2

FUNDAMENTALS OF EXPERT SYSTEMS

This chapter introduces the basic concepts of expert systems. The hierarchical process of developing expert systems is presented, as well as the essential characteristics of expert systems are presented. More specific details of the concepts introduced in this chapter are covered in subsequent chapters.

2.1   EXPERT SYSTEMS PROCESS

This book is organized in the structure of a strategic process for developing successful expert systems. Figure 2.1 presents the hierarchy of topics as they are presented here and in the subsequent chapters. The strategic process is recommended for anyone venturing into the technology of expert systems from the standpoint of training, research, or applications. This chapter covers the basic concepts of expert systems technology. A basic understanding of these concepts is essential to getting the most out of expert systems. More specific details of the concepts presented in this chapter are discussed in appropriate sections of the subsequent chapters. Chapter 3 covers problem analysis. To be effective, the right problems must be selected for expert systems implementation. The principle of ''garbage in, garbage out'' is also applicable here. Wrong problems lead to incorrect implementation of expert systems.

Chapter 4 covers knowledge engineering. Knowledge acquisition is a critical aspect of the expert systems effort. If the knowledge collected is garbage, the best that can be expected from a system is garbage. Chapter 5 presents probabilistic and fuzzy reasoning. Chapter 6 presents fuzzy systems techniques for handling uncertainty in expert systems. Chapter 7 presents neural networks.

Chapter 8 covers neural-fuzzy networks. Chapter 9 presents the technique of evolutionary computing. Chapter 10 presents an application to manufacturing. Chapter 11 presents an application to forecasting.
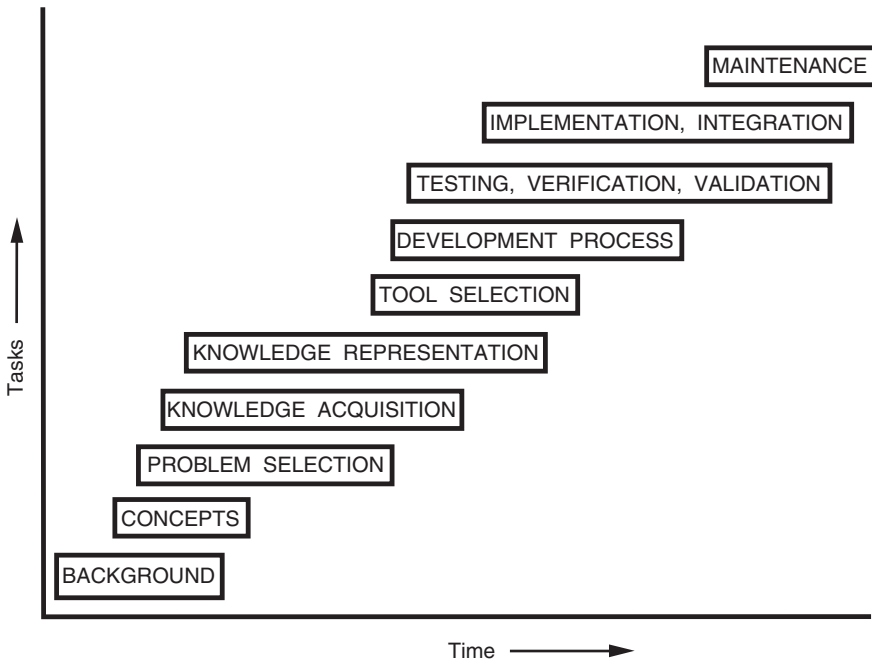
Figure 2.1. Hierarchy of expert systems development process.

2.2   EXPERT SYSTEMS CHARACTERISTICS

By definition, an expert system is a computer program that simulates the thought process of a human expert to solve complex decision problems in a specific domain. This chapter addresses the characteristics of expert systems that make them different from conventional programming and traditional decision support tools. The growth of expert systems is expected to continue for several years. With the continuing growth, many new and exciting applications will emerge. An expert system operates as an interactive system that responds to questions, asks for clarification, makes recommendations, and generally aids the decision-making process. Expert systems provide expert advice and guidance in a wide variety of activities, from computer diagnosis to delicate medical surgery.

Various definitions of expert systems have been offered by several authors. A general definition that is representative of the intended functions of expert systems is:

An *expert system* is an interactive computer-based decision tool that
uses both facts and heuristics to solve difficult decision problems based on
knowledge acquired from an expert.

An expert system may be viewed as a computer simulation of a human expert. Expert systems are an emerging technology with many areas for potential applications. Past applications range from MYCIN, used in the medical field to diagnose infectious blood diseases, to XCON, used to configure computer systems. These expert systems have proven to be quite successful. Most applications of expert systems will fall into one of the following categories:

- Interpreting and identifying
- Predicting
- Diagnosing
- Designing
- Planning
- Monitoring
- Debugging and testing
- Instructing and training
- Controlling

Applications that are computational or deterministic in nature are not good candidates for expert systems. Traditional decision support systems such as spreadsheets are very mechanistic in the way they solve problems. They operate under mathematical and Boolean operators in their execution and arrive at one and only one static solution for a given set of data. Calculation-intensive applications with very exacting requirements are better handled by traditional decision support tools or conventional programming. The best application candidates for expert systems are those dealing with expert heuristics for solving problems. Conventional computer programs are based on factual knowledge, an indisputable strength of computers. Humans, by contrast, solve problems on the basis of a mixture of factual and heuristic knowledge. Heuristic knowledge, composed of intuition, judgment, and logical inferences, is an indisputable strength of humans. Successful expert systems will be those that combine facts and heuristics and thus merge human knowledge with computer power in solving problems. To be effective, an expert system must focus on a particular problem domain, as discussed below.

2.2.1   Domain Specificity

Expert systems are typically very domain specific. For example, a diagnostic expert system for troubleshooting computers must actually perform all the necessary data manipulation as a human expert would. The developer of such a system must limit his or her scope of the system to just what is needed to solve the target problem. Special tools or programming languages are often needed to accomplish the specific objectives of the system.

2.2.2   Special Programming Languages

Expert systems are typically written in special programming languages. The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process. The major advantage of these languages, as compared to conventional programming languages, is the simplicity of the addition, elimination, or substitution of new rules and memory management capabilities. Some of the distinguishing characteristics of programming languages needed for expert systems work are:

- Efficient mix of integer and real variables
- Good memory-management procedures
- Extensive data-manipulation routines
- Incremental compilation
- Tagged memory architecture
- Optimization of the systems environment
- Efficient search procedures

2.3   EXPERT SYSTEMS STRUCTURE

Complex decisions involve intricate combination of factual and heuristic knowledge. In order for the computer to be able to retrieve and effectively use heuristic knowledge, the knowledge must be organized in an easily accessible format that distinguishes among data, knowledge, and control structures. For this reason, expert systems are organized in three distinct levels:

1. *Knowledge base* consists of problem-solving rules, procedures, and intrinsic data relevant to the problem domain.
2. *Working memory* refers to task-specific data for the problem under consideration.
3. *Inference engine* is a generic control mechanism that applies the axiomatic knowledge in the knowledge base to the task-specific data to arrive at some solution or conclusion.

These three pieces may very well come from different sources. The inference engine, such as VP-Expert, may come from a commercial vendor. The knowledge base may be a specific diagnostic knowledge base compiled by a consulting firm, and the problem data may be supplied by the end user. A knowledge base is the nucleus of the expert system structure. A knowledge base is not a data base. The traditional data base environment deals with data that have a static relationship between the elements in the problem domain. A knowledge base is created by knowledge engineers, who translate the knowledge of real human experts into rules and strategies. These rules and
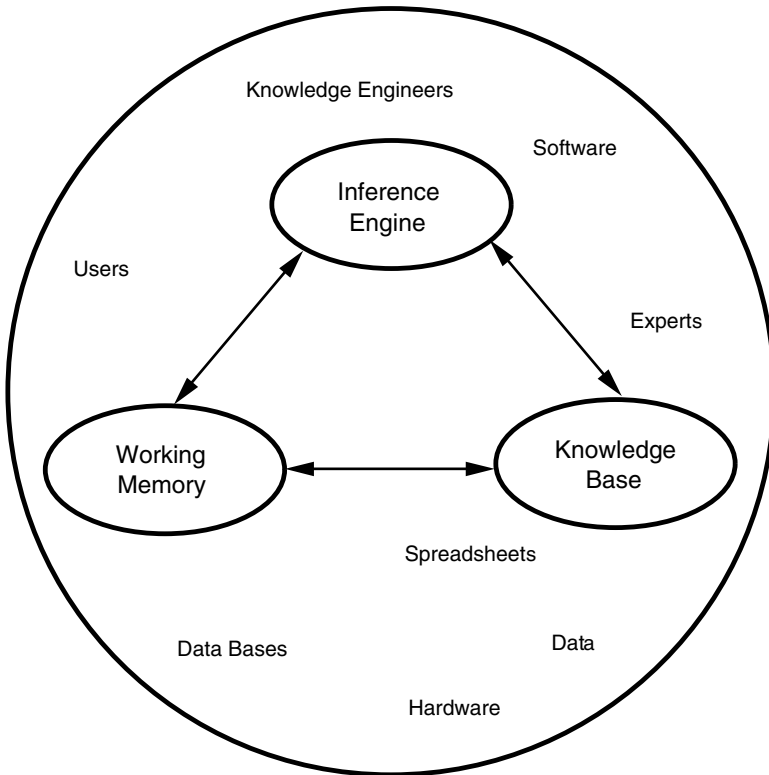
Figure 2.2. Expert systems organization and operating environment.

strategies can change depending on the prevailing problem scenario. The knowledge base provides the expert system with the capability to recommend directions for user inquiry. The system also instigates further investigation into areas that may be important to a certain line of reasoning but not apparent to the user.

The modularity of an expert system is an important distinguishing characteristic compared to a conventional computer program. Modularity is effected in an expert system by the use of three distinct components, as shown in Figure 2.2.

The knowledge base constitutes the problem-solving rules, facts, or intuition that a human expert might use in solving problems in a given problem domain. The knowledge base is usually stored in terms of if–then rules. The working memory represents relevant data for the current problem being solved. The inference engine is the control mechanism that organizes the problem data and searches through the knowledge base for applicable rules. With the increasing popularity of expert systems, many commercial inference engines are coming onto the market. A survey of selected commercial infer-
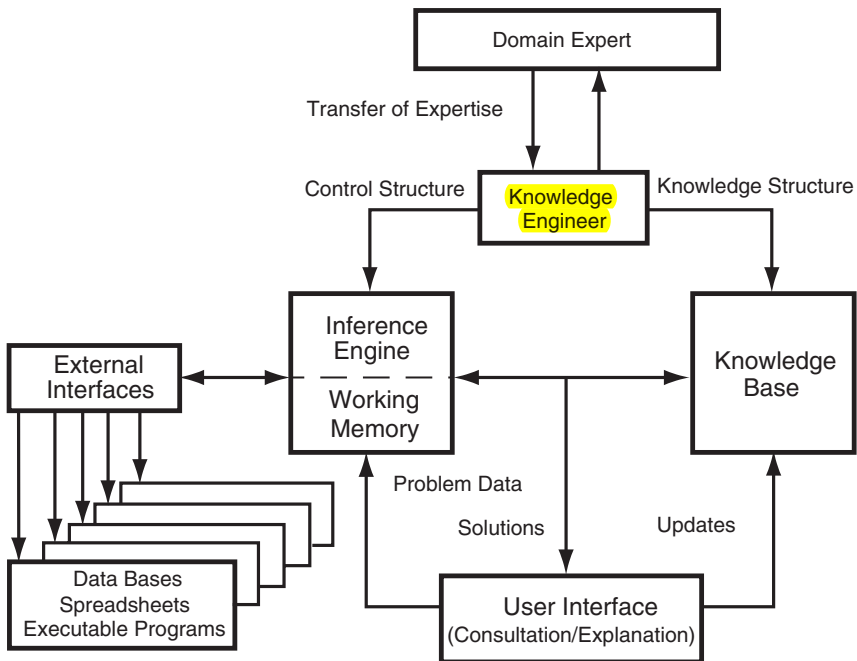
Figure 2.3. Integration of expert systems components.

ence engines is presented in the Appendix at the end of this book. The development of a functional expert system usually centers around the organization of the knowledge base. A functional integration of expert systems components is shown in Figure 2.3.

A good expert system is expected to grow as it learns from user feedback. Feedback is incorporated into the knowledge base as appropriate to make the expert system smarter. The dynamism of the application environment for expert systems is based on the individual dynamism of the components. This can be classified as follows:

- *Most dynamic: Working memory.* The contents of the working memory, sometimes called the data structure, changes with each problem situation. Consequently, it is the most dynamic component of an expert system, assuming, of course, that it is kept current.
- *Moderately dynamic: Knowledge base.* The knowledge base need not change unless a new piece of information arises that indicates a change in the problem solution procedure. Changes in the knowledge base should be carefully evaluated before being implemented. In effect, changes should not be based on just one consultation experience. For example, a rule that is found to be irrelevant under one problem situation may turn out to be crucial in solving other problems.

- *Least dynamic: Inference engine.* Because of the strict control and coding structure of an inference engine, changes are made only if absolutely necessary to correct a bug or enhance the inferential process. Commercial inference engines, in particular, change only at the discretion of the developer. Since frequent updates can be disruptive and costly to clients, most commercial software developers try to minimize the frequency of updates.

2.3.1   The Need for Expert Systems

Expert systems are necessitated by the limitations associated with conventional human decision-making processes, including:

1. Human expertise is very scarce.
2. Humans get tired from physical or mental workload.
3. Humans forget crucial details of a problem.
4. Humans are inconsistent in their day-to-day decisions.
5. Humans have limited working memory.
6. Humans are unable to comprehend large amounts of data quickly.
7. Humans are unable to retain large amounts of data in memory.
8. Humans are slow in recalling information stored in memory.
9. Humans are subject to deliberate or inadvertent bias in their actions.
10. Humans can deliberately avoid decision responsibilities.
11. Humans *lie, hide*, and *die.*

Coupled with these human limitations are the weaknesses inherent in conventional programming and traditional decision-support tools. Despite the mechanistic power of computers, they have certain limitations that impair their effectiveness in implementing human-like decision processes. Conventional programs:

1. Are algorithmic in nature and depend only on raw machine power
2. Depend on facts that may be difficult to obtain
3. Do not make use of the effective heuristic approaches used by human experts
4. Are not easily adaptable to changing problem environments
5. Seek explicit and factual solutions that may not be possible

2.3.2   Benefits of Expert Systems

Expert systems offer an environment where the good capabilities of humans and the power of computers can be incorporated to overcome many of the limitations discussed in the previous section. Expert systems:

1. Increase the probability, frequency, and consistency of making good decisions
2. Help distribute human expertise
3. Facilitate real-time, low-cost expert-level decisions by the nonexpert
4. Enhance the utilization of most of the available data
5. Permit objectivity by weighing evidence without bias and without regard for the user's personal and emotional reactions
6. Permit dynamism through modularity of structure
7. Free up the mind and time of the human expert to enable him or her to concentrate on more creative activities
8. Encourage investigations into the subtle areas of a problem

Expert Systems Are For Everyone. No matter which area of business one is engaged in, expert systems can fulfill the need for higher productivity and reliability of decisions. Everyone can find an application potential in the field of expert systems. Contrary to the belief that expert systems may pose a threat to job security, expert systems can actually help to create opportunities for new job areas. Presented below are some areas that hold promise for new job opportunities:

- Basic research
- Applied research
- Knowledge engineering
- Inference engine development
- Consulting (development and implementation)
- Training
- Sales and marketing
- Passive or active end user

An active user is one who directly uses expert systems consultations to obtain recommendations. A passive user is one who trusts the results obtained from expert systems and supports the implementation of those results.

2.3.3   Transition from Data Processing to Knowledge Processing

What data has been to the previous generations of computing, knowledge is to the present generation of computing. Expert systems represent a revolutionary transition from the traditional data processing to knowledge processing. Figure 2.4 illustrates the relationships between the procedures for data processing and knowledge processing to make decisions. In traditional data processing the decision maker obtains the information generated and performs an explicit analysis of the information before making his or her decision. In
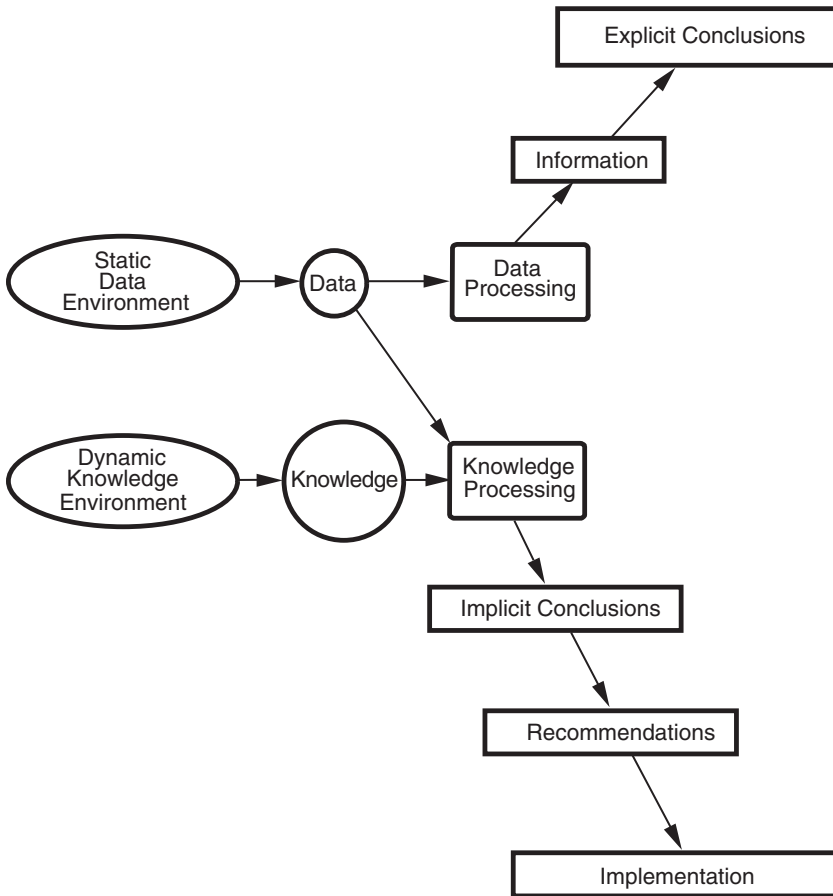
Figure 2.4. Data processing versus knowledge processing.

an expert system knowledge is processed by using available data as the processing fuel. Conclusions are reached and recommendations are derived implicitly. The expert system offers the recommendation to the decision maker, who makes the final decision and implements it as appropriate. Conventional data can now be manipulated to work with durable knowledge, which can be processed to generate timely information, which is then used to enhance human decisions.

2.4 HEURISTIC REASONING

Human experts use a type of problem-solving technique called heuristic reasoning. Commonly called rules of thumb or expert heuristics, it allows the

expert to arrive at a good solution quickly and efficiently. Expert systems base their reasoning process on symbolic manipulation and heuristic inference procedures that closely match the human thinking process. Conventional programs can only recognize numeric or alphabetic strings and manipulate them only in a preprogrammed manner.

2.4.1 Search Control Methods

All expert systems are search intensive. Many techniques have been employed to make these intensive searches more efficient. Branch and bound, pruning, depth-first search, and breadth-first search are some of the search techniques that have been explored. Because of the intensity of the search process, it is important that good search control strategies be used in the expert systems inference process.

2.4.2 Forward Chaining

This method involves checking the condition part of a rule to determine whether it is true or false. If the condition is true, then the action part of the rule is also true. This procedure continues until a solution is found or a dead end is reached. Forward chaining is commonly referred to as data-driven reasoning. Further discussions of forward chaining are presented in subsequent chapters.

2.4.3 Backward Chaining

Backward chaining is the reverse of forward chaining. It is used to backtrack from a goal to the paths that lead to the goal. Backward chaining is very good when all outcomes are known and the number of possible outcomes is not large. In this case, a goal is specified and the expert system tries to determine what conditions are needed to arrive at the specified goal. Backward chaining is thus also called goal-driven. More details are provided on the backward chaining process in Chapter 5.

2.5 USER INTERFACE

The initial development of an expert system is performed by the expert and the knowledge engineer. Unlike most conventional programs, in which only programmers can make program design decisions, the design of large expert systems is implemented through a team effort. A consideration of the needs of the end user is very important in designing the contents and user interface of expert systems.

2.5.1 Natural Language

The programming languages used for expert systems tend to operate in a manner similar to ordinary conversation. We usually state the premise of a problem in the form of a question, with actions being stated much as when we verbally answer the question, that is, in a ''natural language'' format. If, during or after a consultation, an expert system determines that a piece of its data or knowledge base is incorrect or is no longer applicable because the problem environment has changed, it should be able to update the knowledge base accordingly. This capability would allow the expert system to converse in a natural language format with either the developers or users.

Expert systems not only arrive at solutions or recommendations, but can give the user a level of confidence about the solution. In this manner, an expert system can handle both quantitative and qualitative factors when analyzing problems. This aspect is very important when we consider how inexact most input data are for day-to-day decision making. For example, the problems addressed by an expert system can have more than one solution or, in some cases, no definite solution at all. Yet the expert system can provide useful recommendations to the user just as a human consultant might do.

2.5.2 Explanations Facility in Expert Systems

One of the key characteristics of an expert system is the explanation facility. With this capability, an expert system can explain how it arrives at its conclusions. The user can ask questions dealing with the what, how, and why aspects of a problem. The expert system will then provide the user with a trace of the consultation process, pointing out the key reasoning paths followed during the consultation. Sometimes an expert system is required to solve other problems, possibly not directly related to the specific problem at hand, but whose solution will have an impact on the total problem-solving process. The explanation facility helps the expert system to clarify and justify why such a digression might be needed.

2.5.3 Data Uncertainties

Expert systems are capable of working with inexact data. An expert system allows the user to assign probabilities, certainty factors, or confidence levels to any or all input data. This feature closely represents how most problems are handled in the real world. An expert system can take all relevant factors into account and make a recommendation based on the best possible solution rather than the only exact solution.

2.5.4 Application Roadmap

The symbolic processing capabilities of AI technology lead to many potential applications in engineering and manufacturing. With the increasing sophisti-
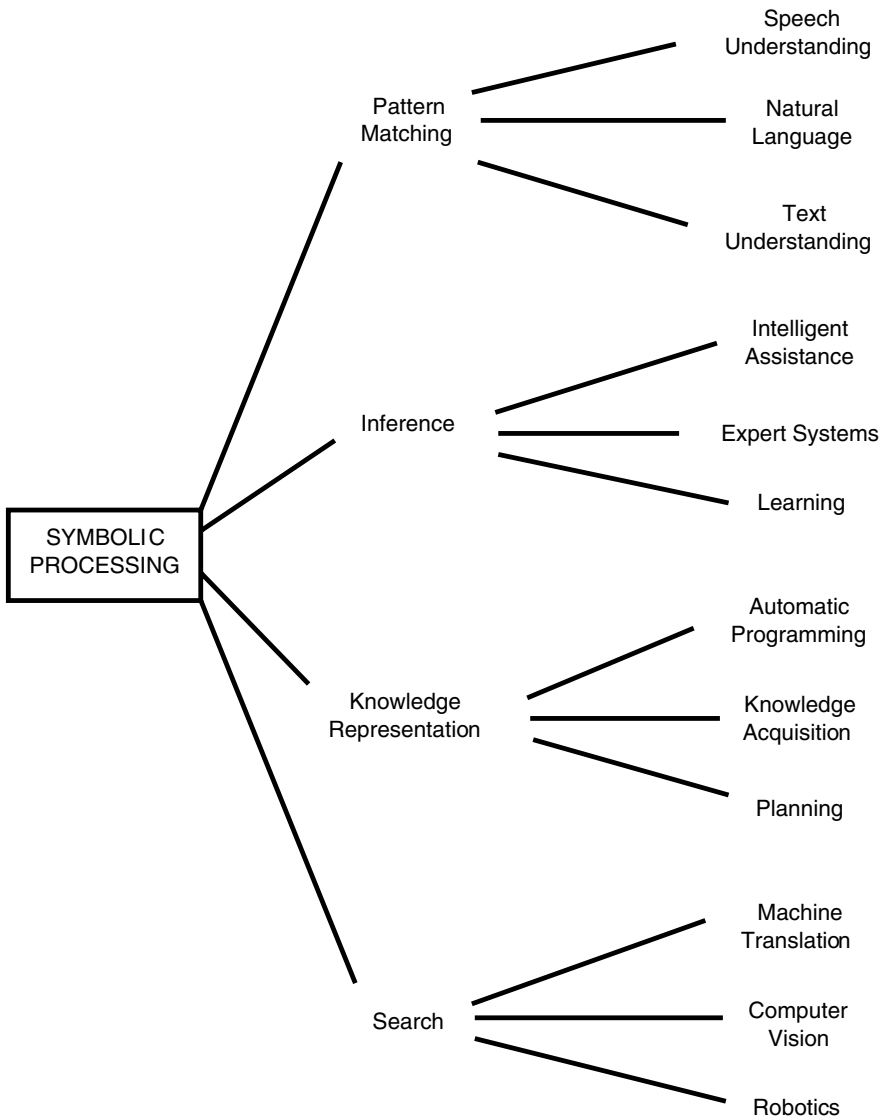
Figure 2.5. Application roadmap for expert systems.

cation of AI techniques, analysts are now able to use innovative methods to provide viable solutions to complex problems in everyday applications. Figure 2.5 presents a structural representation of the application paths for artificial intelligence and expert systems.

2.5.5 Symbolic Processing

Contrary to the practice in conventional programming, expert systems can manipulate objects symbolically to arrive at reasonable conclusions to a prob-
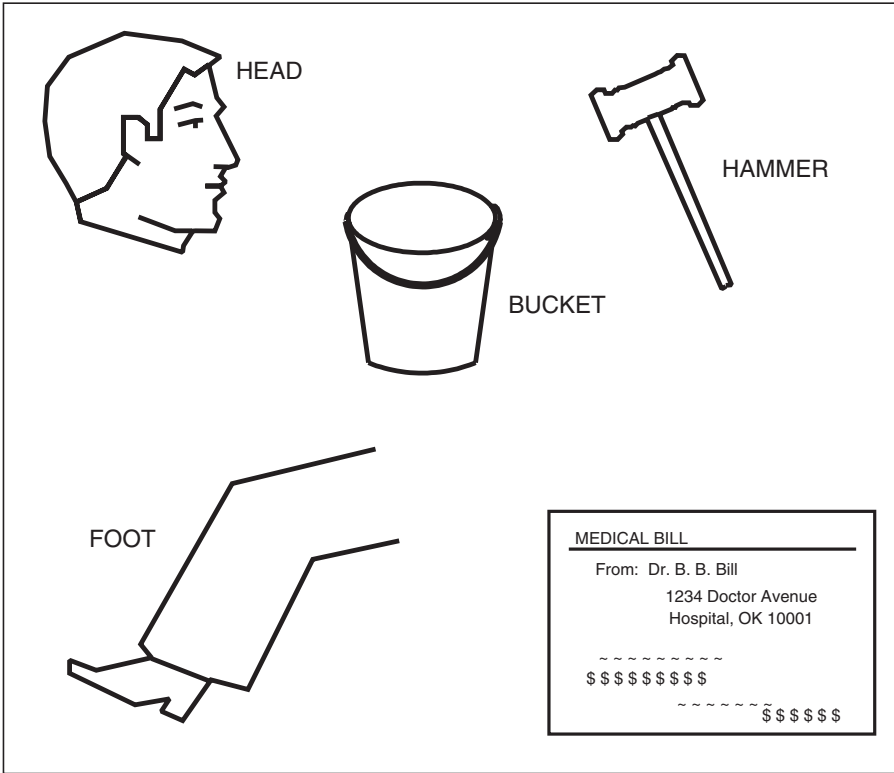
Figure 2.6. Collection of common objects.

lem scenario. The object drawings in this section are used to illustrate the versatility of symbolic processing by using the manipulation of objects to convey information. Let us assume that we are given the collection of five common objects as shown in Figure 2.6. The objects are Head, Hammer, Bucket, Foot, and Bill (as in doctor's bill). We can logically arrange a subset of the set of given objects to convey specific inferences. In Figure 2.7, four of the five objects are arranged in the order Hammer, Head, Foot, and Bucket. This unique arrangement may be represented by the equation presented below:

$$\text{Hammer} \sim \text{Head} = \text{Foot} \sim \text{Bucket}$$

It is desired to infer a reasonable statement of the information being conveyed by the symbolic arrangement of objects in Figure 2.7. Figure 2.8 presents an alternative arrangement of another subset (hammer, foot, foot, and bill) of the given objects. This alternative arrangement may be represented by the equation shown below:
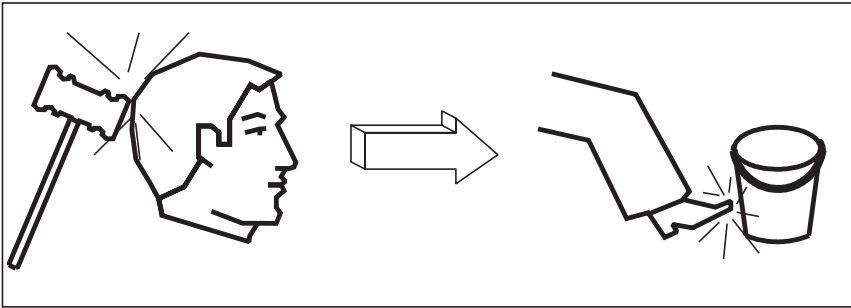
Figure 2.7. Arrangement of common objects.

Hammer ~ Foot = Foot ~ Bill

It is desired to infer a reasonable statement from Figure 2.8. It should be noted that ordinary mathematical reasoning concerning the equation hammer ~ foot = foot ~ bill might lead to Hammer = Bill. However, in artificial intelligence symbolic reasoning, the context of the arrangement of the objects will determine the proper implication.

Figure 2.7: If Hammer smashes Head, then victim kicks the bucket (i.e., dies). In this case, the action part of the statement relates to an action (a fatal one) by the victim of the assault.

Figure 2-8: If Hammer smashes Foot, then assailant foots the bill. In this case, the action part of the statement relates to a compensatory action (restitution) by the assailant.

Using a finite set of symbolic objects, we can generate different pieces of information with different permutations of the objects. A particularly interesting aspect of symbolic processing is noted in Figure 2.8. The object Foot
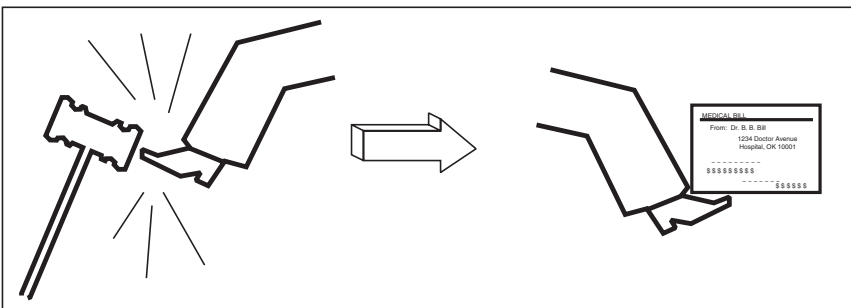

Figure 2.8. Alternate arrangement of objects.

conveys one meaning when concatenated with one given object (Hammer) and another totally different meaning when concatenated with another object (bill). In fact, the identification of the object Bill is itself symbolically conveyed by the contents of the medical bill in Figure 2.6. With the illustrated capability of symbolic processing, very powerful AI-based tools can be developed for practical applications. However, more research and development efforts will be needed before many of those practical applications can be realized.