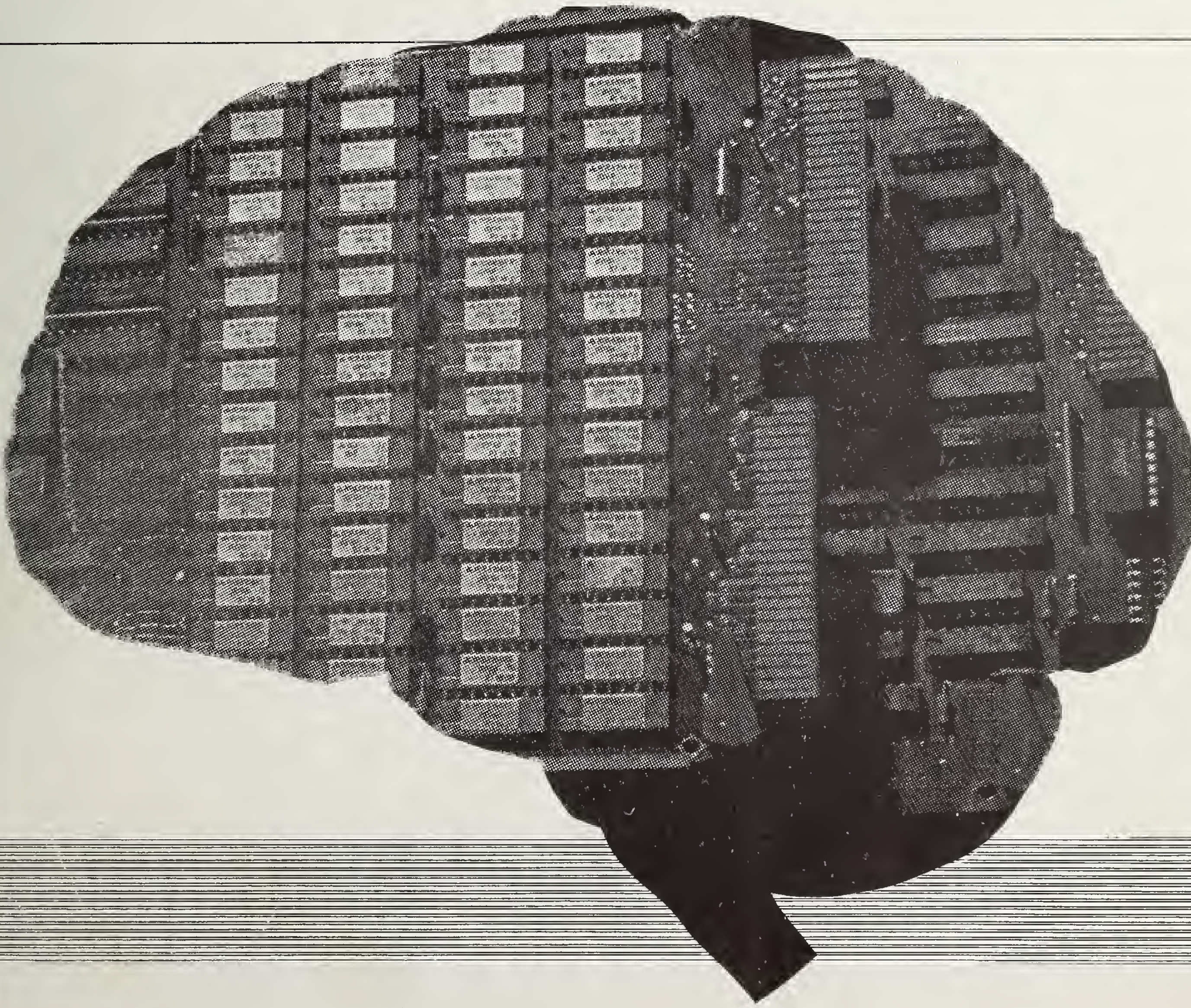
U.S. Department of Commerce
National Bureau of Standards

NBSIR 82-2505

AN OVERVIEW OF EXPERT SYSTEMS

May 1982



Prepared for

National Aeronautics and Space Administration Headquarters
Washington, D.C. 20546

NBSIR 82-2505

AN OVERVIEW OF EXPERT SYSTEMS

William B. Gevarter

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Manufacturing Engineering
Industrial Systems Division
Metrology Building, Room A127
Washington, DC 20234

May 1982

Prepared for:
National Aeronautics and Space Administration
Headquarters
Washington, DC 20546

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Preface

Expert systems is probably the "hottest" topic in Artificial Intelligence (AI) today. In the past, in trying to find solutions to problems, AI researchers tended to rely on search techniques or computational logic. These techniques were successfully used to solve elementary or toy problems or very well structured problems such as games. However, real complex problems are prone to have the characteristic that their search space tends to expand exponentially with the number of parameters involved. For such problems, these older techniques have generally proved to be inadequate and a new approach was needed. This new approach emphasized knowledge rather than search and has led to the field of Knowledge Engineering and Expert Systems.

This report provides a current overview of Expert Systems -- what it is, techniques used, existing systems, applications, who is doing it, who is funding it, the state-of-the-art, research requirements and future trends and opportunities.

This report is in support of the more general NBS/NASA report, "An Overview of Artificial Intelligence and Robotics."

Acknowledgements

I wish to thank those people at Stanford University, XEROX PARC, MIT and elsewhere who have been instrumental in developing the knowledge engineering field, and have contributed time and source material to help make this report possible. I particularly would like to thank Margie Johnson who has done a heroic job typing this series and facilitating their publication.

TABLE OF CONTENTS

List of Tables

List of Figures

I. Introduction

In the 70's, it became apparent to the AI community, that search strategies alone, even augmented by heuristic* evaluation functions, were often inadequate to solve real world problems. The complexity of these problems were usually such that (without incorporating substantially more problem knowledge than had heretofore been brought to bear) either a combinatorial explosion occurred that defied reasonable search times, or that the ability to generate a suitable search space did not exist. In fact, it became apparent that for many problems, that expert domain knowledge was even more important than the search strategy (or inference procedure). This realization led to the field of "Knowledge Engineering," which focuses on ways to bring expert knowledge to bear in problem solving.** The resultant expert systems technology, limited to academic laboratories in the 70's, is now becoming cost-effective and is beginning to enter into commercial applications.

---

*Heuristics are "rules of thumb," knowledge or other techniques that can be used to help guide search.

**One important aspect of the knowledged-based approach is that the combinatorial complexity associated with real-world problems is mitigated by the more powerful focussing of the search that can be obtained with rule-based heuristics usually used in expert systems as opposed to the numerical heuristics (evaluation functions) used in classical search techniques. In other words, the rule-based system is able to reason about its own search effort, in addition to reasoning about the problem domain. (Of course, this also implies that the search strategy is incomplete. Solutions may be missed, and an entire search may fail even when there is a solution "within reach" in the problem space defined by the domain.)

II. What is an Expert System?

Feigenbaum, a pioneer in expert systems, (1982, p. 1) states:

> An "expert system" is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field.
>
> The knowledge of an expert system consists of facts and heuristics. The "facts" constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in a field. The "heuristics" are mostly private, little-discussed rules of good judgment (rules of plausible reasoning, rules of good guessing) that characterize expert-level decision making in the field. The performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses.

III. The Basic Structure of an Expert System

An expert system consists of:

1) a knowledge base (or knowledge source) of domain facts and heuristics associated with the problem;
2) an inference procedure (or control structure) for utilizing the knowledge base in the solution of the problem;
3) a working memory - "global data base" - for keeping track of the problem status, the input data for the particular problem, and the relevant history of what has thus far been done.

A human "domain expert" usually collaborates to help develop the knowledge base. Once the system has been developed, in addition to solving problems, it can also be used to help instruct others in developing their own expertise.

Thus, Michie (1980, pp. 3-5) observes:

...that there are three different user-modes for an expert system in contrast to the single mode (getting answers to problems) characteristic of the more familiar type of computing:

(1) getting answers to problems -- user as client;

(2) improving or increasing the system's knowledge -- user as tutor;

(3) harvesting the knowledge base for human use -- user as pupil.

Users of an expert system in mode (2) are known as "domain specialists." It is not possible to build an expert system without one...

An expert systems acts as a systematizing repository over time of the knowledge accumulated by many specialists of diverse experience. Hence, it can and does ultimately attain a level of consultant expertise exceeding* that of any single one of its "tutors."

It is usual to have a natural language interface to facilitate the use of the system in all three modes. Normally, an explanation module is also included, allowing the user to challenge and examine the reasoning process underlying the system's answers. Figure 1 diagrams a typical (though somewhat idealized) expert system. When the domain knowledge is stored as production rules, the knowledge base is often referred to as the "rule base," and the inference engine the "rule interpreter."

An expert system differs from more convential computer programs in several important respects. Duda (1981, p. 242) observes that, in an expert system, "...there is a clear separation of general knowledge about the problem (the rules

---

*There are not yet many examples of expert systems whose performance consistantly surpasses that of an expert. And currently, there are even fewer examples of expert systems that use knowledge from a group of experts and integrate it effectively. However the promise is there.

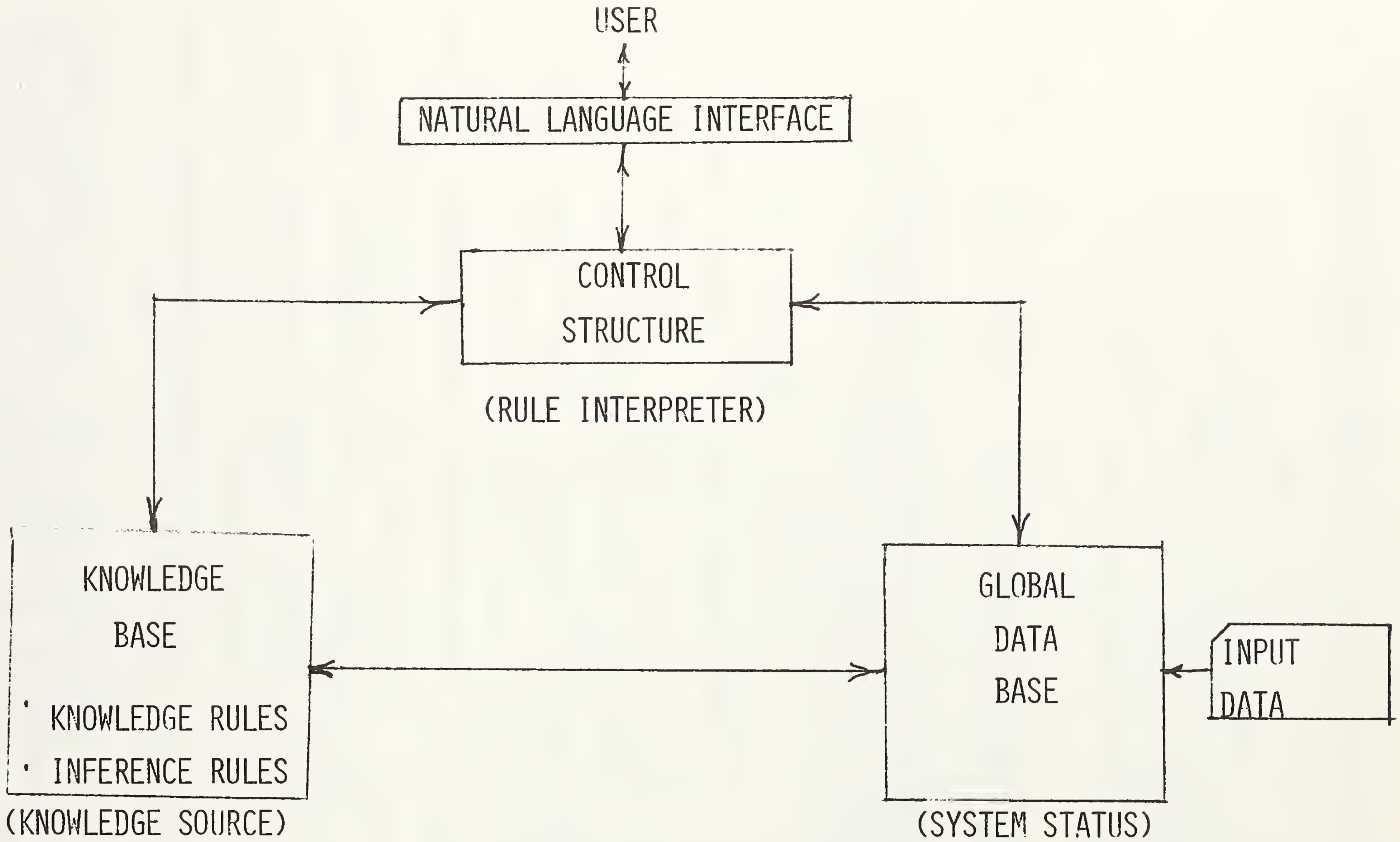


FIGURE 1

BASIC STRUCTURE OF AN EXPERT SYSTEM

forming a knowledge base) from information about the current problem (the input data) and methods for applying the general knowledge to the problem (the rule interpreter)." In a conventional computer program, knowledge pertinent to the problem and methods for utilizing this knowledge are all intermixed, so that it is difficult to change the program. In an expert system, "...the program itself is only an interpreter (or general reasoning mechanism) and [ideally] the system can be changed by simply adding or substracting rules in the knowledge base."

IV. The Knowledge Base

The most popular approach to representing the domain knowledge needed for an expert system is by production rules (also referred to as "SITUATION-ACTION rules" or "IF-THEN rules"). Thus, often a knowledge base is made up mostly of rules which are invoked by pattern matching with features of the task environment as they currently appear in the global data base.

The rules in a knowledge base represent the domain facts and heuristics - rules of good judgment of actions to take when specific situations arise. The power of the expert system lies in the specific knowledge of the problem domain, with potentially the most powerful systems being the ones containing the most knowledge.

Duda (1981, p. 242) states:

> Most existing rule-based systems contain hundreds of rules, usually obtained by interviewing experts for weeks or months... In any system, the rules become connected to each other by association linkages to form rule networks. Once assembled, such networks can represent a substantial body of knowledge....

An expert usually has many judgmental or empirical rules, for which there is incomplete support from the available evidence. In such cases, one approach is to attach numerical values (certainty factors) to each rule to indicate the degree of certainty associated with that rule. (In expert system operation, these certainty values are combined with each other and the certainty of the problem data, to arrive at a certainty value for the final solution.)

Michie (1980, p. 6) indicates that the cognitive strategies of human experts in more complex domains are based "...not on elaborate calculations, but on the mental storage and use of large incremental catalogs of pattern-based rules." Thus, human chess masters may be able to acquire, organize and utilize as much as 50,000 pattern-based rules in achieving their remarkable performance. Michie (p. 20-21) indicates that such rules are so powerful that only some 30 rules are needed for expert system performance for a chess subdomain such as King and Knight against King and Rook, which has a problem space size of roughly 2,000,000 configurations. He further observed for chess that the number of rules required grows slowly relative to the increase in domain complexity. Thus, in chess and other complex domains (such as industrial routing and scheduling) it appears that well-chosen pattern sets may maintain control over otherwise intractable explosions of combinatorial complexity.

V. The Inference Engine

> The problem-solving paradigm, and its methods, organizes and controls the steps taken to solve the problem. One commonplace but powerful paradigm involves the chaining of IF-THEN rules to form a line of reasoning. If the chaining starts from a set of conditions and moves toward some (possibly remote) conclusion, the method is called forward chaining. If the conclusion is known (e.g., it is a goal to be achieved), but the path to that conclusion is not known, then working backwards is called for, and the method is backward chaining. (Heuristic Programming Project, 1980, p. 6)

The problem with forward chaining, without appropriate heuristics for pruning, is that you would derive everything possible whether you needed it or not. Backward chaining works from goals to subgoals (by using the action side of rules to deduce the condition side of the rules). The problem here, again without appropriate heuristics for guidance, is the handling of conjunctive subgoals. In general to attack a conjunction, one must find a case where all interacting subgoals are satisfied, a search for which can often result in a combinatorial explosion of possibilities. Thus appropriate domain heuristics and suitable inference schemes and architectures must be found for each type of problem to achieve an efficient and effective expert system.

> The knowledge of a task domain guides the problem-solving steps taken. Sometimes the knowledge is quite abstract--for example, a symbolic model of "how things work" in the domain. Inference that proceeds from the model's abstractions to more detailed (less abstract) statements is called model-driven inference. Always when one is moving from more abstract symbolic statements to less abstract statements, one is generating expectations, and the problem-solving behavior is termed expectation driven. Often in problem solving, however, one is working "upwards" from the details or the specific problem data to the higher levels of abstraction (i.e., in the direction of "what it all means"). Steps in this direction are call data driven. If you choose your next step either on the basis of some new data or on the basis of the last

problem-solving step taken, you are responding to events, and the activity is called <u>event driven</u>. (Heuristic Programming Project 1980, p. 6).

As indicated earlier, an expert system consists of three major components, a set of rules, a global data base and a rule interpreter. The rules are actuated by patterns, (which match the IF sides of the rules) in the global data base. The application of the rule changes the system status and therefore the data base, enabling some rules and disabling others. The rule interpreter uses a control strategy for finding the enabled rules and deciding which rule to apply. The basic control strategies used may be top down (goal driven), bottomup (data driven), or a combination of the two that uses a relaxation-like convergence process to join these opposite lines of reasoning together at some intermediate point to yield a problem solution.

VI. Uses of Expert Systems

The uses of expert systems are virtually limitless. They can be used to:

- diagnose
- monitor
- analyze
- interpret
- consult
- plan
- design
- instruct
- explain
- learn
- conceptualize

Thus they are applicable to:

- Mission planning, monitoring, tracking and control
- Communication
- Signal analysis
- Command and control
- Intelligence analysis
- Targeting
- Construction and manufacturing
  - design, planning, scheduling, control
- Education
  - instruction, testing, diagnosis
- Equipment
  - design, monitoring, diagnosis, maintenance, repair, operation, instruction

- Image Analysis and Interpretation
- Professions (law, medicine, engineering, accounting, law enforcement)
  - Consulting, instruction, interpretation, analysis
- Software
  - Specification, design, verification, maintenance, instruction
- Weapon Systems
  - Target identification, electronic warfare, adaptive control.

VII. Architecture of Expert Systems

A. Introduction

One way to classify expert systems is by function (e.g. diagnosis, planning, etc). However, examination of existing expert systems indicate that there is little commonality in detailed system architecture that can be detected from this classification.

A more fruitful approach appears to be to look at problem complexity and problem structure and deduce what data and control structures might be appropriate to handle these factors.

The Knowledge Engineering community has evolved a number of techniques which can be utilized in devising suitable expert system architectures. These techniques* are described in the following portions of this section.

The use of these techniques in existing expert systems is illustrated in Table 1**. Table 1 describes the basic approach taken by each of these expert systems and indicates how the approach translates into key elements of the Knowledge Base, Global Data Base and Control Structure. A listing of the systems in Table 1, together with an indication of their basic control structures, is given in Table 2.

Table 2 represents the expert system control structures in terms of the search direction, the control techniques utilized and the search space transformations employed. The approaches

---

*This chapter is largely derived from information contained in the excellent tutorial by Stefik et al. (1982).

**Tables 1-1 to 1-4 are shown on the following pages. Table 1-5 to 1-17 are at the back of this report.

TABLE 1-1

Characteristics of Example Expert Systems

SYSTEM: DENDRAL

INSTITUTION: Standford University

AUTHORS: Feigenbaum & Lederberg

FUNCTION: Data Interpretation

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Generate plausible structural representations of organic molecules from mass spectrogram data	1. Derive constraints from the data. 2. Generate candidate structures 3. Predict mass spectrographs for candidates 4. Compare with data	Rules for deriving constraints on molecular structure from experimental data Procedure for generating candidate structures to satisfy constraints Rules for predicting spectrographs from structures	Mass spectrogram data Constraints Candidate structures	Forward chaining Plan, generate and test.

TABLE 1-2

Characteristics of Example Expert Systems

SYSTEM: AM

INSTITUTION: Stanford University

AUTHORS: Lenat

FUNCTION: Concept Formation

		KEY ELEMENTS OF		
PURPOSE	APPROACH	KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Discovery of mathematical concepts	Start with elementary ideas in set theory. Search a space of possible conjectures that can be generated from these elementary ideas. Choose the most interesting conjectures and pursue that line of reasoning.	Elementary ideas in finite set theory. Heuristics for generating new mathematical concepts by combining elementary ideas. Heuristics of "interestingness" for discarding bad ideas.	Plausible candidate concepts.	Plan, generate, and test.

TABLE 1-3

Characteristics of Example Expert Systems

SYSTEM: R1

INSTITUTION: CMU

AUTHORS: McDermott

FUNCTION: Design

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Configure VAX computer systems (from a customer's order of components).	Break problem up into the following ordered subtasks: 1. Correct mistakes in order. 2. Put components into CPU cabinets. 3. Put boxes into unibus cabinets and put components in boxes. 4. Put panels in unibus cabinets. 5. Lay out system on floor. 6. Do the cabling. Solve each subtask and move on to the next one in the fixed order.	Properties of (roughly 400) VAX components. Rules for determining when to move to next subtask based on system state. Rules for carrying out subtasks (to extend partial configuration). (Approximately 800 rules total)	Customer order. Current task. Partial configuration (System state).	"MATCH" (data driven) (no backtracking)

SYSTEM: MYCIN

INSTITUTION: Stanford University

AUTHORS: Shortliffe

FUNCTION: Diagnosis

TABLE 1-4

Characteristics of Example Expert Systems

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Diagnosis of bacterial infections and recommendations for antibiotic therapy.	Represent expert judgmental reasoning as condition-conclusion rules together with the expert's "certainty" estimate for each rule.	Rules linking patient data to infection hypotheses.	Patient history and diagnostic tests.	Backward chaining thru the rules.
	Chain backwards from hypothesized diagnoses to see if the evidence supports it.	Rules for combining certainty factors.	Current hypothesis	Exhaustive search.
		Rules for treatment.	Status.	
	Exhaustively evaluate all hypotheses.		Conclusions reached thus far, and rule numbers justifying them.	
	Match treatments to all diagnoses which have high certainty values.			

Table 2

Characteristics of Systems in Table 1

			Control Structure															
			Search Direction				Control							Search Space Transformations				
SYSTEM	FUNCTION	DOMAIN	Forward	Backward	Forward and Backward	Event Driven	Exhaustive Search	Generate and Test	Guessing	Relevent Backtracking	Least Committment	Multilines of Reasoning	Network Editor	Multiple Models	Break into Sub-Problems	Hierarchical Refinement	Hierarchical Resolution	Meta Rules
MYCIN	Diagnosis	Medicine		x			x											
DENDRAL	Data Interpr.	Chemistry	x					x										
EL	Analysis	Elec. Circuits	x						x	x								
GUIDON	C.A.I.	Medicine				x												
KAS	Knowl. Acquis.	Geology	x										x					
META-DENDRAL	Learning	Chemistry	x					x										
AM	Concept Formation	Math	x					x										
VM	Monitoring	Medicine				x	x											
GA1	Data Interpr.	Chemistry	x					x										
R1	Design	Computers	x				x								x			
ABSTRIPS	Planning	Robots		x												x		
NOAH	Planning	Robots		x							x				x			
MOLGEN	Design	Genetics			x				x	x	x				x	x		x
SYN	Design	Elec. Circuits	x											x				
HERARSAY II	Signal Interpr.	Speech Unders.			x						x	x					x	
HARPY	"	"	x															
CRYSALIS	Data Interpr.	Crystallography				x		x										x

used in the various expert systems are different implementations of two basic ideas for overcoming the combinatorial explosion associated with search in real complex problems. These two ideas are:

(1) Find ways to efficiently search a space,

(2) Find ways to transform a large search space into smaller manageable chunks that can be searched efficiently.

It will be observed from Table 2 that there is little architectural commonality based either on function or domain of expertise. Instead, expert system design may best be considered as an art form, like custom home architecture, in which the chosen design can be implemented using the collection of techniques discussed below.

B. <u>Choice of Solution Direction</u>

1. <u>Forward Chaining</u>

When data or basic ideas are a starting point, forward chaining is a natural direction for problem solving. It has been used in expert systems for data analysis, design, diagnosis, and concept formation.

2. <u>Background Chaining</u>

This approach is applicable when a goal or a hypotheses is a starting point. Expert system examples include those used for diagnosis and planning.

3. <u>Forward and Backward Processing Combined</u>

When the search space is large, one approach is to search both from the initial state and from the goal or hypothesis state and utilize a relaxation type approach to match the solutions at an intermediate point. This approach is also useful when the search space can be divided hierarchically, so both a bottom up and top down search can be appropriately combined. Such a combined search is particularly applicable to complex problems incorporating uncertainties, such as speech understanding as exemplified in HEARSAY II.

4. Event Driven

This problem solving direction is similar to forward chaining except that the data or situation is evolving over time. In this case the next step is chosen either on the basis of new data or in response to a changed situation resulting from the last problem solving step taken. This event driven approach is appropriate for real-time operations, such as monitoring or control, and is also applicable to many planning problems.

C. Reasoning in the Presence of Uncertainty

In many cases, we must deal with uncertainty in data or in knowledge. Diagnosis and data analysis are typical examples.

1. Numeric Procedures

Numeric procedures have been devised to handle approximations by combining evidence. MYCIN utilizes "certainty factors" (related to probabilities) which use the range of 0 to 1 to indicate the strength of the evidence. Fuzzy set theory, based on possibilities, can also be utilized.

2. Belief Revision or "Truth Maintenance"

Often, beliefs are formed or lines of reasoning are developed based on partial or errorful information. When contradictions occur, the incorrect beliefs or lines of reasoning causing the contradictions, and all wrong conclusions resulting from them, must be retracted. To enable this, a data-base record of beliefs and their justifications must be maintained. Using this approach, truth maintenance techniques can exploit redundancies in experimental data to increase system reliability.

D. Searching a Small Search Space

Many straightforward problems in areas such as design, diagnosis and analysis have small search spaces, either because 1) the problem is small or 2) the problem can be broken up into small independent subproblems. Often a single line of reasoning is sufficient and so backtracking is not required. In such cases, the direct approach of exhaustive search can be appropriate, as was used in MYCIN and R1.

E. Techniques for Searching a Large Search Space

1. Hierarchical Generate and Test

State space search is often formulated as "generate and test" - reasoning by elimination. In this approach, the system generates possible solutions and a tester prunes those solutions that fail to meet appropriate criteria. Such exhaustive reasoning by elimination can be appropriate for small search spaces, but for large search spaces more powerful technique are needed. A "hierarchical generate and test" approach can be very effective if means are available for evaluating candidate solutions that are only partially specified. In these cases, early pruning of whole branches (representing entire classes of solutions associated with these partial specifications) is possible, massively reducing the search required.

"Hierarchical generate and test" is appropriate for many large data interpretation and diagnosis problems, for which all solutions are desired, providing a generator can be devised that can partition the solution space in ways that allow for early pruning.

2. <u>Dependency-Directed Backtracking</u>

In the "generate and test" approach, when a line of reasoning fails and must be retracted, one approach is to backtrack to the most recent choice point (chronological backtracking). However, it is often much more efficient to trace errors and inconsistencies back to the inferential steps that created them, using dependency records as is done in MOLGEN. Backtracking that is based on dependencies and determines what to invalidate is called dependency-directed (or relevant) backtracking.

3. <u>Multiple Lines of Reasoning</u>

This approach can be used to broaden the coverage of an incomplete search. In this case, search programs that have fallible evaluators can decrease the chances of discarding a good solution from weak evidence by carrying a limited number of solutions in parallel, until which of the solutions is best is clarified.

F. Methods for Handling a Large Search Space by Transforming the Space

1. Breaking the Problem Down Into Subproblems

a. Non-Interacting Subproblems

This approach (yielding smaller search spaces) is applicable for problems in which a number of non-interacting tasks have to be done to achieve a goal. Unfortunately, few real world problems of any magnitude fall into this class.

b. Interacting Subproblems

For most complex problems that can be broken up into subproblems, it has been found that the subproblems interact so that valid solutions cannot be found independently. However, to take advantage of the smaller search spaces associated with this approach, a number of techniques have been devised to deal with these interactions.

(1) Find a Fixed Sequence of Subproblems So That No Interactions Occur

Sometimes it is possible to find an ordered partioning so that no interactions occur. The R1 system (see Table 1-3) for configuring VAX computers successfully takes this approach.

(2) Least Commitment

This technique coordinates decision-making with the availability of information and moves the focus of problem-solving activity among the available subproblems. Decisions are not made arbitrarily or prematurely, but are postponed until there is enough information. In planning problems this is exemplified by methods that assign a partial ordering of operators in each subproblem and only complete the ordering when sufficient information on the interactions of the subproblems is developed.

(3) Constraint Proprogation

Another approach (used by MOLGEN) is to represent the interaction between the subproblems as constraints. Constraints can be viewed as partial descriptions of entities, or as relationships (subgoals) that must be satisfied. Constraint proprogation is a mechanism for moving information between subproblems. By introducing constraints instead of choosing particular values, a problem solver is able to pursue a least commitment style of problem solving.

(4) Guessing or Plausible Reasoning

Guessing is an inherent part of heuristic search, but is particularly important in working with interacting subproblems. For instance, in the least commitment approach the solution process must come to a halt when it has insufficient information for deciding between competing choices. In such cases, heuristic guessing is needed to carry the solution process along. If the guesses are wrong, then dependency-directed backtacking can be used to efficiently recover from them. EL and MOLGEN take this approach.

2. Hierarchical Refinement into Increasingly Elaborate Spaces -- Top Down Refinement

Often, the most important aspects of a problem can be abstracted and a high level solution developed. This solution can then be iteratively refined, successively including more details. An example is to initially plan a trip using a reduced scale map to locate the main highways, and then use more detailed maps to refine the plan. This technique has many applications as the top level search space is suitably small. The resulting high level solution constrains the search to a small portion of the search space at the next lower level, so that at each level the solution can readily be found. This procedure is an important technique for preventing combinatorial explosions in searching for a solution.

3. <u>Hierarchical Resolution into Contributing Sub-Spaces</u>

Certain problems can have their solution space hierarchically resolved into contributing subspaces in which the elements of the higher level spaces are composed of elements from the lower spaces. Thus, in speech understanding, words would be composed of syllables, phrases of words, and sentences of phrases. The resulting heterogenous subspaces are fundamentally different from the top level solution space. However the solution candidates at each level are useful for restricting the range of search at the adjacent levels, again acting as an important restraint on combinatorial explosion. Another example of a possible hierarchical resolution is in electrical equipment design where subcomponents contribute to the black box level, which in turn contribute to the system level. Similarly, examples can be found in architecture, and in spacecraft and aircraft design.

G. Methods for Handling a Large Search Space by Developing Alternative or Additional Spaces

1. Employing Multiple Models

Sometimes the search for a solution utilizing a single model is very difficult. The use of alternative models for either the whole or part of the problem may greatly simplify the search. The SYN program is a good example of combining the strengths of multiple models by employing equivalent forms of electrical circuits.

2. Meta Reasoning

It is possible to add additional layers of spaces to a search space to help decide what to do next. These can be thought of as strategy and tactical layers in which meta problem solvers choose among several potential methods for deciding what to do next at the problem level. The strategy, focusing and scheduling meta rules used in CRYSALIS and the use of a strategy space in MOLGEN fall into this category.

H. Dealing with Time

Little has been done in the way of expert systems that deal with time explicitly. The following are approaches to dealing with time in terms of time intervals.

1. Situational Calculus

Situational calculus was an early approach by McCarthy and Hayes (1969) for representing sequences of actions and their effects. It uses the concept of "situations" which change when sufficient actions have taken place, or when new data indicates a situational shift is appropriate. Situations determine the context for actions and, through the use of "frames,"* can indicate what changes and what remains the same when an action takes place. VM uses the situation approach for monitoring patient breathing.

2. Planning with Time Constraints

NOAH was an early parallel planner which dealt with interacting subgoals. The method of least commitment and backward chaining initially produced a partial ordering of operators for each plan. When interference between subgoal plans was observed, the planner adjusted the ordering of the operators to resolve the interference to produce a final parallel plan with time ordered operators. DEVISER (Vere, 1981) is a recent derivative of NOAH which extends this parallel planning approach to treat goals with time constraints and durations. The principal output of DEVISER is a partially ordered network of parallel activities for use in planning a spacecraft's actions during a planetary flyby.

---

*A frame is a data structure for describing a stereotyped situation.

VIII. Existing Expert Systems

Table 3 is a list, classified by function and domain of use, of most of the existing major expert systems. It will be observed that there is a predominance of systems in the Medical and Chemistry domains following from the pioneering efforts at Stanford University. From the list, it is also apparent that Stanford University dominates in number of systems, followed by CMU, MIT and SRI, with a dozen scattered efforts elsewhere.

The list indicates that thus far the major areas of expert systems development have been in diagnosis, data analysis and interpretation, planning and design. However, the list also indicates that a few pioneering expert systems already exist in quite a number of other functional areas. In addition, a substantial effort is underway to build expert systems as tools for constructing expert systems.

DENDRAL (Lindsay et al., 1980), which produces molecular structural representations from mass spectrogram data, has been the most widely used expert system. It has subsequently been generalized to CONGEN to produce a set of structural candidates from whatever constraining data is available.

Feigenbaum (1982, p. 16) states that the most knowledge intensive system is INTERNIST, a medical diagnosis system which considers almost 500 diseases and contains over 100,000 pieces of knowledge.

Table 3

Existing Expert Systems by Function

Function	Domain	System*	Institution
Diagnosis	Medicine	PIP	M.I.T.
	"	CASNET	Rutgers U.
	"	INTERNIST	U. of Pittsburgh
	"	MYCIN	Stanford U.
	"	PUFF	"
	Computer Faults	DART	Stanford U./IBM
Data Analysis and Interpretation	Geology	DIPMETER ADVISOR	M.I.T./Schlumberger
	Chemistry	DENDRAL	Stanford U.
	Chemistry	GA1	"
	Geology	PROSPECTOR	SRI
	Protein Crystallography	CRYSALIS	Stanford U.
Analysis	Electrical Circuits	EL	M.I.T.
	Symbolic Mathematics	MACSYMA	M.I.T.
	Mechanics Problems	MECHO	Edinburgh
	Naval Task Force Threat Analysis	TECH	Rand/NOSC
Design	Computer System Configurations	R1	C.M.U.
	Automatic Programming	PECOS	Yale
	Circuit Synthesis	SYN	M.I.T.
	Chemical Synthesis	SYNCHEM	SUNY Stoneybrook
Planning	Chemical Synthesis	SECHS	U. of Cal. Santa Cruz
	Robotics	NOAH	SRI
	"	ABSTRIPS	SRI
	Planetary Flybys	DEVISER	JPL
	Errand Planning	OP-PLANNER	Rand
	Molecular Genetics	MOLGEN	Stanford U.
Learning from Experience	Chemistry	METADENDRAL	Stanford U.

TABLE 3 CONTINUED

Function	Domain	System*	Institution
Concept Formation	Mathematics	AM	CMU
Signal Interpretation	Speech Understanding	HEARSAY II	CMU
	"	HARPY	"
	Machine Acoustics	SU/X	Stanford U.
	Ocean Surveilance	HASP	System Controls Inc.
Monitoring	Patient Respiration	VM	Stanford U.
Use Advisor	Structural Analysis Computer Program	SACON	Stanford U.
Computer Aided Instruction	Electronic Troubleshooting	SOPHIE	B.B.N.
	Medical Diagnosis	GUIDON	Stanford U.
Knowledge Acquisition	Medical Diagnosis	TEIRESIAS	Stanford U.
	Medical Consultation	EXPERT	Rutgers
	Geology	KAS	SRI
Expert System Construction		ROSIE	Rand
		AGE	Stanford U.
		HEARSAY III	USC/ISI
		EMYCIN	Stanford U.
		OPS 5	CMU
Image Understanding		VISIONS	U. of Mass.
		ACRONYM	Stanford U.

* References to these systems can be found in Duda (1981), Stefik, et al. (1982) and Buchanan (1981)

IX. Tools for Building Expert Systems

To aid in the building of expert systems, special programming tools have recently begun to be developed. These are listed in Table 4. The most ambitious is AGE (Attempt to Generalize). AGE (Nii and Aiello, 1979) has isolated a number of inference, control and representation techniques from a few previous expert systems and has reprogrammed them for domain independence. AGE, itself an expert system, also guides people in the use of these modules in constructing their own individualized expert systems. AGE also provides two predefined configurations of components. One called the "Blackboard framework" is for building programs that are based on the Blackboard model, as was used in HEARSAY II. The Blackboard model uses the concepts of a globally accessible data structure, called a blackboard, and independent sources of knowledge which cooperate in forming hypotheses. The other predefined configuration, called the "Backchain framework," is for building programs that use backward chaining production rules like those used in MYCIN.

Table 4

Programming Tools for Building Expert Systems

Tool	Organization	Nature
OPS 5	CMU	A programming language builton top of LISP designed to facilitate the use of production rules.
EMYCIN	Standford U.	A domain independent version of MYCIN, which accompanies the backward chaining and explanation approach with user aids.
KAS	SRI	Supervises interaction with an expert in building or augmenting an expert system knowledge base in a network form implemented for PROSPEC-TOR.
ROSIE	RAND	A general rule-based programming language that can be used to develop large knowledge bases. Translates near-English into INTERLISP.
AGE	Stanford U.	A sophisticated expert system to aid users in building expert systems.
HEARSAY III	USC/Information Sciences Institute	A generalized domain-independent extension of HEARSAY II. Includes a "context" mechanism, and an elaborated "blackboard" and scheduler.
UNITS	Stanford U.	A knowledge representation language and interactive knowledge acquisition system. The language provides both for "frame" structures and production rules.

Table 4 (continued)

Programming Tools for Building Expert Systems

Tool	Organization	Nature
TEIRESIAS	Stanford U.	A expert system that facilitates the interactive transfer of knowledge from a human expert to the system via a (restricted) natural language dialog.

X. Constructing An Expert System

Duda (1981, p. 262) states that to construct a successful expert system, the following prerequisites must be met:

- there must be at least one human expert acknowledged to perform the task well

- the primary source of the expert's exceptional performance must be special knowledge, judgment, and experience

- the expert must be able to explain the special knowledge and experience and the methods used to apply them to particular problems

- the task must have a well-bounded domain of application

Randy Davis (MIT) at IJCAI-81* noted that a good expert system application:

- doesn't require common sense

- takes an expert a few minutes to a few hours

- has an expert available and willing to be committed.

Hayes-Roth (1981, p. 2) adds that "...the problem should be nontrivial but tractable, with promising avenues for incremental expansion."

Having found an appropriate problem and an accessible expert, it is then necessary to have available an appropriate system-building tool, such as those described in the last chapter. Realistic and incremental objectives should then be set. Major pitfalls to be avoided in developing an expert system are choosing a poor problem, excessive aspirations, and inadequate resources.

---

*The International Joint Conference on Artificial Intelligence, Vancouver, August 1981.

The time for construction of early expert systems was in the range of 20-50 man-years. Recently, breadboard versions of simple expert sysems have been reported to have been built in as little as 3 man-months, but a complex system is still apt to take as long as 10 man-years to complete. Using present techniques, the time for development appears to be converging towards 5 man-years per system. Most systems take 2-5 people to construct, but not more. (It takes one to two years to develop an engineer or computer scientist into a knowledge engineer.)

Randy Davis (at IJCAI-81) indicated that the stages of development of an expert system can be considered to be*:

1. System design
2. System development (conference paper level)
3. Formal evaluation of performance
4. Formal evaluation of acceptance
5. Extended use in prototype environment
6. Development of maintenance plans
7. System release.

*Thus far, no current system has completed all these stages.

XI. Knowledge Acquisition and Learning

A. Knowledge Acquisition

The key bottleneck in developing an expert system is building the knowledge base by having a knowledge engineer interact with the expert(s). Expert systems can be used to facilitate the process. Some of these expert systems are indicated in Table 3, with the KAS system being elaborated upon in Table 1-7.

The most ambitious of these systems is TEIRESIAS (Davis and Lenat, 1982) which supervises interaction with an expert in building or augmenting a MYCIN rule set. TEIRESIAS uses a model of MYCIN's knowledge base to tell whether some new piece of information "fits in" to what is already known, and uses this information to make suggestions to the expert. An appropriate expert may not always be continuously available during the construction of the expert system, and in many cases may not have all the expertise desired. In these cases other approaches to acquiring the needed expertise is desirable.

B. Self-learning and Discovery

Michie (1980, p. 11) observes that "The rule-based structure of expert systems facilitates acquisition by the system of new rules and modification of existing rules, not only by tutorial interaction with a human domain specialist but also by autonomous 'learning'." A typical functional application is "classification," for which rules are discovered by induction for large collections of samples (Quinlin, 1979). Michie (1980, p. 12) provides a list of examples of various "learning" expert systems.

DENDRAL, for obtaining structural representations of organic molecules, is the most widely used expert system. As the knowledge acquisition bottleneck is a critical problem, a META-DENDRAL expert system (outlined in Table 1-8) was written to attempt to model the processes of theory formation to generate a set of general fragmentation rules of the form used by DENDRAL. The method used by META-DENDRAL is to generate, test and refine a set of candidate rules from data of known molecule structure-spectrum pairs. For META-DENDRAL and several of the other learning expert systems, the generated rules were found to be of high quality (Feigenbaum, 1980 and Michie, 1980).

Another attempt at modeling self-learning and discovery is the AM Program (Davis and Lenat, 1982) for discovery of mathematical concepts, beginning with elementary ideas in set theory. AM (outlined in Table 1-2) also uses a "generate and test" control structure. The program searches a space of possible conjectures that can be generated from the elementary ideas in set theory, chooses the most interesting, and pursues that line of reasoning. The program was successful in rediscovering many of the fundamental notions of mathematics, but eventually began exploring a bigger search space than the original heuristic knowledge given to it could cope with. A more recent project - EURISKO - is exploring how a program can devise new heuristics to associate with new concepts as it discovers them.

XII. Who is Doing It

The following is a list by category of the "principal players" in expert systems. In each category, the listing roughly reflects the amount of effort in expert systems at that institution. Stanford University is the major center of effort in expert systems.

Universities

Stanford
MIT
CMU
and scattered efforts at perhaps a dozen other universities.

Non-Profit

SRI
RAND
JPL

Government

NRL AI Lab, Washington, D.C.
NOSC, San Diego, CA

Industrial

Fairchild
Schlumberger
Machine Intelligence Corp., Sunnyvale, CA
Xerox PARC
Texas Instruments
Teknowlege, Palo Alto, CA
DEC
Bell Labs
IntelliGenetics, Palo Alto, CA
TRW
BBN
IBM
Hewlett Packard, Palo Alto, CA
Martin Marietta, Denver, CO
Hughes
AMOCO
JAYCOR, Alexandria, VA
AIDS, Mt. View, CA
Systems Control, Inc., Palo Alto, CA

XIII. Who is Funding It

To date, the government has been the principal source of funds of work in expert systems. The funding sources in the government for expert systems, roughly in decreasing order of expenditure, are:

DARPA
NIH (National Insitutes of Health)
NSF
ONR
NLM (National Library of Medicine)
AFOSR
USGS
NASA

DARPA and NIH have been the primary funders of expert systems to date.

Obtaining precise figures for funding of expert systems (ES) is virtually impossible because ES is not carried as a separate funding category. In addition, expert systems are often embedded in other AI systems such as image understanding systems. Further, with artificial intelligence becoming heavily knowledge-oriented, a substantial portion of current AI systems and activities can be viewed as having expert system components.

Nevertheless, a rough estimate of the current total U.S. government yearly funding for expert systems research and development would be in the order of 10 million dollars. Of this expenditure, approximately several million is spent by DARPA to support basic research.

NIH funds the AIM (Artificial Intelligence in Medicine) network (NIH, 1980) and its users at a little over three million dollars a year. This nationally shared computing resource is devoted entirely to designing AI applications for the biomedical sciences. The community of projects using this resource is expert systems oriented. Approximately one third of the three million dollar expenditure in the AIM area can be considered to be for direct research, the balance being for applications, experimentation and system support.

NSF, focussed more on basic research, funds approximately one million dollars per year in the expert systems area. Other government agencies probably spend another two to three million dollars per year to support a variety of potential applications.

Finally, government contractors using IRAD (Independent Research and Development) funds (associated with their prime contracts) probably spend another one to two million dollars a year in this area.

XIV. Summary of the State-of-the-Art

Buchanan (1981, pp. 6-7) indicates that the current state of the art in expert systems is characterized by:

- Narrow domain of expertise

  Because of the difficulty in building and maintaining a large knowledge base, the typical domain of expertise is narrow. The principle exception is INTERNIST, for which the knowledge base covers 500 disease diagnoses. However, this broad coverage is achieved by using a relatively shallow set of relationships between diseases and associated symptoms. (INTERNIST is now being replace by CADUCEUS, which can diagnose simultaneous unrelated diseases).

- Limited knowledge representation languages for facts and relations

- Relatively inflexible and stylized input-output languages

- Stylized and limited explanations by the systems

- Laborious construction

  At present, it requires a knowledge engineer to work with a human expert to laboriously extract and structure the information to build the knowledge base. However, once the basic system has been built, in a few cases it has been possible to write knowledge acquisition systems to help extend the knowledge base by direct interaction with a human expert, without the aid of a knowledge engineer.

- Single expert as a "knowledge czar."

  We are currently limited in our ability to maintain consistency among overlapping items in the knowledge base. Therefore, though it is desirable for several experts to contribute, one expert must maintain control to insure the quality of the data base.

In addition, most systems exhibit fragile behavior at the boundaries of their capabilities, so that occasionally even some of the best systems come up with wrong answers. Another limitation is that for most current systems only their builders or other knowledge engineers can successfully operate them.

Nevertheless, Randy Davis (at IJCAI-81) observed that there have been notable successes. A methodology has been developed for explicating informal knowledge. Representing and using empirical associations, four systems have been routinely solving difficult problems - DENDRAL, MACSYMA, MOLGEN and PUFF - and are in regular use. The first three all have serious users who are only loosely coupled to the system designers. DENDRAL, which analyzes chemical instrument data to determine the underlying molecular structure, has been the most widely used program (see Lindsay et al., 1980). R1, which is used to configure VAX computer systems, has been reported to be saving DEC several millions of dollars per year, and is now being followed up with XCON.

In addition, as indicated in Table 3, several dozen systems have been built and are being experimented with.

XV Current Problems and Issues

Buchanan (1981, p. 11) states, "Because of the increased emphasis on large knowledge bases, the three issues of explanation, acquisition and validation are becoming critical issues for expert systems."

Explanation

Explanation is needed because users cannot be expected to know or understand the whole program.

Knowledge Acquisition

Feigenbaum (1982, p. 13) states, "...knowledge acquisition is the criticial bottleneck problem in Artificial Intelligence." Knowledge acquisition is difficult and time consuming. The most difficult part is helping the expert to initially structure the domain. The knowledge engineer takes an active role in the knowledge acquisition process - interpreting and integrating the experts answers to questions, drawing analogies, posing counter-examples, and raising conceptual difficulties.

Duda (1981, p. 264) observes, "Past efforts to speed knowledge acquisition have been along three lines: (1) to develop smart editors that assist in entering and modifying rules, (2) to develop an intelligent interface that can interview the expert and formulate the rules, and (3) to develop a learning system that can induce rules from examples, or by reading textbooks and papers." Duda also notes "...that it is difficult for experts to describe exactly how they do what they do, especially with respect to their use of judgment, experience, and intuition... We need to develop more expressive languages that allow the expert to articulate more of the nuances and details of thought processes." Diverse sources of knowledge are also often required, but there is currently no good way to integrate these sources in reaching a solution.

A few knowledge-acquisition systems do exist, such as TEIRESIAS, that are interactive and semi-automatically steer the expert to the needed piece of knowledge to introduce into the expert system under development. However, these existing knowledge acquisition systems have only been used to expand and improve a knowledge base after a vocabulary and knowledge representation had already been chosen and upon which the basic knowledge base had already been built. The knowledge-acquisition problem remains extremely difficult and a major impediment.

Validation

All complex computer programs tend to have errors and are therefore difficult to certify. At the moment, empirical studies (such as has been used to validate MYCIN as a superior diagnostician and therapist) may be the best we can hope for. However, the credibility of the system can be increased if the system is made intelligible and understandable, so that the user can be made responsible for the system and be able to modify it to his or her own satisfaction. More fundamentally, a methodology of validation needs to be developed.

Other problems are:

Lack of Adequate and Appropriate Hardware

Feigenbaum (1980, p. 10) stated that "...applied AI is machine limited." This is still true, though special LISP machines and more general large, fast computing machines are beginning to become available.

Inadequate Special Knowledge Engineering Tools

Though software packages such as EMYCIN and OPS-5 are beginning to become available, there is much room for improvement and extension to capture more of the existing expert systems approaches and architectures and make them readily available to the new expert systems builders. Further, the concepts and techniques thus far developed need to be systematically drawn together and synthesized into higher-order patterns, so that a firm base for future systems can be built, and reinventing the proverbial wheel can be avoided.

Orderly Development and Transfer

To capture the interest of domain experts and develop a major expert system requires continuous funding over several years, which has not always been available. Further, there is as yet no orderly system in the research funding agencies for effectively taking a successful research project and moving it on to appropriate applications.

Shortage of Knowledge Engineers

The field is relatively new and few knowledge engineers are currently being trained by the universities. Because of the huge number of potential applications, shortages of knowledge engineers currently exist and probably will continue to exist for some time.

XVI Research Required

Buchanan (1981, pp. 8-14) indicates that research is required to develop:

- Improved knowledge acquisition systems
- Learning by example
- Better explanation systems and friendlier user interfaces
- More adequate knowledge engineering tools
- Better expert system architectures and inference procedures
- More efficient and workable techniques for working with multiple experts and knowledge sources
- More adequate methods for dealing with time
- The ability to make appropriate assumptions and expectations about the world
- The ability to exploit causal physical and biological models and couple them with other knowledge
- General methods for planning
- Analogical reasoning
- Methods for coupling formal deduction into expert systems
- Parallel processing approaches
- Better knowledge representation methods

XVII Future Trends

Figure 2 lists some of the expert system applications currently under development. It will be observed that there appear to be few domain or functional limitations in the ultimate use of expert systems.

Figure 3 (based largely on Hayes-Roth IJCAI-81 Expert System tutorial and on Feigenbaum, 1982) indicates some of the future opportunities for expert systems. Again no obvious limitation is apparent.

It thus appears that expert systems will eventually find use in most endeavors which require symbolic reasoning with detailed professional knowledge - indeed most of the world's work. In the process, there will be exposure and refinement of the previously private knowledge in the various fields of application. Feigenbaum (1980, p. 10) states that, "The gain to human knowledge by making explicit the heuristic rules of a discipline will perhaps be the most important contribution of the knowledge-based systems approach."

On a more near-term scale, in the next few years we can expect to see expert systems with thousands of rules. In addition to the increasing number of rule-based systems we can also expect to see an increasing number of non-rule based systems as not all problems are homogeneous enough to be readily cast in the production system framework. We can also expect much improved explanation systems that can explain why an expert system did what it did and what things are of importance.

By the late 80's, we can expect to see intelligent, friendly and robust human interfaces. Much better system building tools

Figure 2

Expert System Applications Now Under Development

- Medical diagnosis and prescription
- Medical knowledge automation
- Chemical data interpretation
- Chemical and biological synthesis
- Mineral and oil exploration
- Planning/scheduling
- Signal interpretation
- Military threat assessment
- Tactical targeting
- Space defence
- Air traffic control
- Circuit diagnosis
- VLSI design
- Equipment fault diagnosis
- Computer configuration selection
- Speech understanding

Figure 3

Future Opportunities for Expert Systems

- Building and Construction

  Design, planning, scheduling, control

- Equipment

  Design, monitoring, control, diagnosis, maintenance repair, instruction.

- Command and Control

  Intelligence analysis, planning, targeting, communication

- Weapon Systems

  Target identification, adaptive control, electronic warfare

- Professions

  (Medicine, law, accounting, management, real estate, financial, engineering)

  Consulting, instruction, analysis

- Education

  Instruction, testing, diagnosis, concept formation and new knowledge development from experience.

- Imagery

  Photo interpretation, mapping, geographic problem-solving.

- Software

  Instruction, specification, design, production, verification, maintenance

Figure 3 (continued)

- Home Entertainment and Advice-giving

  Intelligent games, investment and finances, purchasing, shopping, intelligent information retrieval

- Intelligent Agents

  To assist in the use of computer-based systems

- Office Automation

  Intelligent systems

- Process Control

  Factory and plant automation

- Exploration

  Space, prospecting, etc.

should also be available. By 1990, we can anticipate knowledge acquisition systems which, after being given a basic domain context, can rapidly guide a human expert in forming the needed expert system knowledge base. Somewhere around the year 2000, we can also expect to see the beginnings of systems which semi-autonomously develop knowledge bases from text. The result of these developments may very well herald a maturing information society where expert systems put experts at everyone's disposal. In the process, production and information costs should greatly diminish, opening up major new opportunities for societal betterment.

References

1. Michie, Donald, "Knowledge-based Systems," University of IL at Urbana-Champaign, Report 80-1001, Jan. 1980.

2. Feigenbaum, E. A., "Knowledge Engineering: The Applied Side of Artificial Intelligence," Computer Science Dept., Memo HPP-80-21, Stanford University, July, 1980.

3. Feigenbaum, E. A., "Knowledge Engineering for the 1980's," Computer Science Dept., Stanford University, 1982.

4. Nii, H. P., and Aiello, N., "AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs," Proceedings of the Sixth International Conf. on Artificial Intelligence, (IJCAI-79), Tokyo, Aug., 20-23, 1979, pp. 645-655.

5. Buchanan, B. G., "Research on Expert Systems," Stanford University Computer Science Department, Report No. STAN-CS-81-837, 1981.

6. Hayes-Roth, F., "AI The New Wave - A Technical Tutorial for R&D Management," (AIAA-81-0827), Santa Monica, CA: Rand Corp., 1981.

7. Duda, R. O. "Knowledge-Based Expert Systems Come of Age," Byte, Vol. 6, No. 9, Sept. 81, pp. 238-281.

8. Heuristic Programming Project 1980, Computer Science Dept., Standford University, 1980.

9. Stefik, M., et al., "The Organization of Expert Systems: A Prescriptive Tutorial", XEROX, Palo Alto Research Centers, VLSI-82-1, January 1982.

10. Quinlin, J. R., "Discovering Rules by Induction from Large Collections of Examples," in Expert Systems in the Micro-Electronic Age, D. Michie (Ed.), Edinburgh: Edinburgh University Press, 1979, pp. 168-201.

11. Davis, R, and Lenat, D. B. Knowledge-Based Systems in Artificial Intelligence, New York: McGraw-Hill, 1982.

12. Lindsay, R. K., et al., Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project, New York: McGraw-Hill, 1980.

13. McCarthy, J., and Hayes, P. J., "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in Machine Intelligence 4, Meltzer, B., and Michie, D. (Eds.)., Halsted Press, 1969, pp. 463-502.

14. Vere, S., Planning in Time: Windows and Durations for Activities and Goals, Pasadena, CA: JPL, Nov. 1981.

15. The Seeds of Artificial Intelligence: Sumex-AIM, NIH Publ. No. 80-2071, Bethesda, MD: NIH Div. of Research Resources, March 1980.

SYSTEM: EL

INSTITUTION: MIT

AUTHORS: Stallman & Sussman

FUNCTION: Analysis

TABLE 1-5

Characteristics of Example Expert Systems

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	KEY ELEMENTS OF: GLOBAL DATA BASE	KEY ELEMENTS OF: CONTROL STRUCTURE
Steady State analysis of resistor-diode-transistor circuits to determine voltages and currents.	Use assertions in the data base to trigger the rules which create new assertions in the data base. Assume operating states for transistors and diodes. Introduce variables for the parameters (e.g., e for voltage) at one node in the circuit and use electrical laws to symbolically compute parameters at other nodes. Make conjectures when no further rules are applicable. Observe contradictions and revise conjectures and conclusions dependent upon them.	Rules that represent general electrical principles. Rules for making conjectures. Rules for deciding what to forget when contradictions occur.	Facts about the particular circuit being analysed represented as assertions in an associative data base. History of conjectures used and conclusions dependent upon them. Problem status.	Forward reasoning Guesses when needed Relevant backtracking Priority-oriented queue-based control.

SYSTEM: GUIDON

INSTITUTION: Stanford University

AUTHORS: Clancy

FUNCTION: Computer-Aided Instruction (CAI)

TABLE 1-6

Characteristics of Example Expert Systems

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Teach facts and problem-solving strategies for: - diagnosing and treating meningitis and bacterima or - pulmonary function analysis	Use MYCIN or PUFF knowledge base and add rules for teaching medical diagnosis	MYCIN or PUFF knowledge base 200 additional rules for: - guiding dialog with student - presenting medical diagnostic strategies - constructing a student model - responding to the students initiations	Student model thus far Current interchange Status Relevant history of interchange	Event driven

SYSTEM: KAS

INSTITUTION: SRI

AUTHORS: Reboh

FUNCTION: Knowledge Acquisition

TABLE 1-7

Characteristics of Example Expert Systems

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Supervises interaction with an expert in building or augmenting a expert system knowledge base in a network form. Implemented for PROSPECTOR.	Take existing PROSPECTOR knowledge base and knowledge of mechanisms in PROSPECTOR, and add networks for expressing new knowledge and rules for interaction with a human domain expert. The main technique is to consider the KAS system as a general purpose editor, given knowledge of the specific network.	Prospector knowledge base. Inference networks for expressing judgmental knowledge. Semantic networks for expressing the meaning of the propositions employed in the rules. Taxonomic networks for representing basic knowledge among the terms in the domain. Knowledge of various mechanisms employed in PROSPECTOR for representing and using knowledge. Consistency mechanisms.	Partial candiates to be completed.	Network editor -creates, modifies or deletes various nodes and arcs in the networks. Grammar-driven command language to prompt user in the proper use of command language.

TABLE 1-8

Characteristics of Example Expert Systems

SYSTEM: META-DENDRAL

INSTITUTION: Stanford University

AUTHORS: Buchanan and Feigenbaum

FUNCTION: Learning from Experience

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
To generate a set of general fragmentation rules, of the form used by DENDRAL, given sets of known structure-spectrum pairs.	Generate, test and refine a set of candidate rules from known molecule structure-spectrum pairs.	Rules for: -interpreting spectral data and summarizing results. -generating candidate fragmentation rules from the evidence. -generalizing or specializing candidate rules to better fit the evidence.	User supplied context Input data -known structure-spectrum pains -plausible candidate fragmentation rules	Plan, generate, and test.

TABLE 1-9

Characteristics of Example Expert Systems

SYSTEM: VM (Ventilator Manager)

INSTITUTION: Stanford University

AUTHORS: Fagan

FUNCTION: Monitoring and interpreting real-time data

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Interpret the clinical significance of data from a physiological monitoring system of patient breathing.	1. Start with an initial patient state (context or situation) 2. Use initialization rules to determine expectations and unacceptable measurement limits for that state. 3. Run status rules to derive physiological states for use in therapy. 4. Run transition rules to see if state has changed each time new set of periodic measurements arrive. 5. Repeat 2 and 3 for each new state.	Transition rules. Initialization rules. Status rules Therapy rules.	Sets of periodic measurements of patient. Patient current state (context or situation) Expectations and unacceptable limits for measurements in current state. Recent patient history during monitoring. Physiological status.	Event driven. Exhaustive search for each state. State triggered expectations.

TABLE 1-10

Characteristics of Example Expert Systems

SYSTEM: GA1

INSTITUTION: Stanford U.

AUTHORS: Stefik

FUNCTION: Data Interpretation

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Infer a complete molecular structure from measurement of molecular pieces (resulting from digestion by an enzyme).	1. Refine the data. 2. Determine an initial set of generator constraints from data. 3. Combine data and constraints to generate candidate structures. 4. Prune candidate approaches during generation process that are inconsistant with general rules for molecular structures. 5. Test candidates to see if they satisfy data.	Data correction rules. Rules for determining an initial set of generator constraints. Generation rules. Rules for pruning inconsistant candidate classes. Rules for testing candidate molecular structures.	Measurement data. Corrected data. Derived set of generator constraints Partial solutions. Candidate molecular structures.	Hierarchical generate and test. Early pruning of inconsistant approaches.

TABLE 1-11

Characteristics of Example Expert Systems

SYSTEM: ABSTRIPS

INSTITUTION: SRI

AUTHORS: Sacerdoti

FUNCTION: Planning

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	KEY ELEMENTS OF: GLOBAL DATA BASE	KEY ELEMENTS OF: CONTROL STRUCTURE
Devises plans for a robot to move objects between rooms.	Do hierarchical planning by first devising a top level plan based on the key aspects of the problem, then successively refining it by considering less critical aspects of the problem. Recipe: 1. Fix abstraction levels for solutions (plans). 2. Problem solution proceeds top down (most abstract to most specific). 3. Complete solution at one level and then move to next level below.	Criticality assignments of elements in robot planning domain. Configuration of the rooms. Objects and their properties in the domain. Rules for decrementing criticality level. Heuristic search rules for each level.	Goal Initial state of system (criticality at maximum). Plans thus far. Current criticality level	Goal directed (backward chaining at each level). Top down refinement of plans using hierarchical abstract search spaces.

SYSTEM: NOAH

INSTITUTION: SRI

AUTHORS: Sacerdoti

FUNCTION: Planning

TABLE 1-12

Characteristics of Example Expert Systems

		KEY ELEMENTS OF		
PURPOSE	APPROACH	KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Robot Planning System (assigns a time-ordering to operators in a plan.)	Expand, in parallel, individual plans for interacting subgoals, but initially assign only a partial time-ordering to operators. Stop when interference between the partial subgoal plans is observed, and adjust the ordering of the operators as needed to resolve the interference.	Operators. Rules for recognizing interference between plans. Rules for resolving interferences.	Subgoals Partial ordering of operators in subgoal plans. Interference between plans.	Least committment. Backward chaining.

SYSTEM: MOLGEN

INSTITUTION: Stanford U.

AUTHORS: Stefik*

FUNCTION: Design

TABLE 1 -13

Characteristics of Example Expert Systems

*Another Molgen version by Friedland under development.

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Designing molecular genetic experiments	Represent interactions between subproblems as constraints. Formulate constraints as goals to be solved. Use constraint propogation to reveal interactions between subproblems. Suspend problem-solving as necessary, until sufficient information is derived from the interchange of constraints (least committment, opportunistic expansion). Use heuristic guessing to make choices when there is otherwise no compelling reason to do so. Retract guesses as necessary when an unresolvable problem is encountered.	Explicit meta-level problem-solving operators to reason with constraints. Problem-solving rules. Rules for guessing. Rules for discovering interactions between subproblems via constraint propogration.	Partial solutions. History of guesses and their effects. Constraints.	Constraint propogation. Least committment. Heuristic guessing. Relevant backtracking. Use of meta-rules to reason with constraints. Hierarchical refinement. Difference reduction.

TABLE 1-14

Characteristics of Example Expert Systems

SYSTEM: SYN

INSTITUTION: MIT

AUTHORS: Sussman, Steele, & Dekleer

FUNCTION: Design

PURPOSE	APPROACH	KEY ELEMENTS OF: KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Circuit synthesis	Use propogation ideas of EL.	Electrical laws.	Circuit.	Forward reasoning.
-determines values for components in electrical circuits.	Switch to equivalent representations of circuit portions when needed to overcome blockages in the propogation of constraints.	Rules for changing slices	Slice being considered	Changing representations as needed to continue analysis.
	(These "slices" - multiple views of circuit portions - provide redundant paths for information to travel. Slices combine the strengths of multiple models).	Rules for creating appropriate slices.	Set of deduced values.	

SYSTEM: HEARSAY II

INSTITUTION: CMU

AUTHORS: Erman, Hayes-Roth, Lesser, Reddy

FUNCTION: Signal Interpretation

TABLE 1-15

Characteristics of Example Expert Systems

PURPOSE	APPROACH	KEY ELEMENTS OF KNOWLEDGE BASE	KEY ELEMENTS OF GLOBAL DATA BASE	KEY ELEMENTS OF CONTROL STRUCTURE
Speech under-standing	Break the problem up hierarchically into levels (heterogeneous abstract search spaces) with sentences at the top and signal measurement parameters at the bottom. Do both bottom up and top-down processing in a relaxation approach to extend and combine partial candidates. Carry several candiates at each level (parallel lines of reasoning) to keep from being too focussed and missing the correct interpretation. Use a separate knowledge source (KS) for each level. Use a 7 level "blackboard" to display hypotheses. Have KS's communicate via the blackboard. Have KS's when activated: 1. create and extend hypotheses on blackboard, 2. record the evidential support between levels, 3. assign credibility level Use "opportunistic scheduling" of computational resources for changing the breadth of search depending on conditions of uncertainty resulting from interaction of KS assigned credibility ratings and scheduler-assigned priorities of pending KS activations.	Language knowledge. KSs for creating: 1. labeled segments for signal parameter measurements, 2. syllable hypotheses from segments, 3. word hypotheses from syllables, 4. word sequence hypotheses, 5. phrase hypotheses, 6. predictions of words following phrases, 7. sentence level interpretations for the infomation retrieval system. KSs for controlling - number of word hypotheses, - number of word-sequence hypotheses. KSs for rating - credibility of hypotheses, - consistency between segment hypotheses and word-phrase pairs	Hypotheses on 7 level blackboard. Record of evidential support between levels Credibility levels of hypotheses. Results thus far. Agenda queue of pending KS activations. Top-down vs bottom-up consistencies between word-phrase pairs and segment hypotheses.	Combination of top-down and bottom up processing. Opportunistic scheduling. Least committment. Variable-width search.

SYSTEM: HARPY

INSTITUTION: CMU

AUTHORS: Lowerre

FUNCTION: Signal interpretation

TABLE 1 -16

Characteristics of Example Expert Systems

		KEY ELEMENTS OF		
PURPOSE	APPROACH	KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Speech understanding.	Represent the set of all possible utterances in HARPY's domain by production rules which relate signal syllables to words. Add juncture rules at word boundaries. Use a compiler to combine the syntax, lexical and juncture knowledge into a single large transition network in which each path from a start node to an end node represents a sequence of segments for some sentence.	Transition network.	Input speech.	Data driven "Beam Search" thru network.

SYSTEM: CRYSALIS

INSTITUTION: Stanford University

AUTHORS: Englemore & Terry

FUNCTION: Data Interpretation

TABLE 1-17

Characteristics of Example Expert Systems

PURPOSE	APPROACH	KEY ELEMENTS OF KNOWLEDGE BASE	GLOBAL DATA BASE	CONTROL STRUCTURE
Automatic interpretation of protein electron-density maps.	1. Use a skeletonization algorithm to convert the electron density map to a line-skeleton representation. 2. Use a algorithm to partition skeleton graph into chemical side chains and backbone elements. 3. Use rules to hypothesize (with associated confidence levels) and test atoms and super atoms based on the partioned skeleton graph and the known chemical model of the protein under study. 4. Use hierarchical meta-rules to select appropriate KS rule set. Recipe to develop hypotheses 1. Use hypothesis toeholds to fire strategy rules to determine which task rule set to consider next. 2. Match task rules against the "type" column of the event list to focus on one event and to choose the set of KS rules to consider. 3. Match the hypothesis state and the "where" portion of the event to the KS rules to increment the hypothesis and add to the event list. 4. Repeatedly cycle on 2 and 3 until no further matches occur, then return control to strategy-rule interpreter.	Strategy rule set. Task rule set. KS rule set. Skeletonization algorithm. Partitioning algorithm.	"Blackboard" which includes: -a hierarchically orgainized hypothesis data structure -support for the current hypothesis from the partioned skeleton graph and the chemical model. State of the hypothesis. -"toehold" opportunities for accelerated development of the hypothesis. Event list of recent changes by type and location.	Event driven. Hypothesize and test. Hierarchical rule interpreters -strategy -task -KS

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 82-2505	2. Performing Organ. Report No.	3. Publication Date May 1982

4. TITLE AND SUBTITLE

AN OVERVIEW OF EXPERT SYSTESM

5. AUTHOR(S)

William B. Gevarter

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)	7. Contract/Grant No.
NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234	8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This report provides an overview of Expert Systems - currently the hottest topic in the field of Artificial Intelligence. Topics covered include what it is, techniques used, existing systems, applications, who is doing it, who is funding it, the state-of-the-art, research requirements, and future trends and opportunities.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Applications; Artificial Intelligence; Expert systems; Forecast; Intelligent computer programs; Knowledge engineering; Machine Intelligence; Overview; Research; State-of-the-Art; Funding sources.

13. AVAILABILITY	14. NO. OF PRINTED PAGES
[X] Unlimited ☐ For Official Distribution. Do Not Release to NTIS ☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. [X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161	73 15. Price $9.00