

## Chapter 9

# Building Knowledge-Driven DSS and Mining Data

### INTRODUCTION

Some people claim “controlling knowledge leads to power.” Even if that claim is true, companies only “win” when relevant knowledge is shared among employees and other stakeholders. Today, sharing knowledge when making decisions is more important than most people recognize. One way to share knowledge is to build document-driven DSS. Another way is to build computerized systems that can store and retrieve knowledge codified as probabilities, rules, and relationships. Specialized software can process this knowledge and assist managers in making decisions. Specialized decision support and artificial intelligence (AI) tools can also help create knowledge. An umbrella term that describes these systems is knowledge-driven Decision Support Systems (DSS). These DSS provide suggestions to managers, and the dominant component is a “knowledge” capture and storage mechanism. Knowledge and suggestions are the two major themes that link these different knowledge tasks.

Knowledge-driven DSS, suggestion DSS, rule-based DSS and intelligent DSS are overlapping terms for DSS built using artificial intelligence technologies. Expert system development shells and data mining tools are often used to create these systems. Also, business and decision support analysts conduct special decision studies to identify relationships in very large databases using data mining or knowledge discovery tools. When a manager or knowledge worker uses a DSS with a data mining tool, the results from an analysis may suggest relationships and new knowledge.

This chapter is an introduction to and overview of knowledge-driven DSS technologies and applications. The first part of the chapter emphasizes expert system technologies and the second part emphasizes data mining techniques and tools. The overall thrust is to provide a foundation for building knowledge-

driven DSS with specialized artificial intelligence tools. These technologies have been “hyped” by some vendors as solutions to a wide variety of problems, but artificial intelligence technologies are still “leading-edge” capabilities for most businesses. At some point in the future, all managers and knowledge workers may be using knowledge-driven DSS and mining data, but that future is still over the horizon, waiting to be implemented.

So, the focus in this chapter is on examining how software can be used to store, process, find, and derive knowledge to support business decision making. The following sections emphasize terms, characteristics of knowledge-driven DSS, project management and examples of knowledge-driven DSS, an introduction to data mining, examples of data mining, and development tools evaluation.

## **KEY TERMS AND CONCEPTS**

Holsapple and Whinston (1996) discuss artificially intelligent DSS that “make use of computer-based mechanisms from the field of artificial intelligence.” These DSS provide suggestions for business decision makers. When the dominant component of a DSS uses artificial intelligence (AI) technologies, including expert system technologies and some data mining tools, to assist business decision makers, one can call the system a knowledge-driven DSS. Artificial intelligence is a branch of computer science that studies how computer software can imitate the cognitive activities of people. Every application of AI technologies should not be called a decision support system. This section discusses key terms associated with knowledge-driven DSS.

### **Knowledge-Driven DSS and Management Expert Systems**

Knowledge-driven DSS store and apply knowledge for a variety of specific business problems. These problems include classification and configuration tasks, such as loan credit scoring, fraud detection, and investment optimization.

Until recently, human experts had to perform classification and configuration tasks without computer support. Most of us identify a human expert as someone who is very knowledgeable in a particular area or subject. This human expert knows the appropriate questions to ask in order to draw a particular conclusion. In a similar way, one major type of expert system is a computer program that asks questions and reasons with the knowledge stored for the program about a narrow, specialized subject. This type of program attempts to solve a problem or give advice using heuristics.

In general, expert systems are programs with specialized problem-solving expertise. The expertise consists of three components: 1) knowledge of symptoms and indicators related to a particular topic or domain; 2) understanding of the relations among symptoms and of problems and solutions within that domain; and 3) “skill” or methods for solving some of the problems (cf., Power, 1985). An expert system is a knowledge-intensive program that captures the expertise of a human in a limited domain of knowledge and experience. It assists decision makers by asking relevant questions in a problem

domain and by recommending actions and explaining reasons for adopting an action.

An expert system can explain the reasoning behind a conclusion it has reached. This explanation capability is extremely important in auditing and validating the results from a knowledge-driven DSS. It also helps ensure that the system is in compliance with applicable policies, regulations, or legal requirements.

Using knowledge-driven DSS and management expert systems results in a number of benefits. Such systems can improve consistency in decision making, enforce policies, and regulations, distribute expertise to nonexpert staff, and retain valuable expertise for a company when experts retire or resign.

### **Data Mining and Knowledge Discovery**

Data mining and knowledge discovery are “hot” topics in the Information Systems and Marketing trade press. For many years companies have been storing large amounts of data, and more recently, companies have built large data warehouses. Now managers want to take advantage of the data they have collected by analyzing it, using statistical and artificial intelligence tools (cf., Berry & Linoff, 1997). Data mining techniques can help managers discover hidden relationships and patterns in data. Some analysts feel data mining can help a company gain a competitive advantage. Data mining tools can be used for both hypothesis testing and knowledge discovery. When vendors discuss data mining, they may be selling a set of end-user tools or a decision support capability or both. Managers and business analysts can perform data mining activities. Target users of these tools include financial analysts, statisticians, and marketing researchers. People who use these tools should have experience interpreting data.

### **Other Important Terms**

Artificial Intelligence researchers have a specialized vocabulary that has accumulated in the past 30 years. The following are some major terms that are relevant to developing knowledge-driven DSS. A *development environment* is used by a knowledge-driven DSS designer and builder. A development environment typically includes software for creating and maintaining a knowledge base and software called an inference engine. An *inference engine* reasons with a set of rules created by a developer.

A *domain expert* is a key person in a knowledge-driven DSS development project. A domain expert is the person who has expertise in the domain in which a specific system is being developed. A domain expert works closely with a *knowledge engineer* to capture the expert’s knowledge in a knowledge base. This process is used especially for capturing rule and relationship information in a computer readable format.

*Knowledge* refers to what one knows and understands. It is sometimes categorized as unstructured, structured, explicit, or implicit. Knowledge-driven DSS are built using explicit, structured knowledge. *Knowledge acquisition* is the

extraction and formulation of knowledge derived from various sources, especially from experts. A *knowledge base* is a collection of organized facts, rules, and procedures. A knowledge base has a description of the elements in the process along with their characteristics, functions, and relationships. It also contains rules about the actions to implement as a result of certain events. A knowledge base can also obtain its information from external programs and databases. When dealing with a particular task or problem, a knowledge-driven DSS constructs a number of hypotheses based on the external information supplied, its own knowledge, and the rules in its knowledge base.

If managers and MIS professionals want to build knowledge-driven DSS, they must have some familiarity with major AI terms. The key to success is learning some of the basic jargon and staying focused on the broader objective of building DSS that use software with “artificial reasoning” capabilities.

## CHARACTERISTICS OF KNOWLEDGE-DRIVEN DSS

There are a number of characteristics that are common to knowledge-driven DSS. First, this category of software assists—it doesn’t replace—managers in specific problem-solving tasks. Second, the systems use knowledge stored as rules, relationships or probabilities. Third, people interact with a knowledge-driven DSS when they are performing a specific decision task. Fourth, knowledge-driven DSS base recommendations on human expertise and derived knowledge and assist in performing very limited tasks. Fifth, a knowledge-driven DSS processes stored task relevant information and does *not* “think”.

A knowledge-driven DSS differs from a more conventional model-driven DSS in the way knowledge is presented and processed. This difference exists because expert systems attempt to simulate human reasoning processes. A model-driven DSS has a sequence of predefined instructions for responding to an event. In contrast, a knowledge-driven DSS, based on expert system technologies, attempts to reason about a response to an event using its knowledge base and logical rules for problem solving. Expert system technologies use representations of human knowledge. These representations are expressed in a special purpose language such as OPS5, PROLOG or LISP. Expert systems can also perform standard numerical calculations or data retrieval. An expert system development environment uses heuristic methods to obtain a recommendation. A heuristic is an approximate method that identifies varying amounts of uncertainty in conclusions. A conventional model-driven DSS uses mathematical and statistical methods to obtain a more precise solution.

Figure 9.1 shows the components of a knowledge-driven DSS. The model component is called an inference engine and it is the software that actually performs the reasoning function. The inference engine is the software that uses the knowledge represented in the data or knowledge base to draw its conclusions. The design of the inference engine may limit the ways in which knowledge can be represented in the knowledge base so that certain shells are only suitable for particular types of applications. In small systems, this is sometimes called the shell of the expert system, though the shell can be considered to be everything except the knowledge base itself.

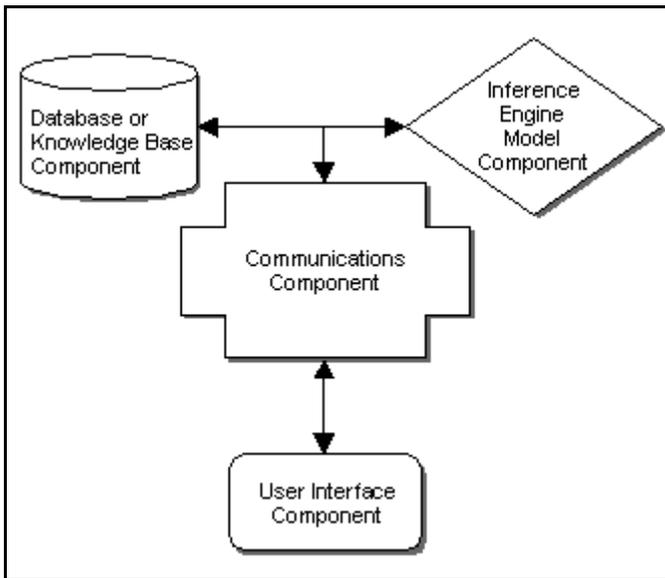


Figure 9.1 Components of a Knowledge-Driven DSS.

In comparing and identifying knowledge-driven and model-driven DSS, one should remember that knowledge-driven DSS have a knowledge base and an inference engine, and model-driven DSS have a structured database and quantitative models. If one compares Figure 9.1 to Figure 6.1, it is clear that the components in knowledge-driven DSS are similar to the general DSS architecture.

## MANAGING KNOWLEDGE-DRIVEN DSS PROJECTS

Knowledge-driven DSS should be initiated with a decision-oriented diagnosis and if the feasibility analysis is positive, then a small project team should complete a rapid prototyping development process. Many knowledge-driven DSS are built using rules and an expert system shell development environment. A knowledge engineer works with a domain expert to elicit rules and relationships. The testing and validation of the system may involve using prior examples and cases from the domain.

Several general rapid prototyping approaches for developing expert systems and knowledge-driven DSS have been proposed. Waterman (1986) proposed the following widely accepted approach:

1. Identification of a domain;
2. Conceptualization;
3. Formalization;
4. Implementation; and
5. Testing.

These five stages are highly interrelated and interdependent. An iterative process continues until the knowledge-driven DSS consistently performs at an acceptable level.

### Choosing a Knowledge-Driven DSS Project

If a business decision problem cannot readily be solved and supported using traditional methods, it may be appropriate to try an expert system solution. How does one choose an appropriate knowledge-driven DSS project? In general, the “telephone test” can be used to help determine if a task can be supported with a knowledge-driven DSS built using expert systems technologies. What is the “telephone test?” To apply the test, one asks “Can a domain expert solve the problem and support decision making using a telephone exchange with a decision maker?” To answer this question, it is often helpful to ask the domain expert to interact with a potential user of a proposed DSS over the telephone and to record the interaction that occurs. The domain expert should be told to ask structured rather than open-ended questions. Based on that exchange, if the answer is “Yes,” a telephone exchange works; then a knowledge-driven DSS, based on expert systems technologies, can be developed to support the decision maker. On the other hand, if the decision maker is unable to describe the problem verbally, or if the expert is consistently unable to recommend a reasonable solution, then development of a knowledge-driven DSS will likely be unsatisfactory. The telephone test ensures that the expert is not gaining additional information about a problem from other senses and that the user is able to adequately describe the problem in words.

### Using Rules for Knowledge-Driven DSS

Managers and developers should be familiar with the concept of a rule. A rule-based expert system has a large number of interconnected and nested IF-THEN statements or “rules” that are the basis for storing the knowledge in the system. Many expert system development environments store knowledge as rules.

The following is an example of a rule:

```
IF INCOME > $45,000 (condition)
AND IF SEX = "M" (condition)
THEN ADD to Target list (action)
```

A rule is a formal way of specifying a recommendation, directive, or strategy, expressed in an “ ‘ if premise, ‘ then’ conclusion” structure. Rules are one way of expressing declarative knowledge. For example, *if* a car won't start, *and* the lights are dim, *then* the car may have a dead battery. Thus, rules are relationships rather than instructions. Note this structure is different than the “if-then” structure used by procedural programming languages.

There are two ways an inference engine can manipulate rules. The first is forward chaining, where an inference engine starts from known facts and looks at the left-hand “if” side of the rules to find any matches and proceeds to find further rules that apply to the user’s responses. The second method is known as backward chaining. This technique involves starting the inference engine with a hypothesized solution by looking at the right-hand “then” statements, then working backwards to find the starting conditions that are necessary to arrive at that solution and see how they match with the user’s responses.

Let’s examine the two approaches from a different perspective. Suppose someone wants to fly from Waterloo, Iowa, to Beijing, China. To find a “chain” of connecting flights one can search in one of two ways:

1. Start with flights that arrive in Beijing and work backwards to eventually find a chain to Waterloo. This is a goal-driven, backward-chaining search.
2. Or, start by listing all flights leaving Waterloo and mark intermediary cities. Look for flights out of intermediaries until all paths to Beijing are found. This approach is working with forward-chaining toward a goal. This is a data-driven, forward-chaining search.

### **Advantages and Limitations of Rules**

Using an inference engine with rules is the most common development environment for knowledge-driven DSS. This preference for a rule technology is because rules are easy for managers and domain experts to understand. Also, using rules, it is easy to provide explanations to users of the DSS. From a developer’s perspective, modification and maintenance of a knowledge base of rules is relatively easy. Also, a developer can combine information about uncertainty in conclusions with rules. Finally, a number of rule-based development environments are available for implementing systems. There are, however, a number of major limitations of using a rule-based development approach. First, and most important, complex knowledge is difficult to represent using rules. Also, when rules are used, the knowledge represented tends to be superficial. Knowledge-driven DSS builders usually like developing systems based on rules, but using rules will not work for all applications.

### **KNOWLEDGE-DRIVEN DSS EXAMPLES**

More than 100 commercial expert systems were developed in the mid-80s in the first wave of enthusiasm about business applications of AI technologies. Many of these systems had fallen into disuse by the early 1990s. The systems generally failed because of managerial problems like lack of system acceptance or turnover of development staff. However, some systems were major successes (cf., Gill, 1995). Two classic examples of successful business expert systems are *TAXADVISOR* and *XCON*. More recent examples include a scheduling system for the Tomakomai paper mill, a customer support system at Compaq Computer, and an insurance plan selection system for Meiji Mutual Life Insurance Company.

*TAXADVISOR* was an expert system designed to assist an attorney with tax and estate planning for clients with large estates. The system collected client data and inferred actions the client needed to take to improve their financial profile, including insurance purchases, retirement actions, transfer of wealth, and modifications to gift and will provisions. *TAXADVISOR* used knowledge about estate planning based on attorneys' experiences and strategies as well as more generally accepted knowledge from textbooks. The system used a rule-based knowledge representation scheme controlled by backward chaining. *TAXADVISOR* was implemented in *EMYCIN* (cf., Michaelsen and Michie, 1983, Waterman, 1986). Accounting firm Coopers and Lybrand developed an early commercial expert system in this domain called *ExpertTax*.

*XCON* (eXpert CONfigurer of VAX 11/780 computer systems) was developed to configure computer systems. Based upon a customer's order, it recommended what components needed to be included to produce a complete operational system, and it determined the spatial relationships among all of the components. *XCON* was implemented in an expert system shell called *OPS5* and developed through a collaboration between researchers at Carnegie-Mellon University and Digital Equipment Corporation (now Compaq). This commercial expert system configured VAX computers on a daily basis and was for many years the largest and most mature rule-based expert system in operation (cf., McDermmot, 1984).

Scheduling and control systems are needed in the paper production industry to ensure that all machines in the mill operate correctly. At Ohji Paper Company's Tomakomai Mill, an expert system developed by Toshiba is used to schedule the paper production machines. The Tomakomai Mill consists of ten paper-making machines, energy supply plants, and pulp supply plants. Two hundred paper products are produced per month. Each product has a specified production volume and due date, and requires a specified machine to produce it. In the Tomakomai Mill, a millwide production management system exists. The system has a planning level and a control/operation level. The scheduling system receives product orders from the headquarters office, makes a schedule, and delivers it to the other planning systems. Each system schedules and optimizes its operations based on the paper-making schedule. The paper production scheduling system consists of an expert system for automated scheduling and a data management system. The scheduling systems are implemented with an expert shell utility, *ASIREX*. This scheduling system has been in practical use since January 1989. The greatest advantage reported for this system is that it speeds up scheduling. The scheduling time for a monthly schedule was reduced from three days to two hours (cf., Nakayama and Mizutani, 1990; <http://itri.loyola.edu/kb/toc.htm>).

Toshiba also sells a small knowledge system with 110 rules called *MARKETS-I*. It is a decision support system to determine the suitability of opening a convenience store at a particular site.

Compaq Computer Corporation created and implemented a very successful Customer Support Intelligent System to provide computer users expert diagnosis and recommendations about problems. In 1989, the service department started by installing a call logging system. Then, an expert system was added. Most

types of problems could be categorized as hardware, network, software, or general information (cf., Dhar and Stein, 1997).

The Meiji Mutual Life Insurance Company is one of the oldest life insurance companies in Japan with assets of around \$ 74 billion. Meiji offers a wide range of insurance and pension products. In addition, the company is aggressively involved in developing and introducing new products. However, with the increasing number of products, the company was finding it difficult to ensure that all the insurance sales staff had the expertise and the latest knowledge required to provide the best advice and service to its customers. To overcome this problem, Meiji used XpertRule to develop the *Life Insurance Plan Selection Expert System*. The system can select the most suitable product, along with a reason for the choice, from Meiji's range of 37 individual-oriented products. Meiji began research into expert systems in 1986. Before using XpertRule, the company had completed a Lisp-based insurance plan selection system. This system, however, had a high delivery and maintenance cost and was not suited for distribution to all branches. Meiji adopted XpertRule because it allows for easy knowledge base construction. The knowledge base contained 47 decision tasks. The rules for selecting each plan were developed as a separate task. The system was structured so that when the details of a customer are entered, the system assesses the suitability of all the plans and reports on the best five. The system takes less than four seconds to make.

The above examples suggest the wide variety of knowledge-driven decision support applications that are possible to construct using expert systems technologies. A number of Web sites have examples of knowledge-driven DSS that provide a more concrete idea of what is possible. For example, the Department of Labor has developed an interactive compliance assistance tool called Elaws designed to help users understand their rights as employees or employers at URL <http://www.dol.gov/elaws>.

## DATA MINING AND CREATING KNOWLEDGE

In the 1970s, companies employed business analysts who used statistical packages like SAS and SPSS to perform trend analyses and cluster analyses on data. As it became possible and affordable to store large amounts of data, managers wanted to access and analyze transaction data like that generated at a retail store cash register. Bar coding and the World Wide Web have also made it possible for companies to collect large amounts of new data.

Database marketing has also benefited from mining data. The information incorporated in the database marketing process is the historical database of previous mailings and the features associated with the (potential) customers, such as age, zip code, and their past responses. Data mining software uses this information to build a model of customer behavior that can be used to predict which customers are most likely to respond to a new catalog. By using this information, a marketing manager can target the customers likely to respond (cf., Thearling, 1998).

For many years, companies had statisticians study company data. When a statistician looks at the data, he or she makes a hypothesis about a relationship,

then performs a query on a database and uses statistical techniques to prove or disprove the hypothesis. This has been called the “verification mode” (IBM, 1998). Data mining software works in a “discovery mode” and looks for patterns. A hypothesis is not stated before the data is analyzed.

There are two main kinds of models in data mining: predictive and descriptive. Predictive models can be used to forecast explicit values, based on patterns determined from known results. For example, from a database of customers who have already responded to a particular offer, a model can be built that predicts which prospects are likeliest to respond to the same offer. The predictive model is then used in a DSS. Descriptive models describe patterns in existing data, and are generally used to create meaningful subgroups such as demographic clusters. Once a descriptive model is identified it may be used for target marketing or other decision support tasks.

### **Data Mining Techniques and Tools**

There are a wide variety of tools for data mining. The decision about which technique to use depends on the type of data and the type of questions that managers want answered by that data. Many commercial data mining software packages include more than one data mining tool. A summer 2000 Kdnuggets.com poll indicated SPSS's (spss.com) Clementine is the most-used data mining software package. It is targeted to business users and is a visual rapid modeling environment. Advanced sources of information on data mining tools include Berry and Linoff (1997) and Dhar and Stein (1997). This section examines five common categories for data mining tools: Case-Based Reasoning, Data Visualization, Fuzzy Query and Analysis, Genetic Algorithms, and Neural Networks (cf., Greenfield, 2000).

#### *Case-Based Reasoning*

Case-based tools find records in a database that are similar to specified records. A user specifies how strong a relationship should be before a new case is brought to her attention. This category of tools is also called memory-based reasoning. Software tries to measure the “distance” based on a measure of one record to other records and cluster records by similarity. This technique has been successful in analyzing relationships in free-form text. The Web site [www.ai-cbr.org](http://www.ai-cbr.org) is a resource for the artificial intelligence and case-based reasoning technology fields.

A five-step problem-solving process is used with case-based tools:

**Presentation:** a description of the current problem is input to the system.

**Retrieval:** the system retrieves the closest-matching cases stored in a database of cases.

**Adaptation:** the system uses the current problem and closest-matching cases to generate a solution to the current problem.

**Validation:** the solution is validated through feedback from the user of the environment.

**Update:** if appropriate, the validated solution is added to the case base for use in future problem solving (cf., Allen, 1994).

### *Data Visualization*

These tools graphically display complex relationships in multidimensional data from different perspectives. Visualization is the graphical presentation of information, with the goal of providing the viewer with a qualitative understanding of the information contents. Data visualization tools are data mining tools that translate complex formulas, mathematical relationships, or data warehouse information into graphs or other easily understood models. Statistical tools, like cluster analysis or classification and regression trees (CART), are often part of data visualization tools. Decision support analysts can visualize the clusters or examine a binary tree created by classifying records. In marketing, an analyst may create “co-occurrence” tables or charts of products that are purchased together. A good visualization is easy to understand and interpret, and it is a reasonably accurate representation of the underlying data.

### *Fuzzy Query and Analysis*

Fuzzy data mining tools allow users to look at results that are “close” to specified criteria. The user can vary what the definition of “close” is to help determine the significance and number of results that will be returned. This category of data mining tools is based on a branch of mathematics called fuzzy logic. The logic of uncertainty and “fuzziness” provides a framework for finding, scoring, and ranking the results of queries. Fuzzy Tech, a company that develops Fuzzy query software, has a Web site with excellent information on this tool at <http://www.fuzzytech.com/index.htm>.

### *Genetic Algorithms*

Genetic algorithms are optimization programs similar to the linear programming models discussed in Chapter 10. Genetic algorithm software conducts random experiments with new solutions while keeping the “good” interim results. A sample problem is to find the best subset of 20 variables to predict stock market behavior. To create a genetic model, the 20 variables would be identified as “genes” that have at least two possible values. The software would then select genes and their values randomly in an attempt to maximize or minimize a performance or fitness function. The performance function would provide a value for the fitness of the specific genetic model. Genetic optimization software also includes operators to combine and mutate genes. This quantitative model is used to find patterns, like other data mining techniques.

### *Neural Networks*

Neural network tools are used to predict future information by learning patterns from past data. According to Berry and Linoff (1997), neural networks are the most common type of data mining technique. Some people even think that using a neural network is the only type of data mining. For example, with appropriate input data a neural network could be trained to predict the price or net asset value of a mutual fund in the next quarter. A neural network could also be trained to categorize applicants for admission to a college into various “success” categories.

Vendors make many claims for neural networks. One claim that is especially questionable is that neural networks can compensate for low quality data. Neural networks attempt to learn patterns from data directly by repeatedly examining the data to identify relationships and build a model. Neural networks build models by trial and error. The network guesses a value that it compares to the actual number. If the guess is wrong, the model is adjusted. This learning process involves three iterative steps: predict, compare, and adjust. Neural networks are commonly used in a knowledge-driven DSS to classify data and, as noted, to make predictions. Figure 9.2 shows that various inputs (from  $I_1$  to  $I_j$ ) are transformed by a network of simple processors. The processors combine and weight the inputs and produce one or more output values ( $O_1$  to  $O_k$ ).

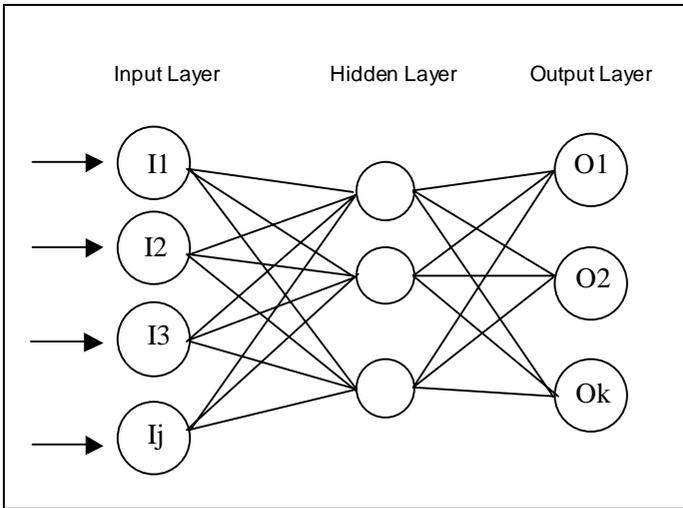


Figure 9.2 Neural Network Example.

### Data Mining Process

Data mining and knowledge discovery attempt to identify predictive relationships and provide managers with descriptive information about the subject of a database. There are a number of prescribed data mining processes. To make the best use of data mining, one must first make a clear statement of objectives. Researchers at IBM have described data mining as a three-phase process of data preparation, mining operations, and presentation. Analysts at the Gartner Group describes it similarly as a five-stage process:

1. Select and prepare the data to be mined.
2. Qualify the data via cluster and feature analysis.
3. Select one or more data mining tools.
4. Apply the data mining tool.
5. Apply the knowledge discovered to the company's specific line of business to achieve a business goal (Gerber, 1996).

These alternative processes can guide a special decision support study that uses data mining. In general, the first step is to select and prepare the data to be mined. Some data mining software packages include data preparation tools that can handle at least some of the preparation that needs to be done to the data. The second step is qualifying or testing the data using cluster and feature analysis software. This step takes some business knowledge about the question that one is trying to answer. This is the step where bias in the data should be detected and removed (IBM, 1998). In the third step an appropriate data mining tool is selected and used. Finally, the results are presented to decision makers, and if the results seem useful, a decision is influenced and one hopes business goals are achieved.

## DATA MINING EXAMPLES

What are some examples of data mining decision support applications? Some applications include: predicting which customers are likely to buy which products and when; improving credit/loan/mortgage risk analysis; identifying new untapped market segments that might be profitable; predicting what securities to buy/sell and when; improving customer service, support, satisfaction, and loyalty; understanding what factors affect profit and productivity; and detecting fraud earlier to avoid losses.

One specific example is conducting a special decision support study to identify characteristics of users of ATM cards at points of sale. Some people never use their ATM cards at points of sale; others use their cards only a couple of times per month; and some use their cards quite frequently. Frequent users generate the most revenue for the financial institution that issues the card. At one company, genetic data mining was used to evolve prediction models for several levels of card usage, based on parameters such as customer age, average checking account balance, and average number of checks written per month. Using these models of frequent users, the financial institution was able to target people matching the frequent-user profile for promotional campaigns (cf., <http://www.ultragem.com/sample.htm>).

Firstar Bank used data mining to determine which customers were likely to be interested in a new service. Data mining allowed Firstar to do target mailings, saving the company time and money compared to broad mailings to all customers. As a result of the targeted mailing, the response rate to the mailings increased by a factor of four (Freeman, 1997).

Siemens uses a DSS, built with case-based reasoning, to aid technical customer support services staff. The program uses the results of previous customer inquiries to help answer quickly questions from current inquiries.

As the result of a data mining project done at ShopKo, managers discovered that the sale of film does not cause the sale of a camera; however, the sale of a camera generally causes the sale of film. Data mining may find relationships that managers already knew existed. One hopes new knowledge and relationships are also discovered. American Century Investments used data mining to find information to help them cross-sell financial products to existing customers (cf., <http://www.spss.com>).

### **Keys to Success**

Developers at American Century shared a number of lessons they learned from this project. One lesson is that senior executive support, as well as IT support is necessary for success. Another lesson is that business issues must drive project development. If the project will not benefit the company, resources should not be allocated to it. They found that data mining often yields specific results rather than general rules. The quality of the data had a direct effect on the usefulness of the results. Finally, MIS staff at American Century found that data mining requires that analysts have statistical skills, business skills, and analytical skills in order for the company to get the most benefit from the tools.

### **EVALUATING DEVELOPMENT PACKAGES**

The following five criteria should be carefully considered when evaluating vendor software for either mining data or building knowledge-driven DSS.

*Cost.* With the significant costs of technology and the rapid advancement of new technologies, companies want affordable packages. A development environment with multiple tools is often better than purchasing a more specialized development package. In general, MIS staff want to learn software that can be applied to a wide variety of problems

*Scalability.* Companies need development software that will easily integrate with existing software applications and hardware platforms. Many knowledge-driven DSS need to be distributed to users, so Web technologies are often appropriate. Some observers want more managers and analysts to have data mining tools, so a distributed, scalable solution is also an issue in statistical analysis and knowledge discovery.

*Security.* With the increase in shared data, there is increasing concern regarding the security of DSS knowledge and large databases. Both rule bases and behavioral data that will be mined need to be protected. Security is easily overlooked in developing knowledge applications.

*Development features.* Knowledge-driven DSS are not usually standard “off-the-shelf” packages. It is important that packages allow for easy development of customized capabilities, rule input, and maintenance. If uncertainties, frames, or other capabilities are part of the development environment, then the package needs to help ensure that features and capabilities are used appropriately.

*Ease of installation and use.* Managers and MIS staff want software packages that are easy to install and require minimal training. This criterion is especially important with end-user data mining tools.

### **CONCLUSIONS AND COMMENTARY**

Knowledge-driven DSS and mining data are at the decision support frontier in organizations. During the 1980s, unrealistic expectations were created for expert systems and the recent “hype” about data mining has also created some skepticism. Managers and IS staff need to investigate how these technologies

might solve real business problems, but caution should be used in selling knowledge-driven DSS and data mining projects.

Data mining techniques and tools are not fundamentally different from the older quantitative model-building techniques. The methods used in data mining are extensions and generalizations of analytical methods known for decades. Neural networks are a special case of what is called projection pursuit regression, a method developed in the 1940s. Classification and regression tree (CART) methods were used by social scientists in the 1960s (cf., <http://www.twocrows.com/iwk9701.htm>). The computing technology used to implement these underlying methods has, however, greatly improved.

For the foreseeable future, modest knowledge-driven DSS projects can provide some benefits and can help MIS staff develop experience using expert system and data mining tools. It is important for large companies to have projects in this category of DSS, but only modest resources should be committed in most companies. The list of possible applications in this chapter should guide the selection of new projects.

Some observers would include document-driven DSS as knowledge management tools. This may be the case, but more importantly, both types of systems can support decision makers.