ED 310 770 IR 052 829

AUTHOR	Fenly, Charles; Harris, Howard
TITLE	Expert Systems: Concepts and Applications. Advances in Library Information Technology, Issue Number 1.
INSTITUTION	Library of Congress, Washington, D.C.
REPORT NO	ISBN-0-8444-0611-2
PUB DATE	88
NOTE	44p.
AVAILABLE FROM	Cataloging Distribution Service, Library of Congress, Washington, DC 20541.
PUB TYPE	Information Analyses (070) -- Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; *Computer System Design; *Expert Systems; Feasibility Studies; Library Automation; *Library Technical Processes; Literature Reviews; National Libraries; Technological Advancement
IDENTIFIERS	*Library of Congress

ABSTRACT

The Processing Services department of the Library of Congress initiated a project to learn about expert systems technology and to examine potential applications of expert systems to functions in their department, e.g., acquisitions, cataloging, and serials control. (An expert system is defined as an artificial intelligence computer program which uses knowledge and inference to address problems that human experts would normally solve in a particular domain of expertise.) The project and this report consist of two parts. Focusing on expert systems technology, the first part includes information gathered through a literature review to develop a working understanding of the concepts of expert systems, and includes sections on artificial intelligence; the characteristics of expert systems; uses of expert systems, including a discussion of applications in librarianship; how expert systems function; and the process and tools for developing expert systems. The second part reports on a study of the feasibility of using expert systems for technical processing in the Library of Congress, which collected data through a series of interviews and onsite visits to determine potential candidates for the application of expert systems technology. The report includes discussions of: (1) the methodology used; (2) the characteristics of a suitable expert system domain; (3) potential applications, i.e., shelflisting assistant, series consultant, and subject cataloging consultant; and (4) operations that were ruled out as potential applications areas, i.e., cataloging in publication, decimal classification, descriptive cataloging, National Union Catalog (NUC), exchange and gift work, ordering, overseas operations, and serials management. (24 references) (SD)

EXPERT SYSTEMS
Concepts and Applications

Prepared by Charles Fenly, Library of Congress in association with Howard Harris, RMG Consultants, Inc.

Advances in Library Information Technology
Issue Number 1

Cataloging Distribution Service, Library of Congress • Washington, D.C. • 1988

Library of Congress Cataloging-in-Publication Data

Fenly, Charles, 1946-
Expert systems.

(Advances in library information technology, ISSN 0899-1227 ; issue no. 1)
1. Libraries--Automation. 2. Expert systems (Computer science) 3. Library science--Data processing. 4. Library of Congress. 5. Libraries, National--United States--Automation. I. Harris, Howard, 1946 II. Title. III. Series.
Z678.93.E93F46 1988 025'.03'0285 88-600201
ISBN 0-8444-0611-2

ISBN 0-8444-0611-2

For sale by the Cataloging Distribution Service
Library of Congress, Washington, D.C. 20541 U.S.A.
(202) 287-6100

TABLE OF CONTENTS

FOREWORD

The technical services managers of the three national libraries, the National Agricultural Library, the National Library of Medicine, and the Library of Congress, have discussed on several occasions the issue of using expert systems in library technical processing. Because of our interest in this topic, we determined that each national library would make a contribution to furthering our understanding of the potential applicability of this technology.

As our part of this exploration, the management of the Processing Services department of the Library of Congress initiated a project to learn about expert systems technology and to examine the potential for applying expert systems to technical processing functions within the department. The project was carried out by Charles Fenly of the Library of Congress staff and Howard Harris of RMG Consultants, Inc., under the direction of a project review group consisting of Mary S. Price, Director for Bibliographic Products and Services, Lucia J. Rather, Director for Cataloging, Robert C. Sullivan, Director for Acquisitions and Overseas Operations, Donald P. Panzera, department Executive Officer, and myself.

Mr. Fenly's background is in librarianship, and Mr. Harris' background is in librarianship and library automation consulting. Since neither has a background in expert systems technology, the first phase of the project was devoted to a literature review which they conducted in order to develop a working understanding of expert systems. In the second phase of the project, they applied this working understanding to a study of department operations in order to recommend possible candidates to be considered for the application of expert systems.

The present report is the result of a learning experience for all involved and is by no means intended to provide a definitive analysis of this complex topic. However, after reviewing the working paper which resulted from the first phase of the project, I felt that the information included provided a useful synopsis of expert systems technology which might be of interest within the library community. I considered the working paper which came out of the second phase, though its results and recommendations may be directly applicable only to the Library of Congress, to be of potential interest as well. Accordingly, I decided that this consolidated and revised version of the two papers would be published by the Cataloging Distribution Service.

Henriette D Avram
Assistant Librarian for Processing Services

INTRODUCTION

This paper is based on a project which was carried out in two phases. In the first phase we conducted a literature review to develop a working understanding of the concepts of expert systems for our own use in carrying out the second phase of the project and to provide a common understanding of the subject for the members of the review group. The findings of this phase of the project were reported in a working paper entitled Working Understanding of Expert Systems Technology, which was presented to the review group on August 10, 1987

In the second phase of the project we conducted a series of interviews and on-site visits within a variety of technical processing operations in order to attempt to determine whether any of these operations were promising candidates for the application of expert systems technology. The findings of this phase of the project were reported in a working paper entitled Opportunities for the Application of Expert Systems Technology in Processing Services at the Library of Congress This paper was presented to the review group on December 17, 1987.

Charles Fenly prepared the present report by combining and extensively revising the two working papers This report is in two parts. Part I, entitled Expert Systems Technology, is intended to provide a general overview of the state of expert systems technology as of late 1987. Part II, entitled Expert Systems in Library of Congress Technical Processing A Feasibility Study, describes the investigation we conducted in Processing Services of the Library of Congress to identify potential candidates for expert systems and presents the results and recommendations of that investigation

PART I: EXPERT SYSTEMS TECHNOLOGY

Part I of this report is an overview of expert systems technology. Because expert systems constitutes one of the applications of artificial intelligence (AI), the first section of Part I provides a brief discussion of some important AI concepts. In the second section some of the characteristics and benefits of expert systems are identified. The third section is devoted to a discussion of the uses which have been made of expert systems and includes a brief review of expert systems in librarianship. The fourth section describes how the main components of an expert system function, and the fifth section discusses the expert system development process, including a description of expert system building tools.

1. ARTIFICIAL INTELLIGENCE

Expert systems is one of the major application areas within the field of artificial intelligence, or AI. "Intelligence" includes such elements as the ability to learn or understand from experience, the ability to acquire and retain knowledge, and the use of the faculty of reason in solving problems. AI is the subfield of computer science concerned with understanding intelligence and developing computer programs which exhibit intelligence. Basic components of artificial intelligence research which are particularly important to an understanding of expert systems are search, knowledge representation, and artificial intelligence languages and hardware (Hunt 1986).

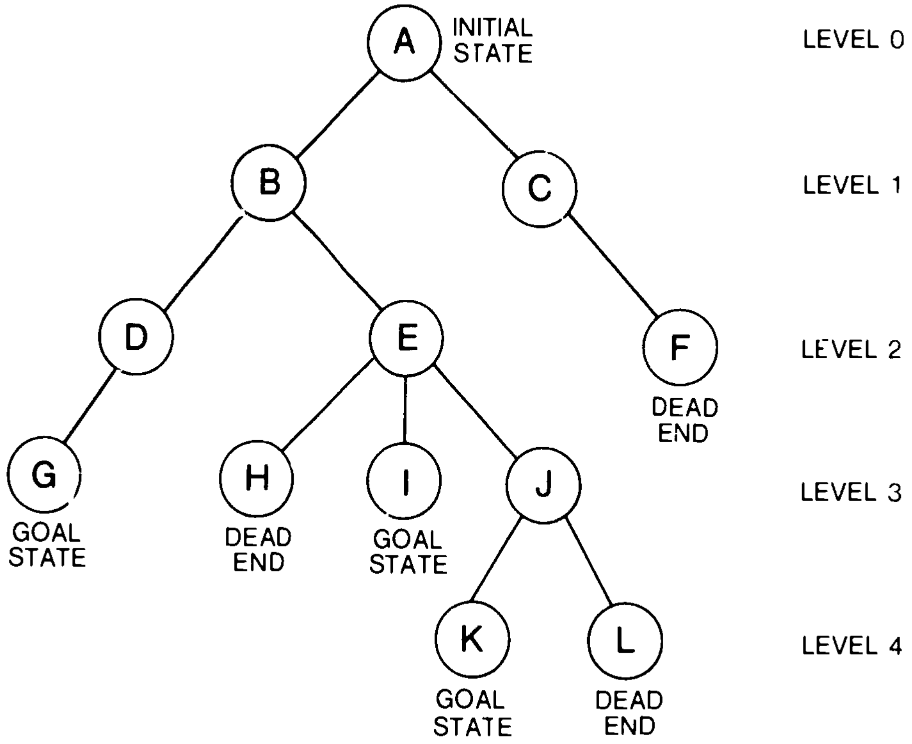
Artificial intelligence problem solving can be viewed as a search among alternative solutions to a problem in an attempt to determine the best solution. The search proceeds under the guidance of one or more control strategies, or search techniques, from an initial state to a goal state. The implicit set of all possible paths which the search might take is called the <u>search space</u>.

In any AI program substantial enough to address a real-world problem, the search space would be far too large to depict graphically. However, the underlying concept can be visualized by representing a small-scale search space as a <u>search tree</u>. Such a tree is shown in Figure 1-1. The search proceeds from the initial problem state (level 0) through the various levels until a goal state is reached. Often, there are alternative paths to a goal state. For example, in Figure 1-1 a goal state may be reached by following, among others, paths A-B-D-G or A-B-E-I. Some paths, such as A-B-E-H, may lead to dead ends, which may force the search to backtrack until another path can be followed or may represent an unsolved problem.

One of the greatest challenges in AI research has been the development of efficient and effective methods of limiting the enormous search spaces associated with real-world problems. Techniques have been designed to limit the search space by using a variety of formal search strategies or by building in shortcuts derived from information about the nature and structure of problems or tasks associated with a particular domain of knowledge. Such limiting strategies and shortcuts are referred to as <u>heuristics</u>. Heuristic problem-solving is one of the most important concepts in AI. An example of a simple heuristic to assist in the selection of a wine for dinner would be "with fish drink white wine." Application of this heuristic would not insure that the best possible wine was chosen, but it would greatly reduce the number of wines which had to be considered by immediately eliminating from contention all non-white wines.

FIGURE 1-1

SEARCH SPACE REPRESENTED AS A SEARCH TREE



A fundamental component of intelligence is knowledge. In AI, knowledge representation focuses on methods for efficiently modeling knowledge in such a manner that it is easily accessible for application to problem-solving within the context of an artificial intelligence computing system. A great deal of early AI research was focused on development of systems which possessed general problem-solving knowledge. Embodying such a capability in a computing system proved extremely difficult, however, and the attention of many researchers was redirected toward systems whose knowledge was specific to a particular domain. Significant challenges related to knowledge include determining what knowledge is required for a given set of problems or tasks and determining which of the alternative methods available for representing knowledge in a computing system is most suitable for addressing a given situation.

Artificial intelligence languages have constituted an important area of AI research. The two most well-known AI programming languages are LISP and PROLOG. These languages are able to accommodate the specialized requirements of AI for symbol manipulation, deduction, and implementation of various strategies for searching alternative paths from initial state to goal states. In addition to these and other AI programming languages, specialized programming environments known as knowledge engineering languages are widely used.

Hardware for artificial intelligence applications is of two types: conventional computer systems at the mainframe, minicomputer, and microcomputer levels, and specialized computing systems known as AI workstations or LISP machines.

Most early AI development was carried out on conventional mainframe computer or minicomputer systems, and these classes of computers continue to be used heavily today. For example, the Digital Equipment Corporation VAX minicomputers are popular AI development machines. In addition, a wide assortment of AI programming and knowledge engineering language software is available for use on microcomputers.

AI workstations are computing systems which are designed to address the requirements of artificial intelligence applications. These machines have a number of specialized features which facilitate AI work. For example, they have high-speed processors and large memory capabilities which enable them to deal with the heavy demands of AI search and knowledge representation. Their high-resolution, bit-mapped displays allow for development of sophisticated graphics. Their advanced software environments, which include AI programming languages, knowledge engineering languages, and extensive programming support facilities, address the specialized programming needs of AI development. Some major manufacturers of these systems are Symbolics, LISP Machine, Inc. (LMI), Xerox, and Texas Instruments (Mishkoff 1985).

2. EXPERT SYSTEMS

An expert system is an artificial intelligence computer program which uses knowledge and inference to address problems of the sort which human experts would normally solve in a particular domain of expertise. The knowledge in an expert system consists both of the commonly accepted facts in the domain and the heuristic knowledge, or rules of thumb, which the best experts use to facilitate decision-making Expert systems typically function as advisors or consultants to assist human users in making decisions or solving problems within the domain in which the system operates (Hunt 1986, Frenzel 1987).

2.1. Components of an expert system

An expert system consists of the following basic components:

(1) A **knowledge base** of facts related to the domain;

(2) An **inference engine**, or rule interpreter, which controls the search of the knowledge base;

(3) A **working memory**, or data base, which keeps track of data input, new facts inferred, and the like, in the solution of the problem being worked on; and

(4) A **user interface**, which allows for easy interaction with the system by its intended users and by system developers. A very important feature of the user interface is an **explanation facility**, which allows a user of the system to query the system's reasoning process and facilitates system debugging.

These components are discussed in more detail in section 4.

2.2. Differences between expert systems and conventional programs

The mere fact that a computer program yields a result comparable to that which an intelligent person would achieve does not make it an expert system. Expert systems differ from conventional programs in several important respects, among which are:

(1) Knowledge: A conventional program manipulates data while an expert system manipulates knowledge. In an expert system knowledge is represented symbolically, with symbols being strings of characters which stand for real-world concepts, such as "main entry," "infection," "H7101 regulator." These symbols are organized into a knowledge base of facts about the domain. The expert system solves problems by a process of searching and pattern-matching among these symbols. (Knowledge representation is discussed in detail in section 4.1);

(2) Heuristic problem-solving: A conventional program solves problems through a repetitive algorithmic process, whereas an expert system uses heuristic and inferential reasoning. A heuristic is a shortcut or rule-of-thumb learned through experience which an expert applies to eliminate unproductive paths toward the solution of a problem. The algorithmic approach is intended to guarantee a solution; the heuristic approach does not guarantee a solution, but it allows problem-solving to take place in domains where the search space is so large that an algorithmic approach would be impossible. For example. it is the use of heuristics which allows human beings successfully to complete a game of chess despite the fact that there are an estimated 10^{120} possible combinations of moves;

(3) Program structure: In a conventional program factual knowledge about the problem being addressed tends to be implicit and intermixed in the program code with procedural instructions for processing data. In an expert system the knowledge base and the control structure, the inference engine, are separate. The expert system knowledge base can therefore be updated without impacting upon the inference engine, making program modification and debugging much easier than with conventional programs. In addition, it is possible for different knowledge bases to function with the same inference engine (although for large-scale problems the inference engine will probably need at least some tailoring to each knowledge base);

(4) Self-knowledge: An expert system can keep track of and display to the system user the logical path by which it arrived at a problem solution. A conventional program does not explain how it achieved its results, and the logical process it followed is often difficult to track through its code.

2.3. Benefits of using expert systems

A number of writers on expert systems technology feel that expert systems have the potential to benefit in a significant way organizations which apply them appropriately. Some of the most commonly identified potential benefits are:

(1) Expert systems make scarce expertise more widely available within the organization, thereby helping non-experts achieve expert-like results;

(2) They free human experts for other activities besides repeatedly solving the problems which an expert system could address;

(3) They promote a standardized, consistent approach to solving relatively unstructured tasks;

(4) They enhance organizational effectiveness and efficiency by making readily available solutions to difficult problems which might otherwise require time-consuming research or consultation with experts to solve;

(5) They provide a means for capturing and storing valuable knowledge that might be lost if an expert left the organization;

(6) They provide a means for permanent retention of highly complex knowledge, since machine knowledge does not deteriorate with time or disuse as human knowledge tends to;

(7) They perform at a consistently high level tasks which humans might perform inconsistently due to fatigue or loss of concentration.
(Beerel 1987, Frenzel 1986, Waterman 1986)

2.4. An imaginary expert system consultation

Typically, a user consults an expert system by interacting with a computer, making selections from menus, keying responses to queries posed by the system, and the like. To illustrate the form this consultation takes, the following imaginary consultation with a hypothetical cataloging expert system is offered. This consultation is based upon a very simplified and limited application of rule 22.5C of AACR 2. Data keyed by the user is supplied in boldface.

[The user enters the system, which responds:]

Welcome to the Personal Name Heading Consultant.

If you want me to formulate a name heading, type h. Otherwise, type m for a menu of other services I can perform for you.

****h**

Enter the name for which you want me to formulate a heading:

****Nelson Salasar Marques**

If Marques is a single surname, type a
If Salasar Marques is a compound surname, type b
If the name does not contain a surname element, type c
If you aren't sure how to answer this question, type help

****b**

Do you know which element of the compound surname this person prefers to be known under?

****no**

Is this person's language Portuguese?

**why [The user is asking the system why it wants this piece of information.]

[The system responds by displaying the relevant rule from its knowledge base:]

IF (1) the surname is a compound surname, and
(2) the person's preference is unknown, and
(3) the person's language is Portuguese
THEN enter under the last element.

Is this person's language Portuguese?

**yes

The name heading which I have formulated is:
Marques, Nelson Salasar
I have also formulated a required cross-reference:
Salasar Marques, Nelson

In the case of this example, the rule displayed in response to the user's "why" query makes it clear how the form of name chosen was determined. In cases where the user did not understand the system's result, the query "how" could be entered. This would prompt the system to display the sequence of rules upon which its result was based.

3. USES OF EXPERT SYSTEMS

3.1. Expert system functions and domains

Expert systems have been developed to perform a variety of functions in a wide range of domains. The following is a commonly-accepted list of the broad functional categories of expert systems a tion and an example of a possible application area of each:

Category	Application
Interpretation	Image analysis
Prediction	Weather forecasting
Diagnosis	Medical diagnosis
Design	Computer configuration
Planning	Job-shop scheduling
Monitoring	Power plant regulation
Debugging	Software correction
Repair	Automobile maintenance
Instruction	Intelligent tutoring
Control	Battlefield management

(Hayes-Roth 1983)

Some examples of the broad domain categories in which expert systems have been developed are:

Aerospace
Agriculture
Chemistry
Computers
Education
Electronics
Energy management
Engineering
Finance
Geology
Information management
Law
Manufacturing
Mathematics
Medicine
Meteorology
Military science

Existing expert systems range from small-scale efforts which barely qualify as expert systems to very large and carefully documented research projects and production systems. Since many systems are known only within the organizations where they were originated, it is impossible to estimate how many expert systems have been developed or are in use today. One source which includes systems "commercially available, proprietary programs used in house, and projects still in the prototype stage" identifies 475 systems (Walker 1986). None of the expert systems in the area of librarianship known to the authors of this report are included.

3.2. Some examples of expert systems

The systems described in this section are examples of successful applications of expert systems technology.

PUFF is an expert system developed at Stanford University. PUFF is capable of interpreting the results of respiratory tests in order to assist a physician in diagnosing the presence and severity of lung disease in a patient.

XCON and XSEL are two expert systems developed jointly by Digital Equipment Corporation (DEC) and researchers at Carnegie-Mellon University which have knowledge of DEC's VAX computers. XCON is used to help configure a customer's order for a large-scale computer system by determining what components are required to produce a complete system. XSEL helps DEC sales personnel select the components necessary to meet a customer's needs and designs a floor layout for them.

DELTA was developed by General Electric to assist maintenance personnel in troubleshooting and repairing diesel electric locomotives. DELTA helps technicians diagnose problems and guides them through the entire repair procedure.

DIPMETER ADVISOR, developed by Schlumberger, the oil field services company, is capable of making expert inferences about geological formations by interpreting data supplied by a dipmeter, a device which takes measurements in an oil borehole.

3.3. Library applications of expert systems

A review of the library and information science literature reveals a strong interest in expert systems. The most advanced research work discussed in the literature is in the area of information retrieval. A number of articles discuss such uses for expert systems as intelligent gateways to online databases and as intelligent online interfaces to assist the user in making effective use of a complex information retrieval system (for example, Burton 1985, Fidel 1986, Kehoe 1985, and Shoval 1985). Some work has also been done in the use of expert systems for reference assistance (for example, Smith 1986, Waters 1986).

In such traditional technical processing areas as acquisitions, cataloging and classification, and serials control, relatively little work has been discussed in the literature, although some attention has been focused on the potential for use of expert systems in cataloging and classification. Molholt (1986) has argued that there is a close relationship between the characteristics of frame-based expert systems (discussed in section 4.1 of this report) and the way information scientists organize knowledge, for example, when constructing thesauri. Jones (1984) suggests that there is potential for the use of expert systems in classification work.

In descriptive cataloging, efforts have been made to apply expert systems to cataloging rules. Chang has developed a system which questions the user to elicit the cataloging problem being experienced and directs the user to the appropriate rule in AACR 2. At this time, as McCone notes (1987), this system is essentially an automated index to AACR 2. The LIBLAB project at Linkoping University has also considered the potential for applying expert systems technology to AACR 2 (Hjerppe 1985). The experience gained by researchers on this project has led them to conclude that far more research is needed into the nature of cataloging expertise before an operational expert system to assist in the cataloging process would be feasible. Merely attempting to convert the rules in AACR 2 into expert system format is likely to be of little use, since a number of factors besides rule application (for example, interpretation of the document being cataloged) are highly important to expert cataloging. The complexity of interpretation in the cataloging process is also illustrated in the description offered by Jeng (1986) of a conceptual model of an expert system for determining title proper.

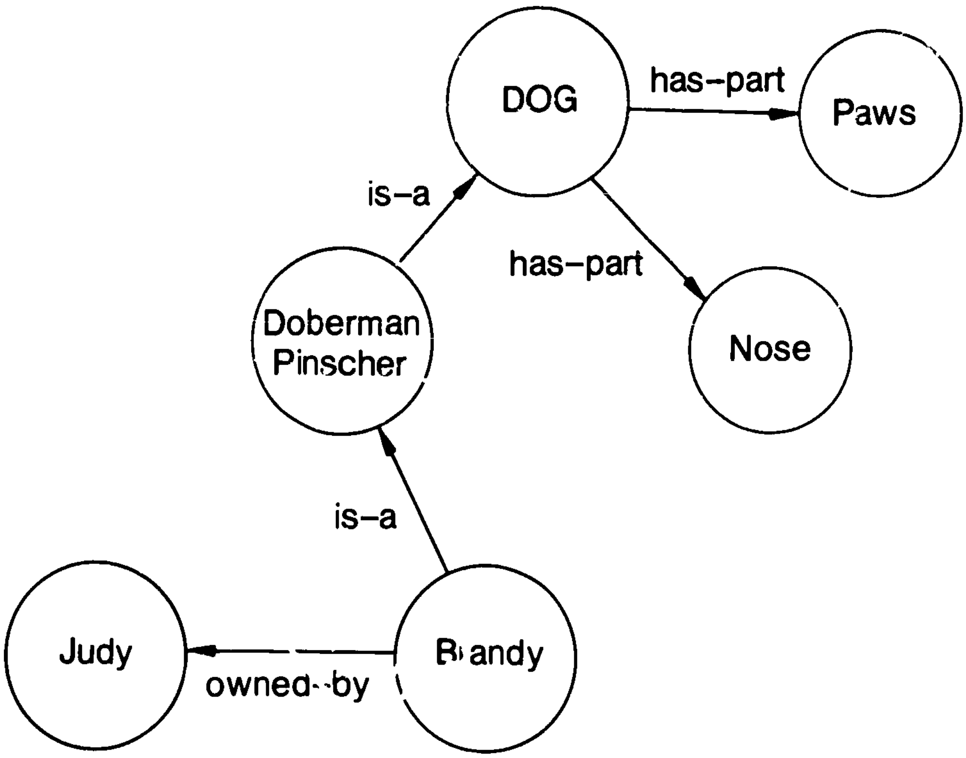
4. HOW EXPERT SYSTEMS FUNCTION

In this section, the components of an expert system listed in section 2.1 are discussed in more detail.

4.1. The knowledge base

The knowledge base is the component of an expert system where facts pertinent to problem-solving in the domain are represented. Methods for representing expert knowledge in the knowledge base may be either declarative (representing facts or assertions) or procedural (representing actions) or a combination of the two. Some examples of declarative methods are semantic networks, logical representation schemes,

FIGURE 4-1
EXAMPLE OF A SEMANTIC NET



object-attribute-value triplets, and frames. The leading example of the procedural method is the production rule approach. Each of these methods is discussed in this section.

A semantic network (or semantic net) is a graphical representation of properties and relationships of objects, situations, concepts, and the like. The semantic net consists of points called nodes connected by links called arcs which describe the relationships between the nodes.

Figure 4-1 is an illustration of a semantic net. Among the relationships represented in this net are the following: "Brandy is a Doberman pinscher," "A Doberman pinscher is a dog," and "Dogs have paws." Examples of arcs which are illustrated are is-a and has-part. These express how the nodes which they connect relate and allow for the inference of new facts. For example, from this semantic net one could infer that Brandy is a dog and that Brandy has paws, even though neither of these facts was explicitly stated.

An important feature of the semantic net is property inheritance: nodes lower in the net can inherit properties from higher nodes, so that properties applying to all levels of a hierarchy need not be repeated at each level. In the example, the parts of a dog can be stored once at the "Dog" level rather than having to be repeated at the breed and the individual dog levels.

Among logical representation schemes, the most commonly employed is predicate logic. In predicate logic a proposition consists of objects, persons, concepts, and the like (arguments) about which something is stated (the predicate). For example, the proposition "A component of a cataloging record is the series area" might be stated in the form of predicate logic as

has component (cataloging record, series area)

where "cataloging record" and "series area" are the arguments and "has component" is the predicate which in this case expresses a relationship between the arguments. Predicate logic lends itself well to inferences. For example, if we add another proposition "A component of the series area is the title proper of series," stated as

has component (series area, title proper of series)

it can be inferred from these two propositions that title proper of series is a component of a cataloging record, although this was not explicitly stated.

In the object-attribute-value triplet method of representing factual knowledge, objects are entities in the domain, attributes are properties associated with the object; these attributes may possess values. For example, in the triplet "cataloger-productivity-2.4 units per hour," "2.4 units per hour" is the value of the attribute "productivity" associated with the object "cataloger." An advantage of this method of knowledge representation is that it facilitates data-gathering by the system through questions posed to the user in the form "What is the [value] of [attribute x] of [object y]?"

Frames are very powerful and versatile data structures which are especially good for representing stereotyped knowledge about an object, concept, or event. Frames can be readily organized into a hierarchical network of nodes and relationships like a semantic net, with a frame constituting each node. A frame is subdivided into a collection of attributes, called "slots." Values may then be associated with the attributes. In some cases default values may be assigned. Slots can also have associated with them procedural attachments which are executed when information in the slot changes. Examples of such procedural attachments are the "if-added" procedure, which executes when new information is placed in the slot, and the "if-needed" procedure, which executes when information is needed in the slot but is not available.

To consider how a system using frames might work, suppose that a frame-based library order system included a frame called "Special Order." Such a frame would include slots for the elements of information needed to process a special order, such as order number, bibliographic information, price, vendor code, and claim date. In most of these slots explicit values would be inserted. Slots whose initial values could be predicted, such as claim date, might have such values generated by default. Some slots might have procedural attachments. For example, an "if-added" procedure attached to the vendor code slot could search a vendor data base for the full name and address associated with the code.

The production rule method is the most commonly used technique for expert system knowledge representation. Production rules typically take the form IF-THEN where the

IF portion describes a condition, antecedent, or situation, and the THEN portion describes the resulting action, consequence, or response. A production rule from the knowledge base of an expert system in cataloging might read:

IF forms of a name vary in fullness,
THEN choose the form most commonly found.

In a rule-based expert system, facts known about the current situation are compared against the domain knowledge expressed as a set of such rules. When the IF portion of a rule is satisfied, the THEN portion is executed. This may result in a new fact being inferred and added to the working memory for possible matches with the IF portion of other rules or may cause the action specified by the THEN portion to be taken.

4.1.1. Uncertainty

Since expert systems deal with the real world, they must often cope with knowledge which is uncertain or incomplete. Various techniques have been employed to deal with uncertainty. These include certainty factors, fuzzy logic, and probability. In addition, nonmonotonic reasoning systems have been developed to provide a means for the revision of knowledge which new knowledge shows to be untrue (Rolston 1988).

Certainty factors are closely associated with MYCIN, perhaps the most famous of all expert systems (Buchanan 1984). In MYCIN the certainty factor (CF) is an expression of confidence in the truth of a particular fact derived from the expert's experience or from any evidence which may be available. The CF is a number ranging from -1 (complete certainty that the information is false) to +1 (complete certainty that the information is true). Embodied in a production rule the certainty factor takes the form shown in the following example:

IF 1) The stain of the organism is gram positive, and
2) The morphology of the organism is coccus, and
3) The growth conformation of the organism is chains
THEN There is suggestive evidence (.7) that the identity of the organism is streptococcus.

The CF is the value ".7" in the THEN portion of the rule, indicating a 70% degree of confidence in the validity of the conclusion.

When the system derives a conclusion by chaining together a number of rules qualified by certainty factors, it must apply the various CF's to a mathematical formula to determine an overall certainty factor for the final result.

Fuzzy logic attempts to deal numerically with imprecise concepts such as tall, short, old, young. An expert assigns a number between 0 and 1 to express the degree of likelihood that something is a member of a particular set. For example, considering the concept "young," the number ".9" might be assigned to express a high degree of likelihood that a person under eighteen would be a member of the set of young people. ".6" might be assigned to indicate that a person of the age of twenty-nine is somewhat likely to be a member of the "young" set. Though the process of assigning such values may be largely intuitive, once they have been assigned they can be manipulated in a well-defined mathematical manner.

Probability is a method of representing uncertainty based on statistical decision theory, for example, Bayes' Theorem. While very sound theoretically, this method can be utilized only if there are sufficient data associated with a particular assertion to make a mathematical calculation of its probability. This may often not be the case in the domains to which expert systems are applied.

A nonmonotonic reasoning system (NMRS) keeps track of tentative beliefs, pieces of knowledge which are potentially incorrect because they are based on assumptions. Should new knowledge be added which shows a tentative belief to be incorrect, that belief and any beliefs that are dependent on it are revised. An example of an NMRS is the Truth Maintenance System (TMS), which functions as a knowledge base management system. Each time a new piece of knowledge is added, TMS is called and takes action as necessary to revise dependent beliefs so that knowledge base consistency is maintained (Rolston 1988).

4.2. The inference engine

The inference engine is the control structure which organizes, controls, and executes the steps followed by the expert system in searching its knowledge base in order to arrive at a solution to the problem being worked on. There are two basic search approaches which may be employed: blind search and heuristic search.

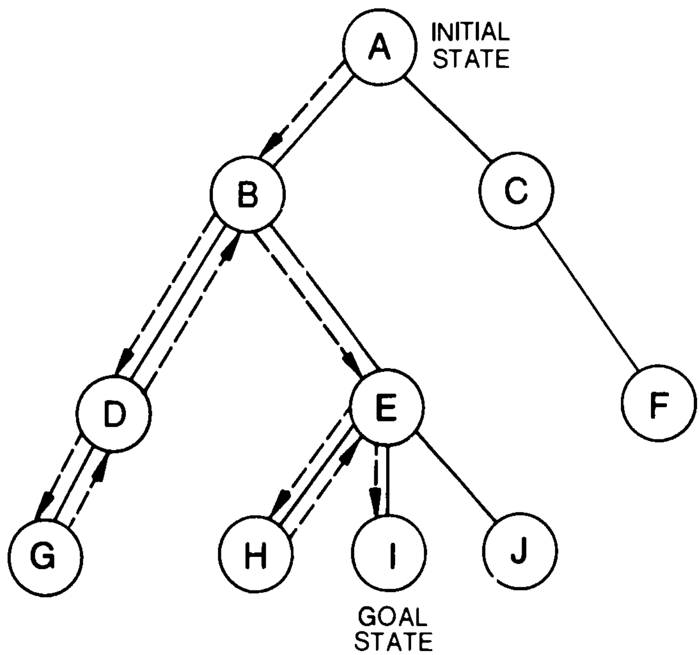
In the absence of a guiding search strategy, the blind search involves consideration of all possible paths from the problem's initial state to a goal state. This might be acceptable for small problems, but would be hopelessly inefficient for large-scale problems. This is due to the phenomenon known as combinatorial explosion. Combinatorial explosion is the potential for a geometric expansion of possibilities at each level of a search tree. A good example is provided by the seemingly very simple game of tic-tac-toe. There are nine possibilities for the first move. For the second move, there are eight possible responses to each of the nine possible first moves, for a total of seventy-two possible second moves. For each of these seventy-two moves, there are seven possible responses, for a total of 504 possible third moves, and so on. Remarkably, the total number of possible game states in tic-tac-toe is 9!, or 362,880.

To make them more efficient, blind searches are guided by search strategies. These include depth-first search, breadth-first search, and forward and backward chaining.

The depth-first search pursues a single path in the search tree until either a goal state, a dead end, or an arbitrarily designated cutoff depth is reached. If a dead end or cutoff point is reached, the system will backtrack to a point where another path can be pursued. Depth-first search is potentially more economical in its use of memory than breadth-first search. However, a serious deficiency of this method is that it does not insure an optimal solution. In a large-scale problem an arbitrary cutoff depth is necessary to prevent the system from using tremendous amounts of time pursuing unproductive search paths to great depths. But if the cutoff point is too shallow, a solution will never be reached, and if the cutoff point is too deep, the solution path will be non-optimal (Shapiro 1987). Figure 4-2 provides an example of a depth-first search.

The breadth-first search examines all nodes in each level of the search tree before moving on to the next level. This method will find the shortest path to a solution (if one exists) but is not practical if the solution is deep in the search tree, since

FIGURE 4-2
DEPTH-FIRST SEARCH



Search path ABDGDBEHEI illustrates a depth-first search

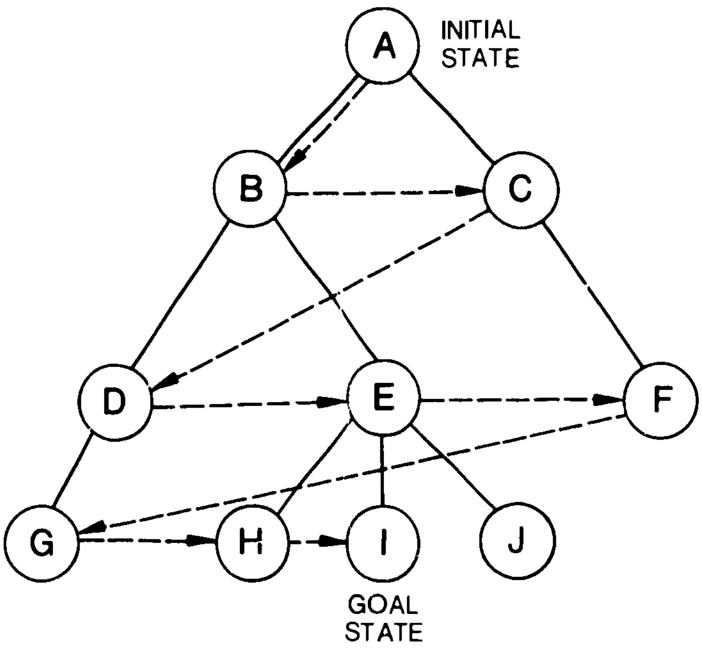
successively deeper levels of the search tree are subject to combinatorial explosion, and each level must be generated before the next level can be examined (Shapiro 1987, Mishkoff 1985). Figure 4-3 provides an example of a breadth-first search.

Forward chaining is a data-driven search method. The inference engine starts with known facts which it attempts to match with facts in the knowledge base. When such matches occur, new facts are inferred, which can then be matched with other facts. This process continues until no new conclusions can be reached. At this point either a goal state has been reached or, if there were no facts to support a goal state, the search has failed; in either case, additional goals can then be sought. The backward chaining method starts from a potential goal state and works backward through the search tree seeking facts which support that goal. If the available facts do not support the goal, the search fails, at which point another potential goal is selected, and the search is tried again. Forward chaining is more appropriate when the number of initial states is greater than the number of goal states. When the number of goal states exceeds the number of possible initial states, backward chaining is more efficient (Frenzel 1986, 1987).

Though the above-discussed blind search techniques are sometimes used, the expert system searching process is generally modified by the use of heuristic search to limit the number of alternative solution paths which must be considered. Heuristic search generally involves a process of evaluating the current node in the search tree and predicting the

FIGURE 4-3

BREADTH—FIRST SEARCH



Search path ABCDEFGHI illustrates a breadth-first search

quality of succeeding nodes as to their desirability as subsequent nodes in the path to the goal. Two examples of the various heuristic search techniques which might be employed are <u>difference reduction</u> and <u>hill-climbing</u> (Mishkoff 1985).

<u>Difference reduction</u> (also referred to as <u>means-ends analysis</u>) uses a combination of forward and backward chaining to shorten the distance between the current node and a goal state by setting subgoals. For example, suppose that a certain heuristic would attain the desired goal state. Suppose further that it is not possible to apply the heuristic from the current node, but that there is a nearby node from which it could be applied. Employing the concept of difference reduction, the ultimate goal would be temporarily set aside in favor of the subgoal of reaching the nearby node enabling use of the desired heuristic. By applying this process repeatedly, smaller and smaller subproblems, each with search spaces much smaller than the original problem, can be solved. When all the subproblems are solved, the main problem is also solved.

Using the <u>hill-climbing</u> approach, when the evaluation of a node reveals that it is not the goal state, the difference between that node and the goal state is calculated. A comparison of the sequence of calculated differences indicates whether the search is moving closer to or farther away from the goal state. If movement is away from the goal state, the search backtracks until a new search path can be taken.

4.3. Working memory

The working memory is the dynamic memory where the current status of an expert system consultation is stored. It contains the initial information provided to the system to enable the search process to start. As rules are examined and executed, the working memory is updated to contain new facts inferred, values established, and the like, which are then available for further use in the decision-making process. The working memory also keeps track of which rules the system has examined and executed and in what sequence, so that the reasoning process employed can be provided to the user if required.

4.4. User interface

The user interface is software which permits interaction between the user and the expert system. The interface may contain pre-formulated questions and menus to facilitate the collection of data needed by the system in order to conduct the search of its knowledge base. The interface also provides the means of displaying the solution reached by the system.

For the expert system to be most useful, the user interface should include an explanation facility. This permits the user to ask the system to display the reasoning process by which a particular result was achieved. The explanation facility not only enhances the credibility of the system but also greatly facilitates debugging when unexpected or erroneous results are produced.

5. EXPERT SYSTEM DEVELOPMENT PROCESS

5.1. Defining a problem suitable for an expert system

Before embarking upon an expert system development project, a problem suitable for the application of this technology must be identified. Expert systems are not well-suited for many types of problems and should be applied only when they are possible, justified, and appropriate (Waterman 1986).

For an expert system to be possible, the task to be carried out must have certain characteristics:

(1) The task must require only cognitive (i.e., not physical) skills and must not require "common sense" reasoning. (It has proven virtually impossible to develop an expert system with common sense);

(2) There must exist human expertise related to the problem. There should be experts in the domain who can articulate their methods and who are in general agreement as to what constitute solutions to the problems the expert system would be intended to solve;

(3) The task must fall within a reasonable range of difficulty. If it is so difficult that it cannot be taught to a novice by an expert or that it takes days or weeks for an expert to carry out, the size and cost of an expert system to tackle the task would be prohibitive. (However, if a large problem can be segmented, its component parts might be suitable for expert systems);

(4) The task must be reasonably well-understood. If basic research is necessary to find solutions to the problem, it is not possible to develop an expert system to approach it.

For an expert system to be justified, conditions such as the following should be satisfied:

(1) There exists a scarcity or anticipated loss of human expertise. For example, the impetus for Ford Europe's expert system research and development effort was the shortage of experts in customer service workshops (Bernold 1986);

(2) A system performing a portion of the total problem would be beneficial to the organization;

(3) There are prospects for a reasonable payoff in terms of dollar savings or returns.

For an expert system to be appropriate, the problem to be solved should have such characteristics as these:

(1) The nature of the problem should be such that it lends itself to symbolic manipulation and heuristic solutions. Incorrect or non-optimal results must be tolerable. Otherwise, conventional algorithmic programs, more efficient and possibly less expensive to develop, might be more appropriate;

(2) The problem must be one which is not too simple. Prerau (1985) has suggested that the ideal problem for an expert system would be one which a human expert could solve in a range of time between a few minutes and a few hours. Expert systems cost too much to develop to apply them to problems which can be solved in seconds; such problems are better candidates for solution by algorithmic programs or perhaps even by manual techniques such as flowcharts or decision-logic tables.

5.2. Developing the expert system

This section provides a very brief outline of the complex process of expert system development. For a detailed discussion of this topic, see Waterman (1986) or Rolston (1988).

Expert systems are developed in an incremental fashion, that is, the system starts out as a small prototype which gradually develops to take on increasingly complex tasks as the organization of the system improves and the amount of knowledge represented increases. The key figures in the development process are the knowledge engineer, the expert systems development specialist, and the domain expert, the human expert in the particular realm of expertise in which the expert system is intended to operate.

Once a problem suitable for an expert system has been identified, the development process begins with identification of the major features of the problem by the knowledge engineer and the expert. The next step involves determining what concepts and strategies aptly describe the process of solving problems within the particular domain of expertise. Once this has been accomplished, the knowledge engineer can begin to express the most important concepts in a formal manner within the framework of an expert system building tool. These formal concepts are then embodied in a working prototype program, which covers a small portion of the problem which will ultimately be handled by the expert system.

The prototype is tested, typically by the domain expert, to evaluate such characteristics as its usability, its reasoning processes, and the appropriateness of its decisions. Working with the domain expert, the knowledge engineer makes revisions and additions necessary to improve performance, and the system is tested again. In some cases, all or much of what has been done may have to be discarded and the process re-started from the prototype stage. The process proceeds in this fashion, with the system growing by increments until it has arrived at an appropriate scope and has achieved a degree of speed, accuracy, and efficiency which permits it to be field-tested. During field testing, new problems with the system are likely to come to light, necessitating further refinements to the system before it can be considered for production use.

Perhaps the most daunting and time-consuming aspect of expert system development is knowledge acquisition. This is the process whereby the knowledge engineer obtains the expert knowledge which is to be represented in the expert system. The techniques employed in gathering this knowledge may include extensive reading of documentation related to the domain, observing the domain expert in problem-solving activities, and engaging in a series of very intense and structured interviews with the domain expert. By these techniques the knowledge engineer attempts to ascertain exactly what steps, in minute detail, the expert takes in solving problems typical of those which the expert system is intended to address.

A variety of factors make knowledge acquisition difficult. Among these is the fact that the expert may never have conceptualized the process by which a particular conclusion is reached, so, when asked to explain how a problem was solved, responds in terms too vague or general to be represented in a method amenable to machine manipulation. Especially difficult to capture may be the heuristics which distinguish the expert from the novice. In an on-the-job situation, the expert may apply such heuristics almost instinctively.

5.3. Expert system building tools

There are a large number of programming environments, or expert system building tools, now available to assist the knowledge engineer in the construction of an expert system. These tools fall into two main classes: programming languages and knowledge engineering languages, or shells. Associated with these programming environments are a variety of support facilities.

5.3.1. Programming languages

Programming languages that are used to build expert systems are of two types:

(1) problem-oriented languages, such as C and PASCAL, which were designed for conventional software development, and

(2) symbol-manipulation languages, such as LISP and PROLOG, which were designed to represent and perform operations upon concepts expressed as symbols, for example, as list structures or logical representations of concepts.

Developers have used virtually all the major programming languages to create expert systems. For example, C, with its speed and flexibility, has become increasingly popular as an expert system building tool (Frenzel 1987). However, the symbol-manipulation languages possess special characteristics which make them especially suitable for use as expert system development tools. LISP, for example, features flexible symbol manipulation, automatic memory management, and uniform treatment of program code and data.

In the United States, LISP is by far the most widely-used AI programming language. Developed in the 1950's by John McCarthy, one of the pioneers of AI, LISP has retained its popularity due to its versatility and powerful capabilities. Many versions of LISP are available for all classes of hardware, and, as noted above, there exists a special class of computer known as the LISP machine which has specialized features to support LISP programming. In Europe, PROLOG is the most popular AI language. PROLOG is based on predicate logic and contains its own built-in inference engine. As with LISP, many versions of PROLOG are available. A good capsule discussion of how LISP and PROLOG function may be found in Frenzel (1986). Besides these two a number of other AI programming languages have been developed.

5.3.2. **Knowledge engineering languages**

Knowledge engineering languages, sometimes referred to as shells, are specialized programming environments tailored for expert system development. Components include a knowledge representation facility, an inference engine, and a variety of support capabilities. Some shells, such as those which were developed by removing the knowledge base from an existing expert system, are relatively specialized, emphasizing one particular knowledge representation scheme and one principal inferencing technique. More generalized and versatile are the large hybrid tools, which support multiple knowledge representation schemes and inferencing techniques and feature very sophisticated support facilities (Rolston 1988)

Commercially available knowledge engineering languages exhibit a great deal of variety with respect to such considerations as hardware requirements, knowledge representation methods and inferencing techniques employed, and support facilities. Furthermore, existing tools are subject to modification, and new tools are being brought onto the market. Selection of a tool for a development project must therefore be based on a careful analysis of the most current information.

An example of a shell which runs on the LISP machine class of hardware is KEE, developed by Intellicorp. KEE supports frame-based, rule-based, and other knowledge representation methods. Its inference engine uses both forward and backward chaining. An example of one of the many shells designed to run on the IBM PC microcomputer is EXSYS, developed by EXSYS, Inc. This tool supports rule-based knowledge representation with a built-in certainty factor mechanism. Its inference engine is capable of both backward and forward chaining.

A "Catalog of Expert System Tools" may be found in Waterman (1986). This catalog provides brief descriptions of a large number of AI programming languages and knowledge engineering languages.

5.3.3. Support facilities

Support facilities consist of a set of adjunct capabilities for a specific programming environment. Typical support capabilities include: debugging aids, input/output (I/O) facilities, explanation facilities, and knowledge base editors (Waterman 1986).

Debugging aids include tracing facilities which allow programmers to monitor the path of the system search and break packages which allow the programmer to stop and examine system execution at a predetermined point.

I/O facilities which may be provided include capabilities for run-time knowledge acquisition and operating system accessibility. Run-time knowledge acquisition is a facility whereby the expert system can ask the user to supply information which cannot be found in the knowledge base. Operating system accessibility allows the expert system to communicate with the local operating system, to request initiation of other programs and systems, to provide information to those systems and programs, and to receive information in response from them.

Explanation facilities may be provided as part of the knowledge engineering language environment. Retrospective reasoning explains how the system reached a particular intermediate or final state; this represents the most prevalent form of explanation facility. A hypothetical reasoning facility allows the user to understand an outcome's sensitivity to a particular fact or element of system knowledge. A counterfactual reasoning capability is a type of explanation facility which can explain why the system failed to reach an expected conclusion.

Knowledge base editors may vary from very basic text editing capabilities to highly developed support capabilities. Automatic bookkeeping capabilities document relevant information for knowledge base changes such as the identity of the responsible individual and the date of the change. Syntax checking facilities monitor such aspects of input statements as spelling and grammar. Consistency checking attempts to identify semantic conflicts between newly entered information and existing system knowledge.

5.4. Development environment

The environment for development of expert systems includes the hardware, software, and human resources required to build them.

Hardware environments include specialized LISP machines, and microcomputers, minicomputers, and mainframe computers. The choice of a hardware environment depends on the software tools chosen, the complexity of the expert system application, and other constraints such as cost and the availability of existing computers.

Microcomputers at present may not be adequate to support the development of expert systems with large knowledge base requirements or complex search strategies. The selection of tools for mainframe computers appears more limited than that for either LISP machine or minicomputer alternatives. Unless cost considerations mandate use of already available hardware, either a LISP machine or a minicomputer appears to be the development machine of choice.

The software alternatives should be evaluated in the context of the particular application to be developed. As noted above, the expert system building tools which are commercially available vary widely. Careful research is needed to insure selection of a tool which is well-matched to the knowledge representation and control strategy requirements of the application to be developed. A fundamental decision is whether to select a knowledge engineering language, which may be easier and faster for system developers to use, or an AI programming language, which may allow developers to tailor the system more precisely to the needs of the application.

Human resource requirements must be evaluated in the context of the organization in which development work is to take place. Important considerations include the degree of knowledge engineering expertise already existing in the organization and how ambitious the expert system development program is to be. To provide an illustration, a start-up project team for development of expert systems within an organization without in-house knowledge engineering expertise might require a staff of the following size and composition:

(1) One project manager, experienced in AI project management, drawn from outside the organization;

(2) One or more domain experts possessing the expert knowledge to be incorporated in the expert system;

(3) One knowledge engineer, perhaps obtained on contract from an outside source;

(4) One or two knowledge engineers in training from within the organization;

(5) One or two experienced AI programmers, perhaps obtained on contract from an outside source;

(6) One or two AI programmers in training from within the organization;

(7) Clerical support.

5.5. System development limitations and pitfalls

Expert systems technology is not yet fully mature. As a result, there are certain fundamental difficulties which may impact upon any expert system development project. In addition, there are pitfalls to beware of in the expert system development process (Waterman 1986).

At this stage of their development, expert systems have certain inherent limitations. These limitations must be taken into account when evaluating the feasibility of any development effort planned for the near future. Some of these inherent shortcomings are:

(1) Limited ability to represent either temporal or spatial knowledge;

(2) Inability to perform common sense reasoning;

(3) The inability of an expert system to recognize its own limitations;

(4) The difficulty expert systems have in dealing with erroneous or inconsistent information.

Expert system building tools also have some significant limitations at this time. Chief among these are:

(1) Inability of the tool to perform knowledge acquisition, so that this remains the most time-consuming aspect of system development;

(2) Inadequacy of the tool in helping to refine a system's knowledge base, so that a large effort is required to obtain a small improvement in system performance; and

(3) Inflexibility and lack of generality of the expert system building tool, so that, for example, particular types of knowledge cannot be represented well, mixed knowledge representation schemes cannot be handled easily, or adequate user interfaces may be difficult to develop.

In addition to these fundamental limitations related to the current state-of-the-art in expert systems, there are a variety of potential mistakes and pitfalls in the processes of planning and developing an expert system in a particular domain of expertise. Some of these are:

(1) Choosing the wrong problem. The problem may be too complex to be solved within the constraints of available resources or so large in scope that a system to address it would be unmanageable;

(2) Choosing an inappropriate system development tool;

(3) Trying to develop an expert system without calling upon experienced knowledge engineers;

(4) Planning to deliver a full-fledged working expert system by an established deadline;

(5) Trying to develop a system with an expert who cannot devote adequate time to the project, who cannot communicate his or her expertise adequately, or who is not committed to the project;

(6) Using an expert who is not a legitimate and recognized authority in the problem domain, so that high-quality rules are not forthcoming and system credibility is compromised;

(7) Making improper use of multiple experts, so that inconsistencies and shallow reasoning creep into the system;

(8) Failing to test constantly during development, so that it becomes apparent after a great deal of effort has been invested that fundamental concepts have been overlooked.

As a result of the limitations inherent in the current state-of-the-art of expert systems technology, most systems operating today function as <u>assistants</u> (performing a useful but limited subset of an expert's task) or <u>colleagues</u> (performing a relatively significant subset of an expert's task). As Rolston (1988) observes: "Few existing systems could actually come close to replacing a human expert in a complex domain."

PART II
EXPERT SYSTEMS IN LIBRARY OF CONGRESS TECHNICAL PROCESSING
A FEASIBILITY STUDY

Part II of this report documents the second phase of our investigation of the potential for using expert systems technology within the Processing Services department of the Library of Congress. This phase of the investigation utilized the understanding of expert systems technology developed in the first phase of the project and documented in Part I of this report.

The purpose of the second phase was to conduct a preliminary investigation to determine from among a number of functional areas within Processing Services whether there were any promising candidates for the application of expert systems technology and, if so, what these potential applications were and what benefits might accrue to the department should expert systems be implemented in these areas. The scope of our investigation was limited to library technical processing functions, such as acquisitions, cataloging, and serials control. We made no attempt to evaluate the potential usefulness of expert systems for other types of activities carried out within the department, such as marketing or financial management.

This phase of our work was intended to identify promising candidates, not to make all of the determinations which would be necessary before actual development could commence. We have therefore not subjected potential applications to detailed cost/benefit analysis, nor have we engaged in systems design or made specific hardware and software recommendations. However, we have discussed the value of such expert system applications and have characterized the applications in ways which may suggest either a system design or particular hardware or software. Such descriptions are intended only to illustrate how an application might function and do not substitute for a formal identification of requirements.

Our findings are based solely on circumstances which apply within Processing Services of the Library of Congress and may therefore have no applicability to the technical processing operations of other institutions.

6. METHODOLOGY

We began this phase of the investigation by consulting the directors for acquisitions and overseas operations, bibliographic products and services, and cataloging, to identify those technical processing operations which they felt were most promising for consideration during this phase of work and to identify potential resource personnel. Based upon the recommendations of the directors and upon subsequent interviews with resource personnel, we conducted investigations in each of the following operational areas:

(1) Cataloging in Publication

(2) Decimal Classification

(3) Descriptive Cataloging

(4) National Union Catalog

(5) Exchange and Gift

(6) Order

(7) Overseas Operations

(8) Serials Management

(9) Subject Cataloging

(10) Shelflisting

The principal method of gaining information about each operation was to conduct initial interviews in each operational area with an individual who could provide an overview of the work performed in the unit and follow-up interviews as necessary with other personnel. For each of the areas judged to contain potential applications of expert systems technology we interviewed and observed the work of individuals who are regarded as experts. In addition to the staff of operating units, we consulted with specialists of the Automation Planning and Liaison Office and the Office for Descriptive Cataloging Policy.

7. DETERMINATION OF EXPERT SYSTEMS FEASIBILITY

Expert system feasibility studies may utilize any of a number of approaches to determine whether a particular operation is suitable for consideration as an expert system application area. In this investigation we posed two fundamental questions for each Processing Services operation under consideration as a candidate for an expert system:

(1) Does this operation constitute a suitable domain for an expert system?

(2) How would an expert system in this domain benefit the department?

7.1. Characteristics of a suitable expert system domain

In section 5.1 we presented some general criteria for determining whether a problem would be suitable for the application of an expert system. For this phase of the project we needed a somewhat more detailed list of criteria for evaluating the suitability for expert systems of the various domains considered. The following lists of "essential" and "highly desirable" characteristics considered in studying the potential for applying expert systems to particular operations within Processing Services are based largely upon an especially comprehensive set of such criteria developed by Prerau (1985).

7.1.1. Essential characteristics of a suitable expert system domain

These are characteristics which a domain must exhibit, at least to some extent, in order to be considered a viable candidate for the application of an expert system. We therefore made a judgement about each of the following with respect to each operation evaluated.

(1) Tasks to be performed and problems to be solved in the domain require expert knowledge, judgement, and experience;

(2) The task requires primarily symbolic (rather than algorithmic) reasoning;

(3) The task requires the use of heuristics;

(4) The task typically takes an expert a few minutes to a few hours to perform;

(5) The task is relatively narrow, well-bounded and self-contained;

(6) Some degree of incorrect or non-optimal results can be tolerated;

(7) The need for the task is projected to continue for several years;

(8) The domain is fairly stable, with changes tending to be gradual and evolutionary;

(9) No radical changes which would redefine the task or establish an alternative means of performing it are being planned;

(10) There are recognized experts working in the domain today.

7.1.2. **Highly desirable characteristics for a suitable expert system domain**

If a domain possesses the following highly desirable characteristics, the potential for applying an expert system to it is greatly enhanced. We therefore attempted to make a judgement concerning each of these factors with respect to each operation evaluated. In some cases, there was not enough information to make a definitive assessment regarding each of these factors.

(1) The task is decomposable, so that development can begin with a small subset of the complete task;

(2) Some degree of incomplete task coverage can be tolerated, at least during system development;

(3) There is written documentation covering the domain;

(4) Test cases are available;

(5) The user interface is not likely to require excessive effort;

(6) The skills required to perform the task are taught to novices;

(7) Experts would agree on whether the system's results were accurate or not;

(8) System inputs and outputs can be clearly defined;

(9) The task cannot be handled satisfactorily by conventional (algorithmic) programming approaches;

(10) The number of important concepts related to the task being addressed does not exceed a few hundred;

(11) There is an expert available to work with a development project. The expert has credibility, has a long period of experience in the domain, could commit substantial time to system development, can communicate his or her expertise effectively, and is cooperative and easy to work with.

7.2. Benefits

Once a domain has been determined as suitable for application of expert systems technology, a determination is required of the benefits which might accrue to the organization if an expert system were put into place. We used the list of expert system benefits which appears in Part I, section 2.3, of this report, in our evaluation of benefits which might result from the application of an expert system in a given functional area.

8. POTENTIAL APPLICATIONS

The most promising opportunities for the application of expert systems which we identified were:

(1) A Shelflisting Assistant:

(2) A Series Consultant;

(3) A Subject Cataloging Consultant.

Each of these is discussed in detail in this section. A discussion of our reasons for not selecting the other operational units which we investigated as potential application areas for expert systems is contained in section 9.

8.1 Shelflisting Assistant

8.1.1. Background information

Shelflisting is a highly labor-intensive task at the Library of Congress. About 90 staff members and supervisors are necessary under present procedures to accomplish the shelflisting of some 170,000 items per year.

At its most basic, shelflisting is an essentially algorithmic process. A relatively simple table is used to translate the designated cataloging data item into an alphanumeric "cutter number" to complete the call number. For example, applying the cutter table to the name "Galbraith, John Kenneth," one can quickly derive a cutter of ".G35." If shelflisting were no more complex than this, it would clearly fail to meet several of the criteria for a suitable expert system domain. In practice, however, two very significant considerations complicate the process.

First, an objective of this process is fitting the item shelflisted into its proper alphabetical place within the assigned classification. Because of the enormous size of the

Library of Congress shelflist, the number indicated by the cutter table is frequently not appropriate. In the example above, the cutter ".G35" may have already been used. Or, if there are large numbers of authors in the particular classification, the cutter ".G35" may not put a work by "Galbraith, John Kenneth" into the proper alphabetical sequence. The number yielded by applying the cutter table is therefore merely suggestive; the process of fitting the item into its appropriate slot takes place at the manual shelflist itself.

The other major complicating factor is that a large percentage of items are not cuttered simply by a single cataloging data item such as main entry. For example, the classification schedules require two cutters in some class numbers: an item might be cuttered first for its geographical or subject coverage, then for the main entry heading. Or a class number might have a special subarrangement unique to it. In some classes, cutters ".A" and ".Z" are reserved for special purposes. The person shelflisting the work must therefore determine by consulting the classification schedules and other pertinent documentation which bibliographic data elements must be used and how they are to be used in completing the call number in a manner consistent with other items in that classification.

8.1.2. Conceptual view of the Shelflisting Assistant

The Shelflisting Assistant proposed here is an interactive system which would appropriately complete the call number in most instances. It would be capable of detecting anomalous shelflisting patterns and calling these to the attention of its operator. It would be easily updated both to correct deficiencies in its own operation and to allow for new developments within the classification scheme. It would feed back to the user its results in a manner which would facilitate a quick determination of the accuracy of those results, and it would be capable of displaying to the user the rules it used to achieve a given result.

The system as we conceive of it would require a data base and an expert system.

As already noted, proposing specific system hardware and software is outside the scope of this project. However, in order to visualize how this system might work, a possible approach to design and hardware configuration of the data base is described. The required data base for shelflisting would consist of records to which new shelflisting decisions could be compared. Each record in this shelflisting data base must contain a subset of data from the full bibliographic record including the LC call number and all fields from which the cuttering in the LC call number was derived. Such a data base could reside on the Library's mainframe, on a departmental minicomputer, or on a workstation. At the workstation level one way of constructing a stand-alone version of this data base might be to load the relevant bibliographic data items from each MARC and PREMARC record onto CD-ROM. Between successive editions of this CD-ROM data base a workstation would need to consult both the CD-ROM data base and a smaller dynamic data base of all shelflisting decisions made since the most recent issue of the static CD-ROM data base. This dynamic data base might reside on hard disk.

The expert system would represent the classification schedules in the form of rules which would specify how the cutter should be determined in the case of each class number. This does not imply that there would be a rule for each class number. Rather, there might be a rule for each unique way of cuttering which would identify the class numbers handled in that manner. In the more typical cases, one rule might identify the means by which many class numbers would be handled. In other cases, a special rule might be necessary for a single class number. The expert system would also contain rules for actually deriving the cutter number as well as for invoking any intermediate tables which may be required in order to construct the complete cutter number.

The system should have a very easy-to-use interface. It would ask the operator for the class number assigned by the subject cataloger. By comparing that information to its rule the system would know what bibliographic information it needed and could then specifically request this from the operator. With this information the system would then approach the relevant portion of the data base to determine where the item being shelflisted would properly fit. The system would "know," for example, that in class "D849" a work cuttered by the name "Samarin" should fit between works by "Saito" (D849.S2) and "Sanguinetti" (D849.S224). Accordingly, it might complete the call number by assigning cutter number ".S22."

To enhance the system's credibility and facilitate the prevention of error, the user interface might display the system's result in context. For example, the new shelflist record might be displayed with the two records before it in the data base and the two records after it, so that the user can be satisfied that the new item has in fact been fitted in properly. If the end result achieved by the system seemed odd or erroneous, the interface would be capable upon request of displaying the rules it used to derive the number.

The system would be capable of noting certain anomalies. For example, if its rule for a certain class number called for single cuttering but the items already in the data base under that class were double cuttered, it would note this and call it to the operator's attention.

8.1.3. Feasibility of the Shelflisting Assistant

In evaluating the domain of shelflisting against the criteria discussed in section 7.1 of this report, it is clear that many of the characteristics listed are satisfied. The task is narrow and well-bounded. The domain is very stable. There are experts performing the work who have experience and credibility and who can communicate their expertise. The task could be readily decomposed for prototyping of a small subset of the complete domain. Inputs and outputs of the process can be very well-defined.

Some of the evaluation criteria, however, are not so clearly well-satisfied. It might legitimately be asked whether the tasks performed in this domain truly require expertise and the use of symbolic knowledge and heuristics. Although shelflisting does not require as high a level of expertise as the other domains described in this report as candidates for expert systems, we believe that expertise is needed, due to the complexity of interpreting and applying the classification schedules and related documentation and the necessity of interpreting the patterns and practices implicit in the shelflist itself. The complicating aspects of the work which were described above insure that the work requires symbolic reasoning and some degree of heuristic decision-making and is not merely algorithmic.

As for the suitability criterion which refers to tolerance of incorrect or non-optimal results, it is clear that outright errors must be avoided in completing the call number. It is an accepted fact given the current state-of-the-art that expert systems make mistakes. We believe, however, that the model that the investigators have proposed has safeguards against excessive error. Most important is the fact that it would interact with a human operator and display its results in a manner which would allow that person to evaluate them in a suitable context. Another safeguard is the system's proposed ability to recognize anomalies.

8.1.4. Benefits of the Shelflisting Assistant

The chief benefit which might result from implementing a system such as the Shelflisting Assistant would be an enhancement of the productivity of the shelflisting operation. Each item could be shelflisted more quickly for the following reasons: (1) Routine consultation of the manual shelflist would be eliminated--the system would "fit" the new item into its proper slot; (2) Routine consultation of the classification schedules and other documentation would be eliminated--the expert system would contain that information.

Another benefit which might be anticipated would be consistency. Once the rules had been refined and were working properly, they would yield consistent results. This is important in a domain which, though requiring expertise, is fairly repetitive and production-oriented, so that the risk of error or inconsistency due to fatigue and loss of concentration is always present.

Finally, as the system evolved, it would eventually include in a form readily accessible to less-experienced staff members the knowledge of complex, unusual, and problematic shelflisting situations which presently must be handled by or in consultation with an experienced expert. Non-experts could use the system to achieve expert-like results, and the knowledge of the most experienced experts would be retained if they left the organization.

8.2. Series Consultant

8.2.1. Background information

Some experienced observers suggest that series work is the most problematic aspect of descriptive cataloging at the Library of Congress. Although some 200 monographic catalogers must deal with series as a part of their work, few of these catalogers are equipped to handle the most difficult decisions without consultation. Frequently, the specialists in the Office for Descriptive Cataloging Policy, especially two with particular expertise in series work, must resolve the most complex cases. The problem is sufficiently serious that this office has decided to embark upon a two-year training effort designed to insure that each monographic cataloging section will have at least one series expert. Further evidence supporting the contention that series is an especially problematic area is provided by the NACO libraries, who have had more trouble achieving independent status for series authority work than for other categories of work they submit to LC.

Many factors make series work a problem. Some of these are:

(1) A series is a serial and may therefore display all the difficulties characteristic of serials, such as title changes, numbering peculiarities, and the like. Monographic catalogers are often unfamiliar with strategies for coping effectively with these serial problems;

(2) The rules and procedures related to series are numerous and complex;

(3) Series practices have changed significantly over the years, making it difficult to relate new pieces to existing series which were established under differing rules and procedures from those which currently apply; and, very importantly

(4) A series cannot be treated in a vacuum but rather must be dealt with in the context of the existing catalog of serial and monographic bibliographic records, with all its complexity and diversity.

An important consideration with respect to series work is that, because of cooperative cataloging, increasing numbers of catalogers at other institutions are now experiencing the same problems with series work which are so perplexing for LC catalogers.

8.2.2. Conceptual view of the Series Consultant

The system proposed would interact with a cataloger to provide guidance and assistance in carrying out the following broad categories of tasks:

(1) Establishing a new series, complete with proper heading, references, and treatment, based on the appropriate cataloging rules and procedures.

This function might be carried out along these lines: The cataloger would be prompted to supply data appropriate to the variable fields of a series authority. If the cataloger needed help with any component, such as how to qualify the heading, the expert system's knowledge base, which would contain rule and procedure information about series, could be immediately queried, preferably through easy to manipulate means such as menus. The system would also have the knowledge necessary to supply or suggest some of the appropriate fixed field data elements, treatment information, and cross-references either automatically or with minimal cataloger effort. Once the necessary data had been formulated, the expert system would generate a series authority either in manual or machine-readable form for addition to the data base. If the system could access the series authority file, it might be possible for the authority record to be automatically uploaded, eliminating the need for duplicate keying.

(2) Resolving complex series questions and problems

The expert system would include the knowledge and heuristics which the best experts currently use in determining how to deal with all the troublesome aspects of series work, such as how to deal with problematic changes in the way a series is presented, how to interpret ambiguous information, how to relate a new piece to a series established under earlier practices, and the like. The system might assist the cataloger in identifying the exact nature of the problem by displaying increasingly detailed levels of menus. The system would be capable of requesting whatever information it needed to evaluate the problem. Eventually, the system would either recommend a solution or recognize that it lacks the knowledge necessary to address that particular problem.

Though no attempt has been made to describe in full detail how this system might work, it is clear that one feature which would greatly enhance its usefulness is the capability for a consultation to be suspended. Should the consultation reach a point at which the expert system needs information which the cataloger cannot readily supply, the cataloger should be able to suspend the consultation and resume it later at the point of suspension.

8.2.3. Feasibility of the Series Consultant

Of the possible expert system applications which we have identified, the Series Consultant most closely resembles expert systems which have been successfully developed in other domains. As described, this system would perform five of the broad functions which are typically cited as being appropriate for expert systems: design, diagnosis, debugging, repair, and instruction (Hayes-Roth 1983).

We believe that series work satisfies every one of the criteria set forth in section 7.1 of this report for suitability as an expert system domain. Handling complex series problems requires substantial expert knowledge and experience. The reasoning employed is chiefly symbolic, and our interview with a series expert made it clear that many heuristics are applied. The task is narrow and deep, as an expert system task should ideally be. Finally, there are experienced, credible, and articulate experts working in this domain.

8.2.4. Benefits of the Series Consultant

This is a domain in which expertise is scarce. An expert system such as we have described would make this scarce expertise more widely available, helping all catalogers achieve expert-like quality and consistency in this difficult aspect of their work. Beyond catalogers at the Library of Congress, such an expert system could be made available to assist participants in the National Coordinated Cataloging Program (NCCP) and NACO who have a need to perform series work which conforms to LC practice.

In addition, because expertise in this domain is scarce, there is the danger of loss of expertise should the most knowledgeable experts leave the organization. The Series Consultant would provide a means for retaining this knowledge. Retention of knowledge is an important issue in this domain for another reason. Because many of the more difficult series problems are seen only rarely, humans, even though they may receive special training in series work, may forget from one occurrence to the next how some of these are to be handled. The Series Consultant's knowledge would not be subject to loss through disuse.

Finally, the Series Consultant should make a positive contribution to organizational efficiency. It would facilitate prompt and accurate resolution of difficult problems without extensive consultation of documents or human experts.

8.3. Subject Cataloging Consultant

8.3.1. Background information

Some 80 professional level staff members are engaged in the work of subject heading assignment and classification. The work of subject catalogers is challenging due to the size and complexity of the subject heading and classification structures into which newly-cataloged items must be fitted. These structures are supported by a very large body of documentation, and in the course of interviewing subject cataloging experts, we determined that good practice requires frequent and at times extended consultation of this documentation. In addition, experts in the Office of the Principal Subject Cataloger may have to be consulted in the case of particularly difficult or unusual problems.

The process of assigning subject headings is complicated by the need to make such determinations as (1) whether any permutation of the term selected by the cataloger to represent a subject concept has been established for use as a heading or as a reference; (2) what the precise form of the subject heading is--the order of words and the number and case of each word; and (3) whether a heading may be subdivided by such means as geographic subdivisions, free floating subdivisions, or other subdivisions specific to the main heading, and if so, precisely what form such subdivisions must take.

The process of determining an item's classification is also complicated by a variety of factors. For example, the same subject may be classified very differently depending on what aspect of the subject is being dealt with. For example, a book about copper as a chemical element is classified in "QD," but a book about copper metallurgy is classified in "TN." Classification is also complicated by the structure of the classification schedules, which employ such techniques as the use of numerous tables for deriving cutter numbers to refine precisely the representation of a topic.

8.3.2. Conceptual view of the Subject Cataloging Consultant

The Subject Cataloging Consultant would replace all of the documentation issued by the Library of Congress in support of subject cataloging. The expert system component of the Consultant would include subject heading and classification policies, interpretations, and procedures. In addition to the expert system component, the Consultant would interact with a number of machine-readable data bases. These would include the bibliographic, name, and subject authority files, already available in a well-defined machine-readable record structure, and the classification schedules, for which work to develop a machine-readable record structure is under way. Data bases to represent the most commonly used geographic subdivisions and the free-floating subdivisions would have to be developed. In addition, a machine-readable thesaurus would be required.

The system would receive input from the cataloger in the form of a term expressing a subject concept. The system would stem this term and match it against a thesaurus. The system could also receive input in the form of an authorized subject heading either known to the cataloger or derived from a search of the subject authority file. Using this input the system would conduct a search in the subject authority file and suggest to the cataloger an authorized subject heading or headings. For terms established but not authorized for use as headings (such as cross references) the system would locate the appropriate authorized heading and proceed. For instances in which broader or narrower terms were available, these headings would be displayed for the cataloger. If the subject term input included words implying some limitation by geographic scope, the system would, for those headings which may be divided geographically, attempt to verify the form of the geographic subdivision by consulting the geographic subdivision data base and complete that portion of the heading. If the subject heading were one with which free-floating subdivisions are used, the system could, under the cataloger's guidance, search the data base of such subdivisions for appropriate free-floating subdivisions for use in conjunction with the heading assigned. At any point the system would allow the cataloger to request and view a set of bibliographic records using a given heading.

In addition to assisting with subject heading assignment, the Consultant would attempt to classify the item being processed, using a classification number either associated with the primary subject heading or located by performing a thesaurus-assisted search of the classification schedules. The expert system component would guide the completion of the classification number by applying the classification schedules with their associated tables.

8.3.3. Feasibility of the Subject Cataloging Consultant

In evaluating the domain of the Subject Cataloging Consultant against the criteria discussed in section 7.1 of this report, it is clear that a number of the characteristics listed are satisfied.

The work requires substantial expertise due to the complexity of interpreting and applying correct subject cataloging policy and practice for subject headings and classification using a large body of documentation. These complicating aspects of the work entail symbolic reasoning and heuristic decision-making.

Within the domain of subject cataloging there appears to be only a limited degree of tolerance for incorrect or non-optimal results. We believe, however, that the model we have proposed has safeguards against excessive error. The system proposed would routinely consult all of the sources required by good subject cataloging practice. In addition, the system is intended as a consultant which would interact with a human user and display its results in a manner which would allow them to be appropriately evaluated.

The domain of subject cataloging is relatively stable, with radical change rare and new and revised headings and classifications fitting within well-defined existing structures. Inputs and outputs of the system could be clearly defined. Finally, there are experts performing the work who have experience and credibility and who can communicate their expertise.

It might be asked whether this domain is too large and too broad to be a viable candidate for the application of expert systems. We feel that the tasks the expert system component would be called upon to perform appear on a conceptual level to be relatively narrow and well-bounded. Furthermore, the domain would seem to lend itself to segmentation for prototyping. This is not to suggest, however, that development of such a system as we have described would be easy. The amount of information to which this system would require access is considerable, and the work necessary to implement the necessary data bases and the thesaurus so that these would be available for interaction with the expert system would probably be extremely challenging.

8.3.4. Benefits of the Subject Cataloging Consultant

The Subject Cataloging Consultant would potentially benefit the Library by enhancing the productivity of the subject cataloging operation. It should allow items to be cataloged more quickly, since routine consultation of an enormous body of documentation would be eliminated. Further, the quantity of documentation currently employed almost ensures that shortcuts such as private files and annotations of dated material are in common use. Implementation of the Consultant would make such shortcuts available to all subject catalogers based on the most up-to-date information.

Another significant benefit which could be anticipated is consistency. Once the system was in place and operating successfully, it might yield somewhat more consistent results than may be possible at present. This benefit is significant in a domain which requires the application of expertise in a large and somewhat production-oriented environment. In such an environment the risk of error or inconsistency due to variations in practice or to fatigue and loss of concentration is always present.

Finally, as with the other applications we have recommended, this system would provide a means for retention of complex knowledge and scarce expertise now subject to loss when an expert leaves the organization

9. OPERATIONS NOT CHOSEN AS POTENTIAL APPLICATION AREAS

We did not consider the other Library of Congress operational units which we examined to be as promising for the application of expert systems technology as those described above. In this section, we have provided a brief summary of the considerations which appeared to rule out each of these as suitable application areas at this time.

Cataloging in Publication: The work of the Cataloging in Publication Division includes such tasks as maintaining liaison with publishers who participate in the program, receiving and preparing pre-publication materials submitted by publishers for CIP cataloging, receiving and preparing books published with CIP data for final processing, and maintaining the Library's pre-assigned card number program. These tasks are performed by a small number of personnel and are mostly high-volume in nature, requiring minutes or seconds to complete. Thus, expert system technology does not appear to be feasible or potentially beneficial within this operation.

Decimal Classification: The work of this division consists of the subject classification of a title using a numeric classification scheme. Although it may be possible to develop an expert system in this area, especially to provide assistance in synthesizing decimal numbers, possibly the most difficult aspect of this work, the benefits of such a system appear to be too small to justify the effort, given the small number of personnel who perform this work.

Descriptive Cataloging: General: The work of descriptive cataloging includes such tasks as identifying for each title cataloged a set of bibliographic elements which characterize that title, formulating these elements into a standardized bibliographic record, formulating uniform access points to each bibliographic record and creating authority records to document these, and performing associated maintenance work.

Descriptive cataloging is performed by a large number of personnel, and the amount of time required for completion of the total process involved for each item cataloged falls within the time frame appropriate for expert systems. However, the process consists of a large number of discrete steps, each of which an experienced cataloger may perform with little difficulty in a short amount of time. For example, although it is possible to envision how the highly rule-based processes related to choice of access points might be implemented as an expert system, in practice, these cataloging decisions might typically be made in less time than would be required to interact with an expert system.

Descriptive Cataloging: Name Authority Work: A name authority consultant similar to the series consultant which we described and recommended would probably be feasible, but less beneficial, since series authority work is regarded as more difficult than name authority work.

National Union Catalog (NUC): The principal work of this operation consists of editing for conformity with standard cataloging practice paper and machine-readable cataloging contributed by libraries for the National Union Catalog. In general, this work does not have any unique technical requirements beyond those found in descriptive cataloging. Therefore, NUC might benefit from an application of expert systems technology developed for cataloging. However, since a more basic level of cataloging expertise and higher levels of production are characteristic of NUC by comparison to other cataloging units, separate evaluation criteria might be required to assess the benefits of an expert system for cataloging within NUC.

Exchange and Gift: Work within this division includes a variety of activities relating to establishing and servicing exchange agreements and soliciting and processing gifts. The number of personnel performing each type of work is fairly small, and many of the activities of the division require rapid, high-volume performance of individual tasks. Accordingly, the potential for an expert system applied to any of the tasks within this operation to yield substantial benefits does not appear great. In addition, development work is proceeding on an automated acquisitions system which will address some of the needs of this operation, making this an inappropriate time to consider expert system development.

Order: The Order Division is responsible for the processing of both special and blanket orders and for subscriptions. As with Exchange and Gift, tasks performed are varied, and the number of people performing each task is small The blanket order process possesses some characteristics which suggest that it might be an appropriate application area for expert systems technology Selection of a blanket order vendor, ongoing assessment of vendor performance, and determination of whether to renew a blanket order with the current vendor all represent complex decisions which might be assisted by an expert system. However, the decisions related to the blanket order process which are most challenging and thus most likely to benefit from the use of expert systems are made in conjunction with departments other than Processing Services, so that consideration of an expert system in this area was outside the scope of our investigation In addition, some of the needs of the Order Division will be addressed by the automated acquisitions system now being developed. For these reasons, this division does not seem to be a potential application area for expert systems technology at present.

Both the Exchange and Gift and Order Divisions might benefit from using an expert system which captured, maintained, and interpreted the Library's collection development policies. Capturing the expertise of a small number of individuals with many years of experience and knowledge of the Library's collections and its collection development policies would seem to provide a strong impetus for considering the development of an expert system. However, the responsibility for the definition, maintenance, and use of the Library's collection development policies rests with persons outside of Processing Services, so that it was not within the scope of our investigation to conduct a detailed analysis of this potential application area

Overseas Operations: The overseas offices perform both acquisitions and cataloging tasks. The acquisitions component entails selection and purchase of materials for both the Library and for selected research library customers. At present, work is in progress to implement an automated system designed to support this function. The cataloging component of overseas operations might benefit from any expert system technology introduced for cataloging operations at the Library. Otherwise, development of expert systems for the overseas offices as a group or for any office in particular might be extremely difficult. The building of an expert system requires incremental development of the components of the system and close working relationships between the development team and the domain expert. This process might present considerable difficulties, given that the overseas offices are remote both from the Library and from each other.

Serials Management: Serials management involves physically sorting serials material, routing material both within and beyond the Serial Record Division, and maintaining accurate records of individual copies of serials material received by the Library. Rapid and accurate bibliographic identification of newly-received material and an efficient means for locating the corresponding serial record and recording information relevant to a new piece are among the important issues for serials management. Currently, new automated capabilities are being developed to support this operation. This fact coupled with the high volume of activity within serials management and the corresponding need for very rapid performance of individual tasks suggests that applying expert systems technology to this area may not be feasible at this time.

REFERENCES

Beerel, Annabel C. Expert Systems: Strategic Implications and Applications. Chiches· Ellis Horwood; New York: Halsted Press, 1987.

Bernold, Thomas, ed. Expert Systems and Knowledge Engineering: Essential Elements of Advanced Information Technology. Amsterdam; New York: North-Holland, 1986

Buchanan, Bruce G. and Shortliffe, Edward H., eds. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Reading, Mass.: Addison-Wesley, 1984.

Burton, Hilary D. "The Intelligent Gateway Processor: Effective Computer Access to Distributed Information Resources," Quarterly Bulletin of the International Association of Agricultural Librarians and Documentalists, v. 30, no. 3 (1985): p. 55-60.

Fidel, Raya. "Toward Expert Systems for the Selection of Search Keys," Journal of the American Society for Information Science, v. 37, no. 1 (Jan. 1986): p. 37-44.

Frenzel, Louis E. Crash Course in Artificial Intelligence and Expert Systems. Indianapolis: Howard W. Sams, 1986.

Frenzel, Louis E. Understanding Expert Systems. Indianapolis: Howard W. Sams, 1987.

Hayes-Roth, Frederick, Waterman, Donald A., and Lenat, Douglas B., eds. Building Expert Systems. Reading, Mass.: Addison-Wesley, 1983.

Hjerppe, R., Olander, B., Marklund, K. Project ESSCAPE: Expert Systems for Simple Choice of Access Points for Entries: Applications of Artificial Intelligence in Cataloging. Linkoping, Sweden: Linkoping University, Department of Computer and Information Science and University Library, 1985.

Hunt, V. Daniel. Artificial Intelligence & Expert Systems Handbook. New York: Chapman & Hall, 1986.

Jeng, Ling-Huey. "An Expert System for Determining Title Proper in Descriptive Cataloging," Cataloging & Classification Quarterly, v. 6, no. 2 (Winter 1986): p. 55-70.

Jones, Kevin P. "The Effects of Expert and Allied Systems on Information Handling: Some Scenarios," Aslib Proceedings, v. 36, no. 5 (May 1984): p. 213-217.

Kehoe, Cynthia A. "Interfaces and Expert Systems for Online Retrieval," Online Review, v. 9, no. 6 (Dec. 1985): p. 489-505.

McCone, Gary K. "Expert Systems, or, Librarian Ex Machina," LITA Newsletter, no. 30 (Fall 1987): p. 3-4.

Mishkoff, Henry C. Understanding Artificial Intelligence. Indianapolis: Howard W. Sams, 1985.

Molholt, Pat. "The Information Machine: A New Challenge for Librarians," Library Journal, v. 111, no. 16 (Oct. 1, 1986): p. 47-52.

Prerau, David S. "Selection of an Appropriate Domain for an Expert System," The AI Magazine, v. 6, no. 2 (Summer 1985): p. 26-30.

Rolston, David W. Principles of Artificial Intelligence and Expert Systems Development. New York: McGraw-Hill, 1988.

Shapiro, Stuart C., editor-in-chief. Encyclopedia of Artifical Intelligence. New York: Wiley, 1987.

Shoval, Peretz. "Principles, Procedures, and Rules in an Expert System for Information Retrieval," Information Processing & Management, v. 21, no. 6 (1985): p. 475-487.

Smith, Karen F. "Robot at the Reference Desk?" College & Research Libraries, v. 47, no. 5 (Sept. 1986): p. 486-490.

Walker, Terri C., and Miller, Richard K. Expert Systems 1986: An Assessment of Technology and Applications. Madison, Ga.: SEAI Technical Publications, 1986.

Waterman, Donald A. A Guide to Expert Systems Reading, Mass.: Addison-Wesley, 1986.

Waters, Samuel T. "Answerman, the Expert Information Specialists: An Expert System for Retrieval of Information from Library Reference Books," Information Technology and Libraries, v. 5, no. 3 (Sept. 1986): p. 204-212.