Report 81-31
Stanford -- KSL

Scientific DataLink

Expert Systems Research: Adapting Technical Knowledge for Computer-Based Consultation Systems.
Edward H. Shortliffe, Jul 1981

card 1 of 1

Heuristic Programming Project
Report No. HPP 81-31

July 1981

Strasbourg 1981

Sixth Symposium on Legal Data Processing

Saloniki, Greece   1-3 July 1981

EXPERT SYSTEMS RESEARCH: ADAPTING TECHNICAL KNOWLEDGE FOR COMPUTER-BASED CONSULTATION SYSTEMS

Paper presented by Edward H. Shortliffe, MD, PhD
Stanford University School of Medicine, Stanford, California 94305

1 Introduction

During the quarter century since the birth of "artificial intelligence" (AI), attempts to develop symbolic models of human reasoning processes have been a major focus of the ongoing research. It is only in the last half-dozen years or so, however, that several related AI research themes have come together in the formation of what is now known as "expert systems research" [1]. In this brief paper I would like to review the key aspects of AI and expert systems work. My goal is to acquaint the reader with the field and to suggest ways in which this line of research is relevant to the development of computer-based legal consultation systems.

1.1 Artificial Intelligence

Artificial Intelligence has been described as "the study of ideas that enable computers to do the things that make human beings seem intelligent" [29]. Unstated in this definition is the implicit assumption that the computer should have the ability to reason symbolically (rather than by combining numbers statistically or by using other computational manipulations that underlie conventional computer programs). Related assumptions are that intelligent programs should be able to acquire new knowledge and apply it appropriately; they should also be able to manipulate and communicate ideas.

Expert systems comprise only one of the application areas in which AI researchers work. Other research fields include:

(1) robotics (the development of machines that can intelligently respond to their environment in an adaptive fashion, generally to accomplish some physical task such as manipulations on an assembly line)

(2) automatic programming (the development of computer programs that are themselves capable of writing programs if they are given a symbolic description of what the new program should do)

(3) game playing (the development of programs that can play complex games; this has been a fertile area for the theoretical study of heuristic search techniques and the psychology of human problem solving)

(4) vision or scene analysis (the development of programs that can "understand" what is seen in a picture, e.g., systems to interpret medical x-rays pictures or the scene on. a television screen)

(5) natural language understanding (the development of programs that can understand free text, e.g., respond to questions, analyze printed texts, or translate from one language to another)

(6) speech understanding (the development of systems that are able to understand language and to translate analogue signals, such as microphone input, into their text equivalent)

(7) information processing psychology (the field of AI in which the researchers are primarily motivated by a desire to understand how human beings reason or solve problems; the computer becomes a useful experimental tool for testing theories and modeling human performance)

(8) intelligent computer-aided instruction (the development of teaching programs that understand the domain at a level of detail that allows them to form a model of the student and to customize the instructional session appropriately).

1.2 Knowledge Engineering

Of particular relevance to the study of legal reasoning is the AI research field that deals with the construction of knowledge-based consultation systems. "Knowledge" is the key word here, a concept that must be distinguished from "data". Computers have long been used to store data, but isolated datapoints do not become "knowledge" until they have been analyzed and summarized. We have accordingly suggested that there are at least four types of knowledge that should be distinguished from conventional statistical data. These characterize the information that must be available to an expert-level consultation system [25]:

(1) knowledge derived from data analysis (largely numerical or statistical);

(2) judgmental or subjective knowledge -- the kind that experts recognize is based on their own experience but which may be difficult to verify without complex and time-consuming studies;

(3) common sense, scientific, or theoretical knowledge -- these kinds of knowledge are often simple facts (e.g., cars are for transportation, Saloniki is in Northern Greece), but are typically symbolic in nature and must be "known" by an expert in the field;

(4) high-level strategic knowledge or "self-knowledge" -- this is the kind of knowledge that often distinquishes an expert from a well-trained novice in a field (e.g., an attorney's knowledge about how best to appeal to a certain kind of juror, or a cardiologist's favorite technique for deciding which diagnostic tests to use in assessing a patient with a new complaint of chest pain.

An expert system, then, is a computer program that contains, and can apply, the knowledge of a specialized domain. It uses this knowledge to make suggestions to users of the program who may not have the system's full range of expertise. The construction of such programs is known as "knowledge engineering" [12],[17]. Such efforts typically require close collaboration between human experts in the field and computer scientists familiar with expert systems research. In order to simulate the encounter between a non-expert and a human consultant, expert systems must contain all four kinds of knowledge I have outlined above. Thus they are much more than conventional data retrieval systems.

Fig. 1 shows a simplified schematic view of an expert system. The program itself sits between the domain expert (or experts) who have collaborated in construction of the system, and the intended system user. Since the human expert may not be generally available to all those who would like his advice, the expert system becomes a surrogate that "learns" about the field from the expert and serves as a consultant in his place. As the figure shows, there are three key types of information transfer during the encounter between an expert and the non-expert seeking his advice:

(1) the expert requests relevant information about the case under consideration;

(2) he offers a recommendation or conclusion based upon the data that are available; and

(3) if the non-expert requests it, he explains the basis for the decisions he has made.

Conventional approaches to computer-based consultation typically address only the first two of these items. It is when a system designer wishes to

KNOWLEDGE-BASED CONSULTATION

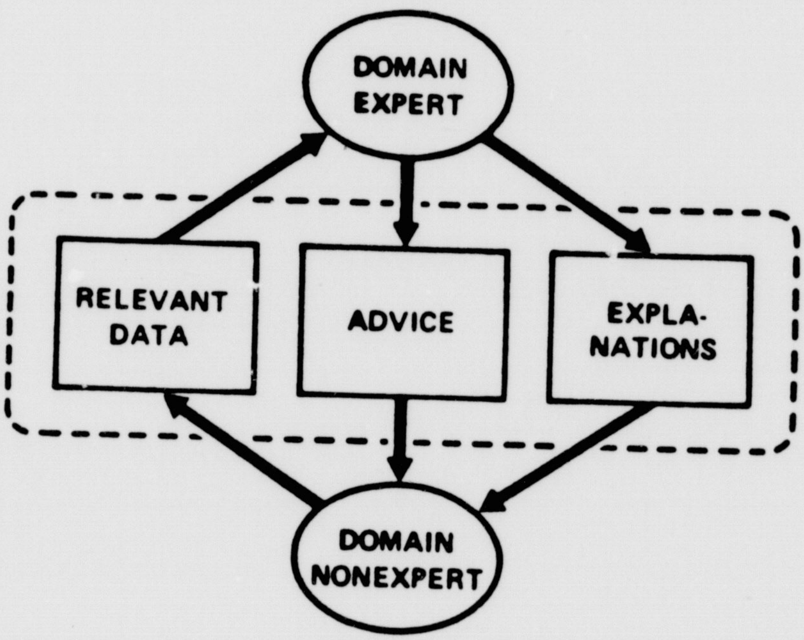

Figure 1

construct programs that can explain the basis for their decisions, in terms that the user can easily understand, that the techniques of knowledge engineering become particularly pertinent.

Although my own work has tended to deal with medical expert systems [22], the same issues that concern physicians are relevant to other potential users of consultation programs. I would be surprised if lawyers did not ask very much this same set of questions about a proposed consultation program:

(1) Do I need this system's help?

(2) Will it help me without being dogmatic?

(3) Does it justify its recommendations so that I can decide for myself what to do?

(4) Is it fast and easy to use?

(5) Is it designed to make me feel comfortable when I use it?

In the remainder of this paper, I shall indirectly address these questions by discussing the major themes in expert systems research. The goal of most such systems is, simply, to be able to answer all five questions in the affirmative.

2 The MYCIN System

One early example of an expert system is the MYCIN program, developed at Stanford University in the mid-1970's [24]. This six-year experiment spawned several newer projects, but it remains a good example of many of the issues that arise in considering the design and construction of knowledge-based programs. In the discussion below, I will use MYCIN frequently to provide examples of the recurring themes in expert systems research.

MYCIN was designed to advise physicians regarding the selection of antibiotics for patients with severe infections. This is an important clinical problem, and studies have shown that physicians do not always do well when selecting a treatment plan from among the large number of available antimicrobial agents [13]. When designing MYCIN, however, we were aware that the simple need for a consultation system would not be enough to guarantee its acceptance by physicians. It was also important that it reach decisions comparable to those of infectious disease experts, and that it be able to explain the basis for its decisions. Other issues outlined in the list of questions above also needed to be addressed, and artificial intelligence techniques seemed to offer potential solutions to these design considerations [22].

3 Knowledge Representation

I have described the four kinds of knowledge that an expert system typically requires in order to provide consultations in a specialized domain. An active area of AI research is the development of new techniques for encoding and manipulating such symbolic (non-numeric) knowledge. The major methodologies in use are production rules, frames, and semantic networks

[29], or combinations of these. The details of these options are not particularly important because most knowledge can be encoded in whichever formalism a system builder wishes to use. What <u>is</u> important is that the representation technique accomplish the following:

(1) It should allow domain knowledge to be kept separately from the computer program itself; in this way, system knowledge can be added or changed without reprogramming;

(2) It should facilitate the formulation of explanations that convey a line of reasoning that a human observer can understand and critique;

(3) It should facilitate a level of understanding of system performance such that an expert, observing the system in operation, can identify and easily correct missing knowledge, mistaken strategies, or frank errors; and

(4) It should of course interact with a reasoning methodology in such a way that the system reliably and predictably reaches good decisions and gives excellent advice.

In the case of MYCIN, we chose to use production rules [7] to encode the knowledge of infectious disease therapy [8]. Stanford researchers had previously gained experience with production rules in a large system that reasoned about chemical structure [2], and as early as 1970 had surmised that knowledge of the law could be encoded using such techniques [3]. A production rule is simply a conditional statement that indicates the circumstances under which a particular conclusion may be drawn. For example:

If: 1) The infection which requires therapy is meningitis, and
2) The patient has evidence of a serious skin or soft tissue infection, and
3) Organisms were not seen on the stain of the culture, and
4) The type of infection is bacterial

Then: There is evidence that the organism (other than those seen on cultures or smears) which might be causing the infection is staphylococcus-coag-pos (.75) or streptococcus (.5)

MYCIN contains about 550 rules such as this that deal with the diagnosis and treatment of bacteremia (bacteria in the blood) and meningitis (bacteria in the cerebrospinal fluid). Although the rules are encoded in the machine using a computer language, routines have been written to translate them into simple English so that they may be displayed and understood by the user as shown above. The strengths with which the conclusions may be drawn are indicated by numerical weights that we call certainty factors (CF's) [23]. Most expert systems that deal with uncertain inferences have been forced to develop scoring systems such as this for keeping track of the weight of evidence in favor of competing hypotheses.

Sprowl has experimented with production rules to represent legal knowledge in his system for drafting legal documents [27]. An example from his paper:

If: 1) the goods have not been delivered to the buyer, and
2) (the buyer has repudiated the contract to sell OR the buyer has manifested an inability to perform his obligations under the contract to sell or sale), and
3) the seller gives notice of his election to rescind to the buyer

Then: the seller may totally rescind the contract or sale.

As McCarty has observed [15], the main difficulty in constructing a computer-based legal consultation system is almost certainly the representation problem. As is true in other complex domains, the vagaries of human actions and beliefs interact with common sense knowledge of the world and precise domain knowledge in ways that defy simple delineation. The challenge at this stage of expert systems development is therefore to constrain the task domain in realistic ways that make the challenge for the program tenable but still allow its decisions to be useful for solving real-world problems. This point has also been demonstrated recently by Clancey in his efforts to use the knowledge in MYCIN for tutoring medical students [5]. Rules that had been adequate for excellent performance in a consultation system were found to be seriously deficient when they were used for teaching [4]. Fagan has also shown that MYCIN-like rules need substantial modification to deal with inference in settings where parameters are changing rapidly over time and techniques for analyzing temporal trends are crucial [11].

4 Knowledge Acquisition

As scientists gained experience in the construction of expert systems, it became apparent that the identification and encoding of expert knowledge was one of the most complex and arduous tasks encountered. The experts in a field often have difficulty distilling their knowledge in a form that is easy to structure formally. It is remarkable how superb a job many specialists can do at their work without really understanding how they perform accurately and reliably.

In the case of MYCIN, we needed to obtain rules regarding antibiotic selection and bacterial identification by talking with collaborating infectious disease experts. We quickly learned that people can identify and express their knowledge best in the context of an actual problem solving session. Thus we presented difficult cases to the experts, observed their performance as they gathered additional information that they felt was needed when deciding how to treat, and asked pertinent questions when the experts seemed to make leaps in logic or to ignore data that might have seemed important. In this way, key "rules" were obtained and then encoded using the production rule formalism that we had devised.

One member of our group became fascinated with the idea of an intelligent program that could actually guide this kind of knowledge-gathering session with an expert. The program he developed, known as TEIRESIAS [6], used the rules already known to MYCIN, and a set of learning strategies, to help an expert identify missing or erroneous knowledge and to update MYCIN's rules interactively. TEIRESIAS is an example of one form of "machine learning" (learning by being told, as opposed to learning by experience or by analogy).

The further refinement of knowledge acquisition methods remains an active area of expert systems research. Presently, however, there are major problems in identifying and encoding relevant domain knowledge. The most appropriate domains for expert systems research at this time may therefore be fields in which the knowledge is already highly structured and well specified[1]. I suspect that the most challenging issues in legal reasoning

[1]Our current research project, ONCOCIN [26], is a cancer chemotherapy consultation program. We chose this field for our work precisely because the protocols for treating these patients are already formalized, written down, and subject to rigorous analysis.

are precisely those in which formal knowledge of the law accounts for only a small part of the process of case analysis. It will be difficult to extract the more judgmental and "artistic" aspects of this process from the heads of expert attorneys.

5 Models of Reasoning

Once knowledge has been captured from experts, books, or other sources, and once it has been encoded using some representation scheme, the program must have an effective method for searching the knowledge base for relevant facts and tying them together in ways that simulate the human reasoning process. There are two key issues: 1) control of the reasoning process, and 2) management of uncertainty.

5.1 Control of the Reasoning Process

Entire books have been written on techniques for traversing a symbolic search space (e.g., [19]), and it will not be possible to discuss the approaches in any detail here. The approach we used in MYCIN is known as goal-directed reasoning or backward-chaining. MYCIN's rules are only loosely related to one another before a consultation session begins (i.e., the system builder does not explicitly tell the program how they relate to one another or when they should interact). MYCIN selects the relevant rules and chains them together as it considers a particular patient. Two rules chain together if the "conclusion" portion of one helps determine the truth value of a condition in the "premise" portion of the other. The resulting reasoning network is created dynamically.

MYCIN "reasons backward" from its recognized goal of determining therapy for a patient. It therefore starts by considering rules for therapy selection, but the premise portion of each of those rules in turn sets up new questions or subgoals. These new goals then cause new rules to be invoked and a reasoning network is thereby developed. When the truth of a premise condition is best determined by asking the physician rather than by applying rules (e.g., to determine the value of a laboratory test likely to be known by the doctor), a question is displayed. The physician enters the appropriate response and the program continues to select additional rules. Once information on the patient is obtained, some rules will fail to be applicable. In this way the invoked applicable rules will provide a customized patient-specific reasoning network for the case under consideration[2].

When reasoning steps are triggered by observations rather than goals, the problem solving process proceeds forward from data to conclusions -- so-called "data-driven" reasoning. Many expert systems use this alternative approach, and there are psychological data to suggest that "forward reasoning" more closely approximates the way in which human beings solve complex problems.

---

[2] Popp and Schlink have described a consultation program for lawyers, named JUDITH [20], which uses a similar goal-oriented reasoning mechanism.

5.2 Management of Uncertainty

When individual inference steps are less than certain, a new level of complexity is added to the interpretation of conclusions reached by a reasoning program. Not only are conventions needed for assigning weights to the individual steps or rules, but when two or more pieces of evidence support the same conclusion, some combining mechanism is needed for determining the net strength of the hypothesis. Expert system researchers have found formal probability theory less than satisfactory for these purposes, although some have attempted to adapt it for symbolic reasoning systems [9]. This issue has been particularly relevant for MYCIN where the knowledge expressed in a rule is seldom definite but tends to include "suggestive" or "strongly suggestive" evidence in favor of a given conclusion. In order to combine evidence regarding a single hypothesis but derived from a number of different rules, it was necessary to devise a numeric system for capturing and representing an expert's measure of belief regarding the inference stated in a rule. Conditional probabilities were not adequate for this purpose [23], and we devised instead the system of "certainty factors" mentioned above. These numbers lie on a -1 to +1 scale, with -1 indicating absolute disproof of a hypothesis, +1 indicating its proof, and 0 indicating the absence of evidence for or against the hypothesis (or equally weighted evidence in both directions). The relationship of the model to formal probability theory and the methods for combining evidence from diverse sources (rules and user estimates) have been described [23].

6 Generation of Good Advice

The ultimate test of the validity of the reasoning model used in an expert system is its ability to reach accurate conclusions and thereby to give valuable advice. The section on evaluation below mentions some of the difficulties that arise in attempting to show that a program is performing at the level of an expert in the field. There is a related controversy among expert systems researchers. Some do not believe it will be possible for computer-based consultants to function at the level of human experts until we better understand and can model the reasoning processes used by intelligent problem solvers. Others have argued that the details of a reasoning model are not important so long as the system reaches good decisions and can explain its reasoning in understandable terms. However, psychological studies of human problem solving (e.g., Elstein's work in the field of medicine [10]) are increasingly being studied by knowledge engineers as they attempt to develop systems that are more robust in their ability to use their knowledge for diverse purposes.

7 Explanation of Decisions

As I stressed in the discussion of Fig. 1, a crucial aspect of the interaction between an expert consultant and his client is the ability for the non-expert to request explanations regarding the recommendation he is receiving. My colleagues and I recently surveyed several hundred physicians and found that they cited explanation capabilities as a principal requirement for clinically acceptable computer-based consultation systems. In fact, most physicians felt that an explanation capability was more important than absolute program accuracy since they felt the latter was unrealistic. If a program can explain the basis for its conclusions, the system user is comforted because he knows he can request an explanation, examine the reasoning carefully, and decide for himself whether to follow the advice he has been offered. I firmly believe this is an important capability for expert systems in any domain; we must recognize that we are constructing <u>tools</u> for use by individuals who are often highly trained themselves and are not likely to be willing to turn over all their decisions to a computer program, regardless of how well it has been validated.

One of the great advantages of the rules used in MYCIN is the way in which they facilitate the development of mechanisms for explaining and justifying system performance. These capabilities also contribute greatly to MYCIN's educational role [4]. Since all questions asked by MYCIN are generated by a rule that is under consideration, and since all rules can be translated into English for display, a rule becomes an easily understood explanation if a physician wants to know why MYCIN has asked a particular question during a consultation session (Fig. 2). Similarly, rules become the basis for responding to free text questions that the physician may enter at the end of a consultation when he is trying to review the overall reasoning steps used by MYCIN (Fig. 3). As shown in the figure, we have developed some simple techniques for understanding free text questions entered by a physician obtaining a consultation [21]. MYCIN can thereby analyze a question and recover pointers to the rules that it used to make specific

PAGINATION ERROR. TEXT COMPLETE

BLANK PAGE

```
35) Is this a hospital-acquired infection?
** WHY
[i.e. WHY is it important to determine whether the infection was acquired
while the patient was hospitalized?]

      [2.0] This will aid in determining the organisms (other than those
            seen on cultures or smears) which might be causing the
            infection.
  It has already been established that
      [2.1] the infection which requires therapy is meningitis, and
      [2.2] organisms were not seen on the stain of this pending csf
            culture, and
      [2.3] the type of the infection is bacterial, and
      [2.4] neurosurgery is not one of the relevant items from the
            history of pt538, and
      [2.5] the age of pt538 is greater than 10 days
  Therefore, if
      [2.6] the infection was acquired while the patient was hospitalized
  then:
      there is evidence that the organism (other than those seen on
      cultures or smears) which might be causing the infection is
      e.coli (.75), staphylococcus-coag-pos (.3),
      pseudomonas-aeruginosa (.3), klebsiella-pneumoniae (.5)

  [back to question 35...]
  **
```

Figure 2

```
** WHY DID YOU TREAT E.COLI?

ITEM-1 is E.coli associated with meningitis.  I decided to treat ITEM-1 on
the basis of clinical evidence only.

The following rules concluded about: the likelihood that e.coli is one of
the organisms (other than those seen on cultures or smears) which might be
causing the infection.

RULE545 was used to conclude that e.coli is one of the organisms (other than
     those seen on cultures or smears) which might be causing the infection
     (.74).
RULE557 was used to conclude that e.coli is one of the organisms (other than
     those seen on cultures or smears) which might be causing the infection
     (.93).

The parameters most frequently used in these rules:
  the infection which requires therapy
  whether a smear of the culture was examined
  whether organisms were seen
  the type of the infection
```

Figure 3

decisions during the consultation. Despite the power that rules have provided for facilitating explanations in an expert system, we have learned that there are still serious limitations to the MYCIN approach. Clancey has discussed some of these in his thesis [4], and explanation remains an active area of ongoing theoretical research in our laboratory.

8 Validation and Evaluation

As the earliest expert systems have begun to mature, it has become necessary to consider the development of techniques for demonstrating that they perform at the level of an expert in their field. MYCIN has been evaluated now in three large studies, and we have discovered that the design of validation experiments is itself an area of considerable research interest. The details of MYCIN's evaluations will not be presented here, but a number of specific points are of interest. First, any evaluation is difficult because there is so much difference of opinion in this medical domain, even among experts. Hence, it is unclear how to select a "gold standard" by which to measure the system's performance. Actual clinical outcome cannot be used because each patient is treated in only one way and because a poor outcome in a gravely ill patient cannot necessarily be blamed on the therapy that had been selected.

Second, although MYCIN performed at or near expert level in almost all cases, the evaluating experts in an early study had serious reservations about the clinical utility of the program. It is difficult to assess how much of this opinion is due to actual inadequacies in system knowledge or design and how much is related to inherent bias against any computer-based consultation aid. In a subsequent study we attempted to eliminate this bias from the study by having the evaluators unaware of which recommendations were MYCIN's and which came from actual physicians [30]. In that setting MYCIN's recommendations were uniformly judged preferable to, or equivalent to, those of five infectious disease experts who recommended therapy for the same patients.

There will eventually be several additional questions to be answered regarding MYCIN and systems like it. Each question requires its own evaluation study. Are the programs used? If so, do the users follow the program's advice? If so, does the user benefit from the encounter with the expert system? Is the system cost-effective when no longer in an experimental form? What are the legal implications in the use of, or failure to use, such systems? The answers to these questions are years away for most consultation systems, but it must be recognized that they are ultimately just as important as whether the decision making methodology leads the computer to accurate and reliable advice.

9 Generalization

I emphasized earlier the importance of keeping the knowledge in an expert system separate from the actual computer program that processes that knowledge and generates advice. One important reason for separating the knowledge base from the program is the ease with which information can be added or corrected. Many expert systems include "editors" that allow a knowledge engineer to modify the knowledge base without having to change the program in any way.

There is a second important advantage to the separation of computer programs from the knowledge base within the machine. One can imagine the development of "interchangeable knowledge bases", each driven by the same computer program. The knowledge bases would have to be structured in accordance with the conventions used by the computer program, but once this was accomplished it should be possible for a single program to generate advice in any one of a number of different domains depending upon which set of rules was used.

This concept is an area of active research for workers in expert systems and is known as "generalization" or the development of "system building tools" [18],[28]. The key idea is to develop a general purpose computer program which can be used by knowledge engineers to build an expert system in any domain desired. Such programs must define conventions for knowledge representation and for the reasoning models used. The builders of new systems must in turn comply with these conventions.

Consider, for example, the system building tool that we have constructed from MYCIN. By removing all knowledge of infectious diseases (i.e., all rules), we were left we a set of programs that we call "Essential MYCIN" or EMYCIN [28]. EMYCIN can in turn be used to build an expert system in a new domain so long as it is possible to structure the knowledge in terms of production rules. Additional code was added to EMYCIN to allow the system designer to produce a knowledge base quickly and accurately. The new features include: 1) a terse, stylized, but easily understood language for writing rules; 2) extensive checks to catch common user errors, such as misspellings; and 3) methods for handling all necessary bookkeeping chores. Several consultation programs have been developed using EMYCIN, including consultants for medical problems and a consultant for structural engineering design.

10 Conclusions

It is readily apparent that expert systems research is relevant to the development of consultation systems that will be useful to attorneys and accepted by them. Indeed, the pioneering work of McCarty [14],[16] and others [20],[27] demonstrates that advanced knowledge engineering research efforts are already actively addressing problems from legal domains. Although the science of artificial intelligence is young, and expert systems research younger still, it is clear that the last decade has seen great strides in our understanding of the important theoretical issues [1]. Polished programs do not yet exist, but it is likely that we will see a first generation of marketable expert systems within the next decade as hardware and software advancements coalesce to permit the development of cost effective products.

References

1. Buchanan, B.G. Research on Expert Systems. Report HPP-81-1, Heuristic Programming Project, Stanford University, February 1981. To appear in Machine Intelligence 10, (D. Michie, ed.).

2. Buchanan, B.G. and Feigenbaum, E.A. DENDRAL and Meta-DENDRAL: their applications dimension. Artificial Intelligence, 11:5-24 (1978).

3. Buchanan, B.G. and Headrick, T.E. Some speculation about artificial intelligence and legal reasoning. Stanford Law Review, 23:40 (1970).

4. Clancey, W.J. Transfer of Rule-Based Expertise Through a Tutorial Dialogue. Doctoral dissertation, Memo STAN-CS-79-769, Computer Science Department, Stanford University, September 1979.

5. Clancey, W.J. Tutoring rules for guiding a case method dialogue. Int. J. Man-Machine Studies., 11:25-49 (1979).

6. Davis, R. Applications of Meta-Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases. Doctoral dissertation, Memo HPP-76-7, Heuristic Programming Project, Stanford University, July 1976.

7. Davis, R. and King, J. An overview of production systems. In Machine Representation of Knowledge (E.W. Elcock and D. Michie, eds), Wiley and Sons, New York, 1976.

8. Davis, R., Buchanan, B.G., and Shortliffe, E.H. Production rules as a representation for a knowledge-based consultation program. Artificial Intelligence, 8:15-45 (1977).

9. Duda, R.O., Gaschnig, J., Hart, P.E., et al. Development of the PROSPECTOR Consultation System for Mineral Exploration. Final report, SRI Projects 5821 and 6415. SRI International, Menlo Park, Ca. 1978.

10. Elstein, A.S., Shulman, L.S., and Sprafka, S.A. Medical Problem Solving: An Analysis of Clinical Reasoning. Harvard University Press, Cambridge, Mass. 1978.

11. Fagan, L.M. VM: Representing Time-Dependent Relations in a Clinical Setting. Doctoral dissertation, Heuristic Programming Project, Stanford University, 1980.

12. Feigenbaum, E.A. The art of artificial intelligence: themes and case studies in knowledge engineering. Proceedings of the 5th International Joint Conference on Artificial Intelligence, pp. 1014-1029, Cambridge, Massachusetts, 1977.

13. Kunin, C.M., Tupasi, T., and Craig, W.A. Use of antibiotics: a brief exposition of the problem and some tentative solutions. Anns. Int. Med. 79:555 (1973).

14. McCarty, L.T. Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning. Harvard Law Review, 90:837-893 (1977).

15. McCarty, L.T. Some requirements for a computer-based legal consultant. Technical Report LRP-TR-8, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey, July 1980.

16. McCarty, L.T. The representation of an evolving system of legal concepts: I. Logical Templates. In Proceedings Third National Conference of the Canadian Society for Computational Studies of Intelligence, Victoria, British Columbia, 14-16 May 1980, 1980.

17. Michie, D. Knowledge engineering. Cybernetics, 2:197-200 (1973).

18. Nii, H.P. and Aiello, N. AGE (Attempt to Generalize): A knowledge-based program for building knowledge-based programs. Proceedings of the 6th International Joint Conference on Artificial Intelligence, pp. 645-655, Tokyo, Japan, August 1979.

19. Nilsson, N.J. Problem Solving Methods in Artificial Intelligence. McGraw-Hill Book Company, Inc. San Francisco, Calif. 1971.

20. Popp, W.G. and Schlink, B. JUDITH: A computer program to advise lawyers in reasoning a case. Jurimetrics, pp. 303-314, Summer 1975.

21. Scott, A.C., Clancey, W.J., Davis, R., and Shortliffe, E.H. Explanation capabilities of knowledge-based production systems. Amer. J. Computational Linguistics, Microfiche 62, 1977.

22. Shortliffe, E.H. Consultation systems for physicians: the role of artificial intelligence techniques. In Proceedings Third National Conference of the Canadian Society for Computational Studies of Intelligence, Victoria, British Columbia, 14-16 May 1980, 1980.

23. Shortliffe, E.H. and Buchanan, B.G. A model of inexact reasoning in medicine. Math. Biosci. 23:351-379 (1975).

24. Shortliffe, E.H. Computer-Based Medical Consultations: MYCIN. Elsevier/North Holland, New York, 1976.

25. Shortliffe, E.H., Buchanan, B.G., and Feigenbaum, E.A. Knowledge engineering for medical decision making: a review of computer-based clinical decision aids. Proceedings of the IEEE, 67:1207-1224 (1979).

26. Shortliffe, E.H., Scott, A.C., Bischoff, M.B., et al. ONCOCIN: An expert system for oncology protocol management. Submitted to Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, British Columbia, August 1981.

27. Sprowl, J.A. Automating the legal reasoning process: a computer that uses regulations and statutes to draft legal documents. American Bar Foundation Research Journal, pp. 1-81, 1979.

28. van Melle, W. A Domain-Independent System that Aids in Constructing Knowledge-Based Consultation Programs. Doctoral dissertation, Memo HPP-80-11, Heuristic Programming Project, Stanford University, June 1980.

29. Winston, P.H. Artificial Intelligence. Addison-Wesley, Reading, Massachusetts, 1977.

30. Yu, V.L., Fagan, L.M., Wraith, S.M., et al. Antimicrobial selectionby a computer: a blinded evaluation by infectious disease experts. J. Amer. Med. Assoc., 242:1279-1282 (1979).