the new knowledge generated by the rule (the conclusion or the validation of the THEN part) is inserted into the memory as a new fact. The inference engine checks every rule in the knowledge base to identify those that can be fired based on what is known at that point in time (the collection of known facts), and keeps doing so until the goal is achieved. The most popular inferencing mechanisms for rule-based systems are forward and backward chaining:

- **Backward chaining** is a goal-driven approach in which you start from an expectation of what is going to happen (i.e., hypothesis) and then seek evidence that supports (or contradicts) your expectation. Often, this entails formulating and testing intermediate hypotheses (or subhypotheses).
- **Forward chaining** is a data-driven approach. We start from available information as it becomes available or from a basic idea, and then we try to draw conclusions. The ES analyzes the problem by looking for the facts that match the IF part of its IF-THEN rules. For example, if a certain machine is not working, the computer checks the electricity flow to the machine. As each rule is tested, the program works its way toward one or more conclusions.

FORWARD AND BACKWARD CHAINING EXAMPLE Here we discuss an example involving an investment decision about whether to invest in IBM stock. The following variables are used:

A = Have $10,000

B = Younger than 30

C = Education at college level

D = Annual income of at least $40,000

E = Invest in securities

F = Invest in growth stocks

G = Invest in IBM stock (the potential goal)

Each of these variables can be answered as true (yes) or false (no).

We assume that an investor has $10,000 (i.e., that A is true) and that she is 25 years old (i.e., that B is true). She would like advice on investing in IBM stock (yes or no for the goal).

Our knowledge base includes the following five rules:

R1: IF a person has $10,000 to invest and she has a college degree,
THEN she should invest in securities.

R2: IF a person's annual income is at least $40,000 and she has a college degree,
THEN she should invest in growth stocks.

R3: IF a person is younger than 30 and she is investing in securities,
THEN she should invest in growth stocks.

R4: IF a person is younger than 30,
THEN she has a college degree.

R5: IF a person wants to invest in a growth stock,
THEN the stock should be IBM.

These rules can be written as follows:

R1: IF A and C, THEN E.

R2: IF D and C, THEN F.

R3: IF B and E, THEN F.

R4: IF B, THEN C.

R5: IF F, THEN G.

Backward Chaining Our goal is to determine whether to invest in IBM stock. With backward chaining, we start by looking for a rule that includes the goal (G) in its conclusion (THEN) part. Because R5 is the only one that qualifies, we start with it. If several rules contain G, then the inference engine dictates a procedure for handling the situation. This is what we do:

1. Try to accept or reject G. The ES goes to the assertion base to see whether G is there. At present, all we have in the assertion base is A is true. B is true. Therefore, the ES proceeds to step 2.
2. R5 says that if it is true that we invest in growth stocks (F), then we should invest in IBM (G). If we can conclude that the premise of R5 is either true or false, then we have solved the problem. However, we do not know whether F is true. What shall we do now? Note that F, which is the premise of R5, is also the conclusion of R2 and R3. Therefore, to find out whether F is true, we must check either of these two rules.
3. We try R2 first (arbitrarily); if both D and C are true, then F is true. Now we have a problem. D is not a conclusion of any rule, nor is it a fact. The computer can either move to another rule or try to find out whether D is true by asking the investor for whom the consultation is given if her annual income is above $40,000. What the ES does depends on the search procedures used by the inference engine. Usually, a user is asked for additional information only if the information is not available or cannot be deduced. We abandon R2 and return to the other rule, R3. This action is called *backtracking* (i.e., knowing that we are at a dead end, we try something else; the computer must be preprogrammed to handle backtracking).
4. Go to R3; test B and E. We know that B is true because it is a given fact. To prove E, we go to R1, where E is the conclusion.
5. Examine R1. It is necessary to determine whether A and C are true.
6. A is true because it is a given fact. To test C, it is necessary to test R4 (where C is the conclusion).
7. R4 tells us that C is true (because B is true). Therefore, C becomes a fact (and is added to the assertion base). Now E is true, which validates F, which validates our goal (i.e., the advice is to invest in IBM).

Note that during the search, the ES moved from the THEN part to the IF part, back to the THEN part, and so on (see Figure 12.6 for a graphical depiction of the backward chaining).

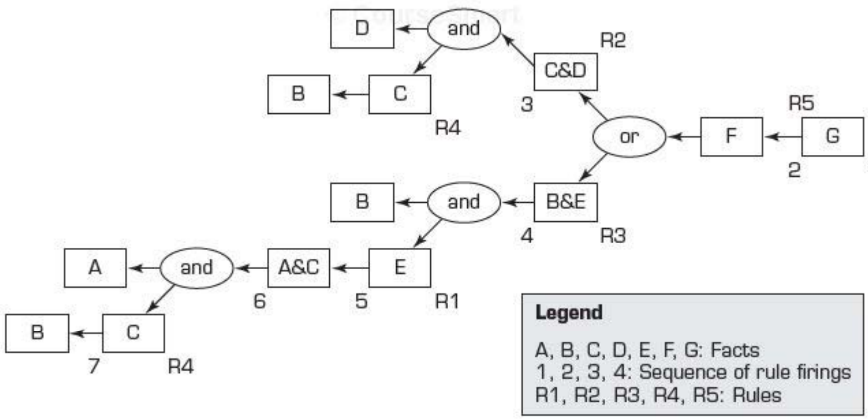

FIGURE 12.6 A Graphical Depiction of Backward Chaining

Forward Chaining Let us use the same example we examined in backward chaining to illustrate the process of forward chaining. In forward chaining, we start with known facts and derive new facts by using rules having known facts on the IF side. The specific steps that forward chaining would follow in this example are as follows (also see Figure 12.7 for a graphical depiction of this process):

1. Because it is known that A and B are true, the ES starts deriving new facts by using rules that have A and B on the IF side. Using R4, the ES derives a new fact C and adds it to the assertion base as true.
2. R1 fires (because A and C are true) and asserts E as true in the assertion base.
3. Because B and E are both known to be true (they are in the assertion base), R3 fires and establishes F as true in the assertion base.
4. R5 fires (because F is on its IF side), which establishes G as true. So the ES recommends an investment in IBM stock. If there is more than one conclusion, more rules may fire, depending on the inferencing procedure.

INFERENCING WITH UNCERTAINTY Although uncertainty is widespread in the real world, its treatment in the practical world of artificial intelligence is very limited. One could argue that

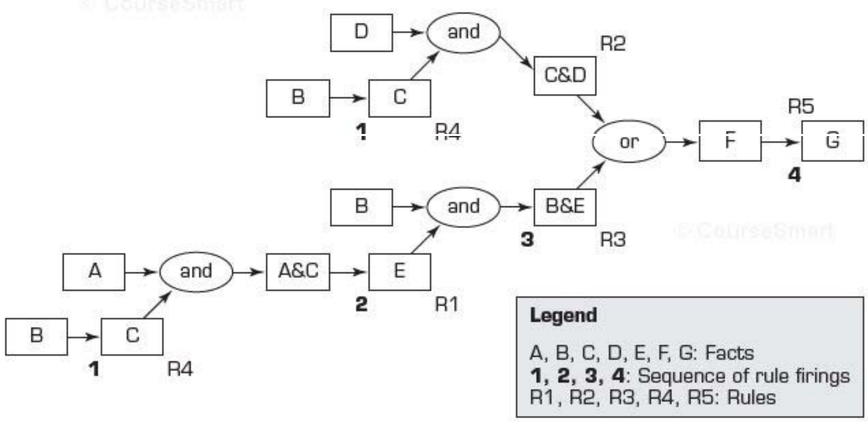

FIGURE 12.7 A Graphical Depiction of Forward Chaining

because the knowledge provided by experts is often inexact an ES that mimics the reasoning process of experts should represent such uncertainty. ES researchers have proposed several methods to incorporate uncertainty into the reasoning process, including probability ratios, the Bayesian approach, fuzzy logic, the Dempster–Shafer theory of evidence, and the theory of certainty factors. Following is a brief description of the theory of certainty factors, which is the most commonly used method to accommodate uncertainty in ES.

The **theory of certainty factors** is based on the concepts of belief and disbelief. The standard statistical methods are based on the assumption that an uncertainty is the probability that an event (or fact) is true or false, whereas certainty theory is based on the *degrees of belief* (not the calculated probability) that an event (or fact) is true or false.

Certainty theory relies on the use of certainty factors. **Certainty factors (CF)** express belief in an event (or a fact or a hypothesis) based on the expert's assessment. Certainty factors can be represented by values ranging from 0 to 100; the smaller the value the lower the probability that the event (or fact) is true or false. Because certainty factors are not probabilities, when we say that there is a certainty value of 90 for rain, we do not mean (or imply) any opinion about no rain (which is not necessarily 10). Thus, certainty factors do not have to sum up to 100.

Combining Certainty Factors Certainty factors can be used to combine estimates by different experts in several ways. Before using any ES shell, you need to make sure that you understand how certainty factors are combined. The most acceptable way of combining them in rule-based systems is the method used in EMYCIN. In this approach, we distinguish between two cases, described next.

Combining Several Certainty Factors in One Rule Consider the following rule with an AND operator:

IF inflation is high, CF = 50 (A)
AND unemployment rate is above 7 percent, CF = 70 (B)
AND bond prices decline, CF = 100 (C),
THEN stock prices decline.

For this type of rule, all IFs must be true for the conclusion to be true. However, in some cases, there is uncertainty as to what is happening. Then the CF of the conclusion is the minimum CF on the IF side:

$$CF(A, B, C) = \text{minimum}[CF(A), CF(B), CF(C)]$$

Thus, in our case, the CF for stock prices to decline is 50 percent. In other words, the chain is as strong as its weakest link.

Now look at this rule with an OR operator:

IF inflation is low, CF = 70 percent
OR bond prices are high, CF = 85,
THEN stock prices will be high.

In this case, it is sufficient that only one of the IFs is true for the conclusion to be true. Thus, if both IFs are believed to be true (at their certainty factor), then the conclusion will have a CF with the maximum of the two:

$$CF\ (A \text{ or } B) = \text{maximum}\ [CF\ (A), CF\ (B)]$$

In our case, CF must be 85 for stock prices to be high. Note that both cases hold for any number of IFs.

Combining Two or More Rules Why might rules be combined? There may be several ways to reach the same goal, each with different certainty factors for a given set of facts. When we have a knowledge-based system with several interrelated rules, each of which makes the same conclusion but with a different certainty factor, each rule can be viewed as a piece of evidence that supports the joint conclusion. To calculate the certainty factor (or the confidence) of the conclusion, it is necessary to combine the evidence. For example, let us assume that there are two rules:

R1: IF the inflation rate is less than 5 percent,
THEN stock market prices go up (CF = 0.7).
R2: IF the unemployment level is less than 7 percent,
THEN stock market prices go up (CF = 0.6).

Now let us assume a prediction that during the next year, the inflation rate will be 4 percent and the unemployment level will be 6.5 percent (i.e., we assume that the premises of the two rules are true). The combined effect is computed as follows:

$$CF(R1, R2) = CF(R1) + CF(R2) \times [1 - CF(R1)]$$
$$= CF(R1) + CF(R2) - [CF(R1) \times CF(R2)]$$

In this example, given CF(R1) = 0.7 and CF(R2) = 0.6

$$CF(R1, R2) = 0.7 + 0.6 - [(0.7) \times (0.6)] = 0.88$$

If we add a third rule, we can use the following formula:

$$CF(R1, R2, R3) = CF(R1, R2) + CF(R3) \times [1 - CF(R1, R2)]$$
$$= CF(R1, R2) + CF(R3) - [CF(R1, R2) \times CF(R3)]$$

In our example:

R3: IF bond price increases,
THEN stock prices go up (CF = 0.85)

$$CF(R1, R2, R3) = 0.88 + 0.85 - [(0.88) \times (0.85)] = 0.982$$

Note that CF(R1,R2) was computed earlier as 0.88. For a situation with more rules, we can apply the same formula incrementally.

Explanation and Justification

A final feature of expert systems is their interactivity with users and their capacity to provide an explanation consisting of the sequence of inferences that were made by the system in arriving at a conclusion. This feature offers a means of evaluating the integrity of the system when it is to be used by the experts themselves. Two basic types of explanations are the why and the how. Metaknowledge is knowledge about knowledge. It is a structure within the system using the domain knowledge to accomplish the system's problem-solving strategy. This section deals with different methods used in ES for generating explanations.

Human experts are often asked to explain their views, recommendations, or decisions. If ES are to mimic humans in performing highly specialized tasks, they, too, need to justify and explain their actions. An explanation is an attempt by an ES to clarify its reasoning, recommendations, or other actions (e.g., 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 (i.e., debugging of the systems by the knowledge engineer).
- Explain situations that were unanticipated by the user.
- Satisfy psychological and social needs by helping the 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.)

Explanation in rule-based ES is usually associated with a way of tracing the rules that are fired during the course of a problem-solving session. This is about the closest to a real explanation that today's systems come, given that their knowledge is usually represented almost exclusively as rules that do not include basic principles necessary for a human-type explanation. Though many explanations have been proposed, in the following section we provide brief descriptions for the two most common types: the why and the how explanations.

WHY EXPLANATIONS A user poses a typical "why?" question to a computer after the computer asks the user to provide some information. For example, using our earlier investment example, we might have the following dialogue:

COMPUTER: What is your annual income?

CLIENT: Why? (Why do you need to know?)

COMPUTER: R2; meaning "In checking R2, I need to know whether your income is above $40,000. If this is true, I will conclude that because you have a college degree, you should invest in growth stocks."

HOW EXPLANATIONS A user poses a typical "how?" question when he or she would like to know how a certain conclusion or recommendation was reached. Simple systems are limited to the final conclusion. More complex systems can handle intermediate conclusions as well. The system can explain why a certain rule was fired; that is, it shows the chain of rules used to reach the conclusion. In the investment problem, the how explanation might look like this:

COMPUTER: Invest in IBM stocks.

CLIENT: How? (How was the conclusion reached?)

COMPUTER: R5–R3–R7–R4; meaning "Given that you have $10,000 to invest and you are younger than 30, then according to R4, you have a college degree. If this is the case, then according to R1, you should invest in securities. For a young investor like you, according to R3, you should invest in growth stocks if you are going to invest at all. Finally, according to R5, if you need to invest in growth stocks, then IBM is your best bet."

The why and how explanations often show the rules as they were programmed and not in a natural language. However, some systems have the capability to present these rules in natural language.

Section 12.7 Review Questions

1. State two production rules that can represent the knowledge of repairing your car.
2. Describe how ES perform inference.
3. Describe the reasoning procedures of forward chaining and backward chaining.
4. List the three most popular methods to deal with uncertainty in ES.
5. Why do we need to incorporate uncertainty in ES solutions?
6. What are the ways by which ES justify their knowledge?

12.8 PROBLEM AREAS SUITABLE FOR EXPERT SYSTEMS

ES 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 12.5. Some ES belong to two or more of these categories. A brief description of each category follows:

- ***Interpretation systems.*** Systems that infer situation descriptions from observations. This category includes surveillance, speech understanding, image analysis, signal interpretation, and many kinds of intelligence analyses. An interpretation system explains observed data by assigning them symbolic meanings that describe the situation.
- ***Prediction systems.*** These systems include weather forecasting; demographic predictions; economic forecasting; traffic predictions; crop estimates; and military, marketing, and financial forecasting.
- ***Diagnostic systems.*** These systems include medical, electronic, mechanical, and software diagnoses. Diagnostic systems typically relate observed behavioral irregularities to underlying causes.
- ***Design systems.*** These systems develop configurations of objects that satisfy the constraints of the design problem. Such problems include circuit layout, building design, and plant layout. Design systems construct descriptions of objects in various relationships with one another and verify that these configurations conform to stated constraints.
- ***Planning systems.*** These systems specialize in planning problems, such as automatic programming. They also deal with short- and long-term planning in areas such as project management, routing, communications, product development, military applications, and financial planning.

TABLE 12.5 Generic Categories of Expert Systems

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 and 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

- ***Monitoring systems.*** These systems compare observations of system behavior with standards that seem crucial for successful goal attainment. These crucial features correspond to potential flaws in the plan. There are many computer-aided monitoring systems for topics ranging from air traffic control to fiscal management tasks.
- ***Debugging systems.*** These systems rely on planning, design, and prediction capabilities for creating specifications or recommendations to correct a diagnosed problem.
- ***Repair systems.*** These systems develop and execute plans to administer a remedy for certain diagnosed problems. Such systems incorporate debugging, planning, and execution capabilities.
- ***Instruction systems.*** Systems that incorporate diagnosis and debugging subsystems that specifically address students' needs. Typically, these systems begin by constructing a hypothetical description of the student's knowledge that interprets her or his behavior. They then diagnose weaknesses in the student's knowledge and identify appropriate remedies to overcome the deficiencies. Finally, they plan a tutorial interaction intended to deliver remedial knowledge to the student.
- ***Control systems.*** Systems that adaptively govern the overall behavior of a system. To do this, a control system must repeatedly interpret the current situation, predict the future, diagnose the causes of anticipated problems, formulate a remedial plan, and monitor its execution to ensure success.

Not all the tasks usually found in each of these categories are suitable for ES. However, thousands of decisions do fit into these categories.

APPLICATION CASE 12.7

Monitoring Water Quality with Sensor-Driven Expert Systems

Environmental concerns are becoming increasingly more important around the world. Constant monitoring of environmental conditions is a challenging task, but a necessary one. If the authorities and organizations involved in the management of environmental resources were able to examine quantitative and qualitative parameters related to environmental conditions, they then might be able to predict undesirable situations and to draw conclusions about adverse trends, enabling them to take countermeasures to prevent catastrophes in a timely fashion.

Hatzikos et al. (2007) described an environmental expert system that monitors seawater quality and pollution in northern Greece through a sensor network called Andromeda. The expert system monitors data collected by sensors at local monitoring stations and reasons about the suitability of water for various uses, such as swimming and cultivation of fish. The sensor data is collected periodically (at different parts of the sea) and transmitted to the evaluation system via a wireless network. The sensors transmit data on the following to the expert system:

- pH
- Temperature
- Conductance
- Salinity
- Dissolved oxygen
- Turbidity

The expert system uses fuzzy logic (see Chapter 13) to infer about the combination of the sensor input values to identify certain environmental conditions and to create and disseminate appropriate alerts. Authorities hope that the expert system will help them make decisions in their battles against water pollution, which is vital for public health and the local economy.

Source: E. V. Hatzikos, N. Bassiliades, L. Asmanis, and I. Vlahavas, "Monitoring Water Quality Through a Telematic Sensor Network and a Fuzzy Expert System," *Expert Systems,* Vol. 24, No. 3, 2007, pp. 143–161.

Section 12.8 Review Questions

1. Describe a sample ES application for prediction.
2. Describe a sample ES application for diagnosis.
3. Describe a sample ES application for the rest of the generic ES categories.

12.9 DEVELOPMENT OF EXPERT SYSTEMS

The development of ES is a tedious process and typically includes defining the nature and scope of the problem, identifying proper experts, acquiring knowledge, selecting the building tools, coding the system, and evaluating the system.

Defining the Nature and Scope of the Problem

The first step in developing an ES is to identify the nature of the problem and to define its scope. Some domains may not be appropriate for the application of ES. For example, a problem that can be solved by using mathematical optimization algorithms is often inappropriate for ES. In general, rule-based ES are appropriate when the nature of the problem is qualitative, knowledge is explicit, and experts are available to solve the problem effectively and provide their knowledge.

Another important factor is to define a feasible scope. The current technology is still very limited and is capable of solving relatively simple problems. Therefore, the scope of the problem should be specific and reasonably narrow. For example, it may be possible to develop an ES for detecting abnormal trading behavior and possible money laundering, but it is not possible to use an ES to determine whether a particular transaction is criminal.

Identifying Proper Experts

After the nature and scope of the problem have been clearly defined, the next step is to find proper experts who have the knowledge and are willing to assist in developing the knowledge base. No ES can be designed without the strong support of knowledgeable and supportive experts. A project may identify one expert or a group of experts. A proper expert should have a thorough understanding of problem-solving knowledge, the role of ES and decision support technology, and good communication skills.

Acquiring Knowledge

After identifying helpful experts, it is necessary to start acquiring decision knowledge from them. The process of eliciting knowledge is called *knowledge engineering*. The person who is interacting with experts to document the knowledge is called a *knowledge engineer*.

Knowledge acquisition is a time-consuming and risky process. Experts may be unwilling to provide their knowledge for various reasons. First, their knowledge may be proprietary and valuable. Experts may not be willing to share their knowledge without a reasonable payoff. Second, even though an expert is willing to share, certain knowledge is tacit, and the expert may not have the skill to clearly dictate the decision rules and considerations. Third, experts may be too busy to have enough time to communicate with the knowledge engineer. Fourth, certain knowledge may be confusing or contradictory in nature. Finally, the knowledge engineer may misunderstand the expert and inaccurately document knowledge.

The result of knowledge acquisition is a knowledge base that can be represented in different formats. The most popular one is if-then rules. The knowledge may also be represented as decision trees or decision tables. The knowledge in the knowledge base must be evaluated for its consistency and applicability.

Selecting the Building Tools

After the knowledge base is built, the next step is to choose a proper tool for implementing the system. There are three different kinds of development tools, as described in the following sections.

GENERAL-PURPOSE DEVELOPMENT ENVIRONMENT The first type of tool is general-purpose computer languages, such as C++, Prolog, and LISP. Most computer programming languages support the if-then statement. Therefore, it is possible to use C++ to develop an ES for a particular problem domain (e.g., disease diagnosis). Because these programming languages do not have built-in inference capabilities, using them in this way is often very costly and time consuming. Prolog and LISP are two languages for developing intelligent systems. It is easier to use them than to use C++, but they are still specifically designed for professional programmers and are not very friendly. For recent Web-based applications, Java and computer languages that support Web services (such as the Microsoft .NET platform) are also useful.

ES SHELLS The second type of development tool, the **expert system (ES) shell**, is specifically designed for ES development. An ES shell has built-in inference capabilities and a user interface, but the knowledge base is empty. System development is therefore a process of feeding the knowledge base with rules elicited from the expert.

A popular ES shell is the Corvid system developed by Exsys (**exsys.com**). The system is an object-oriented development platform that is composed of three types of operations: variables, logic blocks, and command blocks. Variables define the major factors considered in problem solving. Logic blocks are the decision rules acquired from experts. Command blocks determine how the system interacts with the user, including the order of execution and the user interface. Figure 12.8 shows a screenshot of a logic block that shows the decision rules. More products are available from business rules management vendors, such as Haley (**haley.com**), ILOG (**ilog.com**), and LPA's VisiRule (**lpa.co.uk/vsr.htm**), which is based on a general-purpose tool called Micro-Prolog.

TAILORED TURN-KEY SOLUTIONS The third tool, a tailored turn-key tool, is tailored to a specific domain and can be adapted to a similar application very quickly. Basically, a tailored turn-key tool contains specific features often required for developing applications in a particular domain. This tool must adjust or modify the base system by tailoring the user interface or a relatively small portion of the system to meet the unique needs of an organization. For example, both Haley and ILOG offer various tailor-made solutions for insurance, medical, scheduling, and homeland security applications.

CHOOSING AN ES DEVELOPMENT TOOL Choosing among these tools for ES development depends on a few criteria. First, you need to consider the cost benefits. Tailored turn-key solutions are the most expensive option. However, you need to consider the total cost, not just the cost of the tool. Second, you need to consider the technical functionality and flexibility of the tool; that is, you need to determine whether the tool provides the function you need and how easily it allows the development team to make necessary changes. Third, you need to consider the tool's compatibility with the existing information infrastructure in the organization. Most organizations have many existing applications, and the tool must be compatible with those applications and needs to be able to be integrated as part of the entire information infrastructure. Finally, you need to consider the reliability of the tool and vendor support. The vendor's experiences in similar domains and training programs are critical to the success of an ES project.

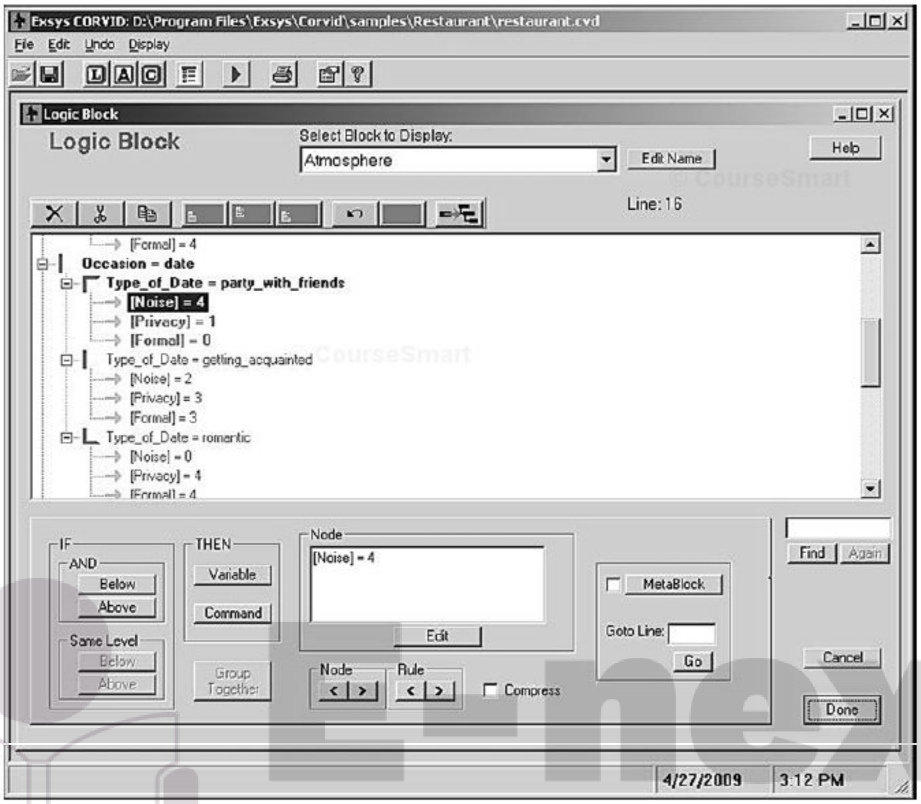

FIGURE 12.8 A Screenshot from Corvid Expert System Shell *Source:* Courtesy of Exsys Inc. Reprinted with permission.

Coding the System

After choosing a proper tool, the development team can focus on coding the knowledge based on the tool's syntactic requirements. The major concern at this stage is whether the coding process is efficient and properly managed to avoid errors. Skilled programmers are helpful and important.

Evaluating the System

After an ES system is built, it must be evaluated. Evaluation includes both verification and validation. Verification ensures that the resulting knowledge base contains knowledge exactly the same as that acquired from the expert. In other words, verification ensures that no error occurred at the coding stage. Validation ensures that the system can solve the problem correctly. In other words, validation checks whether the knowledge acquired from the expert can indeed solve the problem effectively.

Section 12.9 Review Questions

1. Describe the major steps in developing rule-based ES.
2. What are the necessary conditions for a good expert?
3. Compare three different types of tools for developing ES.
4. List the criteria for choosing a development tool.
5. What is the difference between verification and validation of an ES?

12.10 BENEFITS, LIMITATIONS, AND CRITICAL SUCCESS FACTORS OF EXPERT SYSTEMS

Thousands of ES are in use today in almost every industry and in every functional area. For example, Eom (1996) prepared a comprehensive survey of about 440 operational ES in business. His survey revealed that 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. ES in finance and in engineering applications are described in Nedovic and Devedzic (2002) and Nurminen et al. (2003). For many firms, ES have become indispensable tools for effective management, but the application of ES may also have limitations. In this section, we outline the major benefits and limitations of this technology.

Benefits and Limitations of ES

The use of ES can create benefits and at the same time incur limitations, as explained in the following sections.

BENEFITS OF ES The following are some of the benefits of ES:

- ***Increased output and productivity.*** ES can work faster than humans can. For example, the classic XCON enabled Digital Equipment Corp. to increase the throughput of its popular VAX minicomputers configuration orders fourfold.
- ***Decreased decision-making time.*** Using the recommendations of an ES, a human can make decisions much faster. For example, American Express can now make charge approval decisions in less than 5 seconds, compared to about 3 minutes before implementation of an ES. This property is important in supporting frontline decision makers who must make quick decisions while interacting with customers.
- ***Increased process and product quality.*** ES can increase quality by providing consistent advice and reducing the size and rate of errors. For example, XCON reduced the error rate of configuring computer orders from 35 to 2 percent and then even less, thus improving the quality of the minicomputers.
- ***Reduced downtime.*** Many operational ES are used for diagnosing malfunctions and prescribing repairs. By using ES, it is possible to reduce machine downtime significantly. For example, on an oil rig 1 day of lost time can cost as much as $250,000. A system called DRILLING ADVISOR was developed to detect malfunctions in oil rigs. This system saved a considerable amount of money for the company by significantly reducing downtime.
- ***Capture of scarce expertise.*** The scarcity of expertise becomes evident in situations in which there are not enough experts for a task, the expert is about to retire or leave the job, or expertise is required over a broad geographic area. For example, more than 30 percent of all requests for authorization of benefits are approved automatically through eCare, enabling CIGNA Behavioral Health to handle more requests with its existing staff.
- ***Flexibility.*** ES can offer flexibility in both the service and manufacturing industries.
- ***Easier equipment operation.*** An ES makes complex equipment easier to operate. For example, Steamer was an early ES used to train inexperienced workers to operate complex ship engines. Another example is an ES developed for Shell Oil Company to train people to use complex computer program routines.
- ***Elimination of the need for expensive equipment.*** Often, a human must rely on expensive instruments for monitoring and control. ES can perform the same

tasks with lower-cost instruments because of their ability to investigate the information provided more thoroughly and quickly.

- ***Operation in hazardous environments.*** Many tasks require humans to work in hazardous environments. An ES can allow humans to avoid such environments. They can enable workers to avoid hot, humid, or toxic environments, such as a nuclear power plant that has malfunctioned. This feature is extremely important in military conflicts.
- ***Accessibility to knowledge and help desks.*** ES make knowledge accessible, thus freeing experts from routine work. People can query systems and receive useful advice. One area of applicability is the support of help desks, such as the HelpDeskIQ system offered by BMC Remedy.
- ***Ability to work with incomplete or uncertain information.*** In contrast to conventional computer systems, ES can, like human experts, work with incomplete, imprecise, and uncertain data, information, or knowledge. The user can respond with "don't know" or "not sure" to one or more of the system's questions during a consultation, and the ES can produce an answer, although it may not be a certain one.
- ***Provision of training.*** ES can provide training. Novices who work with ES become more and more experienced. The explanation facility can also serve as a teaching device, as can notes and explanations that can be inserted into the knowledge base.
- ***Enhancement of problem solving and decision making.*** ES enhance problem solving by allowing the integration of top experts' judgment into the analysis. For example, an ES called Statistical Navigator was developed to help novices use complex statistical computer packages.
- ***Improved decision-making processes.*** ES provide rapid feedback on decision consequences, facilitate communication among decision makers on a team, and allow rapid response to unforeseen changes in the environment, thus providing a better understanding of the decision-making situation.
- ***Improved decision quality.*** ES are reliable. They do not become tired or bored, call in sick, or go on strike, and they do not talk back to the boss. ES also consistently pay attention to all details and do not overlook relevant information and potential solutions, thereby making fewer errors. In addition, ES provide the same recommendations to repeated problems.
- ***Ability to solve complex problems.*** One day, ES may explain complex problems whose solutions are beyond human ability. Some ES are already able to solve problems in which the required scope of knowledge exceeds that of any one individual. This allows decision makers to gain control over complicated situations and improve the operation of complex systems.
- ***Knowledge transfer to remote locations.*** One of the greatest potential benefits of ES is its ease of transfer across international boundaries. An example of such a transfer is an eye care ES for diagnosis and recommended treatment, developed at Rutgers University in conjunction with the World Health Organization. The program has been implemented in Egypt and Algeria, where serious eye diseases are prevalent but eye specialists are rare. The PC program is rule based and can be operated by a nurse, a physician's assistant, or a general practitioner. The Web is used extensively to disseminate information to users in remote locations. The U.S. government, for example, places advisory systems about safety and other topics on its Web sites.
- ***Enhancement of other information systems.*** ES can often be found providing intelligent capabilities to other information systems. Many of these benefits lead to improved decision making, improved products and customer service, and a sustainable strategic advantage. Some may even enhance an organization's image.

PROBLEMS AND LIMITATIONS OF ES Available ES methodologies may not be straightforward and effective, even for many applications in the generic categories. The following problems have slowed the commercial spread of ES:

- Knowledge is not always readily available.
- It can be difficult to extract expertise from humans.
- The approach of each expert to a situation assessment may be different, yet correct.
- It is difficult, even for a highly skilled expert, to abstract good situational assessments when under time pressure.
- Users of ES have natural cognitive limits.
- ES work well only within a narrow domain of knowledge.
- Most experts have no independent means of checking whether their conclusions are reasonable.
- The vocabulary, or jargon, that experts use to express facts and relations is often limited and not understood by others.
- Help is often required from knowledge engineers who are rare and expensive, and this could make ES construction costly.
- Lack of trust on the part of 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 in some cases. For example, the initial fully developed XCON system could not fulfill about 2 percent of the orders presented to it. Human experts must step in to resolve these problems.
- ES, like human experts, sometimes produce incorrect recommendations.

The Web is a major facilitator of ES that overcomes several of these limitations. The ability to disseminate ES to the masses makes them more cost-effective. Consequently, more money can be spent on better systems.

Gill (1995) discovered that only about one-third of all commercial ES studied survived over 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 clearly indicate that some ES fall short of generally intelligent human behavior. Although there is no recent follow-up research, the rapid progress of information technology can reduce the possibility of failure, and several of these limitations will diminish or disappear with technological improvements over time.

Critical Success Factors for ES

Several researchers have investigated the reasons ES succeed and fail in practice. As with many MIS, a number of studies have shown that the level of managerial and user involvement directly affects the success level of MIS, specifically ES. However, these factors alone are not sufficient to guarantee success, and the following issues should also be considered:

- 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) and not purely quantitative (otherwise, a numeric approach should be used).
- The problem must be sufficiently narrow in scope.

- 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-user job improvement must be considered.
- The impact should be favorable. End-user attitudes and expectations must be considered.
- Management support must be cultivated.
- End-user training programs are necessary.
- The organizational environment should favor adoption of new technology.
- The application must be well-defined and structured, and it should be justified by strategic impact.

Managers attempting to introduce ES technology should establish end-user training programs, demonstrating its potential as a business tool. As part of the managerial support effort, the organizational environment should favor new technology adoption.

Section 12.10 Review Questions

1. Describe the major benefits of using ES.
2. Describe some of the limitations of ES.
3. Describe the critical success factors of ES.

12.11 EXPERT SYSTEMS ON THE WEB

The relationship between ES and the Internet and intranets can be divided into two categories. The first is the use of ES on the Web. In this case, the Web supports ES (and other AI) applications. The second is the support ES (and other AI methods) give to the Web.

One of the early reasons for ES development was its potential to provide knowledge and advice to large numbers of users. Because the Web enables knowledge to be disseminated to many people, the cost per user becomes small, making ES very attractive. However, according to Eriksson (1996), attaining this goal has proven to be very difficult. Because advisory systems are used infrequently, they need a large number of users to justify their construction. As a result, very few ES disseminate knowledge to many users.

The widespread availability and use of the Internet and intranets provide the opportunity to disseminate expertise and knowledge to mass audiences. By implementing ES (and other intelligent systems) as knowledge servers, it becomes economically feasible and profitable to publish expertise on the Web. ES running on servers can support a large group of users who communicate with the system over the Web. In this way, user interfaces based on Web protocols and the use of browsers provide access to the knowledge servers. This implementation approach is described in Eriksson (1996). If you go to the Web site of Exsys (**exsys.com**), you can try the Banner with Brains, which integrates ES capabilities into a Web banner (see Application Case 12.8). Another example is a rule-based system for intelligent online dialogue, developed by German scholars (see Application Case 12.9). Gensym (**gensym.com**) offers a real-time supporting tool called G2 that has been applied to many mission-critical domains, such as chemical, oil, gas, and process manufacturing.

APPLICATION CASE 12.8

Banner with Brains: Web-Based ES for Restaurant Selection

Selecting a restaurant for dating or business in a foreign city has never been easier, thanks to the availability of services over the Web and support from expert systems. At **exsys.com**, you can try a demo system that integrates an ES with a banner. All interactions occur through the banner.

The ES is familiar with restaurants in Albuquerque. When you need to find a restaurant, the system asks about the occasion and the type of food you desire. The preference data is then fed to a spreadsheet of information on the various restaurants. The system creates a probabilistic ranking of the restaurants that meet your needs. It then weighs various factors, based on the specified occasion, and displays up to five restaurants. It also explains why it recommends these restaurants. This kind of application will become increasingly popular in the future.

Sources: Exsys, "Corvid Restaurant Selection Knowledge Automation Expert System," **exsys.com/Demos/Restaurant/restaurant_demo.html** (accessed June 2009); and G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, June 2005.

APPLICATION CASE 12.9

Rule-Based System for Online Consultation

A group of German scientists took advantage of the convenience of the Internet to offer a Web-based online consultation system for intelligent dialogue in assisting and conducting interviews. Traditionally, interviews are conducted face-to-face or over the telephone. The new system is used by a university to interview applicants and assess their chances of being admitted and by companies to screen job applicants. The system consists of a rule-based knowledge base that can dynamically adjust the questions to ask, based on user responses to previous questions. The system can also be used to support social workers in interacting with their clients online.

Sources: S. Mertens, M. Rosu, and Y. Erdani, "An Intelligent Dialogue for Online Rule-Based Expert Systems," 9th International Conference on Intelligent User Interface, January 13–16, 2004; and **expertise2go.com** (accessed March 2006).

ES can be transferred over the Web not only to human users, but also to other computerized systems, including DSS, robots, and databases. Other ES Web support possibilities include system construction. Here, collaboration between builders, experts, and knowledge engineers can be facilitated by Internet-based groupware. This can reduce the cost of building ES. Knowledge acquisition costs can be reduced, for example, in cases in which there are several experts or in which the expert is in a different location from the knowledge engineer. Knowledge maintenance can also facilitate the use of the Internet, which is also helpful to users.

Finally, the Web can greatly support the spread of multimedia-based ES. Such systems, called *intellimedia systems*, support the integration of extensive multimedia applications and ES. Such systems can be very helpful for remote users, such as those in the tourism industry (see Stabb et al., 2002) and in remote equipment failure diagnosis.

The other aspect of the ES–Internet relationship is the support that ES and other AI technologies can provide to the Internet and intranets. The major contributions of AI to the Internet and intranets are summarized in Table 12.6.

Information about the relationships among ES, intelligent agents, and other AI and the Internet is readily available on the Internet. For example, the Web sites of *PC AI* magazine

TABLE 12.6 AI/ES and Web Impacts

Aspects	Impacts from the Web	Impacts on the Web
Knowledge acquisition	Experts in different areas can collaborate over the Internet. Knowledge acquisition can be done at different times to fit the schedules of different experts. Knowledge acquired from different experts can be shared on the Internet to stimulate discussion for enhancement.	Knowledge of Web operations and activities can be acquired and managed for sharing and use.
Expert systems development	Collaborative design of expert systems by a geographically distributed team becomes possible. Outsourcing of the design effort becomes feasible. ES evaluation can be done remotely. The Web provides a unified multimedia user interface for easy system integration. Web services provide an improved platform for designing ES.	ES can be designed to support Web activities, automatic services, and better performance.
Expert systems consultation	Users in remote areas can use the system to solve problems. Expertise can easily be disseminated to a large body of users.	Applications of ES are available for Web browsing and monitoring.

(**pcai.com**) and the American Association for Artificial Intelligence (**aaai.org**) provide sets of hyperlinks to related Web sites. University of Maryland, Baltimore County (**agents .umbc.edu**) provides a collection of resources on intelligent agents. In the future, more applications on the Internet will be available; particularly those that provide automated decision making and real-time decision support (see Technology Insights 12.4).

Section 12.11 Review Questions

1. What are the benefits of deploying an ES on the Web?
2. How can an ES help a decision maker use the Web to find relevant information?
3. Visit **exsys.com** and run two of the demo systems. Describe each system.

TECHNOLOGY INSIGHTS 12.4 Automated and Real-Time Decision Systems

A technology called automated decision systems (ADS) is taking off, and it embodies the best attributes of artificial intelligence and business analytics. ADS are based on business rules, somewhat similar to ES, and, like other DSS technologies, they often involve statistical or algorithmic analysis of data. The main differences between the rules in ADS and the rules in ES are in the way they are created and used. The rules in ES are determined based on the experiences of domain experts and are executed collectively by an inference engine. In contrast, the rules in ADS are often created from historical data using advanced business analytics techniques and are used individually to trigger an automatic decision for a routine business situation. ADS typically make decisions in real-time after weighing all the data and relevant business rules for a particular case (ES can also be used to make real-time decisions, but often through a consultation session with the end user). Sometimes ADS incorporate business process management information, leading some observers to classify them as "smart business process management" systems.

The most salient characteristic of these systems is that they actually make a decision, such as what price to charge a particular customer, whether to grant a loan or an insurance policy, which delivery truck to reroute, or what drug to prescribe to a diabetic patient. In many cases, their decisions are made without any human intervention at all; in others—sometimes for legal or ethical reasons—they work alongside a human expert, such as a doctor. For the most part, these systems are used for decisions that must be made frequently and very rapidly, using information available online. The decision domains are relatively highly structured, with well-understood decision factors.

"Real time" can be looked at from both business and technology perspectives. From the business perspective, "real time" signifies that the users require rapid responses to customer requests. From the technology perspective, "real time" means that the system needs to have enough power to respond quickly to a user request.

Real-time ES are widely used for environmental protection and chemical processes. For example, RTXPS is a real-time ES designed for online dynamic decision support, mission-critical command, and control and communication tasks such as (1) emergency management for technological and environmental hazards, including early warning for events such as floods, toxic or oil spills, tsunamis, landslides, and so on; and (2) complex control and assessment tasks, including coordination of first response, recovery, restoration, and cleanup operations (see **ess.co.at/RTXPS/**).

Sources: C. White, "Intelligent Business Strategies: Near Real-Time and Automated Decision Making," *DM Review Magazine,* October 2002; and T. Davenport, "Decision Evolution," *CIO Magazine,* October 2004.

Chapter Highlights

- Artificial intelligence (AI) is a discipline that investigates how to build computer systems to perform tasks that can be characterized as intelligent.
- The major characteristics of AI are symbolic processing, the use of heuristics instead of algorithms, and the application of inference techniques.
- AI has several major advantages over people: It is permanent, it can be easily duplicated and disseminated, it can be less expensive than human intelligence, it is consistent and thorough, and it can be documented.
- Natural (human) intelligence has advantages over AI: It is creative, it uses sensory experiences directly, and it reasons from a wide context of experiences.
- Knowledge, rather than data or information, is the major focus of AI.
- Major areas of AI include expert systems, natural language processing, speech understanding, intelligent robotics, computer vision, fuzzy logic, intelligent agents, intelligent computer-aided instruction, automatic programming, neural computing, game playing, and language translation.
- Expert systems (ES) are the most often applied AI technology. ES attempt to imitate the work of experts. They capture human expertise and apply it to problem solving.
- For an ES to be effective, it must be applied to a narrow domain, and the knowledge must include qualitative factors.
- Natural language processing investigates techniques that allow users to communicate with computers in a natural language. It includes text-based and voice-based natural language user interfaces.
- An intelligent robot is a computer-based program or machine that can respond to changes in its environment. Most of today's robots do not have the same capabilities as human beings have, but they are improving rapidly.
- Intelligent tutoring systems use AI to help users learn. Artificial intelligence can improve training and teaching.
- The power of an ES is derived from the specific knowledge it possesses, not from the particular knowledge representation and inference schemes it uses.
- Expertise is task-specific knowledge acquired through training, reading, and experience.
- ES technology can transfer knowledge from experts and documented sources to the computer and make it available for use by nonexperts.
- The major components of an ES are the knowledge acquisition subsystem, knowledge base, inference

engine, user interface, blackboard, explanation subsystem, and knowledge-refinement subsystem.

- The inference engine provides reasoning capability for an ES.
- ES inference can be done by using forward chaining or backward chaining.
- Knowledge engineers are professionals who know how to capture the knowledge from an expert and structure it in a form that can be processed by the computer-based ES.
- ES development process includes defining the nature and scope of the problem, identifying proper experts, acquiring knowledge, selecting the building tools, coding the system, and evaluating the system.
- ES are popular in a number of generic categories: interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction, and control.
- The ES shell is an ES development tool that has the inference engine and building blocks for the knowledge base and the user interface. Knowledge engineers can easily develop a prototype system by entering rules into the knowledge base.
- ES have many benefits. The most important are improvement in productivity or quality, preservation of scarce expertise, enhancement of other systems, ability to cope with incomplete information, and provision of training.
- Many ES failures are caused by nontechnical problems, such as lack of managerial support and poor end-user training.
- Although there are several technical limitations to the use of ES, some of them will disappear with improved technology.
- Some ES provide advice in a real time.
- ES and AI provide support to the Internet and intranets.

Key Terms

artificial intelligence (AI) *534*
backward chaining *558*
blackboard *551*
certainty factors (CF) *561*
consultation environment *550*
development environment *550*
expert *542*
expert system (ES) *537*
expert system (ES) shell *567*
expertise *543*
explanation subsystem *552*
forward chaining *558*
fuzzy logic *541*
heuristics *535*
inference engine *558*
inferences rules *557*
intelligent agent (IA) *541*
intelligent computer-aided instruction (ICAI) *540*
intelligent tutoring system (ITS) *540*
knowledge acquisition *550*
knowledge base *551*
knowledge engineer *551*
knowledge engineering *553*
knowledge rules *557*
knowledge-based system (KBS) *532*
knowledge-refining system *552*
natural language processing (NLP) *537*
neural computing *540*
production rules *557*
robot *539*
rule-based expert systems *529*
speech (voice) understanding *539*
theory of certainty factors *561*
Turing test *534*
user interface *551*
visual recognition *540*

Questions for Discussion

1. Compare numeric and symbolic processing techniques and give an example to illustrate their differences.
2. Do you agree with the statement that using speech communication as the user interface could increase people's willingness to use ES? Why or why not?
3. It is said that powerful computers, inference capabilities, and problem-solving heuristics are necessary but not sufficient for solving real problems. Explain.
4. Explain how the Web improves the benefit–cost ratio of ES and enables systems that otherwise are not justifiable.
5. Explain the relationship between the development environment and the consultation (i.e., runtime) environment.
6. Explain the difference between forward chaining and backward chaining and describe when each is most appropriate.
7. What kind of mistakes might ES make and why? Why is it easier to correct mistakes in ES than in conventional computer programs?
8. Review the limitations of ES discussed in this chapter. From what you know, which of these limitations are the most likely to still be limitations in the year 2100? Why?
9. An ES for stock investment is developed and licensed for $1,000 per year. The system can help identify the most undervalued securities on the market and the best timing for buying and selling the securities. Will you order a copy as your investment advisor? Explain why or why not.
10. Given the current status of the Web, discuss how it is changing the availability of ES and how it is being used to embed expertise in other systems.

Exercises

TERADATA STUDENT NETWORK (TSN) AND OTHER HANDS-ON EXERCISES

1. Go to **teradatastudentnetwork.com** and search for stories about Chinatrust Commercial Bank's (CTCB's) use of the Teradata Relationship Manager and its reported benefits. Study the functional demo of the Teradata Relationship Manager to answer the following questions:
   a. What functions in the Teradata Relationship Manager are useful for supporting the automation of business rules? In CTCB's case, identify a potential application that can be supported by rule-based ES and solicit potential business rules in the knowledge base.
   b. Access Haley and compare the Teradata Relationship Manager and Haley's Business Rule Management System. Which tool is more suitable for the application identified in the previous question?
2. We list 10 categories of ES applications in the chapter. Find 20 sample applications, 2 in each category, from the various functional areas in an organization (i.e., accounting, finance, production, marketing, and human resources).

TEAM ASSIGNMENTS AND ROLE-PLAYING

1. Download Exsys' Corvid tool for evaluation. Identify an expert (or use one of your teammates) in an area where experience-based knowledge is needed to solve problems, such as buying a used car, selecting a school and major, selecting a job from many offers, buying a computer, diagnosing and fixing computer problems, etc. Go through the knowledge-engineering process to acquire the necessary knowledge. Using the evaluation version of the Corvid tool, develop a simple expert system application on the expertise area of your choice. Report on your experiences in a written document; use screenshots from the software as necessary.
2. Search to find applications of artificial intelligence and ES. Identify an organization with which at least one member of your group has a good contact who has a decision-making problem that requires some expertise (but is not too complicated). Understand the nature of its business and identify the problems that are supported or can potentially be supported by rule-based systems. Some examples include selection of suppliers, selection of a new employee, job assignment, computer selection, market contact method selection, and determination of admission into graduate school.
3. Identify and interview an expert who knows the domain of your choice. Ask the expert to write down his or her knowledge. Choose an ES shell and build a prototype system to see how it works.

INTERNET EXERCISES

1. Go to **exsys.com** to play with the restaurant selection example in its demo systems. Analyze the variables and rules contained in the example's knowledge base.
2. In 1995, there were about 2,000 Web sites related to AI. Today, there are substantially more. Do a search at Google and describe how many Web sites you find. Categorize the first 20 into groups, or if you used a search engine that grouped them, what groups did you find?
3. Search the Internet using the keyword expert systems and describe what you find. List the first five applications in your search results and compare their differences.
4. Access the Web site of the American Association for Artificial Intelligence (**aaai.org**). Examine the workshops it has offered over the past year and list the major topics related to intelligent systems.
5. Search online to find a few ES development tools not listed in the chapter. Classify them into different categories.

END OF CHAPTER APPLICATION CASE

Business Rule Automation at Farm Bureau Financial Services

Financial service is a major area for ES because the process involves a number of complicated rules. Farm Bureau is a 60-year-old comprehensive financial services provider that offers a broad range of innovative products. With the 2003 merger of three Farm Bureau insurance companies into Farm Bureau Financial Services, the new management team believed that technology would enable the organization to improve the business processes each of the entities had performed in the past. One area the carrier sought to improve was underwriting, and it turned to a rule-based system that would eliminate many of the tasks previously required of underwriters.

"Farm Bureau's goal was to get the carrier's business rules out of the legacy code so the rules would be more manageable," explained Brett Clausen, vice president of the property/casualty companies for Farm Bureau. "The question we asked ourselves initially was 'How could we use this tool to achieve speed to market while at the same time underwriting more effectively and efficiently?'" Farm Bureau identified several benefits it expected to gain with rule-based underwriting. That list included reduction in workloads, better policy issuance times, improved consistency in underwriting risks, adaptable response to market regulatory changes, expense reduction, underwriters being able to focus more on exposures that create a higher liability to the company, and the ability to better manage and monitor results.

The benefits of using rule-based systems are quite clear. "Our challenge early on was a lot of our policies going through were being reviewed," Clausen said. "There was a very low percentage—about 10 percent to 12 percent—of transactions that were going through without being reviewed. We've increased that now to more than 60 percent of transactions going through without being reviewed."

With such an increase, Clausen pointed out that it was easy to see how many rules had changed in the past 2 years. The cautious pace allowed users to identify the business rules that were important and had an impact on business results and to eliminate some of the rules that didn't have an impact on decision making. "We wanted to be conservative and targeted 20 percent as a good pass-through rate," he said. "That would give us the ability to step back and look at what rules were firing and what rules were not firing." Audits were conducted to make sure the policies that were passing through without being looked at were quality business and also to ensure that a policy that the carrier normally would not have written didn't get through.

"We were pleased with that initial juncture, and then we started looking at the rules again," Clausen said. It wasn't hard for Farm Bureau's underwriters and customer service representatives to reexamine the rules because the results of those rules were showing up in their workload. As a result, Farm Bureau has reduced its transaction time by about 75 percent, Clausen claimed. "The amount of transactions that are coming through is phenomenal," he said. "In 2005, we had about 450,000 transactions come through our automated underwriting rules base. Roughly 250,000 of those passed without being looked at. When you look at the impact that has on the human resource perspective, getting our underwriters to focus on where we want them to focus, and our customer service representatives to focus on building relationships with our policyholders, that's a major step."

Questions for the Case

1. Describe the role of business rules at Farm Bureau and explain why those rules are very important.
2. Describe the benefits of using the rule-based system at Farm Bureau and explain why the system can generate these benefits.
3. Evaluate potential tools for implementing such a rule-based system and list the criteria that you would use in tool selection.
4. Find a proper application outside the financial services domain (such as manufacturing or retailing) and evaluate whether a rule-based system will generate the same level of benefits as those in the Farm Bureau system and explain why.

Sources: Compiled from R. Hyle, "Business Rules Streamline Underwriting for Farm Bureau Financial Services," *Tech Decisions,* February 2006; and **fbfs.com** (accessed March 2009).

References

Adomavicius, G., and A. Tuzhilin. (2005, June). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering,* Vol. 17, No. 6.

Byrd, T. A. (1995). "Expert Systems Implementation: Interviews with Knowledge Engineers." *Industrial Management and Data Systems,*Vol. 95, No. 10, 67–75.

ChessBase News. (2003, February 8). "Kasparov vs. Deep Junior Ends in 3-3." **chessbase.com/newsdetail.asp?newsid=782** (accessed June 2009).

Currie, A. (1999). "The History of Robotics." **faculty.ucr.edu/~currie/roboadam.htm** (accessed September 2006).

Davenport, T. (2004, October). "Decision Evolution." *CIO Magazine.*

Eom, S. B. (1996, September/October). "A Survey of Operational Expert Systems in Business (1980–1993)." *Interfaces,* Vol. 26, No. 5.

Eriksson, H. (1996, June). "Expert Systems as Knowledge Servers." *IEEE Expert.*

Feigenbaum, E., and P. McCorduck. (1983). *The Fifth Generation.* Reading, MA: Addison-Wesley.