

17

Approved for public release. Distribution unlimited.

Artificial Intelligence Technology Assessment for the U.S. Army Depot System Command

K. A. Pennock

July 1991

Prepared for the U.S. Department of the Army
under a Related Services Agreement
with the U.S. Department of Energy
Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute



DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes any **warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831;
prices available from (615) 576-8401. FTS 626-8401.

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

Approved for public release.
Distribution unlimited.

ARTIFICIAL INTELLIGENCE TECHNOLOGY ASSESSMENT
FOR THE U.S. ARMY DEPOT SYSTEM COMMAND

K. A. Pennock

July 1991

Prepared for
the U.S. Department of the Army
under a Related Services Agreement
with the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

SUMMARY

This assessment of artificial intelligence (AI) has been prepared for the U.S. Army's Depot System Command (DESCOM) by Pacific Northwest Laboratory.^(a) The report describes several of the more promising AI technologies, focusing primarily on knowledge-based systems because they have been more successful in commercial applications than any other AI technique. The report also identifies potential Depot applications in the areas of procedural support, scheduling and planning, automated inspection, training, diagnostics, and robotic systems.

One of the principal objectives of the report is to help decisionmakers within DECOM to evaluate AI as a possible tool for solving individual depot problems. The report identifies a number of factors that should be considered in such evaluations.

(a) Pacific Northwest Laboratory is operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-ACO6-76RLO 1830.

CONTENTS

SUMMARY	iii
1.0 INTRODUCTION	1.1
1.1 BACKGROUND	1.1
1.2 ADVANTAGES OF ARTIFICIAL INTELLIGENCE	1.4
2.0 METHODS IN ARTIFICIAL INTELLIGENCE	2.1
2.1 RULE-BASED EXPERT SYSTEMS	2.1
2.1.1 Composition of Expert Systems	2.2
2.1.2 The Knowledge Base	2.3
2.1.3 The Inference Engine	2.4
2.1.4 The Human-Computer Interface	2.4
2.1.5 Applications of Expert Systems	2.5
2.2 MODEL-BASED REASONING SYSTEMS	2.5
2.2.1 Qualitative Physics	2.7
2.3 BLACKBOARD SYSTEMS	2.8
2.3.1 Composition	2.9
2.3.2 Problem-Solving Activity	2.11
2.3.3 Advantages	2.12
2.4 NEURAL NETWORKS	2.12
2.4.1 Physical Structure of the Network	2.13
2.4.2 Data Types	2.14
2.4.3 Activation Rates	2.14
2.4.4 Learning Rules	2.14

2.4.5	Computational Rule	2.15
2.4.6	Future Applications	2.16
2.5	ADAPTIVE SEARCH ALGORITHMS	2.16
2.6	GENETIC ALGORITHMS	2.17
3.0	APPLICATIONS IN ARTIFICIAL INTELLIGENCE	3.1
3.1	CONTROL	3.1
3.2	DESIGN	3.3
3.3	DIAGNOSIS	3.5
3.4	TRAINING	3.6
3.5	MONITORING	3.8
3.6	PLANNING	3.9
3.7	PREDICTION	3.11
3.8	REPAIR	3.13
3.9	PERCEPTION AND ANALYSIS	3.14
4.0	TOOLS	4.1
4.1	SOFTWARE TOOLS	4.1
4.2	GENERAL-PURPOSE LANGUAGES	4.3
4.3	ARTIFICIAL INTELLIGENCE PROGRAMMING LANGUAGES	4.4
4.3.1	Functional Languages	4.4
4.3.2	Logical Languages	4.4
4.3.3	Object-Oriented Languages.....	4.5
4.3.4	Knowledge Engineering Tools	4.6
4.3.5	Small Systems	4.7

4.3.6	Large Systems	4.8
4.4	HARDWARE TOOLS	4.8
4.4.1	Personal Computers	4.10
4.4.2	Workstations	4.11
4.4.3	Dedicated Artificial Intelligence Machines	4.11
5.0	EMERGING ISSUES IN ARTIFICIAL INTELLIGENCE	5.1
5.1	PARALLEL PROCESSING	5.1
5.2	PARALLEL HARDWARE	5.1
5.3	SOFTWARE DEVELOPMENT	5.2
6.0	DEPOT APPLICATIONS	6.1
6.1	PROCEDURAL SUPPORT	6.4
6.2	SCHEDULING AND PLANNING	6.5
6.3	AUTOMATED INSPECTION TASKS	6.7
6.4	PERSONNEL TRAINING (INSTRUCTION)	6.9
6.5	DIAGNOSTIC SYSTEMS	6.12
6.6	ENHANCEMENT OF ROBOTIC SYSTEM FLEXIBILITY	6.13
7.0	EVALUATION OF APPLICATIONS	7.1
7.1	DEPOT-RELATED TASKS SUITABLE FOR AI APPLICATIONS	7.1
7.2	APPROPRIATENESS OF APPLICATIONS	7.1
8.0	REFERENCES	8.1

FIGURES

1.	Expert System Components	2.3
2.	Blackboard System Structure	2.10
3.	Neural Network	2.12
4.	Operation of a Genetic Algorithm	2.20
5.	Control Loop.....	3.1
6.	Initial Values to Goal Values	3.2
7.	Planning Example	3.10
8.	Predictive System	3.11
9.	Perceptive System.....	3.15
10.	Expert System Interface	6.5
11.	Perception and Recognition System	6.7

1.0 INTRODUCTION

This report, prepared at the request of the U.S. Army Depot System Command (DESCOM), provides a general technical assessment of artificial intelligence (AI) for an audience with limited knowledge of this specialized field. The assessment is intended to assist managers in deciding whether artificial intelligence is an appropriate tool for solving individual depot problems. It identifies optional approaches and methods viable for developing real-world applications. This report has three main objectives:

1. to acquaint the reader with AI - The characteristics of AI versus traditional computing will be discussed, as well as common methods or techniques of artificial intelligence.
2. to define common applications for AI - General application categories will be discussed to afford the reader a broad understanding of the role of AI in the real world. Potential Depot applications will subsequently be discussed, including references to examples from industry. Finally, a brief checklist for evaluation of candidate problems for expert system development will be provided.
3. to provide the user with a representative list of available hardware and software AI products - The list is not comprehensive; new products are released daily, and existing products are frequently enhanced. The list provided is intended as a starting point for exploration of available commercial products.

Because most successful commercial applications are knowledge-based systems (also known as expert systems), they are given a more thorough discussion than the other, less frequently applied AI techniques. This report describes what knowledge-based systems are and how they are used and also provides a brief outline of the characteristics of other successful applications (Section 7). More in-depth information concerning evaluation of candidate problems is provided in a report entitled A Methodology of Evaluating Potential Knowledge-Based Systems Applications (Melton et. al. 1989), which considers goals, appropriateness, resources, and other considerations involved in the analysis of a potential knowledge-based system application.

1.1 BACKGROUND

Digital computers have been used to perform mathematical calculations for decades. It is the goal of AI to allow computers to go beyond a strictly mathematical framework, extending

the uses of the computer as a human tool. AI is the discipline involved with enabling machines to learn new information, to reason, and to solve problems using knowledge. In other words, AI is a man-made means of representing and processing knowledge.

Knowledge is more than a collection of numerical values. It is a combination of information and experience. Information includes data (i.e., raw input), while experience includes associations, beliefs, and values about this information. Conventional programming techniques rely on extensive manipulation of numerical quantities. AI programming, on the other hand, relies on the manipulation of knowledge. Often, though not exclusively, this knowledge is in the form of symbols, such as words.

A problem that is understood well enough that relationships between facts can be completely and rigidly specified is a candidate for traditional computational techniques. In this case, an algorithm is developed which invariably produces the appropriate response. When a problem is not understood completely, experience is a useful tool. For example, dim lights and a weak radio lead us through experience to believe that a car's battery may be low. However, many other alternatives are possible, such as a film of dirt covering the headlights and a broken antenna.

In certain problem domains, such as the diagnosis of a complex system, the proper application of experiential knowledge can provide significant problem-solving power. The types of problems that an AI approach fits best are precisely those which are poorly understood, inexact, and highly complex. Engine diagnosis, investment decisions, even recognition of a face are among the hundreds of common human tasks that fall into this category.

Because of its reliance on knowledge gained through experience, much of the computation in AI is symbolic in nature. The concepts captured in knowledge may have no precise numerical interpretation, but the symbols which represent them can be combined in specified ways through chains of reasoning. If the lights are dim and the radio is weak, then there is a 92% chance that the battery needs charging. This is an example of experiential knowledge. When a number of these rules interact together, a more exact diagnosis may be made (e.g., faulty wiring from the battery).

For complex systems, the total number of possible difficulties is very great due to a condition known as combinatorial explosion (see Section 2.5). Experiential knowledge is

knowledge about the most likely difficulties, which spares the user the necessity of checking all possible combinations (often clearly an impossible task).

Humans excel at solving certain types of complex problems. An AI approach attempts to simulate the human mechanism of intelligence by codifying the prototypical human responses to such problems (i.e., problems characterized by complexities of immense input or solution sets, uncertainty in the information, or incomplete data).

Using AI methods, people are beginning to address problems that could not be examined in the past because the solutions required manipulation of too much data; for example, development of a vision system for a semi-autonomous robot. There is no possibility of specifying a complete catalogue of all visual scenes for a robot. Not enough memory exists. Changes in the position of the object being viewed, orientation of the robot, lighting variations, and several other factors would alter the scene and therefore require another image to be stored in memory. The number of possible images is essentially infinite, so that using the catalogue approach to visual scene analysis is only possible in the simplest and least useful of cases. AI has been used to reduce complexity by selecting key features that do not change and then drawing inferences about the features. Understanding of natural language and speech analysis are two more examples of problems which are untractably complex due to the sheer quantity of information to be analyzed. It is clear that to contend with such complexity, the simplifications and knowledge which AI utilizes are necessary.

However, extensive data sets are just one type of complexity suited to treatment with AI methods. Another type involves informational uncertainty. Data values may have a degree of uncertainty due to insufficient information, lack of understanding, randomness, or unreliability. In an environment where data may be incomplete or uncertain, natural intelligence performs admirably. Every day, human beings make successful business decisions in an environment that is constantly changing and difficult to predict. Relatively simple decisions, such as how many units of a certain commodity a retail store should purchase, require a combination of uncertain forecasts and unreliable data. Still, humans decision-makers are able to make such choices, often very well. AI methods attempt to mimic human capabilities in these areas by providing a means to process uncertain or partial information.

Another important characteristic of the type of complexity which marks the ideal problem for AI is that part of the information about the problem is heuristic--useful but imprecise. Heuristic knowledge is also known as "rule of thumb" knowledge. Heuristic knowledge is experiential knowledge - knowledge that is correct most of the time, but not all of the time.

In summary, AI approaches are concerned with the same types of problems at which human intelligence excels, i.e., problems characterized by large and frequently complex input and solution spaces, by uncertainty and/or incomplete information, and by a heuristic nature of some or much of the knowledge. These are the types of problems least suited to solution by traditional computational techniques. AI, therefore, may be regarded as complementary to computing as a tool for solving problems.

1.2 ADVANTAGES OF ARTIFICIAL INTELLIGENCE

AI techniques are most commonly used to partially replicate human abilities and the most successful AI applications have tended to augment human performance rather than replace it. They may be implemented to release workers from dangerous or tedious occupations, to mitigate the effects of shortages of expertise forecasted for a variety of occupations, or to increase the quality of available expertise by improving the speed and reliability of expert decisions. The application of AI may also provide cost-effective alternatives for many industrial chores, thereby becoming a competitive necessity. Knowledge-based systems, and to a lesser extent, other AI technologies have been successfully employed in a variety of settings, performing a variety of tasks, both inside and outside government.

The following sections describe several of the more promising AI technologies, general applications for AI, and specific potential Depot applications. Many of the documents listed in the references (Section 8) provided further information on specific technologies and applications.

2.0 METHODS IN ARTIFICIAL INTELLIGENCE

The basic problem-solving methods used in AI are reviewed in this chapter. Over the last thirty years, a number of different approaches have been developed to address a wide range of problems. Each method can be broadly described as a way of organizing, representing, and applying knowledge. The most economically promising of these methods are discussed in the following subsections: rule-based expert systems, model-based reasoning systems, blackboard systems, neural networks, and adaptive search algorithms (in particular, genetic algorithms).

In terms of application, these methods need not be exclusive of one another and, indeed, mixing them may provide the best solutions to a given problem. For example, computer vision systems may include neural networks for low-level feature extraction in tandem with a rule-based system to reason about higher-level structural cues.

2.1 RULE-BASED EXPERT SYSTEMS

A human expert acquires knowledge and skills that enable competent, effective, and innovative problem-solving. There are three features that characterize an expert: specialization, accuracy, and efficiency. An expert can be defined as a person who has amassed a large quantity of specialized knowledge gained from experience in solving problems of a certain type. The expert is also relatively accurate in terms of solutions; he or she will be right often enough to be useful and reliable. (Each domain has generally accepted rates of error. A meteorologist, for instance, may be allowed more of an error margin than an operator of a nuclear power plant.) Given unlimited time, novices may perform as well as experts on some tasks, but an expert is able to arrive at conclusions with relative efficiency. These characteristics--specialization, accuracy, and efficiency--are the same qualities simulated in AI expert systems.

In this context, we are concerned primarily with functional expertise, such as that exhibited by an automobile mechanic or a medical doctor, rather than expertise in broad and poorly defined areas where knowledge and performance are not easily measured, such as international relations or history.

Expert systems, sometimes known as knowledge-based systems, are computer programs that attempt to simulate the performance of an expert on some well-defined, narrowly scoped task. These programs, which represent the first commercial success of the AI industry, are designed to have many of the capabilities of a human expert, such as specialization, accuracy, and efficiency. Though expert systems may not exceed the performance of the best of the experts, they often do perform better than the average expert and therefore serve to increase the overall skill level to a significant degree. Expert systems also interact with users in ways a human expert might, seeking additional information by asking questions and explaining conclusions in language appropriate to the user.

Expert systems differ from traditional computing in several important ways. First, the data used by expert systems is in symbolic form, as well as numeric. Symbols are non-numeric values used to represent concepts; for humans, these are chiefly words. Second, experts systems have a different control structure. Traditional programs combine control with knowledge in the form of algorithms; expert systems, on the other hand, have a separate control structure that knows how to use a body of problem specific knowledge expressed in an appropriate form such as if-then rules. Finally and most importantly, expert systems, unlike traditional computing systems, are constructed to behave like experts, linking together groups of information, forming a chain of reasoning, and describing this knowledge to the user of the system.

2.1.1 Composition of Expert Systems

Expert systems have three primary components: the knowledge base, the inference engine, and the human-computer interface. The knowledge base is the store of information; the inference engine defines ways the information in the knowledge base can be combined; and the human interface is a way to communicate between the computer and the user.

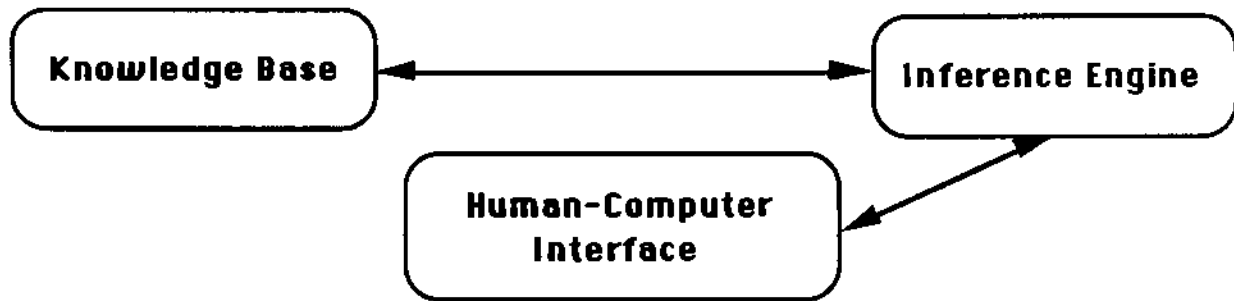


Figure 1. Expert System Components

2.1.2 The Knowledge Base

The knowledge base may be represented in many forms, including frames, logical expressions, semantic nets, and rules. Each of these terms refers to a distinct way of storing pieces of information in symbolic form so that they can be efficiently combined with other pieces of information. Knowledge representation by rules has traditionally predominates and is the focus of this section. Rule-based knowledge systems are composed of both facts and rules.

Facts are objective statements describing a known reality. Rules (heuristics) describe the relations between facts. For instance, a vision system might capture the forms of three objects on a conveyor belt. A knowledge-based expert system might be used to extract features from the images (e.g., size, shape, centroid) and then combine this information through rules (heuristics) to conclude that one object was a box, one a wrench, and one a bomb casing. The rules established in an expert system are used to integrate and relate facts and, ultimately, to draw conclusions.

The facts and rules of the knowledge base embody the expertise. Much of this knowledge is gained from an expert's experience specific to a single domain. This experience determines the rules (heuristics) relevant to the knowledge base. To develop this knowledge base, facts and heuristics are gathered from experts who serve as the system model. How is this done? For small programs and simple problems, the expert may simply purchase a skeleton of an expert system, an expert system minus any domain specific knowledge, and transfer his knowledge to the shell. But for larger problems, a knowledge engineer may be necessary. A knowledge engineer, trained in the process of extracting relevant knowledge

through interview techniques, logically structures this knowledge into a knowledge base. Whichever the case, construction of the knowledge base is often the most difficult and time-consuming chore in creating an expert system.

2.1.3 The Inference Engine

The second major component of an expert system architecture is the inference engine, which controls the reasoning process that infers new knowledge from the existing knowledge base. There is more than one way to create new information by inference, including forward chaining and backward chaining, two methods commonly used in rule-based expert systems.

In forward chaining, the discovery proceeds from known facts to new facts. The illustration of object recognition described above is an example of forward chaining: the facts are used to form a conclusion. Backward chaining involves hypothesizing a particular conclusion and then "reasoning backwards" to ascertain whether facts could support this hypothesis. Forward chaining and backward chaining may be combined in a rule-based system. For example, forward chaining could be used if the nature of the conclusions to be reached are indefinite (thereby making formulation of an hypothesis difficult); backward chaining could be used when the conceivable number of possible inferences made from basic facts is very large, and evaluation of each choice would therefore be time consuming. A control strategy inherent in the expert system must not only control the direction of the reasoning process, but must also provide the inception point and resolve conflicts that may occur if two distinct reasoning paths could possibly be used at the same time.

2.1.4 The Human-Computer Interface

The last component of a rule-based expert system to be discussed is the human-computer interface, which provides three primary functions. First, it serves as a port for knowledge acquisition--from an expert (for domain knowledge) or from a user (for environmental knowledge). Second, it provides explanations to the user for particular choices the expert system has made. Third, it provides utilities for input, output, and online help. This component is usually sophisticated, because a module that acquires and explains knowledge and reasoning to a user is critical to the success of a knowledge-based system.

2.1.5 Applications of Expert Systems

Although expert systems have been applied successfully on many occasions and represent a technology that has matured in recent years, there are still several considerations to be made before an expert system is constructed. The problem to be solved should be well defined, as discussed before. Historians will probably not be replaced by expert systems in the near future. On the other hand, problems that are completely described by mathematical analysis are not appropriate for the domain of expert systems. In addition, some problems are either too simple or too complex to justify the production of an expert system. Even if the problem matches well with current capabilities, it must be determined whether solving the problem with an expert system is economically justifiable.

Expert systems do have many proven advantages. They are consistent, accurate, fast, reliable, and accessible relative to their human counterparts. These qualities may translate directly into cost savings when appropriately applied.

2.2 MODEL-BASED REASONING SYSTEMS

AI methods that reason about the physical system itself and do not simply capture the rules identified by human experts have a wider range of operation. The most detailed form of model based reasoning is an algorithmic (analytic) model of the real system. In AI applications, the model is typically accurate in representing the qualitative behavior of the system and only addresses quantitative behavior of the system to a limited extent. The objective is to create a model explicit enough to provide predictions about the important behaviors of the actual system, but simple enough not to bog developers down in the complexity of complete numerical models.

Model-based reasoning necessitates the construction of a model that behaves, at some level of abstraction, the same as the real system. While modeling can include the modelling of natural systems, the scope of this overview is limited to models of man-made systems, such as engines, pumps, or assembly lines. In this context, models are built to identify complex sequences of behavior and to ensure that the system functions according to its purpose. The components and the interactions among components are identified so a sensed error in the system can be traced to its initial malfunction. Model-based reasoning, therefore, involves

drawing conclusions about the condition or likely future condition of a physical system. The specific goal of the reasoning process, in our context, is to develop the knowledge necessary to enable the system to function according to its intended purpose.

The process of reasoning about physical systems has several components. The first component is a thorough understanding and description of the system. To model a car engine, knowledge and understanding of the engine components and the interrelationships among the components is required. For example, all engine parts, their functions, and their expected interactions must be identified and described. This understanding has several possible levels, from a specified set of deterministic and stochastic equations to common-sense or functional understanding.

The second component in the reasoning process is a determination of the current status of the system. Sensors provide information about key components that describe the system state (i.e., "is it working or not working"). From a functional standpoint, sensors would determine, for example, "the wiper blades are not working."

The third component is the ability to apply an underlying knowledge of the operation of the physical system, given current conditions, to determine some functional classification (i.e., normal, critical, etc.) or probable future state. In the above example, sensors determined that the wiper blades are not functioning. Given hot and dry weather conditions, the model would classify the malfunction as noncritical; if weather conditions changed to rainy, however, the model would reclassify the function of wiper blades as critical.

The final component is the ability to determine actions to be taken to rectify abnormal operation or to maintain normal operating conditions. In the case with the wiper blades, alternative solutions might be offered, such as the following:

- "Check wiper blades. Replace if cracked."
- "Check wiper blades. Are they in normal position?"
- "Check wiring to wiper blades. If wiring is OK, check battery."

Through this iterative reasoning process, the initial malfunction can be found and repaired. This final component may or may not be included in a model that reasons about physical systems, but it is implied if the intention is to troubleshoot for system errors.

These four components (understanding the system, determining the system state, classifying the system status, and determining any necessary actions) interact to produce a model-based reasoning system. The characteristics common to an alternative reasoning approach about physical systems, known as qualitative physics, are discussed below. This approach was selected for discussion because of its increasing popularity and apparent utility.

2.2.1 Qualitative Physics

Qualitative physics is also known as naive physics or common-sense physics. Qualitative physics tries to define a system's mechanical operations without relying on a complex mathematical description. It is a simplification that seeks to retain in the model only those abstract components essential to the functioning mechanism. In a sense, qualitative physics is an attempt to capture the human approach to reasoning about the physical world. Humans do not consciously diagnose their environment in terms of thousands of nonlinear differential equations. Nonetheless, humans operate very well in the real world. Qualitative physics has sought to identify the basic characteristics of the knowledge that humans use when reasoning about their environment and apply those characteristics to computers. As a result, AI can gain a relative advantage in the rapid, accurate, and robust anticipation of physical behavior.

Several characteristics are common to AI systems that are based on qualitative reasoning. The integration of these features describes the basic methods used in qualitative reasoning.

The first idea common to qualitative reasoning approaches is the system. A system is a set of interacting components. Each distinct component must be defined in terms of its characteristics and its behavior--a car engine, for example. The system must then be described in terms of the interactions between the components, essentially answering the question of how a distributor rotor affects a spark plug. These relationships must be defined for every component. The overall behavior of the system then arises from the overall interaction of component behaviors. The description of behaviors and interactions are mathematical simplifications.

These mathematical simplifications give rise to the second common feature, causality. Causality is modeled by the interactions among components described above. The causal relationships are important for defining overall system behavior. For instance, a spark plug

failure may be causally related to failure of a battery. The discovery of a failed battery encourages the verification that the spark plug is really working. Therefore, causality implies that system behavior is predictable with a deeper understanding than simply the modeling of component behavior.

Another common feature is functionality. Again, this concept separates qualitative reasoning from other forms of modeling. From a purely mathematical perspective, components do not possess a purpose, only a structure and other stochastic and deterministic features of operation. In qualitative physics, the mechanism is viewed from the human perspective. A functional requirement or purpose is imparted to the components and to the system. The reasoning process can then be augmented by the requirement that each part performs a task, and this supports reasoning about the overall system behavior.

Finally, qualitative reasoning systems rely on the quantization of information. (chunking continuous data into discrete pieces) to simplify the description. A state is defined as the condition of a system at a particular time and is measured by the value of specific parameters. Behavior is described as a sequence of physical states. States of a system are represented by discrete values, not continuous ones. For example, if the rate of current flow into a circuit element is designated by the symbol di/dt (a time derivative), this symbol may be limited to either + or -, thereby compressing the infinite range of real numbers to two values. These values may be physically significant because of some behavioral change that takes place at this point, while the actual numerical value is unimportant to system behavior.

The features described above provide an introduction to the characteristics of qualitative reasoning systems. These systems, like others that reason about physical mechanisms, can be applied to a range of activity, from simulation and verification to diagnosis and analysis. They seem to offer a solution to the rule-based systems that operate without "deep level" knowledge and, therefore, are more easily confounded. It should be pointed out that the techniques employed by qualitative physics could be applied to many types of systems.

2.3 BLACKBOARD SYSTEMS

A classical approach to problem solving is the process of breaking down a large problem into a group of less complex problems that are more easily identified and defined and

that fit together to form a solution. For example, the operation of a technical conference can be broken down into smaller tasks such as registration, technical abstract review and selection, hotel accommodations, meals and reservations, technical session planning, poster sessions, guest activities, finances, media relations, and other tasks necessary to make the conference run smoothly. The blackboard approach to solving problems draws from this paradigm. A large problem is separated into subtasks using a variety of reasoning strategies and knowledge sources capable of acting separately to achieve a single goal. These tasks are usually at different levels of abstraction and information flows up or down between the levels to assist the problem solving.

Blackboard systems effectively use division of labor and organizational plans to create a powerful and flexible problem-solving strategy. In blackboard systems, division of labor is embodied as a collection of functionally independent knowledge pools. That is, each knowledge pool represents a subtask of information. The organizational plan gathers appropriate knowledge pools together to reason towards a solution. These two components can be envisioned as a group of experts (each represented by a knowledge pool) who know how and when to contribute (as specified by the organizational plan) to a large problem requiring the full range of their expertise.

2.3.1 Composition

The broad scope of activity that characterizes blackboard systems arises from the interaction of three basic components: 1) the group of knowledge sources, 2) the blackboard, and 3) the control structure. Each component contributes synergistically to a solution. These components are described further below.

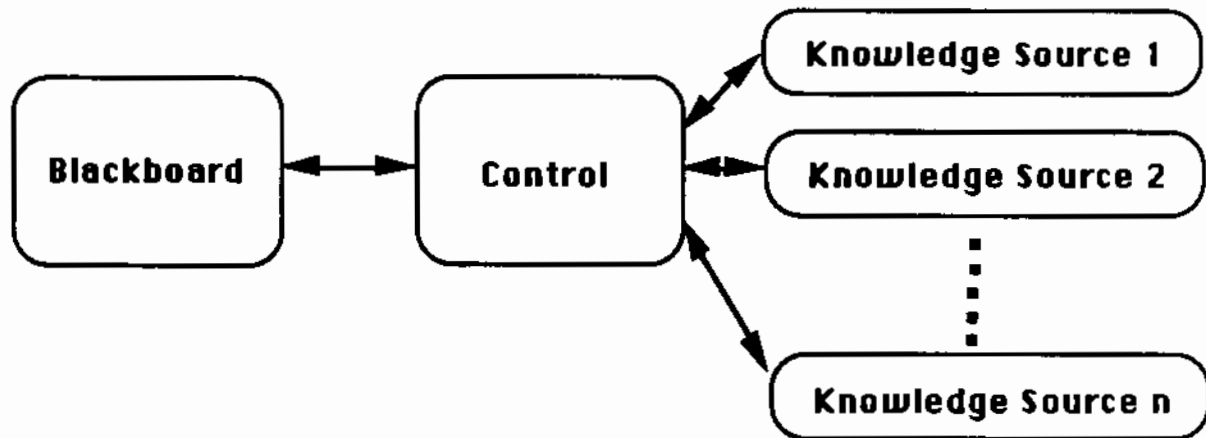


Figure 2. Blackboard System Structure

Knowledge sources, the first component of a blackboard system, can be compared to human experts. Knowledge sources, or pools, contain adequate knowledge about a specific domain, but unlike human experts, do not have even rudimentary understanding of any other domain. This modularity is a key feature of blackboard systems. Knowledge sources are allowed to be ignorant of other domains because each pool of knowledge determines when to apply its expertise to alter the contents of the solution. Knowledge sources are represented as groups of rules or logical assertions which specify a problem state to locate and an action to initiate.

A second component of a blackboard system is the blackboard itself. The blackboard is the global structure that provides the only (indirect) interaction between knowledge sources. It functions as a central repository of solution information. The nature of the solution process has been characterized as incremental, iterative, and opportunistic (Nii 1986). Solutions are incremental in that knowledge sources one at a time alter the contents of the solution represented on the blackboard. This is equivalent to solving each subproblem on the way to the overall solution. Through iterative and incremental transformations, which change the initial state, a solution can be achieved. The solution process is also opportunistic because knowledge sources "grab" control of the blackboard when given opportunities allowed within the organizational framework.

The blackboard represents information as objects. If a solution is thought of as a path composed of a discrete number of steps, then each object is associated with a single point in

that path. This can be an initial point, an incomplete or intermediate solution, or a final goal state. Additionally, this information is hierarchical, so that the solution of one subproblem serves as an initial state, or input, to a higher level of problem design. This implies that there may be multiple boards in the blackboard model, corresponding to the multiple subproblems that are to be solved along the way. Objects may have relationships with other objects, either in the same or different levels of the hierarchy, and these are identified by a set of link labels.

The third and final component is the control structure, which often takes the form of control modules. The details of the control strategy are dependent upon both the task and the implementation, but its constant purpose is specification of the problem-solving behavior of the system as a whole. The effectors of the control strategy select the behavior of the system by their choice of either the blackboard object on which to operate, the particular knowledge source that is to perform the operation, or both.

2.3.2 Problem-Solving Activity

Problem solving in blackboard systems is an iterative process. Initially, a knowledge source makes some change in the data structure of the blackboard. For illustrative purposes, the knowledge sources can be thought of as detectors in a visual system. Some of these detectors respond to simple shapes, such as line segments, curves, or corners. More complicated detectors respond to combinations of primitives, such as boxes, circles, or cones. Even more complex detectors are designed to detect combinations of these boxes, circles, and cones. Therefore, there are three layers in the hierarchy. In analysis of a scene, the most primitive shape detectors will likely take command of the blackboard first, contributing their knowledge about primitive shapes to the global data structure. At each step, the different knowledge sources contribute their opinions about the strength of their potential contributions, and at each step the control structure assists in specifying some combination of shapes and knowledge source to be the object/operator pair. This procedure continues until some predetermined condition is met, presumably the successful identification of the visual scene. In this way, the pieces of the puzzle are put together, beginning at the bottom and working upward to higher levels of analysis.

2.3.3 Advantages

Blackboard systems are best suited to problems that are complex or ill defined (Nii 1986). Complex problems are broken down into detailed processes and simple subsystems. When the subsystems have been divided to a tractable level of complexity, the smaller pieces are analyzed and fit together to form the larger solution. Ill-defined problems are those for which the sequence of transformations leading from initial state to goal state are difficult to identify. There are many characteristics that characterize a problem as either ill-defined or complex. Problems fitting these characterizations may lend themselves to solution by blackboard systems.

2.4 NEURAL NETWORKS

Humans are astonishingly efficient at many tasks that traditional digital computers perform very poorly. Image and speech analysis are two examples. In recognition of this, researchers have studied the structure of the human brain for clues to its operation. The field of artificial neural systems (ANS) was inspired by biological discoveries within the human nervous system. Due to theoretical, algorithmic, and hardware developments, neural networks are currently being applied to several types of problems.

Basically, ANS, also known as connectionist models, parallel distributed models, or neural networks, are collections of simple but highly interconnected processing units.

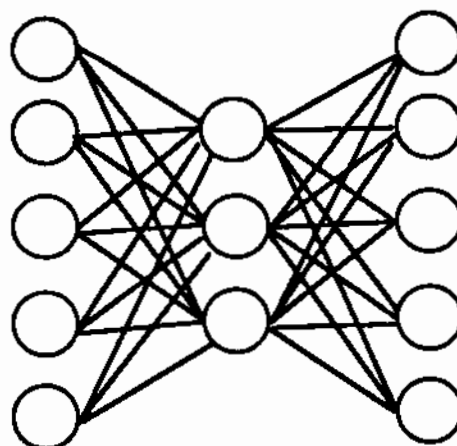


Figure 3. Neural Network

The operation is therefore both parallel, in that all units are capable of receiving and outputting information at the same time, and distributed, in that each unit is connected to many other units.

What neural networks do is easier to describe than how they do it. Neural networks receive many signals from an environment in parallel. Using rules that differ from algorithm to algorithm, they modify their connection weights to produce a desired output. This is learning in a neural network. These networks are capable of storing many associations in the same collection of units. The associations may be images of faces with names, sounds with words, or positional signals onboard an autonomous vehicle with the proper control signal. The key is that connectionist systems are capable of making any mapping between input and output pairs and that multiple associations can be learned and simultaneously stored in the same network.

Each of the simple units receives input, processes the input, and then transmits output. Generally, each unit has weighted connections to other units. The input to a selected unit is the weighted sum of the outputs of the other units connected to it. This sum is typically passed through a nonlinearity (one of several types), and the result is a unit output. In this way, units pass messages in the form of activation values. The system operates by accepting input, then passing activations until it "settles" to a solution. The nature of this solution depends upon the ANS implemented.

A variety of models exist, each with its own algorithmic intent and physical structure. Rather than a comprehensive discussion, we will provide a brief list of features that characterize their diversity: 1) the physical structure of the network, 2) data types, 3) the activation rules for determining how the input received is converted to an output signal, 4) the learning rules for adaptation, and 5) the computational rules.

2.4.1 Physical Structure of the Network

The physical structure of the network refers to topological characteristics and may include design considerations, such as the number of processing units and the number and order of connections. For example, some networks are arranged in strict layers. Connections are allowed only between units in impinging layers. Other networks have each unit connected with every other unit in the network. Some of these characteristics can be network parameters

as well. The number of nodes, for instance, may depend on the problem size and will change from problem to problem. Topological characteristics, interdependently with other system features, determine many important functional properties of the system. These physical characteristics are one feature of neural network diversity in AI systems.

2.4.2 Data Types

Network models are also distinguished by the data ranges and data types that are allowed. Some models limit activation values, weight values, or both to small positive ranges. Some incorporate negative (inhibitory) values, as well. ANS can also be separated into groups that accept analog values and discrete values.

2.4.3 Activation Rates

The activation rules, or processing unit (node), characteristics provide other distinguishing features. Most units are sum and fire accumulators (that is to say that they simply add all the input signal strengths together and send off a an output signal if that sum is greater than a specified value). Neurons however frequently differ in the choice of output function. The output is some function of the input and is usually nonlinear. These nonlinearities have a variety of forms (i.e., a threshold). Output nonlinearity is important because it imparts the capability to create essentially any nonlinear mapping between input and output. This frees the network model to make any association required.

2.4.4 Learning Rules

One of the most intriguing characteristics of neural networks is the ability to learn. The network's learning process creates correspondences between input and output patterns. If the network learns to complete a pattern from incomplete input, it is termed autoassociative. It can recognize and reproduce a pattern even if the patterns do not exactly match. If, on the other hand, the network creates a correspondence between two distinct patterns, it is heterassociative, or simply associative. Other networks map input in pattern classes.

There can be two learning types, supervised and unsupervised. Supervised learning requires a teacher to supply the correct response during the training process. Unsupervised

learning requires no teacher. During supervised learning, the network weights its answers until it independently and correctly responds. During unsupervised learning, the network makes associations based on the underlying algorithm.

To learn, the network is trained by a training set. A representative sample of input patterns is repeatedly entered to the first layer of processing units. Over time, the network learns to respond to these input patterns in a prescribed manner. The error-back propagation model forces the output units to answer correctly (the desired output pattern). It iteratively alters connection weights to units that have a large discrepancy between the actual and the desired outputs. This is a form of supervised learning.

Networks can also learn without supervision. Learning rules, which form the basis for adaptive learning in a neural system, differ substantially among algorithms. Many are based on the Hebbian rule that strengthen a link weight in proportion to the correlated activities between the sending and receiving nodes. The memory capacity and the speed with which the system learns is closely related to the learning rules used. An increase in the complexity of the learning rules will slow the learning speed.

2.4.5 Computational Rules

The final feature of model diversity is the rule set that defines computation. These rules describe the reasoning process to achieve an outcome. Connectionist systems often have two distinct operational phases: learning and computation. Learning is described above as the adaptive process of forming new associations while retaining the old ones. Computation follows learning. It is the phase of network activity whereby a network produces a proper response to a given input. In other words, computation is the dynamic course of activated message passing as arranged to form a solution. The precise nature of this process is different from model to model.

ANS are used in several application areas. Neural systems are frequently used in pattern classification, especially when large data sets are required. Optimization problems are another well-suited application. Optimizations are often composed of many simultaneous mutual constraints. These constraints map very well to the neural network architecture. Robotic movement is yet another application area where the complexity of the task matches well with connectionist capabilities. The multidimensional transfer function between a robot's

sensory inputs and motor outputs can be mapped in a neural network. Sensor fusion and adaptive signal processing are related functions that have also been performed by neural networks.

Connectionist models (neural networks) have several advantages over other computational models. The foremost advantage is parallelism. The speed of a parallel network is essential for many large processing chores. Parallel processing is independent of problem size. A second advantage is a high fault tolerance possessed by connectionist models. Distributed computations imply that the failure of a small number of processing units may not severely affect performance. Finally, neural networks are adaptable. This ability offers improvement of performance over time. For systems responding to new situations (i.e., control applications), the capability to learn online is valuable. Because of these advantages, ANS will be used more and more frequently.

2.4.6 Future Applications

Neural network technology is beginning to make the transition from research to application. Several companies have begun to apply it to areas ranging from the classification of seismic signatures to speech recognition to the guidance of autonomous land vehicles. A major impetus to the variety of new applications will be the development and implementation of new hardware. Currently, neural networks are run as simulation on single-processor digital computers, or on parallel computers, or on special purpose digital array processor. None of these hardware implementations captures the potential throughput of neural networks. Chips are currently being designed and tested to implement neural networks directly. When chip technology is able to construct thousands or millions of interconnections on a single chip, the full computing power of neural networks will become available. The usefulness of this technology should increase correspondingly.

2.5 ADAPTIVE SEARCH ALGORITHMS

Many problems in AI do not have a clearly defined solution strategy. Perhaps a deterministic solution strategy does not exist. In these cases, knowledge of a problem may consist only of certain initial conditions and constraints that describe the nature of the problem

parameters and variables and a final goal state to be reached. As a result, a search for the proper combinations of variables and parameters may be the best way to arrive at an optimal solution.

Search problems are fairly common. A manager trying to decide where, when, and how to spend limited funds is searching for the optimal combination of projects and budgets to maximize profits. Indeed, many optimization problems are found in resource allocation or in optimal control. These search problems involve an initial state, a goal state, and operators that transform intermediate states to new states. The operators change a given state into a new state according to the nature of the search scheme. Then the new state is compared to the goal state. The goal state may be the unique and "correct" combination of variable and parameter values, or it may be the optimal combination of these values as measured by some established criterion.

The primary difficulty with searching through combinations of values is that many domains are susceptible to combinatorial explosion. This implies that the number of possible solutions is too large to search in a reasonable period of time. Therefore, heuristics are developed to streamline the search and to direct it to variable combinations more likely to include the goal state. There are a number of heuristic searches commonly used by AI practitioners to narrow the number of feasible states to be searched. Among these algorithms are the A* algorithms, hill-climbing algorithms, and genetic algorithms.

2.6 GENETIC ALGORITHMS

Genetic algorithms, which perform an optimization, are based roughly on the principles of genetics as applied to evolution. Essentially, the forces of evolution affect survival among populations of plants and animals. Evolutionary success, measured by survival, depends on the adaptability of individuals within a species to their environment. Genetics, then, is a study of the adaption mechanism in living organisms. More specifically, geneticists study the selection of particular genetic units, called genes.

Genetic algorithms cast optimization problems in the framework of a survival struggle: a population of solutions is modified from generation to generation using principles derived from genetics. The best solutions of one generation contribute heavily to the next generation's

solutions. Applications of genetic algorithms are very promising for a number of problems particularly difficult for other search algorithms. Genetic algorithms are ideally suited to problems with a very large number of potential solutions with very little a priori knowledge available. Though genetic algorithms vary widely in form and application, there are three basic steps common to all:

1. construction of an initial population
2. development of a performance measure for each member of the population
3. creation of new populations using genetics principles.

Each step requires some explanation.

The first step, construction of an initial population, translates a given problem into an evolutionary paradigm. A tentative solution state is formed as a vector of state variables that characterize it. These are the parameters of the problem whose optimal combination is the goal of the search. Generally, research indicates it is advantageous, due to the nature of the genetic recombination, to represent these state variables as bit strings for manipulation. For example, if a given variable can assume eight possible values, then it should be represented as three bits in the state variable vector. The length of a state vector for a given problem will then depend on two things: the number of parameters to be considered and the number values each parameter can assume. In the absence of solution structure information, a population of random bit vectors can be generated to represent the initial population. The generation of the initial population is the first part of the genetic paradigm.

After an initial population is created, a performance measure must be devised to evaluate the soundness of each state. This is key to the operation of the algorithm. For population genetics, the measure reflects the capacity of the individual to survive and find a suitable mate. The nature of this metric clearly depends on the nature of the problem. For example, if combinations of workers with different skill levels can accomplish a particular task, the metric may reflect the total production time for each combination of employees. When determining the evaluation criteria, the score for an individual will reflect the fitness of that individual. Thus, the percentage of contribution to the next generation is determined. Not only

do the fittest survive, they also have a greater influence in the structure of the offspring. Those less fit are removed from the population. The creation and weights of performance measures are the second structure to the genetic paradigm.

The final step of genetic algorithms is determining the composition of new generations. Genetic operators are the agents of evolution. They act on each member of the current population to produce the next. These operators apply two basic principles of population genetics, crossover and mutation.

Crossover is a phenomena of sexual reproduction. Two parents mix traits to pass to offspring in varying combinations and amounts. Two state vectors analogously share their traits with a variable number of next generation state vectors. The mixture has a degree of randomness to assure variability. Crossover is the primary means of moving towards optimal solutions in genetic algorithms, preserving the surviving small building blocks and testing new combinations.

Mutation is the second principle of population genetics. In the terms of optimization, mutation is the feature used to escape spurious local minima. In the practice of genetic algorithms, mutation is the flipping of one bit for each predefined interval of total bits. For instance, an average of 1 in 1000 bits might be reversed. Crossover and mutation are demonstrated in the following diagram. Using genetic operators, such as crossover and mutation, the search builds new populations of solutions as it progresses toward an optimal solution.

Genetic algorithms can be classified as an AI method because of its adaptive ability. Like all forms of AI, genetic algorithms use experience to direct a solution. Similar to neural networks, and unlike most other methods, it adapts due to its own experience instead of utilizing the experience of a human in the form of heuristic rules. Also like neural networks, it performs adaption in parallel, thereby efficiently recording environmental information by simultaneously evaluating the worthiness of each offspring.

Genetic algorithms perform across a wide range of applications. They have been used to specify parameters in multicomponent systems, to restore images in medical imaging, to

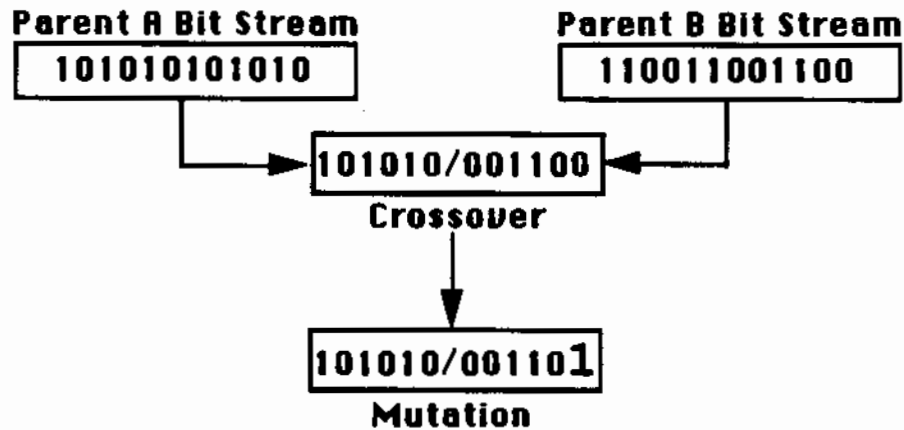


Figure 4. Operation of a Genetic Algorithm

create lisp code for predefined algorithms, and to control the parameters of other genetic algorithms. They are generally better at using available information than other heuristic search mechanisms and require no knowledge of the problem domain other than the knowledge of the parameters and a method for evaluating their combinations. They also have the advantage of parallelism; each candidate can be evaluated in parallel to determine fitness.

Research into genetic algorithms and their applications has recently increased, as in many areas of AI. The expectation is that with more interest will come more uses for this promising class of search algorithms.

3.0 APPLICATIONS IN ARTIFICIAL INTELLIGENCE

This chapter describes the types of problems that may be solved with AI. Selecting a method, such as using a rule-based system, is made with a specific problem in mind. In the following subsections, several general categories of applications are described, which represent the types of problems AI systems are designed to solve: control, design, diagnosis, training, monitoring, planning, prediction, repair, and perception and analysis.

3.1 CONTROL

Control is essential to the proper operation of many mechanisms. The human body has an immensely sophisticated control system, which is demonstrated each time a human hand reaches for a cup of coffee or a computer keyboard. The challenge posed for control is guidance; the behavior of a system is urged toward the proper value. A missile traveling through space can be used as an example of a control application. The missile includes sensors that, at regular intervals, send its location to a control module. The control module measures the actual trajectory against the desired trajectory and adjusts the steering mechanism to reduce the difference between the two. The cycle of sensing, comparing, and correcting continues until the target is reached.

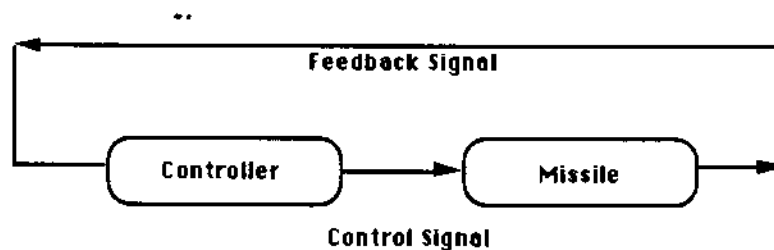


Figure 5. Control Loop

Guiding the missile may be specified for all environmental conditions. Modification of the pitch, roll, yaw, and thrust all affect the missile heading. In this case, the system (missile) and factors affecting the system (pitch, etc.) are clear and well defined.

However with some problems, the relationship between a system's behavior and the factors affecting the system are not so well understood or are so complicated that a solution is

effectively beyond the time constraints of the application. Such problems are well suited for AI, because they imply a complexity beyond a mathematical tractability. In other words, heuristics may be introduced.

Heuristics are useful for reasoning through unlikely or difficult control situations--to handle unexpected or drastic fluctuations in the environment, to simplify immense computations that would otherwise be necessary, or to accommodate a goal (target) modification.

A trajectory can be associated with the behavior of a system through time. It is most simply a sequence of variable values. The first state of the trajectory is the system's initial condition. Using a central heating system as an example, the initial condition is the starting temperature of the room. If the thermostat is turned up, the goal state is the temperature to which the thermostat was set. This would be the endpoint of the trajectory. The temperature values of the system between these two points comprise its trajectory or temporal behavior. The object of the controller is to regularly adjust the system to achieve or maintain a desired path. The controller or thermostat continually monitors the room temperature and turns the heater on or off, depending on fluctuations in the environment.

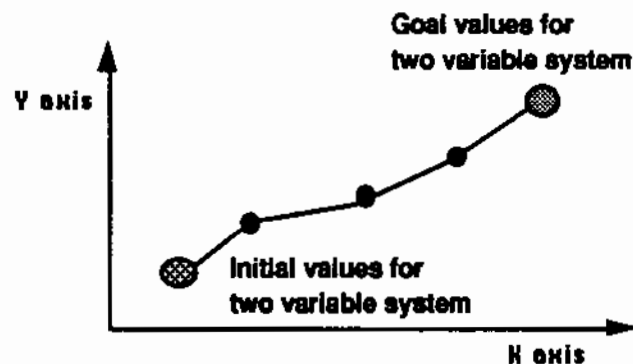


Figure 6. Initial Values to Goal Values

Complexity in a control problem is introduced when 1) many factors are necessary to control system behavior, 2) uncertainty is characteristic of the available knowledge, or 3) unanticipated variations in the environment or goal occur. In very complex systems, control may require the application of rules that seem to fit the current conditions. In some systems, a

functional description that incorporates the relationships among all the variables in the system may not exist. However, certain relationships generally hold for certain variable values. Heuristics of this sort could be added to the control system to handle these situations. A controller may also be able to adapt to, or learn of, new situations and relationships.

Artificially intelligent control systems may be effectively employed where important variables, the so-called "system state variables," can be monitored and their quantities changed in response to a system measurement of fitness. Applications seem most readily apparent in developing flexible robotic systems capable of performing a wide range of tasks that may differ significantly in required behavior.

3.2 DESIGN

The purpose of a good design is optimization. For example, one of the purposes of commercial packaging design is to maximize consumer attention. A good design for a sports car body minimizes the car's drag coefficient. Optimization problems are characterized by goals, the factors that are to be maximized or minimized, and constraints, which are the resource limitations. AI is well suited to optimizing designs with consideration for resource constraints.

In essence, designing a system involves a trial-and-error discovery of what works well. First, materials are arranged in multiple configurations. Then the configurations are analyzed and, based on estimated performance, the best design is selected. AI systems efficiently analyze designs iteratively (i.e., testing and retesting until an optimal design is found).

One example of AI application is in helping design circuit boards. Circuit boards need to perform certain functions very quickly and efficiently. The design of a circuit board must also consider space limitations and production constraints. For instance, at Hewlett-Packard, a system evolved whereby a circuit board design would be submitted to a team of experts, who would examine the design for workability. (Does it perform the expected functions? Does it fit well on the circuit board? Can it easily be manufactured?) This design and review process would take 4 to 6 months. With the implementation of an AI design system, this design

optimization process was reduced to two weeks. Not only did the AI system speed up the process, it also pushed designers to create a more effective circuit board with few changes on the production floor.

In AI, design and planning applications are similar. In each, the input and knowledge entered into the system, as well as the goals, constraints, and solutions, produce an optimal structure. But a designer's creation is basically independent of time restrictions and unconcerned with environmental behavior, while planning requires that each possible outcome at each step in time be considered. A design application is also much more likely to be concrete (i.e., manipulating tangible components and connections instead of behaviors or actions, as with planning). Design can therefore be classified as a distinct application category.

Design optimization allows a design problem to be clearly identified as a fitness measure to judge the product. Constraints act to limit the possible number of products. Consider the design of a repair shop. The product is efficiency; the measuring stick is the turn-around time from request to repair. Designing the layout of the shop involves acquiring the most useful combination of necessary equipment and spatially arranging the equipment stations so that workers in one area least affect workers in the other areas of the shop. The designer, of course, faces several constraints (e.g., limitations of space, time, money, and product availability) that prevent him or her from trying every possible combination of tools, workstations, and spacial arrangements. The designer, guided by experience and common sense, has undoubtedly acquired heuristic knowledge about the quality of certain products and the efficiency of certain arrangements. Therefore, many conceivable combinations may be eliminated immediately. A workstation for power tools, for example, would not likely be placed next to sensitive electronic equipment.

The more expertise a shop designer has, the greater his or her store of heuristic knowledge. The remaining possible combinations form a modified search space that is much smaller than the original. From this narrowed solution space, an optimal design might be found. AI systems have been readily applied to similar tasks and appear to fit the nature of such a problem well.

AI systems that perform a design function may be applied to time-independent problems that have identifiable goals and constraints. The design of an integrated circuit is a good example. Design systems often provide collaborative information to human designers, working with them to form a better product.

3.3 DIAGNOSIS

A clogged fuel filter or an unproductive assembly line are two examples of systems operating at less than peak efficiency due to a failure of one or many parts. The symptoms may be a sputtering engine in a car or slower production time on an assembly line. When a physical, biological, or even social system breaks down, it is necessary to locate the malfunction so a treatment can be prescribed. Diagnosis is the application of knowledge and reasoning to identify such malfunctions. A diagnostician, human or computer, acts as a detective that gathers information through observation and inquiry, analyzes the clues (symptoms), considers likely possibilities, and identifies a suspect.

Diagnostics is one of the earliest and most successful AI applications. Using the diagnosis of a mechanical failure as an example, the primary features and difficulties inherent in a diagnostic task, as well as successes of AI diagnostic systems, will be illustrated. Diagnostic knowledge-based systems have frequently outperformed the average expert in a variety of problem domains in terms of both accuracy and speed. Both humans and knowledge-based systems must reason about a mechanical failure using rules of judgment, evidence of symptoms, mechanical facts, knowledge of cause-and-effect patterns, and experience. A diagnostic system generally uses four steps: 1) knowledge acquisition, 2) reasoning, 3) prescription, and 4) explanation.

Knowledge acquisition consists of augmenting the knowledge already residing in the database. Through dialogue and inquiry, relevant details are gathered which supplement existing information. For a mechanic, this information may include describing mechanical symptoms, dates of malfunctions, data provided in the manual for a particular vehicle, and other pertinent information. New and existing knowledge is combined to more completely characterize the problem.

Reasoning, the second step in diagnosis, consists of trying to fit the facts together to form a tentative hypothesis. As the hypothesis is formed, supporting facts are identified or sought through dialogue and through observation (e.g., of a vehicle and its operation). If the initial hypothesis is refuted by testing or observation, then a new hypothesis is formed. The reasoning continues until halted by diagnosis (e.g., the car has a clogged fuel filter) or by the recognition that a diagnosis will not be forthcoming (e.g., more time and information are needed). This iterative reasoning process drives the development of a diagnosis.

The third stage of the diagnostic process is prescription. Once the problem has been identified, what can be done? Depending on the diagnosis, alternative prescriptions may be offered (e.g., cleaning an old part, installing a new part, buying a new car, or some optimal combination of these). In this prescription step, remedies appropriate for the diagnosis are offered.

The final step in this methodology is explanation. In the case of an automotive mechanical problem, the reasoning and conclusions would be explained to the vehicle's owner. For many diagnostic systems, the explanatory effort is essential to its usefulness; otherwise, the utility of a sound diagnosis may be ignored. Therefore, justifying the diagnosis and the remedy is essential to the success of the diagnostician, whether expert system or mechanic.

Diagnostic systems can be effectively applied to numerous systems so complex as to defy a strictly analytical or mathematical description. Diagnostic systems are also useful when constraints (e.g., time and money) to identifying all possible functional difficulties are prohibitive. The wide range of AI applications extends from machine repair to medical diagnosis and from analysis of government policy to organizational effectiveness.

3.4 TRAINING

Knowledge is a commodity. There is constant need to transfer knowledge from those who have it to those who need it. Instructional expert systems are developed primarily to pass knowledge efficiently from a knowledge base to a human user. Efficiency is a fundamental concern with respect to use of valuable resources (human expertise) in conveying pertinent facts. There are several alternatives to knowledge-based instructional systems (e.g., human

instructors; instructional software that does not use AI technology; and media such as video tapes, audio tapes, and books). An instructional knowledge-based system has definite advantages, including interactive ability that allows queries and provides reasoned responses, reproducibility that allows one instructional system to be distributed to a number of locations, and cost effectiveness.

Instructional AI systems are important alternatives to human experts for several reasons. Human experts charge for their time, a charge likely to increase with the quality and breadth of their expertise. Experts in a particular domain may be a scarce resource and may be difficult to attract, even if funds are available. The quality of human expertise also varies drastically, as does the teaching ability of the expert. On the other hand, a knowledge-based AI system basically requires a one-time fee and is always available when needed. Furthermore, the quality of instruction is constant and predictable. For these reasons, AI is a viable alternative in instructional applications.

Instruction requires three things: 1) knowledge to be taught; 2) an inference mechanism to generate new ideas or facts from existing ones; and 3) an interface unit that accepts and supplies questions, requests, and instruction. Each component must be specially attuned to the task of teaching. For instance, the knowledge base must include facts attuned to the information consumer. It is often appropriate, for example, that novices be taught differently than students who are already familiar with the subject matter.

The human interface of an AI knowledge-based system should have a well-designed query system capable of providing clear and complete explanations. Human users may ask questions, the answers to which are not contained in the store of knowledge. A static form of knowledge representation, such as a book, is of little use if it does not contain the required information. With a knowledge-based system, however, reasoning chains can augment the knowledge base and create information to answer the user's questions. Because almost every domain contains virtually limitless potential questions, reasoning is an immensely useful tool. The number of instructional uses for AI is as great as the number of expert domains, limited only by imagination and resources.

3.5 MONITORING

To monitor a process is to measure and compare. At each step in the process, the monitor is told how a system should run; specific system behaviors are identified as abnormal if they deviate from the proscribed pattern. The monitoring system is not responsible for controlling the process but rather observes and reports system errors.

In the simplest terms, monitoring is a pattern recognition task. The number of possible system errors is matched against the recognized (defined) system patterns. However, as with other AI applications, problems often increase in complexity beyond the simplest case, thereby introducing the need for alternatives to traditional computing.

Monitoring can be thought of as a passive form of control, in which signals are received and then judged either appropriate or inappropriate; however, there is no active involvement in the form of control signals. Small fluctuations are typically unimportant to a monitoring system but are addressed by a control system. In a monitoring system, the knowledge held is predictive, designed to identify trends in the sensing data and to predict the long-term behavior of the system. Control systems, on the other hand, are designed to maintain a particular trajectory and are not concerned with long-term system behavior except as a series of short-term steps.

Two functions are essential in the task of monitoring: 1) recognition of an abnormal state and 2) notification of its existence. A monitoring system acts as an observer, identifying and reporting deviations from specified norms. Such a system may or may not be based upon understanding of the actual process and what is normal, but the greater the knowledge base, the deeper the inferences that can be made about abnormal modes.

Of course, for every increase in complexity, there is an explosion of possible error combinations that makes it less reasonable to specify every possible error. Unforeseen errors may be left off the list of anticipated error states. Unless the monitor can infer knowledge of abnormal behavior from a causal analysis, the system will be unable to properly recognize and report these unexpected errors. An approach that appears increasingly useful permits recognition of abnormal states through "deep" knowledge about the causal behavior of the system. This application is ideal for an AI system designed for a monitoring function.

An AI monitoring system can be used to check performance in a wide variety of applications, and what the system is constructed to monitor is, of course, a design consideration. It may be used to gage the performance of complex and costly machinery or as an early warning system. A monitoring system in a coal-burning power plant, for example, may collect and analyze gage readings from fuel consumption and heat sensors. System monitoring can be applied to almost any temporal behavior of sufficient complexity. All that is required is knowledge of what is not allowed and/or the ability to infer what is not allowed based on reasoning.

3.6 PLANNING

A plan is course or sequence of actions which occur between the input of information and the attainment of a goal. A plan is formulated with a specific goal in mind. Each step of the plan is intended to contribute to goal achievement. As goals become more complex, it is necessary to establish a hierarchy of goals and subgoals to be accomplished. In short, a plan is sequential, purposeful, and hierarchical, often composed of many parts differing in their levels of abstraction.

AI systems can be used as planners in diverse areas. In robotics, planning is important, because a machine that can plan can, in essence, program itself and therefore become much more autonomous. Many management decisions can be cast in a planning template, including common decisions required concerning labor and investment.

Efficient planning requires logical and complete structuring of the problem at a functional level of detail. To be complete, all the pieces must be present. A plan for moving crates from one location to another must include subgoals of checking to see that the crates exist in the proper location, that they are movable (not too unwieldy or heavy), that the new location is available, and that there is enough room for all the crates to be stored at the new location.

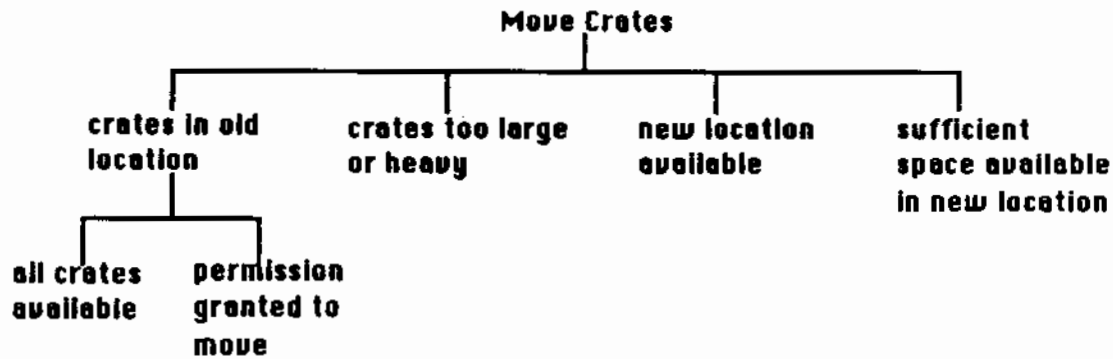


Figure 7. Planning Example

The level of functional detail required differs from task to task. The task of moving crates, for example, would require more complicated specifications if the new location were farther away, giving rise to many more potential pitfalls. In addition to being complete, the plan must capture the structural logic of the task, including capturing the temporal relationships among events (e.g., crates must be picked up before they are set down).

The difficulty is that in real life, an almost limitless supply of surprises may be encountered. In the above example, the robot may pick up an empty box, encounter objects along the way, drop a container, find a closed door, and so on. The planner must be able to replan based on new information. AI is applied to reduce the number of viable alternatives and generate an effective plan of action.

A "good" plan, whether the product of a human mind or a knowledge-based system, must satisfy certain requirements. The first and primary criterion is success: Does the plan accomplish its goal? If so, then other criteria, such as flexibility and efficiency, may be assessed. A flexible plan not only responds to foreseeable events but effectively contends with unforeseen events, either solving the problem despite unanticipated circumstances or minimizing their harmfulness. Efficiency is important because carrying out a plan requires actions to be taken that require expenditures of resources. All else being equal, a plan which optimizes the use of resources is superior to one that does not.

Planning systems require several forms of knowledge: e.g., knowledge of the problem domain (i.e., problem, cause, and effect), how to translate the problem domain into actions that

can achieve a goal, and how to anticipate which actions may enhance an understanding of the environment and its potential hazards. In addition, such systems require effective reasoning, a capability offered by AI.

As complexity of a problem structure increases, the potential for effective use of AI increases. Planning tasks appropriate for application of AI are frequently associated with industrial robots. For each chore assigned to a robot, a set of movements must be planned to enable successful performance. Even for uncomplicated tasks, the planning of the robot's movements can require several layers of subgoals. Other examples of planning tasks suitable for AI applications can be identified by searching for similar problems characterized by a series of stages which must be accomplished sequentially. AI is becoming more and more applicable as planning problems increase in complexity.

3.7 PREDICTION

Predictions are informed guesses based on facts about the current situation and experience with similar situations in the past. A talent for prediction can be extremely useful for solving problems that rely on estimates of future environments. Prediction involves reasoning about temporal tendencies from knowledge of 1) causal relations or 2) historical data or experience. Prediction entails having at least partial knowledge about how a system operates and forecasting when and why failures may occur. For example, if the domain of interest is future prices of a piece of equipment, then relevant factors would likely include recent changes in prices of major components, prices of similar equipment, and interest rates that affect production costs.

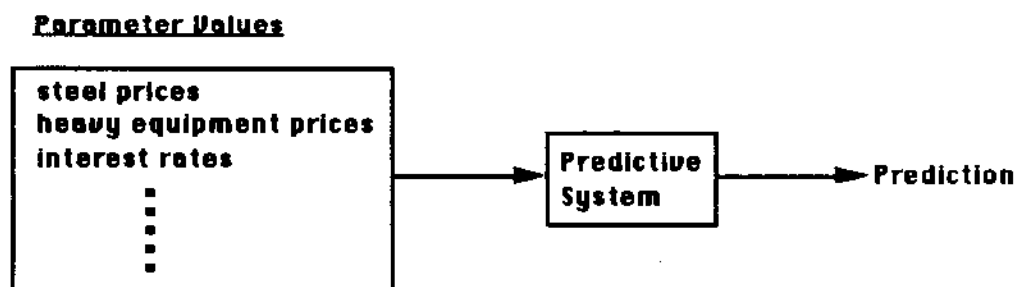


Figure 8. Predictive System

In real-world problems, many variables, some very obscure, may be important for accurately predicting future behavior. In such cases, the application of AI methods may be appropriate because of the complexity that arises from the poorly understood interaction of many environmental variables. In addition, AI may be useful in such tasks because of its ability to use experiential or heuristic knowledge.

Prediction problems essentially consist of converting the knowledge of a process into an estimate of the future status of the system, given a history of events and knowledge of causal relationships and the environmental state. One way to view a predictive problem is as an inductive learning task. From the data available, a global system structure is surmised. This amounts to discovering the rules by which the system operates. Once these rules are understood, future behavior can theoretically be anticipated.

Another way to view prediction is as an autoassociative recall task. An autoassociative memory is capable of retrieving a complete pattern from its store given only a partial stimulus. In other words, given partial values of certain parameters, other parameter values can be predicted. In this paradigm, only patterns are identified, not rules. As an example, consider a study analyzing the use of manpower. Two features of interest may be the number of sick days taken and the number of vacation days given. Perhaps a relationship is discovered such that when the number of allotted vacation days is increased, the number of sick days taken is reduced by twice that amount. However, it should probably not be inferred as a causal rule that changing one parameter automatically changes the other. Many other factors would undoubtedly need to be considered, leading to the need for higher-order statistics. For many types of complex problems, a statistical relationship is all that can be claimed.

The two approaches, inductive learning and autoassociative recall, are generally applied to different types of problems. For systems that are well understood and not too complex, such as mechanical systems, inferring causal relationships and discovering rules of operation may be a valid approach. For more complex systems, such as economic or social systems, prediction through recognition of recurrent patterns is probably the sounder approach. Heuristics, a trademark capability of AI systems, may be added to increase efficiency or to fill knowledge gaps.

Obviously, the ability to predict parts failures, labor shortages, or other events can be an invaluable asset, well worth the investment required for AI technology.

3.8 REPAIR

The repair function involves specification and implementation of steps that return a system to working order. Such tasks can be especially challenging when performed by machine and, therefore, require close cooperation between the engineer and the AI expert. Interfaces between hardware and software must be smoothly designed.

In implementing repairs to a mechanism, AI systems must use not only knowledge of the mechanism itself but also of the system of which the mechanism is a part (e.g., capabilities and limitations). Using AI technology together with careful integration between the machine to be repaired and the machine that will perform the repair, an AI system can be designed to both prescribe a remedy and then physically implement the remedy.

Prescription may be defined as formulation of a plan to correct an identified malfunction. This requires a relatively shallow knowledge of the malfunctioning mechanism compared to the knowledge necessary for diagnosis. A repair system may be simpler and may require far fewer inferences concerning the structure and operation of the faulty mechanism. For example in debugging a computer program or an electronic circuit board, designing a remedy (e.g., removal of an unwanted character or replacement of a part) may be relatively simple once the error has been identified. The difficulty in prescription is to specify a repair operation that is complete enough to allow the implementation component of the system to implement the plan. Prescription, then, is a sophisticated interface between the operational knowledge, which provides a diagnosis, and the physical hardware, which performs a repair.

Implementation, the second of the two repair stages, is commonly associated with the hardware used to perform the physical steps. The implementation component is composed of the interpreter, which translates prescribed steps into required actions, and the hardware designed to carry out the instructions. The mechanical characteristics of the hardware depend entirely on the task to be performed and may include anything from a stationary spot welder to a flexible robotic arm requiring complex movement specifications.

Repair systems have the common goal of replacing human labor in tedious and repetitive work. Robots and robotic systems are closely associated with repair applications. In fact, the development of repair systems are often constrained by the available hardware to perform the physical function. As the hardware becomes more sophisticated, so will the applications designed for AI repair systems.

3.9 PERCEPTION AND ANALYSIS

As machines perform more of the rudimentary tasks formerly assigned to humans, they will require an expanded ability to perceive and analyze the environment around them. Perception is the receipt and transformation of environmental signals into usable information. This involves reducing a host of complex environmental inputs into symbol classes. Analysis is the combination of the new information (symbols) with existing knowledge in a useful manner.

Through perception and analysis, the environment is separated into relevant and irrelevant pieces. The primary objects become distinct from background objects, and the pieces are assembled into a logical whole. For example, spoken words are distinguished from among the auditory signals (e.g., bird songs, music, automobile noises, barking dogs) that strike a receiver's ears; individually recognized symbols are then assembled to create an understanding of the utterance.

Humans beings constantly use their five senses--sight, sound, smell, taste, and touch--to extract detailed information from the world. Similarly, a wide range of task-dependent sensors have been developed for use by machines. For example, an infrared sensor has been designed to detect the radiation from a thermal source and is useful for night vision, and other sensors respond to events such as sound waves or vibrations in the earth's crust. Signals received by sensors such as these are transmitted for processing to other components of the system. In this way, sensors work together to form an expanded characterization in order to recognize an object.

Recognition is the comparison of received impressions with templates stored in memory. A close match implies recognition. The following diagram illustrates the recognition process:

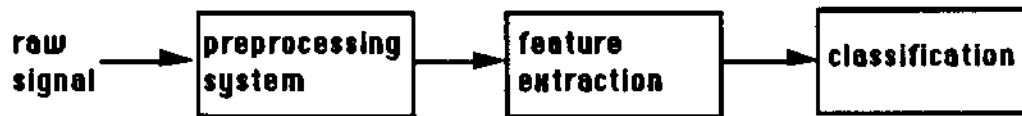


Figure 9. Perception System

After the entity is classified, analysis may take place to determine what it actually "means" for this entity to be present. If the task is natural language understanding, for example, then an identified letter is combined with other identified letters so a whole word can be formed. This is most often a hierarchical analysis. Pieces of low-level information are combined to give a piece of high-level information. This combination process continues as long as necessary for successful recognition. Such operations are knowledge-and memory-intensive and require massive searching and matching time to resolve the patterns to be recognized.

Successful application of machine perception and analysis is often dependent upon a well-chosen combination of AI technologies. The sensors must be chosen to fit the nature of the task. Various AI methods may then be combined to process, or analyze, the incoming information. For instance, neural networks may be chosen to perform low-level processing of signal data, to extract features, and perhaps classify the information. These classes may then be combined with facts and rules of reasoning which reside in a knowledge-based system. The products of these other, similar analyses may be combined using a blackboard system.

There are several applications for machine perception and analysis. In industry, robots may require a vision system to perform general chores. Natural language understanding is important to a variety of goals, including the simplification of human-computer interaction. Such perceptual and analytical systems provide an interesting mixture of AI methods.

4.0 TOOLS

Building machines capable of solving problems using human-like intelligence was one goal of those who developed the first computers. Early efforts had little success. The equipment was primitive and little was then understood about human intelligence. However recently, since development of sophisticated hardware and advancements in the understanding of human intelligence, this goal has been reestablished. AI has become a commercially viable source of solutions for problems previously unrecognized or ignored.

Increased interest has spawned hardware and software improvements that have resulted in advances in technology. Currently, the market is filled with a variety of vendors and an array of products that extend across a wide range of potential applications.

4.1 SOFTWARE TOOLS

Although the purpose of this report is to introduce the general field of AI applications, the tools discussed are primarily used in knowledge-based systems. There are two reasons for this. First, knowledge-based systems (expert systems) are the most mature and the most commercially viable. Therefore, many more products exist for knowledge-based systems than for other AI systems. Products developed for other AI areas are largely untested due to their newness. Second, expert systems shells and knowledge engineering tasks are quite different from other programming tasks. They are, therefore, served well by tools. Other AI methods, such as genetic algorithms, are typically developed from existing products that are not specifically applicable to a particular problem. In other words, C or FORTRAN can be used to develop a genetic algorithm to perform an optimization task on a commonly available workstation or personal computer (PC). A whole new development environment is not required.

Software tools facilitate construction of knowledge-based systems. The number and types of knowledge-based system tools has proliferated in recent years. Today, there exists a wide variety of such tools. Some are as basic as new versions of computer languages, with characteristics that provide advantages for the development of knowledge-based systems. Others are complete and flexible development environments designed for large-scale applications on expensive and specific hardware. It would be impractical to try to describe all

of these systems. Rather, it is useful to divide them into the following three categories: 1) programming languages, 2) small tools knowledge-engineering languages, and 3) large tools knowledge-engineering languages.

Programming languages may be divided into general-purpose languages and AI languages, the main difference being that general-purpose languages are numerically oriented and AI languages are symbol oriented. These languages are not identified by particular proprietary implementation but by characteristics relying on symbolic processing for the development of knowledge-based systems.

In reference to small tools knowledge-engineering languages, "small" refers to both the representational capacity of the system and the capabilities of the hardware platform. In discussion of this category, specific commercial examples will be identified.

The third category, large tools knowledge-engineering languages, represents large and powerful knowledge-based systems tools offering multiple methods for representing knowledge and for specifying program control.

The range of features and capabilities available in knowledge-based systems form a continuum in terms of both price and power.

A variety of tools are necessary to accomplish a variety of tasks. The first decision faced by the builder of a knowledge-based system is whether to build the system from scratch or to purchase a shell. This decision relies on several considerations. If a knowledge-based system requires new methods of knowledge representation or program control or new combinations of established methods, the system will probably have to be built from scratch. On the other hand, the development of a more stereotyped system can proceed by using a less flexible knowledge-engineering tool, such as a PC-based knowledge-engineering language.

Broad guidelines are available for choosing the proper tool to efficiently represent the type of knowledge in question. The type of inferencing or control paradigm needed is also a consideration. A knowledge-based system developer must also recognize that developing a system from scratch will probably be much more difficult than using an established

knowledge-engineering language. The tool selected for a knowledge-engineering task should be the most specific tool that meets all of the task requirements.

4.2 GENERAL-PURPOSE LANGUAGES

General-purpose languages are imperative languages reflecting the operation of the typical sequential computer. They are, in effect, higher-level versions of the architecture because they manipulate computational resources similar to machine language.

General-purpose languages (e.g., C, Pascal, Modula-2, and ADA), which are familiar to almost anyone who writes computer programs, have several advantages in the production of knowledge-based systems. First, because they are designed for general-purpose hardware, they usually run faster on such hardware than languages designed for AI. Second, because of the widespread distribution of both general-purpose hardware and programming languages, these products are broadly available and enjoy a strong support network. Third, many programmers are available who are experienced in using general-purpose languages. Finally, knowledge-based systems using general-purpose languages can easily be integrated with existing software.

These advantages of efficiency and popularity are offset by several disadvantages. First and most important, general-purpose languages do not support symbolic processing, which is the essence of knowledge-based systems. Second, these languages do not have automatic memory management, which is important to the memory-intensive development of knowledge-based systems. Third, these languages, by design, concentrate on data manipulation rather than on the essential algorithm. This may make development of the system more difficult. Languages that more closely follow mathematical models, such as AI languages of LISP and PROLOG, can facilitate algorithm design. The shortcomings of general purpose-languages translate into longer and more difficult knowledge-based system development. When a general-purpose language is used to implement a knowledge-based system application, the application is first developed, then translated into an imperative language.

4.3 ARTIFICIAL INTELLIGENCE PROGRAMMING LANGUAGES

Because of the disadvantages of general-purpose languages, AI programmers and knowledge engineers have used other languages, generally known as AI languages, although they need not be limited to only AI applications. AI languages can be divided into three categories: 1) functional languages, 2) logical languages, and 3) object-oriented languages.

4.3.1 Functional Languages

Functional languages are true functions in the mathematical sense. LISP, the first and most popular AI programming language, is an example of a functional language. There are a variety of dialects of the LISP language available on all common platforms, from dedicated LISP machines, to personal computers.

Functional languages provide easy and flexible symbolic manipulation. The primary data structure, the list, treats all data types as equal, putting alphanumeric symbols on equal footing with numerical variables. Built-in capabilities, such as property lists, facilitate the use of symbols and increase the symbolic computational power. LISP has a concise syntax, which decreases the amount of code required to accomplish a given task. There is evidence that functional languages like LISP are relatively easy to "debug," as well. Extensive editing and debugging aids ease the chore of writing programs. The uniform treatment of data and control structures makes self-modification easier. Another very important feature of LISP is automatic memory management. This feature (called "garbage collection") allows for more efficient use of computer memory.

4.3.2 Logical Languages

Logical languages are powerful tools for constructing AI programs. The most popular member of this group is PROLOG, the name of which was derived from words "programming" and "logic." Though not as old as LISP, PROLOG has a large following among programmers; in fact, PROLOG was chosen as the sole language for development of the fifth-generation computer in Japan.

Like LISP, PROLOG was designed with symbolic processing in mind and is capable of very efficient list processing. It has sophisticated pattern-matching capabilities, as well. PROLOG has a simple syntax when compared with procedural languages, and computation is controlled chiefly by a process of logical deduction.

Programming in PROLOG is conceptually simple. First, a knowledge store must be developed by entering facts and entities into the knowledge base. Then rules must be specified describing how the entities relate. To use the program, queries are written. As a simple example, assume that the following fact is entered: "Carolyn loves everybody." Then assume that the system is asked, "Does Carolyn love Tom?" The program will respond affirmatively. Large collections of facts and rules about the facts can describe complex logical systems.

4.3.3 Object-Oriented Languages

Object-oriented programming is a relatively late arrival in the field of mainstream programming and may therefore be less well known than other language types. The concept of an object-oriented programming language is somewhat nebulous, at least partly because of the variety in the languages claiming to be object-oriented. Although definitions and features are by no means final and definite, certain general characteristics can be identified: specifically, dynamic binding, data abstraction, and inheritance.

The basic idea of object-oriented programming involves the encapsulation of data and data operations. Procedural languages establish coded operations in the form of procedures (i.e., programs that alter data structures that are passed to the procedure). Object-oriented languages hold the data, as well as the data operations, in a single entity called an object. Each data class belongs to an object. These data classes are known as instances. For example, integers and operations performed on integers may occupy an instance of an object class. Each object has two parts, a shared part and a private part. The shared part contains characteristics shared by other instances in the object class. Private parts make an object distinct from all other objects.

A simplified view of programming with an object-oriented programming language is as follows. Computation is performed by sending messages to a particular object. (A message is a request to engage a particular operation.) When an object receives a message, it first

determines whether or not it is able to service it. If it can, it establishes the course of action to be taken and then performs the required steps. After completion, the object returns the result to the sender.

4.3.4 Knowledge Engineering Tools

A knowledge-engineering language is essentially a package of software tools and a software system that facilitates the construction, operation, and maintenance of a knowledge-based system. Using one of the AI programming languages lessens the imposing task of developing a knowledge-based system from scratch, allowing more time for designing the problem solution.

The features of knowledge-based system tools differ according to the product, but the basic components are similar: 1) a knowledge base that holds the store of factual information, 2) an inference engine that specifies the inference and control mechanisms, 3) an explanation facility that provides the rationale for decisions, and 4) a user interface that carries information to and from the system user. Each of these components provides support in building one of the knowledge-based system modules.

To build the knowledge base, commercial tools support particular type(s) of knowledge representation, such as rules, frames, or semantic nets. Larger, more powerful tools allow multiple representations. For creating the inference engine, tools support one or more inferencing strategies to specify the construction method of new facts from existing facts. Again, larger and more powerful systems allow more than one reasoning strategy. A facility for explanation is also typically contained in the commercial tool. This explanation facility will explain the process of machine reasoning to a user by displaying the sequence of firing rules, for example. The pieces of a user interface are also included in most tool packages. Often sophisticated debugging and editing capabilities are part of the complete package. These components speed the time-consuming job of building and testing the knowledge base and the system as a whole.

Several measures of knowledge-based system tools can be used to classify the products available, including the following:

- the number of rules allowed
- the number of representation and inferencing methods allowed
- the generality or specificity of the intended use.

Such measures are not complete. Therefore, the types of knowledge-based systems tools are divided into two broad categories: 1) small systems tools and 2) large systems tools. As personal computers become more and more powerful and operate with more and more memory, this division becomes blurred. However in general, larger hardware systems are used for the more powerful and flexible knowledge-based systems tools.

A few general comments should be made about the purchase and use of off-the-shelf tools. First, although some classifications are useful in examining knowledge-based system tools, the characteristics and capabilities of such tools exist in a continuum; sharp distinctions may be deceptive. The features and merits of each tool should be considered before purchase. Second, the most specific product that fits the task should be purchased. More specificity usually means that less time and effort is required to accommodate the user's specific application.

4.3.5 Small Systems

Small knowledge-based system tools are designed to run on personal computers. (The two main classes of personal computers are the IBM and IBM compatibles and the APPLE Macintosh family.) Using these tools, relatively inexperienced users, often experts in their own fields, have built successful applications without the assistance of knowledge engineers. Less sophisticated developers, or those whose projects do not require the complete package of options offered by large system tools, may find it advantageous to use the smaller tools. They are easier to learn and cost substantially less than large systems. This can be important not only for developers but also for the end users who will purchase the tool.

Small knowledge-based system tools also differ from larger tools in that they support fewer rules than large systems, although the number of rules allowed increases with

expansion of the power on a personal computer. Because of processing power and memory constraints, small systems are less flexible than larger ones. Therefore, they support fewer inferencing schemes.

4.3.6 Large Systems

Large knowledge-based system tools are usually capable of storing vast amounts of knowledge in the form of rules, frames, objects, or whatever forms the knowledge base supports. Large system tools also possess an array of representational schemes and control structures that allow the tool's capabilities to be tailored to a specific problem. In addition, they contain sophisticated features to assist in the development process.

Large systems, because they have more features to both represent knowledge and to combine it in a process of reasoning, are especially useful in construction of large knowledge-based systems with potentially thousands of rules. Large systems are, for the most part, designed to run on larger, more powerful computing equipment, such as mainframes, workstations, and special-purpose LISP machines. Consequently, they require a more significant investment, not only in the purchase of the tool itself but also in the purchase of the supporting equipment and training of personnel.

The choice of which tool to purchase depends on the case and may involve many considerations. The investment in both money and time is much greater for large system tools, so the payoff for the system or systems to be developed should be expected to be high. The strategy of the implementing organization may be a factor. If the organization is committed to development of knowledge-based systems and seeks to build a large and visible system flagship, then a more flexible and costly system may be warranted. On the other hand, if the approach is to put together a series of moderately sized applications, then small systems tools would probably be more appropriate.

4.4 HARDWARE TOOLS

The extension of AI applications has benefited from special-purpose hardware designed to accommodate their unique processing needs. Development of knowledge-based systems, in particular, required an architecture designed to support the AI programming languages (indirectly the symbolic processing), which were used to build these systems. Other AI

applications have their own processing requirements. Neural networks require massive parallelism and employ simple processors with large capacities for interconnection with other processors. Special-purpose array processing boards and even more promising neural network chips have been designed to meet these needs. Genetic algorithms can benefit extensively from parallelism, as well. Model-based reasoning, the success of which depends on the complexity of the model, can be computationally intensive, also requiring a powerful machine.

The development of special machines for symbolic processing has been of special interest to developers of expert systems. Design features of these machines include a large memory and a large virtual address space. A high-speed machine for cycle-intensive operations is also important. Several stand-alone machines have been designed by different vendors to meet these special requirements.

The drawback of specialization is the loss of general programming capabilities. While enhanced symbolic processing machines have demonstrated advantages in symbolic applications, traditional workstations have enjoyed a share in the AI market because of their general-purpose capabilities. There is a trend toward combining knowledge-based systems applications with traditional software systems. This approach has several advantages, including lower cost, broader access, and ease of use.

The three categories of hardware platforms--the personal computer (PC), the workstation, and the specialized symbolic platforms--represent a logical division of processing capabilities. They are discussed in the following subsections. Other classes, such as mainframe computers, appear to be less important to the development of AI applications and so are excluded from this discussion. (Parallel machines, which are becoming more important, are discussed in Chapter 5.)

The distinctions among classes of computers are also becoming blurred. New chip technology has greatly increased the processing power of small systems; and additional software capabilities, such as windowing, have provided them with expanded functionality. Also, new machines, such as the SPARC Stations introduced by SUN, fit somewhere between the boundaries delineated. It is clear that the market for computers has matured tremendously and has filled many of the gaps that formerly existed.

4.4.1 Personal Computers

The dramatic increase in computing power in PCs has made it reasonable to consider them for AI applications. Experimentation with PC parallel computing is ongoing. The introduction of MacIvory, a new LISP chip, marks a major upgrade of minicomputers adapted to larger tasks. Because of the increasing viability of the PC as a hardware tool, the power and flexibility of knowledge-based systems products have increased, as well. This is apparent by the introduction of scaled-down versions of the large products and by the entry of new products into the commercial marketplace. The trend in computing has been toward distributed resources and small stand-alone systems that can be networked to allow communication and software sharing. (The list of small systems tools in Appendix A attests to this trend.)

Minicomputers afford several advantages over other types of platforms. The first is cost. A range of prices exists, depending on the model and quantity of additional equipment to be included; however, \$3,000 to \$5,000 is generally sufficient to purchase substantial computing power. Within this price range, a team of users can each be equipped with computers and software. The distribution of resources may serve to avoid user conflicts that frequently occur when computers must be shared. Another advantage of PCs is reliability. It is very unlikely that many PCs will be inoperative at the same time in a group setting, whereas the down time on a shared computer affects all users simultaneously. Still another advantage is that new users can easily be accommodated. Experimental hardware can be added with little disruption to work. A final advantage is that products developed for small systems immediately have a wide field of potential users. If a knowledge-based system is developed in one location, it may easily be exported to other locations without extensive conversion costs.

4.4.2 Workstations

Workstations have traditionally been considered a tool for general-purpose software development. They should also be seriously considered as a platform for AI applications. Besides their capabilities, such as extensive windowing systems, bit-mapped screens for graphics, multifunction mice, and well-developed operating systems, workstations offer the key potential for integrating knowledge-based systems applications with mainstream applications. For some applications, such as computer vision, this integration is a requisite for

increasing speed and usefulness. For other applications, integration will allow a beneficial mixture of techniques from mainstream programming to database technology, thus providing a wider array of problem-solving tools.

Workstations that have traditionally relied on C, Pascal, FORTRAN or other procedural languages may now include versions of LISP, PROLOG, Smalltalk, and other AI languages. Some are specially altered to increase efficiency. This adds another level of usefulness to these machines, enabling them to compete with dedicated AI machines.

A brief list of workstations and dedicated symbolic processing machines is included in Appendix B. PCs are not discussed in the appendix, because information about them is widely available through magazines that focus on the PC market.

4.4.3 Dedicated Artificial Intelligence Machines

Several manufacturers have dedicated stand-alone machines to symbolic processing. These manufacturers produce hardware platforms for committed users and developers of knowledge-based systems with large applications. These systems are designed to ease the difficulties of developing complex pieces of software and increasing processing speed.

Most computers dedicated to AI and LISP processing share a number of common characteristics. They have high resolution, bit-mapped displays, large physical memories for memory-intensive knowledge processing, and high-speed processors designed for rapid implementation of LISP instructions. These machines also support powerful knowledge-based system development tools that facilitate editing and debugging of long, complex chains of code.

An example of a dedicated symbolic processing machine is the Texas Instruments Explorer II Plus. Texas Instruments utilized state-of-the-art, 1.0-micron LISP chip technology, which condenses almost 600,000 transistors into a square centimeter of area, in developing a unique architecture to create an extremely powerful symbolic processing computer. This architecture is designed to decrease the time required to perform instructions in the LISP language and to facilitate the frequent use of memory and particular LISP data structures. The Explorer II Plus features a 40-MHz clock. It also has a pipelined architecture that performs microinstructions and many LISP macroinstructions in a single-clock cycle. Hardware-supported functions for manipulating complex data structures and memory

management are provided; this hardware support is necessary for rapid symbolic processing with the LISP language. The uniquely designed LISP chip has a 32-bit VLSI processor with a 114K of RAM memory to speed the execution of instructions. Because symbolic processing is generally memory intensive, the Explorer II Plus provides up to 128 megabytes of memory for large applications. There are several other characteristics, such as a high-resolution monitor (1024 by 808 pixels), which are designed with the user in mind.

The description of the Explorer II Plus illustrates many of the features that are essential for truly efficient symbolic processing using LISP. While other companies have designed their own dedicated machines with their own mix of technological features, the basic impetus behind these distinct implementations is the same.

5.0 EMERGING ISSUES IN ARTIFICIAL INTELLIGENCE

5.1 PARALLEL PROCESSING

AI and parallel processing are two relatively new technologies. Because both are in early stages of development, commercial products uniting the two technologies are not readily available. Also, prices for computationally useful parallel processors are still very high. But the trend towards more affordable parallel computers, and even PC-based parallel computing tools, promises that soon such products will be both available and affordable.

The capabilities of parallel processing match closely with the needs of AI applications. Parallel computers provide a number of processors and expanded memory, enabling very large problems to be solved much more efficiently. Evidence shows that natural intelligence relies on massive parallel processing for vast amounts of data. Therefore, machine parallel processing is also a powerful means of achieving the increased processing capabilities required for many tasks requiring AI.

Parallel processing can be used with very large and complex applications and works well with numerically intensive operations (i.e., mathematical modeling of complex systems). Many AI applications (e.g., computer vision systems, speech processing, and large expert systems) would be well served by parallel processing technology. These include inherently parallel computational methods of AI, such as neural networks and genetic algorithms.

AI and parallel processing can form a symbiotic relationship, with each expanding the potential application areas for the other. Most vendors of parallel machines have expressed interest, through invested resources, in supporting the development of products that will make their parallel machines available to members of the AI community. As parallel processing hardware decreases in price and increases in availability, and as more people use parallel programming, it is likely that AI will experience a surge in interest.

5.2 PARALLEL HARDWARE

Four basic classes of computers exist. One class is the sequential computer; the other three are parallel computing models. Any computational device, sequential or parallel, receives and operates on input according to a set of instructions. A sequential computer is a

single-instruction stream, or single data stream model (SISD). At any given moment, a single instruction is manipulating a single piece of information. Parallel machines, on the other hand, have multiple processors and are capable of extended computation. They are capable of operating with multiple instruction streams on a single data stream (MISD), a single-instruction stream on multiple data streams (SIMD), or multiple instruction streams on multiple data streams.

A particularly useful model of parallel computers is the SIMD class. These machines perform the same manipulations on multiple inputs. They are capable of performing many data-intensive tasks with a high degree of parallelism. In SIMD machines, data is passed between processors in one of two ways: through shared memory models or interconnected machines. Shared memory models have a large block of memory available to all processors; data passed among the processors is written to and read from this shared memory. Interconnect models are distinguished by a network of connections between the processors. Ideally, every processor is connected to facilitate sharing of data. However because of manufacturing constraints, this kind of interconnection is not economical, so interconnect models are divided into particular topologies, such as linear arrays, mesh arrays, hypercubes, and perfect shuffle exchanges.

It is necessary to have so many types of parallel machines because, to take full advantage of the multiple processors, different problems are best suited to different architectures. For example, the Connection Machine, with up to 64,000 simple processors, is effective for low-level operations on large data sets and is appropriate for vision and speech applications; the butterfly machine, which contains no more than 256 processors, may be used for less specialized AI applications, such as expert system development.

5.3 SOFTWARE DEVELOPMENT

Software development historically lags behind hardware development. Many machines are still burdened by primitive compiler designs, awkward operating systems, and lack of higher-level parallel language support. Several parallel implementations of common

programming languages have been developed (e.g., Concurrent Common Lisp for Intel's iPSC parallel processor, developed by Gold Hill). As these languages gain in support and popularity, and as improvements and new languages are introduced, the task of programming parallel machines should become less formidable.

6.0 DEPOT APPLICATIONS

Knowledge may be described as the usable encapsulation of the structure or organization of the surrounding environment. Applications in AI are efforts to use the intangible quantity called knowledge to increase the efficiency of a process. As the industrial revolution was founded on mechanization and the resources necessary for it, so the current revolution of the world economy may be built on information and concomitant technologies. AI is a technology that applies knowledge, a specialized form of information, to complex tasks facing industry and government.

How and when to bring AI to bear upon the specific types of problems found within the Army depot system, both currently and in the future, is the subject of this section of the report. Experience acquired in AI applications in industry will be discussed in relation to depot system requirements.

In the next decade, AI technologies will primarily be used to fulfill two functions. First, they will augment the physical capabilities of humans. Flexible, multipurpose robots partly driven by AI technology will inspect assembly line products, test malfunctioning circuit boards, and operate production machinery. Intelligent machines processing sensor signals will be able to perform many tedious, hazardous, or labor-intensive tasks at a savings. The more intelligent the robot, the greater the number of tasks it will be able to perform and the greater the autonomy of its performance. Increased utility in this area will depend on technical advances in both hardware and software.

Second, AI applications will be increasingly used for decision support. This function may include intelligently filtering huge quantities of data from complex domains, constructing expert assistants to help with professional and skilled decisions, and intelligently controlling various processes by automatically adjusting parameters to optimize efficiency.

The role of AI will continue to expand in the development of expert assistance, a technology that is, to a fair extent, already commercially successful. For a variety of reasons (e.g., the need to reduce costs, contend with skill shortages, ensure quality, and speed performance time) decision- support expert and model-based reasoning systems will likely be implemented in greater numbers.

The process has already begun in the world's competitive market place. The Japanese, in their Fifth Generation Computer Project, have invested funds and effort nationwide in knowledge-based systems research and development designed, in part, to support decision-support expert systems technology. Many American companies have successfully implemented such systems, and such technology is expected to become more widespread during the 1990s.

Many challenges exist to the depot's ability to perform its required function over the next few years, and AI may be able to play a substantial role in addressing these problems. As the complexity of the machines used in everyday life increases, so does the knowledge required to use and maintain them. However, forecasters predict a slow rate of growth in the number of skilled, technical personnel. Shortages of experts are expected, just when more expertise is needed to manage the increased complexity brought by technical advancements. As a result, organizations in need of technically competent people, especially organizations less able to compete for them, must find adequate substitutes. AI will be used to reduce this shortage by effectively increasing the skill level of personnel and decreasing decision time, making available experts and skilled staff more productive.

Another anticipated constraint within the public sector of the economy is budget cuts. The size of the federal debt demands that savings be found wherever possible, thus requiring that all areas of government become more cost conscious and efficient. The military, because of the generally perceived decrease in international tensions, may be a politically attractive target for budget savings. The mechanization of knowledge through AI technology may serve to ameliorate budget shortages. The percent of manufacturing costs attributed to knowledge areas such as design, engineering, resource management, and related activities today represents most of a product's costs, and this percentage is constantly increasing. The implication is that the best way to streamline an operation may be to streamline knowledge-dependent activities. When people can be made more efficient by relatively low-cost support tools, important gains in productivity are possible.

Still another observation that supports an expanded role for AI in the coming decade is the current waste of "knowledge resources." It makes sense to conserve the talents and skills of experienced personnel, especially considering the number of years and dollars that an organization invests in such people. With the retirement or other separation of a skilled

professional, the skills and expertise of the individual are often lost. Knowledge-based systems, and to a lesser extent other AI technologies, provide means of capturing expertise developed within the organization, which can then benefit the organization even when the expert leaves the company.

During the past decade, industry has grown increasingly aware of the changes that have collectively been termed the "knowledge revolution." Consequently, efforts have increasingly been made to develop industrial applications of AI techniques. These efforts, particularly in the area of expert system development, have generally proven successful, yielding large gains in productivity and sizable returns on investment. Much remains to be learned, but experience to date encourages a close look at the potentially profitable applications of AI.

Advances in AI are occurring rapidly. One reason is that expenditures for research made during the past two decades are now producing commercial applications. As the field matures, the number of industrial, real-world applications should increase. In addition, advances in AI technology are closely tied to advances made in computing machinery. Because AI is essentially the study of how to contend with complexity, more computational speed translates to wider applications. Advances made in processing power, parallel processing, optical storage and optical computing, and computer networking will doubtless contribute substantially to new applications.

The role of AI problem-solving techniques should also grow because of AI's recently expanding ties with more traditional software and computer hardware. Embedding AI techniques within other applications legitimizes the technology and familiarizes applications-minded people with their capabilities. Distribution of AI products and languages to more mainstream computing environments allows AI techniques to be considered as options for a wider range of problems.

In the following subsections, a representative cross section of depot activities are discussed for the purpose of identifying those that appear appropriate for AI technology. It is hoped that the features identified in these applications will become apparent in other applications still to be identified at individual depot sites.

6.1 PROCEDURAL SUPPORT

In most large organizations, a comprehensive system exists for purchasing, tracking supplies and equipment, and accomplishing numerous other tasks, large and small, which must be accounted for under legal and/or financial regulations. Often, the larger the organization, the greater the number of rules and procedures which must be followed. Certainly, the United States Government is no exception to this rule.

In most cases, a repository of rules exists in the knowledge of financial or administrative experts or secretaries scattered throughout the organization. Several institutions have found it advantageous to collect such procedural information in a single knowledge-based system, the idea being that the user has only one place to go to get the answers to most procedural questions.

There are several significant advantages to having a procedural support expert system. First, having a single location to which users can go (or connect to from their PCs) for information about a variety of regulated procedures may be both cost effective and stress reducing. The user saves time and effort, as do the financial experts, managers, or administrators whose task it is to provide the answers. An expert system is available, theoretically, 24 hours a day. An expert system may also reduce costs by reducing the dissemination of inaccurate information and minimizing error. Finally, there is some evidence that in certain situations, people are more likely to ask an expert system for assistance than to disturb another person. Expert systems designed to support procedural applications offer considerable potential for improvements in efficiency.

AI is clearly the best technology for expert systems for a number of reasons. The task is knowledge intensive, requiring procedural knowledge and efficiency information from a number of sources, from secretaries and other staff members to voluminous documents and regulations. Expert systems allow for easy additions or alterations to reflect changes in the procedures captured by the system. The sophisticated user interface employed by expert systems allows information to pass both ways in a relatively natural form of exchange (see Figure 10).

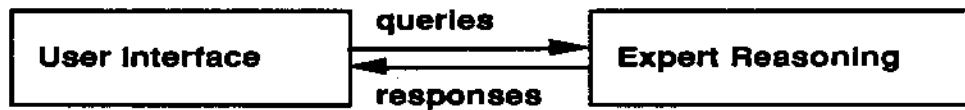


Figure 10. Expert System Interface

The user both asks questions of the expert system and responds to questions posed by the system. This ability to engage in dialogue makes expert system technology easy and comfortable for many people to use.

In most cases, the investment required to develop a procedural support system would not be great. Developing and installing such a system can easily be divided into increments; modules representing different sets of procedures may be built and tested separately and then added to the operational system. A system of this type could be built on a PC system, eliminating the large expenses involved in obtaining development hardware and software and making the system more widely accessible. Commercial expert system tools for PCs should be adequate for building the knowledge base and developing the complete expert system.

Several examples of similar applications, such as the IBM Capital Asset Expert System (CASES), exist in industry (Feigenbaum, McCorduck, and Nil,1988). IBM constructed CASES to simplify the process of disposing of or transferring machinery by providing engineers and managers with information about which forms are to be completed and how, which signatures are required, warnings and tips, departments and/or people to be notified, and other pertinent information. IBM users are reportedly pleased with the application because of the savings and convenience it provides and intend to construct more such systems.

6.2 SCHEDULING AND PLANNING

A variety of applications fall under the heading of resource scheduling and planning. These include applications designed to ensure efficient use of resources, including supplies, machinery, vehicles, and even human resources. Scheduling tasks may involve creating schedules to allow efficient use of frequently occupied machines, scheduling for maintenance and repair, and task scheduling for human resources. Expert systems can be designed to optimize the use of scheduled resources.

Planning systems are basically decision-support systems, assisting experts in predicting events and in coping with predicted outcomes. Expert systems may be useful in a number of depot-related planning activities, including decision support for capital purchases, supply requisitions, performance assessments, manpower assessments, hiring, plant expansion, or reorganization. The goal of an AI application in this domain is optimization. Decision making in the area of scheduling and planning is difficult because of the number of items that must be evaluated simultaneously. For example, when planning for purchase of large capital items, the decision maker must consider numerous factors, including the value of the product's use (its dispensability), the state of the equipment to be replaced, projected economic or strategic advantages of the new equipment, the projected rate of use of the new equipment, and the cost and quality of the equipment to be purchased. Including facts that are less directly relevant, such as trends in market pricing, can potentially improve the decision. The decision maker must also consider numerous constraints that may be imposed upon the decision, such as the current budget, the cost of storage (for items not immediately used), and availability of the product. With so many fragments of information to contend with, considerable expertise is required to make a decision in a complex organizational environment. And when considerable expertise is required, AI may help to improve both the quality and the consistency of the decisions.

Two primary AI technologies are commonly applied to problems of scheduling and planning: expert systems and model-based reasoning. Expert systems represent an appropriate technology because the task is exceedingly knowledge intensive, often in terms of heuristic knowledge, which humans use so profitably. It may be particularly important to capture the expertise of older, more experienced experts who are near retirement. This knowledge can be used both in the decision-making process itself and in tutoring less experienced personnel. Even when a task requires extensive knowledge in a complex domain, the innately modular nature of the expert system eases the burden of construction.

Model-based reasoning approaches are especially effective in addressing planning problems. These systems apply knowledge to produce simulation of a system, and valid simulations allow valid predictions. When plans are contingent upon future events, such predictive systems may be essential for choosing optimal courses of action.

Model-based applications are prevalent in industry. One such application is that of advising management on purchasing of new equipment. Commercial systems are currently on the market which are designed to support the decision process by evaluating proposals and generating alternatives. It is designed for a broad business environment and contains a general base of knowledge to which specific details must be added for each organization. A conceptually related expert system called COCOMO1, sold by Level Five Research, assists managers in estimating labor requirements, time schedules, and costs associated with proposed projects. The system estimates project productivity, providing a quantitative measure of the project's potential worth.

Another industrial example of an expert system--this one related to scheduling of resources--is The Boeing Company's SIMLAB, which generates schedules for the company's flight simulators, ensuring a high level of usefulness. A system constructed by the National Defense Research Establishment in Sweden has been designed to dynamically schedule actions in a plan-guided aircraft. In Finland, a knowledge-based system is being used to paginate newspapers. The number of planning and scheduling activities within any organization is virtually limitless, as is the number of potential AI applications to such tasks. Without doubt, there are many depot activities that represent appropriate applications for expert systems.

6.3 AUTOMATED INSPECTION TASKS

Automated inspection systems are designed to assist and replace humans in sensory and recognition tasks. These tasks are typically tedious, requiring only minimal skill to perform. Essentially, the function of a system designed for inspection is to receive input from sensors aimed at the target and to process the information.

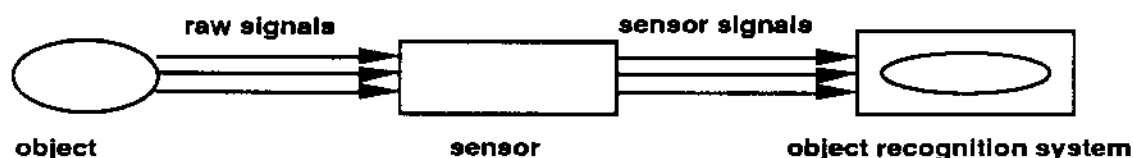


Figure 11. Perception and Recognition System

The output from the system is a categorization of the target: "good" or "bad" in the simplest case. Automated inspection systems could be particularly useful to the depot system for performing tasks such as inspection and automatic testing of electronic components, automated evaluation of nondestructive testing results, quality assurance of incoming parts, quality control of outgoing products, and X-ray screening of debris for selected objects (e.g., live ammunition).

Traditional pattern recognition techniques, in which each scene or set of sensory stimuli is matched with others in memory, are quite expensive in terms of computer time and memory. Historically, they have also been notoriously ineffective in classifying objects when even minor variations exist in parameters such as orientation or lighting. AI methods are now being applied to increase the accuracy of automated classification systems, at the same time increasing their operational speed.

Several AI techniques may conceivably be used, either alone or in combination, to perform an automated inspection function. Neural networks are especially well suited to simplified recognition tasks and have many advantages over traditional pattern recognizers, which rely on single-processor digital computers. Neural networks are fast, robust, and potentially tolerant to small variations in conditions, and they may be implemented with special-purpose array processors. Speed increases by a factor of 1000 have been reported. Even faster are networks implemented on a single chip. Chips with a relatively small number of interconnections for smaller problems have already been developed, and research is continuing on chips capable of holding much larger networks. Development of large-scale applications for neural networks, however, is being impeded by hardware difficulties, including the small number of interconnections available per chip and insufficient storage.

Lockheed has developed a probabilistic neural network (PNN) and used it for a number of tasks, including classification of seismic signatures. Such a network provides quantitative statistical analysis of a problem, as well as the probability of the correctness of a given solution. It notifies the user when there is insufficient information to provide a solution rather than providing inappropriate responses. The Pacific Northwest Laboratory (PNL) has applied neural networks to the classification of signatures for nondestructive testing. Oak Ridge National Laboratory has used them to perform inspection of printed material. Neural networks

are used in one of the New York airports for inspection of passenger luggage. A great deal of research has been performed on vision systems, much of it for the Department of Defense (DOD).

Model-based reasoning and expert systems represent two options for application to more complicated inspection tasks, including those for which there is little regularity in parameters (e.g., sensing angle and environmental conditions) and those for which criteria for acceptance vary widely. (In inspection of welds, for example, two distinct welds may be very different but equally adequate.)

Model-based reasoning systems essentially create models of the environmental factors affecting target viewing and compute what a given target's signature should be. The chief advantage of such a system is that given measurable environmental parameters, the system is very flexible. In effect, what can be modeled can be recognized. However, such systems are computationally expensive, and development of real-time model-based reasoning systems will depend on further technical advances in computation.

Expert systems operate using heuristics defined in terms of sensor output. The expert system weld inspector, for example, would identify certain characteristics of a weld as either good or bad. These characteristics would be recognized within image-by-image processing techniques. By collectively evaluating these pieces of information and applying rules, the expert system would evaluate the state of the weld. Like model-based systems, expert systems may also face time constraints, because each image must be processed and characteristics extracted from the observed image before input to the expert system takes place. Many factors, such as signature clutter, may affect the success rate of an expert system designed to perform automated inspection. Best results may be derived from combining available technologies.

Model-based reasoning and expert system technologies have been applied to complex tasks undertaken for the DOD by companies such as Martin Marietta in Minnesota.

6.4 PERSONNEL TRAINING (INSTRUCTION)

Shortages of critical expertise are forecast for the United States, both in the government and in private industry. Shortages imply competition to attract talent. To compete successfully

for expertise, the depot system may have to increase salaries and benefits, provide quicker advancements, and/or find other ways to induce promising talent to join and discourage capable employees from leaving. Because the government is often less flexible than the private sector in its ability to offer inducements, and because the federal deficit may decrease the amount of money available for such inducements, the government sector might well experience the brunt of a skills shortage.

AI may help to diminish the effects of personnel shortages by serving in the role of teacher. For example, AI methods can be used to provide comprehensive in-house training to unskilled or moderately skilled workers, with the goal of obtaining more expert performance; they can be used to train long-time employees in related job skills or new employees with little or no experience in a particular assignment. Technicians, diagnosticians, and mechanics would be likely consumers for such training products. For example, mechanics could be given online support and instruction for operating a new class of machinery; inexperienced employees could be taught by an expert system to perform and interpret diagnostics on particular types of equipment. In addition, AI systems can be used as skill repositories, storing knowledge of specific skills that are infrequently required (e.g., those required in dealing with a rarely encountered chemical compound or weapon system).

The task of instruction is particularly complex. A system designed to teach must have the ability to solve a wide variety of potential problems, because student questions may range from the simple to the obscure. Such a system must not only be able to perform problem analysis over the whole domain but must also be adept at explanation. Good solutions plus poor explanations equals a poor system. Knowledge-based systems are relatively proficient at describing their own process of reasoning, and the sophistication of their user interface makes the student/teacher exchange quite natural. Within a narrow domain of expertise, therefore, knowledge-based systems can be constructed to have the same strengths as a capable human instructor, if not the flexibility.

In addition, knowledge-based instructors have important advantages over human teachers. Knowledge-based systems can be on duty around the clock to field inquiries. They are consistent in their responses, whereas human instructors are likely to differ significantly in their approach to selected problems and therefore produce students who also exhibit this divergence. It has also been shown that students are actually more likely to approach the

expert system than a human instructor for help; trainees ask questions more frequently and are much more willing to ask a question again when the answer has been forgotten or misunderstood. The application of expert systems appears to offer potential advantages to all parties: the organization, the resident experts, and the trainees.

A major disadvantage to expert system technology is the inflexibility of the expertise residing in the system. Knowledge can be divided into the categories of domain knowledge and global (or general) knowledge. Expert systems are filled with domain knowledge, while human problem-solvers hold both domain knowledge and global knowledge (which may be imported from another domain to aid in solving a particular problem). The knowledge base of an expert system contains facts and rules about the problem domain, but if a question is posed that requires information beyond the scope of that domain, the system fails. Knowledge-based systems have been characterized as "idiot savants," providing excellent performance only for problems that do not require knowledge resources beyond a limited domain.

When the limitations of expert systems present serious difficulties, techniques other than knowledge-based systems can be used. Model-based systems include knowledge of how the domain behaves and why. Such systems can reason out difficult questions by observing the operation of their own internal models. These systems are said to have a "deep" knowledge and are designed to be resilient when confronted by questions requiring knowledge not immediately available.

Another AI technology that can contribute to instruction of employees is natural language processing, the analysis and comprehension of human speech by a machine. Though the technology is far from a complete understanding of human language, applications to specific domains with restricted vocabularies have been moderately successful. Natural language processing can improve the quality of understanding between the machine instructor and the trainee, allowing a natural and successful channel of communication to be developed between them. This technology can also free the trainee from the requirement of learning the system's language, thereby decreasing the overall instruction time.

Here, too, private industry has set the precedent in construction and implementation of artificially intelligent tutorial systems. One such system was built to train employees who operate the recovery boilers at paper plants. Another is used as a knowledge store for

information on how to handle and ship dangerous chemicals. Still another, at one of the nation's leading automotive companies, serves to train mechanics how to diagnose problems with a certain class of robots on the production floor. These and other successful examples illustrate the potential for development of AI applications for depot use.

6.5 DIAGNOSTIC SYSTEMS

Because industry depends so heavily on machines to perform manufacturing processes, diagnostics represents one of the most common and most successful applications for AI. Artificially intelligent diagnostic systems analyze and diagnose problems with machinery and equipment, such as motor vehicles, or with processes, such as maintenance lines. Such systems are typically designed to contend with problems after they have arisen, but they may also be designed for troubleshooting, which involves processing of functional information as warning signals so that problems may be avoided. Diagnostic systems have frequently been constructed to identify failed components or subsystems from the supply of diagnostic information, but they may also be constructed to identify deep or root causes of component or system failures.

There are many conceivable applications for diagnostic systems in a depot environment. Such systems can be constructed for many types of machines (e.g., tanks, missiles, computer hardware), as well as electronic circuit boards and devices. Electronic testing data can be fed directly into an expert system designed to diagnose faults and failures in these devices. Diagnostic systems can be constructed to identify more abstract problems within a software system or within a service line composed of robot workers. Essentially, a diagnostic system can be built for any process or machine with sufficient complexity.

There are numerous reasons to apply AI to diagnostics. As previously discussed, the shortage of skilled workers can be ameliorated by making available skilled professionals more productive. In many documented cases, expert diagnostic systems have been able to do this. By reducing the list of possible failures to those most likely to occur, such a system allows for fewer components and subsystems to be taken apart and tested. In this case, decreasing alternatives increases efficiency. Many diagnostic systems also decrease the time required for diagnosis, because a computer can carry out a reasoning process much faster than its human counterpart. Furthermore, although expert systems are often outperformed by the most expert

of experts, AI diagnostic expert assistants are often able to raise the collective performance of a group of workers because they are more effective than the average worker.

Many different AI technologies have been applied to the task of machine diagnosis. Expert systems have most often been used, perhaps most importantly because of their ability to capture and utilize experiential knowledge. Model-based reasoning systems have also proven to be effective, especially for providing a "root cause" analysis which not only identifies inoperable components but also the reasons behind their failure. Other AI systems that have been implemented to perform diagnosis include object-oriented systems, blackboard systems, and hybrid systems (i.e., combinations of AI technologies).

Examples of AI diagnostic systems abound. One system assists telephone companies in reducing cable failures by analyzing operational and repair data and prescribing preventative actions. It was designed to reduce the load on expert cable analysts who were often overburdened. Texas Instruments developed an expert system for its own use to perform a similar task for a certain type of reactor used in semiconductor fabrication. The Honeywell Corporation built an expert system to help field engineers diagnose and repair problems in air-conditioning systems. Examples of other systems, including an object-oriented fault tree models developed by NASA Ames and a hybrid expert system for monitoring and diagnosis of nuclear power plants built by the Oak Ridge National Laboratory, are in prevalent.

6.6 ENHANCEMENT OF ROBOTIC SYSTEM FLEXIBILITY

Robots are currently in place in many industries and are likely to become even more common within the work environment. Flexibility is a key issue in robotics, because flexible robots can do more tasks, spend less idle time waiting for assignments, and are therefore more efficient than single-purpose robots. For example, while a single-purpose robot may be designed to lift a particular type of crate, a more flexible robot may be capable of lifting a variety of items, large and small, of various shapes. Flexible robots may be provided with a variety of skills, such as welding and painting, and may be designed to perform traditionally human functions, such as making minor repairs on relatively delicate equipment.

Industry has begun to recognize the advantages of more flexible robots. The Japanese have made progress in developing flexible systems and currently have many more

multipurpose robots than the United States. The robotic systems in the United States can be augmented with the help of AI technologies designed to provide them with the following three components: enhanced planning abilities, enhanced perceptual abilities, and enhanced motor control.

No single AI method can achieve all of these goals. An AI system that provides increased flexibility for robots must, in all probability, be a hybrid system, with each component responsible for a certain function just as different parts of the human nervous system perform different functions.

Ideally, a robotic system should be able to generate simple plans and address minor contingencies autonomously; otherwise, the robot must be expressly programmed for every small task it may conceivably be asked to perform. This requires a great deal of a programmer's time and makes the system unable to perform when confronted with even the smallest change in circumstances. Autonomous robots, conversely, save time by reducing the amount of necessary human intervention. An autonomous spill-cleaning robot being developed at the University of Tennessee is an example of efforts along these lines.

Knowledge-based systems and model-based reasoning systems have been developed experimentally to accomplish this type of planning. Flexibility in planning must be considered as a matter of degree. Not every possible occurrence can be anticipated, but reasonably capable systems have been developed to contend with limited but functional environments. Planning of the type typically performed by these systems involves goal trees.

The second component of flexibility, enhanced perceptual abilities, is essential to provide robots with the capacity to respond to a dynamic environment. Sensor technology allows a robotic system to capture a rather complete "picture" of an environment, but analyzing this picture fully enough to allow planning for changes requires an AI component.

Though not all environments can be dealt with by any single system, some models, such as blackboard systems for vision, may be ideal for the multi-sensor vision required for many robotic functions. Problems such as scene analysis require combining diverse sources of knowledge (multiple sensors, internal knowledge, and input from other AI components). For example, a robot required to move warehouse items from one location to another may have to recognize the size and shape of the items to be moved, recall certain characteristics of them

from memory (e.g., fragility, weight), and adjust the amount of force used for grasping the item to the weight of the box. This sort of scene analysis is a common, if not completely defined, task in AI. Expert systems, model-based reasoning systems, and neural networks have each been applied at one level or another to the vision task.

Providing robotic systems with enhanced motor control requires a computational expense and precision that is difficult to attain with conventional means. AI attempts to augment the motor control of robotic systems and thereby make such systems capable of more human-like performance on certain chores. The two tasks involved in providing intelligent control for robotics systems are: 1) to provide contingency control to allow responses to unforeseen external or internal disturbances (e.g., the robot runs into a wall), and 2) to increase the precision of the robot's movements. Expert systems have been built to assist intelligent controllers in coping with unanticipated disturbances. However, they are not designed to increase the precision of the movements. On the other hand, neural networks seem to be well suited to the task of enhancing a robot's range of functions. Their parallel nature provides for rapid generation of control signals and, therefore, for more precise movement. The fault tolerance of artificial neural systems, which stems from the distribution of processing tasks among a large number of simple processing units, is also an important feature for robotic systems. Finally, the adaptive or learning ability of neural networks may eventually greatly simplify the task of programming the controller characteristics.

Creating more flexible robots is one of the most difficult applications for AI because it requires combining several methods and because the subtasks themselves are challenging. Few successful examples as yet exist in industry. However, the amount of research now being conducted in this area attests to a growing commitment to develop more flexible robotic systems. The Ford Motor Company, for example, has invested heavily in robotic systems and, more recently, in AI systems to augment them. It is expected that during the 1990s, implementation of more flexible robots will be both possible and profitable.

7.0 EVALUATION OF APPLICATIONS

In previous sections of this document, AI methods and general applications categories were highlighted. This section is concerned with more thorough identification and analysis of applications specific to the depot system. Applications selected for discussion are those most apt to benefit from increased efficiency and reliability and/or reduced costs.

The following subsections discuss specific depot-related tasks that may be served by technologies and the characteristics of a task that make it suitable for AI applications (or, conversely, that identify it as unsuitable.)

7.1 DEPOT-RELATED TASKS SUITABLE FOR AI APPLICATIONS

Of course, not every option can be explored. Those working within the depot system who have basic knowledge of the characteristics and capabilities of AI systems are probably the ones best able to identify appropriate applications. It is hoped that this document will help to supply interested readers with such functional knowledge.

Industrial applications are sometimes difficult to identify because of the complex nature of the tasks they address. Some industrial chores fall exclusively within the general application categories previously described, such as repair or diagnosis. But often, real-life applications are combinations of these basic categories and sometimes represent combinations of different technologies. For example, robots designed to load and unload munitions at a depot may require a vision system, perhaps using a neural network for recognition of physical objects and a model-based reasoning system to make sense of the scene as a whole. A rule-based system might be used to augment the controller of the machine, enabling it to contend with unforeseen and possibly dangerous situations. These may be united with the other components within the organizational structure of a blackboard system.

7.2 APPROPRIATENESS OF APPLICATIONS

Most successful AI applications in both industry and government consist of expert systems. Expert systems were the first AI technology to become commercially available. They

have been applied to numerous problems and undergone continuous evolution, and today they can be built upon many available retail expert system shells. But although expert systems have provided successful solutions to many problems, they are not appropriate for all problems. Also, development of expert systems can be relatively time consuming and expensive, thereby making them unsuitable solutions to many problems which they fit in theory. The following discussion will highlight the most salient points of general agreement among experts today concerning evaluation of expert systems applications.

Many factors play a role in determining whether or not an expert system should be implemented, and many of these are difficult to measure effectively. A quantitative analysis is usually not simple. Nevertheless, an analysis may be quite useful in encouraging evaluation of the features of a task before construction of the system begins, thus weeding out many projects that are not likely to succeed before time and money are spent. Furthermore, such an evaluation can encourage the consideration of many new applications that share the same characteristics as the application being considered.

The support the project receives before it is undertaken is an essential and often neglected consideration. To have good probability of success, the expert system application must have considerable support from the following three groups: management, consumers, and the expert or experts from whom the knowledge for the system is to be elicited.

Because of the time and cost involved in large development projects, an uncommitted management with overriding, short-term goals can make demands upon the system that the system cannot deliver, or at least make the project more difficult than necessary. To be sure that the project will be allotted the required time and other resources, management must have a realistic view of the project's goals and a supportive role in its development.

It is also advantageous for consumers of the end product to have an accurate and realistic appraisal of the system to be developed. They should reasonably expect that performance of the system will provide a payoff substantial enough to justify their commitment of time and effort, yet not expect results well beyond the actual performance of similar systems. In some cases, potential users may fear or resent an expert system as a replacement or a

competitor. Education, including realistic appraisal of the anticipated performance and role of the expert system within the work environment, can be very useful in generating the necessary support from this group.

Like users, experts may also feel challenged by the introduction of an automated system and therefore should be helped to understand that the tool is not a threat. An expert who is interested in the process of developing the system is likely to be much more useful than one who is not. Individual characteristics (e.g., the ability to communicate effectively or a tendency to pontificate) will also affect the process of eliciting knowledge from the expert. Because knowledge elicitation is often the most difficult task in building an expert system, considerable attention should be devoted to analyzing the expert or experts upon whom the system depends.

Another consideration that arises when the probability of success of an expert system project is to be estimated is the nature of the task itself. In analyzing the task, it is important to recognize that not all problems are created equal. There are several relevant features of the problem domain that require careful consideration: the expertise or knowledge requirements, the problem's level of difficulty, the measurability of generated solutions, and the likely evolution of the task.

Before undertaking development of an expert system, it may be useful to verify that the required expertise actually exists; i.e., that there is a man or woman who is considered to be an expert in this field. If there is such an individual or group of individuals, one should next consider the extent to which this knowledge is heuristic. Heuristics, as described earlier, are useful "rules of thumb," acquired by experts during months or years of experience, which when applied to a problem tend to generate solutions. For the problem to be appropriate for solution through AI technologies, creativity must not be relied on. Creative solutions require spontaneous generation of new rules and the unanticipated synthesis of existing knowledge. Expert systems are not capable of this type of response.

The level of difficulty of the problem also helps to determine whether expert system technology is appropriate to a given task. Tasks to which expert systems are to be applied must not be too large; large problems may require massive stores of information, much of it not clearly related or defined. When the rules required to connect this information reach the tens

of thousands, performance and reliability are likely to suffer significantly. Well-defined problems rather narrow in scope are much better candidates for expert system development. Also, the expert system ideally should not rely heavily on other areas of research, such as general natural language interfaces, which have their own problems and challenges. Adding further complexity to expert system implementation is likely to lead to unnecessary delays.

Just as tasks that are too difficult are inappropriate, so are tasks that are too easy. If a moderately skilled technician with some training can perform as well as a more experienced "expert," the task may be too easy to require the development of an expert system. The problem could probably be addressed through more conventional means. If an algorithm can be developed to solve the problem, then AI techniques, being generally less well understood, should not be used.

The third consideration for task evaluation is that the task should provide for measurability of generated solutions. There are two aspects to inventing a meaningful performance measure: 1) assessment of the solution itself and 2) assessment of the effectiveness of the expert systems solution relative to other methods (e.g., use of human experts).

A panel of experts should generally agree about the quality of a solution. In keeping with this restriction, problem domains that are too subjective should likely be discarded. The day-to-day performance of the expert system should also have a definite measure of success. Ideally, this measure should be defined before the system is implemented. Whether this measure be cost, efficiency, reliability, or some combination of these or other criteria, it should be devised to gage how adequately the system satisfies the ultimate consumer's needs. Initial generation of this measure should serve to shape the development of a useful system and encourage a focused and realistic evaluation of the expert system before it is begun.

It should be noted that 100% accuracy should not be a requirement of the system. Tasks for which absolute perfection are essential should be addressed using an algorithmic approach, which provides solutions that are completely defined.

The final feature to be discussed as an evaluation point concerns the likely evolution (i.e., longevity) of the task for which the expert system is to be built. First, the task should be one that is likely to be around for the foreseeable future. If the task is too short lived, then the

payoff may not be adequate to justify the expenditure of resources. It is reasonable to believe that bigger initial investments require longer periods of repayment. Secondly, if the task is likely to evolve and thereby alter the initial requirements of the system, then the task should ideally be modular in nature. Modularity allows for incremental growth, which may include the addition of subtasks or system alterations. Accommodating incremental growth means that the whole system will not have to be overhauled (usually at the cost of considerable time and expense) when changes are made. Another advantage of modularity accrues in the construction of the system: tasks that can be compartmentalized are more easily tested and evaluated, reducing costs and completion time.

8.0 REFERENCES

- Andress, K., and A. Kak. 1988. "Evidence Accumulation & Flow of Control." AI Magazine, 9(2):75-94.
- Barr, A., P. R. Cohen, and E. A. Feigenbaum. 1989. The Handbook of Artificial Intelligence, Volume IV. Addison-Wesley Publishers, Inc., Reading, Massachusetts.
- Bobrow, D. G. 1985. "Qualitative Reasoning About Physical Systems: An Introduction," in Qualitative Reasoning About Physical Systems, D. G. Bobrow, ed. MIT Press, Cambridge, Massachusetts.
- Butler, C. W., E. D. Hodil, and G. L. Richardson. 1988. "Build Knowledge-Based Systems with Procedural Languages," IEEE Expert, IEEE Expert 3(2):51-58.
- Charniak, E., and D. McDermott. 1985. Introduction to Artificial Intelligence, Addison-Wesley Publishers, Inc., Reading, Massachusetts.
- Cohen, P., and A. Howe. 1988. "How Evaluation Guides AI Research." AI Magazine, 9(4):35-43.
- de Kleer, J., and J. S. Brown. 1985. "A Qualitative Physics Based on Confluences," in Qualitative Reasoning About Physical Systems, Daniel G. Bobrow, ed. MIT Press, Cambridge, Massachusetts.
- Feigenbaum, E., P. McCorduck, and H. P. Nii. 1988. The Rise of the Expert Company. Times Books, New York, New York.
- Franklin, J., and K. Karna. 1989. "Shaping Knowledge Within Artifacts: Expert Systems in Government." IEEE Expert, 4(1):3-5.
- Grefenstette, J. J. 1986. "Optimization of Control Parameters for Genetic Algorithms." IEEE Transactions of Systems, Man, and Cybernetics SMC-16(1):122-128.
- Harmon, P., and D. King. 1985. Expert Systems: Artificial Intelligence in Business. John Wiley & Sons, Inc., New York.
- Holland, J. H., K. J. Holyoak, R. E. Nisbett, and P. R. Thagard. 1986. Induction: Processes of Inference, Learning and Discovery. MIT Press, Cambridge, Massachusetts.
- Lippmann, R. P. 1987. "An Introduction to Computing with Neural Nets." IEEE ASSP Magazine 4(2):4-22.

Melton, R. B., D. M. Devaney, M. A. Whiting, and S. C. Laufmann. 1989. A Methodology for Evaluating Potential KBS Applications. PNL-6919, Pacific Northwest Laboratory, Richland, Washington.

McClelland, J. L., D. E. Rummelhard, and PDP Research Group. 1986. Parallel Distributed Processing Explorations in the Microstructure of Cognition. MIT Press, Cambridge, Massachusetts.

Nii, P. H. 1986. "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architecture." The AI Magazine 7(2):38-53.

Pascoe, G. A. 1986. "Elements of Object-Oriented Programming." BYTE Magazine 11(8):139-144.

Srihari, S. N., C. H. Wang, P. W. Palumbo, and J. J. Hull. 1987. "Recognizing Address Blocks on Mail Pieces Specialized Tools and Problem Solving Architecture." AI Magazine 8(4):25-40.

Slagle, J., and M. Wick. 1988. "A Method for Evaluating Candidate Expert System Applications." AI Magazine 9(4):44-53.

Spang, S., Ed. 1986. "AI Hardware: Part II." The Spang Robinson Report 2(2).

Spang, S., Ed. 1986. "AI and Parallel Processing: Part II." The Spang Robinson Report 2(7).

Winston, P. H. 1984. Artificial Intelligence. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

DISTRIBUTION

<u>No. of Copies</u>		<u>No of. Copies</u>
	<u>OFFSITE</u>	Commander Red River Army Depot ATTN: SDSRR-F Texarkana, TX 75507-5000
2	DOE/Office of Scientific and Technical Information	
25	T. Erwin U.S. Army Depot System Command ATTN: AMSDS-VV, Bldg. 523 Chambersburg, PA 17201-4170	Commander Sacramento Army Depot ATTN: SDGGA-FM-M (Paul Reynard) Sacramento, CA 95813-5010
	Commander Anniston Army Depot ATTN: SDSAN-DEL-R2K (Mike Trowse) Anniston, AL 36201-5001	Commander Seneca Army Depot ATTN: SDSSE-FM (John Frank) Romulus, NY 14541-5001
	Commander Corpus Christi Army Depot ATTN: SDSCC-GM (Jack Brooks) Corpus Christi, TX 78419-6000	Commander Sierra Army Depot ATTN: SDSSI-FMO (Rod Colvin) Herlong, CA 96113-5000
	Commander Letterkenny Army Depot ATTN: SDSLE-CES (Lynn Witthoeft) Chambersburg, PA 17201-4150	Commander Tobyhanna Army Depot ATTN: SDSTO-FM (Joe Kuligowski) Tobyhanna, PA 18466-5000
	Commander Lexington Blue Grass Army Depot ATTN: SDSLB-RMI (Joe Dennis) Lexington, KY 40511-5001	Commander Tooele Army Depot ATTN: SDSTE-FMM (Chris Tillman) Tooele, UT 84974-5000
	Commander New Cumberland Army Depot ATTN: SDSNC-R (Bob Graham) New Cumberland, PA 17070-5000	Commander Mainz Army Depot ATTN: SDSMZ-FM (Karl Thuorkauf) APO New York, NY 09185

No. of
Copies

Commander
DSAFE
ATTN: SDSFE-TOM (Dr. Gil Mezz)
Camp Market
APO San Francisco, CA 96483-0443

ONSITE

31 Pacific Northwest Laboratory

R. E. Rhoads (25)
Publishing Coordination
Technical Report Files (5)