AD-A110 905 STANFORD UNIV CA DEPT OF COMPUTER SCIENCE F/G 9/2
RESEARCH ON EXPERT SYSTEMS,(U)
MAR 81 B G BUCHANAN N00014-79-C-0302
UNCLASSIFIED STAN-CS-81-837 NL

1 OF 1
AD A
110905

END
DATE FILMED
03-82
DTIC

March 1981

Also numbered:
HPP-81-1

Report No. STAN-CS-81-837

Research on Expert Systems

Bruce G. Buchanan

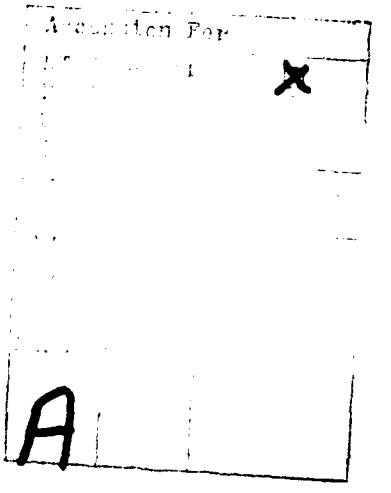
Research sponsored in part by

Office of Naval Research
National Science Foundation
and
Defense Advanced Research Projects Agency

Department of Computer Science

Stanford University
Stanford, CA 94305

Table of Contents



RESEARCH ON EXPERT SYSTEMS

Bruce G. Buchanan[1]
Computer Science Department
Stanford University

1 INTRODUCTION: What Is An Expert System?

All AI programs are essentially reasoning programs. And, to the extent that they reason well about a problem area, all exhibit some expertise at problem solving. Programs that solve the Tower of Hanoi puzzle, for example, reason about the goal state and the initial state in order to find "expert-level" solutions. Unlike other programs, however, the claims about expert systems are related to questions of usefulness and understandability as well as performance.

We can distinguish expert systems from other AI programs in the following respects:

Utility
Performance
Transparency

Designers of expert systems are motivated to build useful tools in addition to constructing programs that serve as vehicles for AI research. This is reflected in the tasks chosen. Solving the Tower of Hanoi puzzle, per se, is not a critical bottleneck in any scientific or engineering enterprise. But integrating mathematical expressions and determining molecular structures are important problems for scientists. Utility is the least important of the three criteria and is perhaps less definitional than a personal bias about whether expertise on trivial matters constitutes expertise at all. In some cases a task is chosen just because of its inherent importance. More often than not, a problem's significance for AI research is also a factor now because expert systems are still constructed by researchers.

The hallmark of expert systems is high performance. Using weak methods to perform any useful task requires expertise. And it requires skill on the part of the designer to shape these programs into "world-class" problem solvers. Thus we see relatively few expert systems and those we do see include considerable domain-specific knowledge codified over months or years. High performance requires that the programs have not only general facts and principles but the specialized ones that

---

[1]This work was supported in part by DARPA contract MDA 903-80-C-0107, NSF grant MCS 7903753 and ONR contract NOOO 14-79-C-0302. The paper is based on an invited lecture at the AISB Summer Workshop, Amsterdam, July, 1980. Mike Genesereth, Doug Lenat, Ed Feigenbaum, and Carroll Johnson provided helpful comments on an early draft. All members of the Heuristic Programming Project at Stanford have contributed to the ideas reported here; my debt to them is substantial. They are also partly responsible for errors in my thinking.

separate human experts from novices. Unfortunately for all of us, specialized expertise includes almost by definition, knowledge that is *not* codified in print. Thus high performance has to be courted with patience.

In addition to utility and performance, I have added transparency, or understandability, as a third characteristic of expert systems. This separates AI programs from very good numerical algorithms. It is not necessary that expert systems are psychological models of the reasoning of experts. However, they must be understandable to persons familiar with the problem. Statistical pattern recognition programs, for example, perform well on many important problems, but there is little illumination to be gained from rehashing algebraic manipulations of Bayes' Theorem.

2 CURRENT STATE

MYCIN [Shortliffe, 1976] represents a prototype of "Level-1" expert systems in many respects because it was built with the three criteria of utility, performance and transparency among its design goals. In the decade or so before MYCIN, roughly 1965-1975, DENDRAL [Lindsay, *et al.*, 1980] and MACSYMA [Moses, 1971] were developed as working tools. Other medical AI programs were developed then, most notably PIP (the MIT present illness program) [Pauker, *et al.*, 1976], INTERNIST [Pople, 1977], and the Rutgers GLAUCOMA program [Weiss, et al., 1978]. And three important organic chemical synthesis programs [Corey & Wipke, 1969], [Wipke, et al., 1977], [Gelernter, *et al.*, 1977] were demonstrated as well. Several specialized programs were also developed for mathematical and management science problems [Hearn, 1971], [Burstall, 1966 (a)(b)], [Kuhn & Hamberger, 1963]. These tasks were chosen partly because of the value of their solutions and partly because of the belief that complicated problem areas were more fruitful than "toy" problems for studying complex reasoning. All of these were initially programmed more as a collection of algorithms and tricks than as a coherent method working with a large body of knowledge.

Out of that early work we, the AI community, came to realize that separating domain-specific knowledge from the problem solving methods was important and essential for knowledge base construction and maintenance. With open-ended problems and ill-defined bodies of knowledge, it was obvious that building a knowledge base was more a matter of iteration and refinement than bulk transfer of facts. This was clearly the case in Samuel's checkers program [Samuel, 1959] and Greenblatt's chess program, [Greenblatt et al,. 1967] and became painfully clear early in the work on DENDRAL. Thus a separate and simple representation of the domain-specific knowledge was essential for successfully transferring expertise to a program. (In the case of MACSYMA, virtually all the knowledge is in the methods, so the distinction is not always a sharp one.)

We also saw from this early work that transferring the judgmental knowledge of experts into a program meant representing the concepts and problem solving methods that the experts use. Clever shortcuts and elegant formalisms are worthless unless the experts can fit their own knowledge into the framework provided by the designer. Only when a program's vocabulary is "natural" to experts can they help refine and augment the knowledge base to bring the system's performance up to their own level of expertise.

We also learned that high performance tools will not be used if the interface to them is clumsy. Since we needed a large amount of feedback to refine the knowledge base, we were obligated to pay attention to human engineering issues as well as problem solving issues.

There has been much experimentation with different ways of representing knowledge. Productions had been very successful in Waterman's poker playing and learning programs [Waterman, 1970] and had proved easy to manipulate in parts of DENDRAL. They fit the MYCIN problem [Davis, et al., 1977] well also. But we now realize that almost any uniform encoding of many, nearly-separate items of knowledge would have allowed us to achieve our goals. Almost any knowledge can be represented in almost any formalism; the main issue is how easily the domain knowledge can be codified and maintained.

Work on MYCIN, DENDRAL and other expert systems also showed the value of a simple control structure. It needs to be powerful enough for reasoning about complex problems. But it cannot itself be so complex that the expert cannot predict the effects of adding new items to the knowledge base. DENDRAL'S forward chaining, data-directed inference is preferable in this respect to MYCIN'S backward chaining, goal-directed inference.

In building useful expert systems, it was also seen to be necessary to consider more of the whole *environment*, in which the program would ultimately be used. High performance is a necessary, but not sufficient, aspect of usefulness. Human engineering issues are important for making the program understandable, for keeping experts interested, for making users feel comfortable. Explanation, help facilities and simple English dialog thus became important. INTERNIST recently incorporated a display-oriented interface with menu selection, for example, to allow more flexible and natural use by physicians [R. Miller, private communication]. Simple, non-heuristic utilities (e.g., [Stefik, 1978]) offer extra capabilities beyond the main focus of the reasoning programs, but are necessary in the total package offered to users. Speed of computation forced rewrites of HEARSAY [Lesser & Erman, 1977] to HARPY [Newell, 1978] and the DENDRAL hypothesis generator into CONGEN [Carhart, et al., 1979]. The whole environment also was seen to include knowledge acquisition and knowledge base maintenance [Davis, 1976].

One of the interesting features of expert systems is their ability to reason under uncertainty. This is essential for reasons of practical utility, since there is no practical application in which the data can be guaranteed to be correct or complete as given. Moreover, in problem areas that are not fully understood we cannot assume that the program's knowledge base is either correct or complete, either in separate entries or as a whole.

SOURCES OF UNCERTAINTY

MISSING OR ERRONEOUS DATA

MISSING OR ERRONEOUS RULES

INACCURATE MODEL

The basic mechanism we have for coping with uncertainty in expert systems is to exploit redundancy. If there are many redundant items of evidence that support the same conclusion, the absence or incorrectness of a few of them will not seriously impair performance. Similarly, if there are many reundant reasoning paths to the same higher-level conclusion then the incorrectness of any path can be mildly confusing but should not seriously throw the program off track.

CORRECTIONS FOR UNCERTAINTY

REDUNDANT DATA

REDUNDANT RULES

EXPERTS' HEURISTICS

CAUTIOUS STRATEGY

Incomplete information is a particularly pervasive problem in empirical problems. Very often programs halt when items are unknown; frequently, too, they ask the user for the missing items. Some systems try to infer the missing information from available facts and relations. Default values are used, too, either with subsequent validation or without. The defaults may be either fixed globally or dependent on the context, e.g., inherited from a parent node that describes the current context in general terms. It is also possible for a program to guess at a plausible value - using heuristic procedures to fill in a context-dependent value, rather than using a value stated somewhere as a default value. Another way of coping with incomplete information is to do the best one can without it. MYCIN tries to infer a value for each relevant fact (or asks for it) but if the fact remains unknown, it reasons to a "best guess" solution using the available facts. If too many facts are missing it advises the user that not enough is known about the case to make any reliable conclusions. CONGEN, too, generates all solutions consistent with the available facts, even though there may not be enough

known to formulate a unique solution. McCarthy's work on circumscription is a formal approach to these kinds of problems [McCarthy, 1980].

ACTIONS AVAILABLE TO COPE WITH INCOMPLETE INFORMATION

Stop	*Use Default*
Ask	Guess
Infer	*Skip and Use Available Information*

PROSPECTOR [Duda, et al., 1978], INTERNIST, CONGEN and MYCIN, are among the best examples of expert systems whose designs encompassed:

- uniform representation of knowledge,
- conceptually simple control structure,
- consideration of the environment of use.

These were mostly done in the period 1975-1980 and thus can be taken as representative of the state of the art of expert systems.

Expert systems crystallize many issues of AI by forcing attention to high performance, actual use, and transparent lines of reasoning. We do understand a little about choosing problem areas that match the current state of the art. As Feigenbaum has written [Feigenbaum, 1977] one of the most critical questions is whether there is an expert available and willing to spend time developing and debugging the knowledge base. Also, the problem should be one which is interesting to the expert (not algorithmic or trivial or already totally understood). At the same time, the problem must be constrained: neither involving an indefinite number of common sense concepts and facts about the world nor involving a very large number of objects and relations in the problem area itself. MYCIN, for example, needs for meningitis about a dozen types of objects (some with multiple instances, such as multiple infections), about 200 attributes associated with those objects, each with 2-100 values (many are yes/no attributes). MYCIN, "knows" 450 rules that relate sets of object-attribute-value triples and another 500-1000 individual facts stored as definitions (e.g., E.coli is gram-negative), lists (e.g., the list of normally sterile sites), and relations (e.g., the prescribed drug for streptococcal infections is usually penicillin).

The state of the art of expert systems technology is advancing, but to be quite realistic we need to look at existing limitations as well as potential power. The following table lists many characteristics of what can currently be done.

EXPERT SYSTEMS: STATE OF THE ART

- NARROW DOMAIN OF EXPERTISE
- LIMITED LANGUAGE FOR EXPRESSING FACTS AND RELATIONS
- LIMITING ASSUMPTIONS ABOUT PROBLEM AND SOLUTION METHODS (HELP REQUIRED FROM A "KNOWLEDGE ENGINEER")
- STYLIZED I/O LANGUAGES
- STYLIZED EXPLANATIONS OF LINE OF REASONING
- LITTLE KNOWLEDGE OF OWN SCOPE AND LIMITATIONS
- KNOWLEDGE BASE EXTENSIBLE BUT LITTLE HELP AVAILABLE FOR INITIAL DESIGN DECISIONS
- SINGLE EXPERT AS "KNOWLEDGE CZAR"

The domain of expertise cannot grow too large because we lack efficient means for building and maintaining large knowledge bases. Donald Michie [private communication] estimated that the average rate of growth for a knowledge base for his AL/X system is about two rules per week, by the time errors are found and corrected. MYCIN'S knowledge base was constructed and debugged over two years, so the rate is comparable. Thus an expert system cannot now cover more than a narrow slice of a domain. The most notable exception is INTERNIST, for which the knowledge base covers about 500 disease diagnoses or about 80% of internal medicine [H. Pople, private communication]. However, this represents a full time commitment for an expert internist, Dr. Jack Meyers, and several colleagues and students over a period of over ten years. Also, it represents a strategy to cover internal medicine in more breadth than depth, using a relatively shallow set of associations between disease states and manifestations.

The representation languages that are available are still limited. Once a commitment is made to a framework, e.g., a hierarchy of objects, it is inevitable that experts will find relations that are difficult to express in that framework. Ad hoc programming removes this difficulty: a clever programmer can find a way to encode anything an expert wants to say. But the loss of uniformity is too high a price to pay for removing the constraint, for an ad hoc knowledge base rapidly becomes unmanageable.

Just as an expert needs help understanding the representational framework, he/she also needs help understanding the problem solving methods used by the program. Someone who is familiar with both the program and the domain, a so-called "knowledge engineer", must provide that help.

Input/output languages and interfaces are improving, but most are still stylized and rather inflexible. In Level-1 systems, the emphasis has been more on demonstrating adequacy of the knowledge bases than on acceptability and ease of use. Understanding totally unconstrained English text is not yet possible, even in technical domains [Bonnet, 1979].

The explanations, too, are stylized. MYCIN, for example, unwinds its goal stack to explain why it needs a piece of information, and does so in the same way for every user. This offers some insight, but is not always acceptable.

Neither the utility programs for knowledge base construction nor the reasoning programs themselves contain much knowledge about their own assumptions and limitations. They offer little guidance about the appropriateness of new problems or the boundaries of their own expertise. One of the marks of wisdom, Socrates told us repeatedly, was knowing when not to claim expertise.

As just mentioned, knowledge bases are constructed laboriously. Several research groups have considered the problem of automating knowledge base construction, or writing routines that carry on a dialog with an expert to elicit knowledge without the help of a knowledge engineer. So far, however, these activities are successful only when the program contains an initial framework to build on.

Although it is desirable to have several experts contributing to a knowledge base, we are currently limited in our ability to maintain consistency among overlapping items. Except for blatant contradictions, the incompatibilities are too subtle for a program to catch, or a knowledge engineer either. So, currently, a single expert must coordinate and monitor the contributions to a knowledge base to insure quality as well as consistency.

In addition to the programs and task areas already mentioned, several others have helped define or extend the concept of expert systems. For example, in the following task areas (and more) expert systems have been constructed and described: computer system configuration (J. McDermott's R1 program), automatic programming [Barstow, 1979], physics problems [Novak, 1976 ; Bundy, et al., 1979], chess [Wilkins, 1980], tutoring or ICAI [Brown, et al., 1975; Clancey, 1979], software consultation [Genesereth, 1978], electronics debugging [Sussman, 1975], protein structure determination [Engelmore & Terry, 1979], signal interpretation [Nii & Feigenbaum, 1978], visual scene understanding [Brooks, et al., 1979].

3 DIRECTIONS OF FUTURE WORK

Much of the new work on expert systems must necessarily be extensions of old work on problem solving, controlling search and inference, representing facts and relations about the world, understanding language and visual scenes, and so forth. In fact, all AI research is relevant for constructing and understanding expert systems. Thus the representation and control issues discussed over the last 25 years will continue to recur in expert systems. The Logic Theorist [Newell, et al., 1957] was presented to the scientific community in 1957; the Advice Taker in 1958, [McCarthy, 1963] Samuel's checkers program [Samuel, 1959] in 1959; and Minsky's structuring of AI in 1961 [Minsky, 1961]. These, and other, early papers have not been outdated. The issues remain with us, and insofar as expert systems are constructed by persons whose primary interest is AI, they will continue to provide us with new wrinkles on old problems.

3.1 REPRESENTATION AND CONTROL

In the immediate future, expert systems will be severely constrained until we understand better how to represent and reason with many kinds of concepts, including the following:

Causal Models	Propositional Attitudes and Modalities
Strategies	Conflicts in Plans, Strategies and Methods
Expectations and Default Knowledge	Multiple Sources of Expertise
Temporal andSpatial Continuity	Parallel Processing
Plans and Approximations	Multiple Sources of Knowledge
Abstraction and Hierarchies	Learning from Experience
Analogies (Formulating and Using)	Focus of Attention on Facts & Relations

None of the items in this list represents a shift in emphasis, or anything that would not have been familiar to the participants of the 1956 Dartmouth Conference [Feigenbaum, 1979]. Many are found in the early papers cited. For each of the issues listed above there has already been substantial work. The point of listing them is to emphasize that much more needs to be done to progress from Level-1 to Level-2 systems. In particular, what are the alternatives available for representing and using these concepts, and under what conditions should we choose one over another? To a very large extent the proof of effective representations of these concepts must lie in their use for high performance problem solving. The concepts are discussed very briefly below.

Causal Models --- The best work in casual reasoning has been in systems developed for analysis of small electronic circuits and simple physical devices (e.g., [deKleer, 1979, Rieger & Grinberg, 1977]. We have much to learn about exploiting causal models of physical and biological devices and coupling the models with other knowledge.

Strategies --- With a cautious problem solving strategy, all relevant, available facts are used by all relevant inference rules (in a data-driven system). In a "quick and dirty" strategy, many facts and inference rules are ignored because they seem less relevant. We want a program's strategy to be sensitive to the problem solving context. And it needs to be represented explicitly and flexibly enough to be scrutinized and modified. Meta-rules in a MYCIN-like system [Davis & Buchanan, 1977] are one way to encode strategies, and use them. What alternatives exist? What are the strengths of each?

Expectations and Default Knowledge --- In complex or open-ended domains we need to be able to make assumptions about the world rather than express all we know explicitly. Non-monotonic logic (e.g., [Doyle, 1980]) offers one paradigm. Frames can be used to represent what is known about "typical" members of classes and used to store expectations for comparison with observed data (see [Minsky, 1975], [Aikins, 1980].

Temporal Continuity --- Reasoning over time requires different representations and mechanisims (e.g., feedback) than static analysis of a situation (see [Fagan, 1980]) Some information decays in certainty or value as it grows older.

Spatial Continuity --- Most work on representing 3-dimensional models of objects is done in the context of vision systems in which a representation of a scene is the final goal. Expert systems need to be able to use those representations to reason efficiently about scenes (see [Kuipers, 1976]). When there are thousands or millions of facts like "the leg bone is connected to the ankle bone", a diagram offers great economies.

Plans and Approximations --- The planning method in GPS is to solve an approximate, more general, problem than the given one and then use the solution as a guide for constructing the desired solution. In NOAH [Sacerdoti, 1974] and MOLGEN [Stefik, 1980; Friedland, 1980] planning exploits abstraction hierarchies and constraints. Sussman [Sussman, 1975] has explored how debugging a plan can lead to a problem solution. Most work on planning has been research done for its own sake. Expert systems need to incorporate those methods and more.

Abstractions and Hierarchies --- Many systems represent and use abstractions and hierarchies. But there is little understanding of the strengths and weaknesses of various techniques. For example, different kinds of inheritance in representation languages [Brachman, 1977] are available but we don't know which to recommend for a new problem without trying some. Diagrams are abstractions of considerable heuristic value that we do not know how to exploit (see [Gelernter, 1959]).

Analogies --- Analogical reasoning is generally regarded as a powerful method for suggesting hypotheses when more constrained generators fail to produce satisfactory ones. Formulating loose analogies is relatively easy but finding those that are useful for a specified purpose is difficult. Using analogies productively is also difficult. Winston's frame-based program finds similarities in stories and situations [Winston, 1979]; Kling exploited structural similarities between an old and new theorem to suggest an economical set of axioms for a resdution theorem prover to use on the new theorem [Kling, 1971].

Propositional Attitudes and Modalities --- Common-sense reasoning and problem solving in open-ended domains often require inferences about believing, knowing, wanting and other concepts that do not necessarily preserve truth value under substitution of equals for equals [McCarthy, 1977]. For example, it may be true that John believes Venus is the Evening Star and not true that John believes Venus is the Morning Star (although they are one and the same). It is also necessary to reason with modal operators such as necessity and possibility.

Conflict in Plans, Strategies and Methods --- As knowledge bases grow larger and planning becomes more complex, we can expect multiple conflicts in planning and problem solving. Are all methods for resolving conflicts *ad hoc*, domain-dependent rules or are there general principles we can use?

Multiple Sources of Knowledge --- The expertise available to an expert system may have to be gathered or stored as spearate "packages", or it may be desirable to do so. The Blackboard model derived from HEARSAY provides one useful framework [Nii & Aiello, 1979]. Maintaining consistency in the whole knowledge base, or coping with inconsistency during reasoning, are problems that still require solutions when working with many knowledge sources.

Parallel Processing --- As tasks increase in complexity and knowledge bases grow in size, expert systems will need to find methods for increasing efficiency. Some problems require distributed control just to avoid the risk of failure of the central processor. Other problems involve inherently parallel subproblems. Distributing the problem solving across many processors is economically feasible but we lack experience in making it work (see [Smith, 1978] [Lesser & Corkill, 1978]).

Learning from Experience --- There has been little progress on methods for improving performance in light of past experience [Buchanan, et al., 1978]. Samuel's work was a tour-de-force that other work has not approached. Any kind of learning still requires special purpose programs. Almost every conceivable expert system can benefit from past experience, at the least from simple records of past successes, and failures.

Focus of Attention on Relevant Facts and Relations --- As the breadth of knowledge increases, problem solvers need context-sensitive mechanisms for focussing attention on parts of the problem and parts of the knowledge base that appear most fruitful [Pople, 1977]. Many methods have been tried but we have little understanding of their relative merits.

In addition to representing and using the general concepts in the above list (and many others besides) future work on expert systems will involve other issues arising more directly from the work on expert systems. Because of the increased emphasis on large knowledge bases, the three issues of explanation, acquisition, and validation are becoming critical issues for expert systems. While they would not have surprised AI researchers in 1956, their importance seems not to have been fully anticipated. Also, we are beginning to see more interest in experimentation with AI programs. These four topics will be discussed briefly in turn, followed by a short discussion of the difficulty of choosing a framework for problem solving.

3.2 EXPLANATION

Explanation is important for an expert system because users cannot be expected to know or understand the whole program. The users are seeking help from the program because they want advice about their problem and will take some action based partly on that advice. They will be held responsible for the actions, in many cases. Therefore they need to be able to understand the rational basis for the programs' decisions.

An important source of explanatory descriptions is a record of what data and hypotheses the reasoning program has considered. Merely keeping a "laboratory notebook", of sorts, is a first step in making the reasoning transparent [Buchanan, 1979]. One kind of interactive explanation is simply question answering [Scott, et al., 1977]. But while answering questions about the contents of the knowledge base is necessary, it is not sufficient for giving users the information they need. In complicated cases the difficulty many lie more in <u>how</u> the program uses what it knows than in <u>what</u> it knows [Swartout, 1977]. Thus the user needs to be able to understand the line of reasoning.

In the MYCIN example in the appendix, part of the dialog contains the prompt for information about burns, for which the user might request an explanation. The response to a "why?" question is MYCIN'S reason why a fact is needed to complete the line of reasoning. In effect, X is needed because then I can conclude Y, already having established other facts that are contained with X in a rule. Work on explanations in MYCIN assumes that the user needs to know specific rules in the knowledge base which have been invoked. It does not take account of individual differences in users'

qualifications or different purposes for asking a question in the first place. A smarter system that can determine and exploit those differences can provide more helpful explanations. In building a tutor for MYCIN'S knowledge base, called GUIDON [Clancey, 1979], we found that students needed more than the conditional rules to understand what is going on. They needed some of the causal descriptions that justified the rules in order to make sense of them and remember them. Thus we conclude that a knowledge base capable of producing excellent results may, nonetheless, be less than satisfactory for pedagogy.

3.3 KNOWLEDGE ACQUISITION

Knowledge acquisition has become recognized as an issue with expert systems because it has turned out to be difficult and time consuming. DENDRAL, for example, was originally "custom-crafted" over many years. Its knowledge of chemistry was carefully molded from material provided by chemists and then cemented into place. We rewrote large parts of the system as the knowledge base changed. After doing this a few times we began looking for ways to increase the rate of transfer of chemistry expertise from chemists into the program. Making procedures highly stylized and dependent on global parameters was a first step, but still required programmers to write new procedures. DENDRAL'S knowledge of mass spectrometry was finally codified in production rules.

Once the vocabulary and syntax for the knowledge base are fixed, the process of knowledge acquisition can be speeded considerably by fitting (sometimes *forcing*) new knowledge into the framework. A programmer, whose title in this role is "knowledge engineer", is still required to explain the program's framework to the expert and to translate the expert's problem solving knowledge into the framework. This is about as far as we have come in building expert systems.

There have been prototype dialog programs that communicate with an expert to provide some of the same help that the knowledge engineer provides. One of the most ambitious, to date, is TEIRESIAS [Davis, 1976,] but even it is limited to helping debug and fill out a knowledge base that has already been largely codified.

Ultimately it would be desirable to have a program learn from nature, as scientists do. As mentioned above, the state of induction programs is not up to widespread use for constructing knowledge bases. However, prototype programs (e.g., [Mitchell, 1977]) again point to future directions for research on expert systems.

An interactive editor that prompts for values of necessary slots is a starting place for a knowledge

acquisition system, but it is not the final product. When a "knowledge engineer" [illegible] an expert, he/she is not passive but:

1. interprets and integrates the expert's answers to questions;

2. draws analogies to help the expert structure the domain or remember important aspects of the domain;

3. poses counter-examples and raises conceptual difficulties.

The most difficult aspect of knowledge acquisition is helping the expert structure the domain initially. Because the knowledge acquisition system has no domain-specific knowledge at the beginning (by definition), the system can only rely on general knowledge about the structure of knowledge bases and specific examples of other knowledge bases as well as what the expert says about the new domain. The knowledge acquisition system has to contain, or have access to, the structure, assumptions, and limitations of the inference mechanism that will use the new knowledge. MYCIN, again, assumes that rules are structured from fact triples, that the rules will be used to infer values of attributes of a primary object, and so forth.

Maintaining a large knowledge base will be every bit as difficult as constructing it in the first place. With problems having no closed solutions, the knowledge base of an expert system should certainly change as experts accumulate more experience and develop new techniques. In medicine, for example, new measuring devices make it possible to detect new states or quantify known parameters more precisely. New microbiological agents are discovered as well as new drugs to treat them.

Maintenance may mean actively seeking problems in the knowledge base that need attention. There may be gaps, where some of many possible combinations of conditions are covered, but not all. There may be overlapping items in the knowledge base, leading to inconsistent or redundant conclusions. Or items may become outdated. An intelligent maintenance system should have both the syntacic and semantic knowledge needed to assign blame to specific items in the knowledge base that appear to be responsible for poor performance and to suggest modifications.

The problems of knowledge base maintenance become more difficult when two or more experts contribute to the knowledge base. In MYCIN, although several physicians contributed, only one physician at any one time could make changes. Thus all recommendations for change went to a knowledge base "czar" who decided how to maintain consistency.

3.4 VALIDATION

Expert systems are beginning to move from the research and development stage into the market place. MACSYMA, DENDRAL and MOLGEN all have serious users who are only loosely coupled to the designers of the programs. Under these circumstances, the developers are expected to provide some objective demonstration that a program performs as well as they claim.

Anyone who has constructed a complex reasoning program knows how difficult it is to anticipate unusual requests and error conditions. We want expert systems to provide assistance in a broad range of unanticipated situations -- that is the strength of an AI approach. But we also want to provide assurance to prospective users that the programs will perform well.

Convincing the external community is different from convincing insiders. Insiders can examine code and perform *gedanken* experiments that carry as much weight as statistics. For the external community, however, we need to develop our own equivalents of rat studies and clinical trials for programs, such as those that new drugs are subjected to. Empirical proof is the best we can hope for: sometimes actual use is the most we can point to [Buchanan & Feigenbaum, 1978].

MYCIN is one program whose performance has been externally validated. There have been different empirical studies of MYCIN'S performance, each simpler than the last but all of them time consuming. In the last of these [Yu, et al., 1979] we were trying to determine how outside experts compared MYCIN'S final conclusions with conclusions of local experts and other physicians. Ten meningitis cases were selected randomly and their descriptions were presented to seven Stanford physicians and one student. We asked them to give their therapy recommendations for each case. Then we collected all recommendations, together with MYCIN'S recommendation for each case and the actual therapy, in a 10 x 10 matrix -- ten cases each with ten therapy recommendations. We asked a panel of eight experts not at Stanford to give each recommendation a zero if, in his opinion, it was unacceptable for the case and a one if the recommendation was acceptable. They did not know which, if any, recommendation came from a computer. The results are shown in the following table.

RATINGS BY 8 EXPERTS ON 10 MENINGITIS CASES PERFECT SCORE = 80 *

MYCIN	52	ACTUAL THERAPY	46
FACULTY-1	50	FACULTY-4	44
FACULTY-2	48	RESIDENT	36
INF. DIS. FELLOW	48	FACULTY-5	34
FACULTY-3	46	STUDENT	24

* Unacceptable Therapy = 0, Equivalent Therapy or Acceptable Alternate = 1

The differences between MYCIN'S score and the scores of the infectious disease experts at Stanford are not significant. But we can claim to have shown that MYCIN'S *recommendations were* viewed by outside experts to be as good as the recommendations of the local experts and all of those better than the recommendations of physicians (and the student) who are not meningitis experts.

So far, I have reviewed many outstanding problems of expert system work. All of these are motivated in one way or another by the three parts of the definition of expert systems I gave initially:

HIGH PERFORMANCE --- obviously requires careful attention to the representation of knowledge, methods of inference and validation that the program does perform well.

UTILITY --- requires a large body of knowledge about a problem of significant size or difficulty and thus requires careful attention to knowledge acquisition and knowledge base maintenance.

TRANSPARENCY --- requires explanation programs using high-level concepts and models familiar to the user. That can tell a user what the program knows, how it uses its knowledge, and why *reasons* as it does.

In addition to the problems just discussed, two other outstanding issues are beginning to influence work on expert systems but have had little influence to date. The first issue, or perhaps project, is experimentation with existing AI systems. The second is choosing a problem-solving framework.

3.5 EXPERIMENTATION

AI is an empirical science, as Newell and Simon have argued convincingly [Newell & Simon, 1976]. The data we work with are programs; the conclusions we hope to draw from studying them include understanding the phenomenon of intelligent action itself. One reason to construct expert systems is to replace arguments about what computers can do by demonstrations. Physicians, chemists, and

mathematicians support our claims that the programs are working on interesting challenging problems. These and other AI programs constitute data points, sometimes more because of their methods than because of their tasks.

We have generalized from the data presented but we have almost totally ignored the value of controlled experiments. The collection of papers on the GPS experiments [Ernst & Newell, 1969], represent the most systematic sets of experiments undertaken in AI. But we must think still more about experimenting with the programs we spend so much time building. At this time we are not even very good at formulating precise questions that can be answered experimentally.

Eventually we will be able to work out a taxonomy of problems and a taxonomy of solution methods. Newell and Simon have taken us farthest in this direction [Newell, 1973], but they will undoubtedly agree we still have less than perfect understanding of our discipline. When the taxonomies exist, then we can begin developing criteria that let us determine the best method for a given problem.

Because construction of expert systems and experimentation with them are both very expensive at the moment, we are beginning to see a trend toward design tools for expert systems. These are tools that help a person design and build an expert system within a given framework. By setting up the framework and providing some knowledge engineering help, the design system can speed up the construction, or modification, of an expert system. Such systems can also speed up our experiments with existing systems.

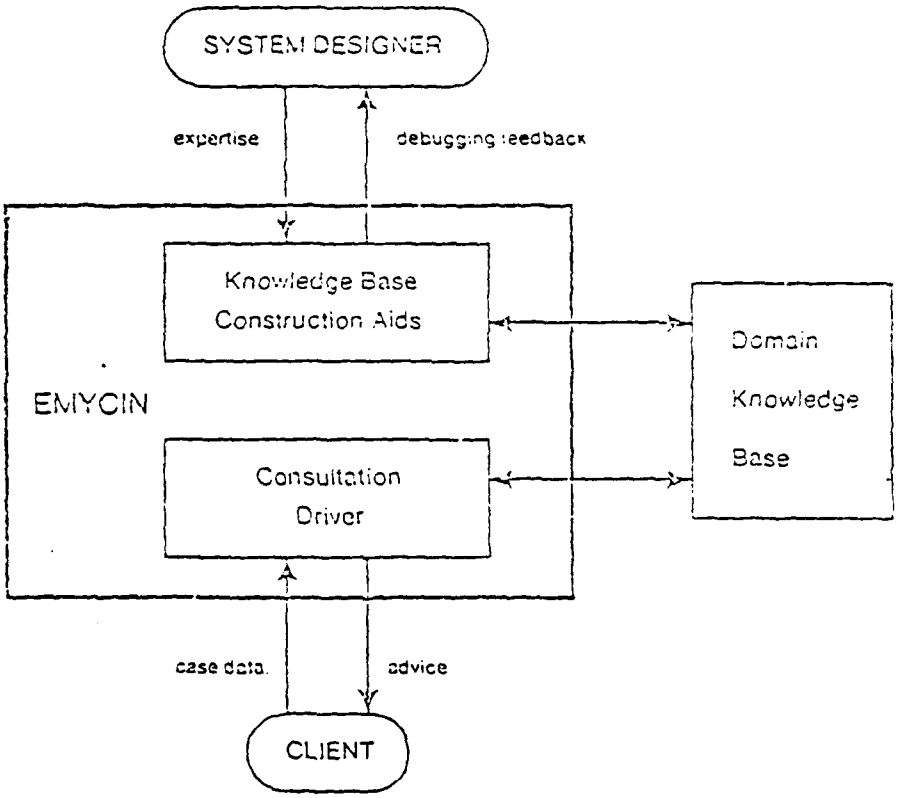
EMYCIN [vanMelle, 1980] is one such design system that helps a person design and build a MYCIN-like expert system. The name stands for "essential MYCIN", the MYCIN system without the medical knowledge. It assumes that production rules are an appropriate representation framework for a person's new knowledge base and that a backward-chaining, or goal-directed, interpreter is an appropriate inference mechanism. If a new problem can be set up as a problem of gathering evidence for and against alternative hypotheses that define subgoals for ultimately satisfying the major goal then EMYCIN is likely to provide some help in constructing an initial prototype expert system to solve the problem.

EMYCIN provides some assistance in structuring a person's knowledge about a problem. This means finding out about the main kinds of objects in the domain and their relationships. What is the primary object about which the expert system should offer advice - a patient, a corporation, an automobile, a computer? What are its parts, and their sub-parts? Also, EMYCIN needs to know about the attributes of those objects and possible values. A computer's manufacturer, a patient's age, a

corporation's size, for example are relevant attributes for most problems involving these primary objects. EMYCIN expects that goals are stated as finding plausible values for one or more attributes.

After EMYCIN has helped a designer build a new knowledge base, and thus a new expert system, it interprets the knowledge base with the inference engine. These two main functions are shown schematically in the figure below. In addition, the rules in the knowledge base can also be compiled into a decision tree for more efficient execution.

THE EMYCIN SYSTEM



Some of the experimental expert systems developed in EMYCIN are PUFF (see [Feigenbaum, 1977]), SACON [Bennett & Engelmore, 1979], and consultants for computer system debugging, nutrition, psycho-pharmacology, nervous disorders, and circuit debugging.

Other similar design tools are OPS4 [Forgy & McDermott, 1977] at Carnegie-Mellon, Hearsay-III [Balzer, et al., 1980] at ISI, AGE [Nii & Aiello, 1979] at Stanford, EXPERT [Weiss & Kulikowski, 1979] written at Rutgers, XPRT [Steels, 1979] at MIT, and RITA & ROSIE at RAND [Anderson & Gillogly,

1976]. Representation languages such as KRL [Bobrow & Winograd, 1977], OWL [Szolovits, et al., 1977], and the UNITS package [Stefik, 1979] have similar motivations of making it convenient to build a new knowledge base, without locking the designer into an interpreter for it.

3.6 CHOOSING A FRAMEWORK

The last outstanding issue is the well-known problem of choosing the right framework for solving a problem before searching for a solution [Amarel, 1968]. Problem solving can be viewed as a two stage process:

- Choose a language, L
- Select the best solution within L

We are beginning to understand how to use heuristic methods to find and select solutions to problems within a given problem solving framework. If expert systems can also suggest new frameworks for solving problems, then they will be useful aids for theory construction as well as for hypothesis formation within an existing theory.

When MYCIN gathers evidence for alternative hypotheses, the choices are fixed in advance in the vocabulary of the rule set and object-attribute value triples. When CONGEN generates chemical structures, it describes them in a given vocabulary of labelled, planar graphs. Extending the vocabulary to include some 3-dimensional information has been and still is a task of great magnitude. When META-DENDRAL proposes rules that codify data, it does so within a fixed and very limited vocabulary.

One of the criticisms of sceptics is that AI programs are not yet touching "real science". This must be false - otherwise only Galileo, Newton, Einstein and a few others could be called real scientists. But the objection is right in one respect: we do not have AI methods for searching a space of frameworks the way we search a space of hypotheses.

Lenat's program, AM [Lenat, 1976], generates new mathematical terms by combining old terms in interesting ways. It is continually expanding its framework, given in the initial concepts of number theory with which it starts. J.S. Brown wrote a concept formation program [Brown, 1972], that added new predicates to cover interesting partitions of the data it noticed. The BACON program [Langley, 1979] defines new concepts from old ones in order to reduce the combinatorics of its search. Although there is much more to the introduction of new theoretical terms in science, these redefinitions offer considerable savings in reducing the number of terms to consider. The heuristics

of when to introduce a new "macro", in this sense, still needs to be much better understood. Beyond that, though, will be the Level-III expert systems that can aid scientists by introducing new theoretical terms into existing languages and creating new explanatory languages.

4 CONCLUSION

AI is still very much in the so-called "natural-history" stages of scientific activity in which specimens are collected, examined, described, and shelved. At some later time a theory will be suggested that unifies many of the phenomena noticed previously and will provide a framework for asking questions. We do not now have a useful theory. The vocabulary that we use to describe existing systems is more uniform and useful than it was a decade ago, however. And the questions that we pose in the context of one program are sometimes answered in another.

Expert systems will provide many more data points for us over the coming years. But it is up to everyone in AI to do controlled experiments, analyze them, and attempt to develop a scientific framework in which we can generalize from examples. At the moment we ourselves lack the vocabulary for successful codification of our own data.

REFERENCES

1. Aikins, J.S. "Prototypes and Production Rules: A Knowledge Representation for Computer Consultation". *Ph.D. Dissertation, Stanford University Computer Science Department.* STAN-CS-30-814, (1980).

2. Amarel, S. "On Representations of Problems of Reasoning about Actions". In D. Michie (ed.), *Machine Intelligence 3.* New York: American Elsevier, 1968.

3. Anderson, R.H. and Gillogly, J.J. "The Rand Intelligent Terminal (RITA) as a Network Access Aid". *AFIP Proceedings 45*, pp. 501-509 (1976).

4. Balzer, R., Erman, L., London, P. & Williams, C. "HEARSAY-III: A Domain Independent Framework for Expert Systems". *Proceedings 1980 AAAI Conference*, Stanford, (1980).

5. Barstow, D. "An Experiment in Knowledge-Based Automatic Programming". *Artificial Intelligence 12*, pp. 73-119 (1979).

6. Bennett, J. & Engelmore, R. "SACON: A Knowledge-Based Consultant for Structural Analysis". *Proceedings IJCAI-79*, (1979).

7. Bobrow, D, and Winograd, T. "An Overview of KRL, a Knowledge Representation Language". *Cognitive Science 1*, pp. 3-46 (1977).

8. Bonnet, A. "Understanding Medical Jargon As If it Were a Natural Language". *Proceedings IJCAI-79*, (1979).

9. Brachman, R.J. "What's In a Concept: Structural Foundations for Semantic Networks". *International Journal Man-Machine Studies 9*, pp. 127-152 (1977).

10. Brooks, R., Greiner, R., & Binford, T. "The ACRONYM Model-Based Vision System". *Proceedings IJCAI-79*, pp. 105-113, (1979).

11. Brown, J.S. "A Symbiotic Theory Formation System". (University of Michigan PhD Dissertation) *University of California, Irvine Computer Science Department*, Technical Report 017 (1972).

12. Brown, J.S., Burton, R.R. & Bell, A.G. "Sophie: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI)". *International Journal Man-Machine Studies 7*, (1975).

13. Buchanan, B.G. "Issues of Representation in Conveying the Scope and Limitations of Intelligent Assistant Programs". In J. E. Hayes, D. Michie, and L. I. Mikulich (eds.). *Machine Intelligence 9.* Chichester: Ellis Horwood Ltd. and New York: John Wiley, 1979.

14. Buchanan, B.G., and Feigenbaum, E.A. (1978) "DENDRAL and Meta-DENDRAL: Their Applications Dimension". *Artificial Intelligence 11*, pp. 5-24 (1978)

15. Buchanan, B.G., Mitchell, T., Smith R.G., and Johnson,C.R., Jr. (1978) "Models of

Learning Systems". In J Belzer (Ed.), *Encyclopedia of Computer [illegible] and Technology, Vol. 11*. New York: Marcel Dekker, Inc. 1978.

16. Bundy, A., Byrd, L., Luger, G., Mellish, C. & Palmer, M. "Solving Mechanics Problems Using Meta-Level Inference". In D. Michie (ed.) *Expert Systems in the Micro Electronics Age*. Edinburgh: Edinburgh University Press, 1979.

17. Burstall, R.M. "Computer Design for Electricity Supply Networks by Heuristic Methods". *Computer Journal 9*, (1966).

18. Burstall, R.M. "A Heuristic Method for A Job Scheduling Problem. *Operations Research Quarterly 17*. (1966).

19. Carhart, R.E. "CONGEN: An Expert System Aiding the Structural Chemist". In D. Michie (ed.) *Expert Systems in the Micro Electronics Age*. Edinburgh: Edinburgh University Press, 1979.

20. Clancey, W. "Tutoring Rules for Guiding a Case Method Dialogue". *International Journal Man-Machine Studies 11*, pp. 25-49 (1979).

21. Cory, E.J., and Wipke, W.T. "Computer Assisted Design of Complex Organic Synthesis". *Science 166*, pp. 178-192 (1969).

22. Davis, R. "Applications of Meta-Level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases". *PhD Dissertation, Stanford University Computer Science Department*, STAN-CS-76-564 (1976).

23. Davis, R. and Buchanan, B.G. "Meta-level Knowledge: Overview and Applications. *Proceedings IJCAI-77*, pp. 920-928 (1977).

24. Davis, R., Buchanan, B.G., and Shortliffe, E.H. "Production Rules as a Representation of a Knowledge-Based Consultation Program". *Artificial Intelligence 8*, (1977).

25. Doyle, J. "A Truth Maintenance System". *Artificial Intelligence 12*, pp. 231-272, (1979).

26. Duda, R.O., Gaschnig, J., Hart, P.E., Konolige, K., Reboh, R., Barrett, P., and Slocum, J. "Development of the PROSPECTOR Consultation System for Mineral Exploration". Final Report, *SRI Projects 5821 and 6415*. SRI International, Menlo Park, California, 1978.

27. Engelmore, R.S. & Terry, A. "Structure and Function of the CRYSALIS System". *Proceedings IJCAI-79*, pp. 250-256, (1979).

28. Erman, L.D. & Lesser, V.R. "Hearsay-II: Tutorial Introduction and Retrospective View". *Carnegie-Mellon University, Computer Science Department Report*. CMU-CS-78-117 (1978).

29. Ernst, G., and Newell, A. "GPS: A Case Study in Generality and Problem Solving". New York: *Academic Press*, 1969.

30. Fagan, L. "VM: Representing Time-Dependent Relations in a Clinical Setting". *PhD Dissertation, Stanford University Computer Science Department*, (1980).

31. Feigenbaum, E.A. "The Art of Artificial Intelligence: Themes and Case Studies in Knowledge Engineering". *Proceedings IJCAI-77*, pp. 1014-1029, (1977).

32. Feigenbaum, E.A. (Chairman, Panel Discussion) "History of Artificial Intelligence Research, 1956-61". *Proceedings IJCAI-79*, pp. 1103-1105, (1979.)

33. Forgy, C. & McDermott, J. "OPS, A Domain-Independent Production System Language". *Proceedings IJCAI-77*, pp. 933-939, (1977.)

34. Friedland, P. "Knowledge-Based Experiment Design in Molecular Genetics" *PhD Dissertation, Stanford University, Computer Science Department.* STAN-CS-79-771, (1979).

35. Gelernter, H.L. "Realization of a Geometry-Theorem Proving Machine". *Proceedings of International Conference on Information Processing*, Paris: UNESCO House, pp. 273-282, (1959). Reprinted in E.A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*. New York: McGraw-Hill, 1963.

36. Gelernter, H.L., Sanders, A.F., Larsen, D.L., Agarival, K.K., Boivie, R.H., Spritzer, G.A., and Searleman, J.E. "Empirical Explorations of SYNCHEM". *Science*, 197 pp. 1041-1049, (1977).

37. Genesereth, M.R. "Automated Consultation for Complex Computer Systems". *Ph.D. Dissertation, Harvard University*, September (1978).

38. Greenblatt, R.B., Eastlake, D., and Crocker, S. "The Greenblatt Chess Program". *Proceedings 1967 Joint Computer Conference 30*, pp. 801-810, (1967).

39. Hearn, A. "REDUCE 2: A System and Language for Algebraic Manipulation". *Proceedings 2nd Symposium on Symbolic and Algebraic Manipulation*, March (1971).

40. deKleer, J. "Causal and Teleological Reasoning in Circuit Recognition". *PhD Dissertation, MIT Department of Electrical Engineering and Computer Science*, MIT AI Lab Report TR-529 (1979).

41. Kling, R.E. "A Paradigm for Reasoning by Analogy". *Artificial Intelligence 2*, pp. 147-178, (1971).

42. Kuhn, A., and Hamberger, M. "A Heuristic Program for Locating Warehouses". *Management Science 9*, (1963).

43. Kuipers, B. "Spatial Knowledge". *MIT AI Lab Memo 359*, (1976).

44. Langley, P. "Rediscovering Physics with BACON.3". *Proceedings IJCAI-79*, pp. 502-507, (1979).

45. Lenat, D. "An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search". *PhD Dissertation, Stanford University Computer Science Department*, STAN-CS-76-570 (1976).

46. Lesser, V.R. & Erman, L.D. "A Retrospective View of the Hearsay-II Architecture". *Proceedings IJCAI-77*, pp. 790-800, (1977).

47. Lesser, V.R. and Corkill, D. "Cooperative Distributed Problem Solving: A New Approach for Structuring Distributed Systems". *University of Massachusetts Computer and Information Science Department Report*, Technical Report 78-07 (1978).

48. Lindsay, R., Buchanan, B.G., Feigenbaum, E.A., and Lederberg, J. "Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project". New York: *McGraw-Hill*, 1980.

49. McCarthy, J. "Programs with Common Sense". *Stanford University, AI Memo AIM-7*, (1963).

50. McCarthy, J. "Epistemological Problems of Artificial Intelligence". *Proceedings IJCAI-77*, pp. 1038-1044, (1977).

51. McCarthy, J. "Circumscription - A Form of Non-Monotonic Reasoning". *Artificial Intelligence 13*, pp. 27-39, (1980).

52. Minsky, M. "Steps Toward Artificial Intelligence". *Proceedings of Institute of Radio Engineers*, January, 1961. 49: 8-30, (1961). Reprinted in E.A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*. New York: McGraw-Hill, 1963.

53. Minsky, M. "A Framework for Representing Knowledge". In P. Winston (ed.) *The Psychology of Computer Vision*. New York: McGraw-Hill, 1975.

54. Mitchell, T.M. "Version Spaces: An Approach to Rule Revision During Rule Induction". *Proceeding IJCAI-77*, pp. 305-310, (1977).

55. Moses, J. "Symbolic Integration: The Stormy Decade". *Communications ACM 8*, pp. 548-560, (1971).

56. Newell, A. "Artificial Intelligence and the Concept of Mind". In R. Schank and K. Colby (eds.) *Computer Models of Thought and Language*. San Francisco: Freeman, 1973.

57. Newell, A. "Harpy, Production Systems and Human Cognition". *Carnegie-Mellon University, Computer Science Department Report*, CMU-CS-78-140 (1978).

58. Newell, A., Shaw, J.C., & Simon, H.A. "Empirical Explorations with the Logic Theory Machine: A Case History in Heuristics". *Proceedings Western Joint Computer Conference*, 1957, 15:218-239, (1957). Reprinted in E.A. Feigenbaum & J. Feldman (eds), *Computers and Thought*. New York: McGraw-Hill, 1963.

59. Newell, A., Barnett, J., Forgie, J.W., Green, C., Klatt, D., Licklider, J.C.R., Munson, J., Reddy, D.R., and Woods, W.A. *Speech Understanding Systems: Final Report of a Study Group*. New York: American Elsevier, 1973.

60. Newell, A. & Simon, H.A. "Computer Science as Empirical Inquiry: Symbols and Search". The 1976 ACM Turing Lecture. *Communications ACM 19*, pp. 113-126 (1976).

61. Nii, H.P. and Aiello, N. "AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs". *Proceedings IJCAI-79*, pp. 645-655 (1979).

62. Nii, H.P. & Feigenbaum, E.A. "Rule-Based Understanding of Signals". In D. Waterman & F. Hayes-Roth (eds.), *Pattern-Directed Inference Systems*. New York: Academic Press, 1978.

63. Novak, G. "Computer Understanding of Physics Problems Stated in Natural Language" *American Journal of Computational Linguistics*, MF53, (1976).

64. Pauker, S., Gorry, A., Kassirer, J., and Schwartz, W. "Towards the Simulation of Clinical Cognition--Taking a Present Illness by Computer". *American Journal of Medicine* 60, pp. 981-996, (1976).

65. Pople, H.E. "The Formation of Composite Hypotheses in Diagnostic Problem Solving--an Exercise in Synthetic Reasoning". *Proceedings IJCAI-77*, pp. 1030-1037, (1977).

66. Rieger, C. & Grinberg, M. "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms". *Proceedings IJCAI-77*, (1977).

67. Sacerdoti, E.D. "Planning in a Hierarchy of Abstraction Spaces". *Artificial Intelligence* 5, pp. 115-135 (1974).

68. Samuel, A.L. "Some Studies of Machine Learning Using the Game of Checkers". *IBM Journal of Research and Development*, 1959, 3:211-229, (1959). Reprinted in E.A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*. New York: McGraw-Hill, 1963.

69. Scott, A.C., Clancey, W., Davis, R., and Shortliffe, E.H. "Explanation Capabilities of Knowledge-Based Production Systems". *American Journal of Computational Linguistics*. Microfiche 62, (1977).

70. Shortliffe, E.H. "Computer Based Medical Consultations: MYCIN". New York: American Elsevier, 1976.

71. Smith, R.G. "Framework for Problem Solving in a Distributed Processing Environment". *Ph.D. Dissertation, Stanford University, Computer Science Department*. STAN-CS-78-667, (1978).

72. Steels, L. "Reasoning Modeled as a Society of Communicating Experts". *MIT AI Lab Memo*, TR 542 (1979).

73. Stefik, M. "Inferring DNA Structures from Segmentation Data". *Artificial Intelligence 11*, pp. 85-114, (1978).

74. Stefik, M. "An Examination of a Frame-Structured Representation System". *Proceedings IJCAI-79*, pp. 845-852, (1979).

75. Stefik, M. "Planning with Constraints". *Ph.D. Dissertation, Stanford University Computer Science Department*. STAN-CS-80-784, (1980).

76. Sussman, G.A. "A Computer Model of Skill Acquisition". New York: American Elsevier, 1975.

77. Swartout, W. "A Digitalis Therapy Advisor with Explanations". *Proceedings IJCAI-77*, pp. 819-825, (1977).

78. Szolovits, P., Hawkinson, L.B., and Martin, W.A. "An Overview of OWL, a Language for Knowledge Representation". *MIT LCS TM 86*, (1977).

79. van Melle, W. "A Domain Independent System that Aids in Constructing Knowledge Based Consultation Programs". *PhD Dissertation, Stanford University Computer Science Department*. STAN-CS-80-820, (1980).

80. Waterman, D.A. "Generalization Learning Techniques for Automating the Learning of Heuristics". *Artificial Intelligence*, 1, pp. 121-170, (1970).

81. Weiss, S. and Kulikowski, C. "EXPERT: A System for Developing Consultation Models". *Proceedings IJCAI-79*, pp. 942-947, (1979).

82. Weiss, S., Kulikowski, C., Amarel, S., and Safir, A. "Method for Computer-Aided Medical Decision Making". *Artificial Intelligence 11*, pp. 145-172, (1979).

83. Wilkins, D. "Using Patterns and Plans in Chess". *Artificial Intelligence 14*, pp. 165-203, (1980).

84. Winston, P.H. "Learning and Reasoning by Analogy: The Details". *MIT AI Lab Memo 520*, (1979).

85. Wipke, W.T., Braun, H., Smith, G.,Choplin, F., and Sieber, W. "SECS--Simulation and Evaluation of Chemical Synthesis: Strategy and Planning". In W.T. Wipke and W. J. House (eds.). Computer Assisted Organic Synthesis. Washington, D.C.: *American Chemical Society*, pp. 97-127, (1977).

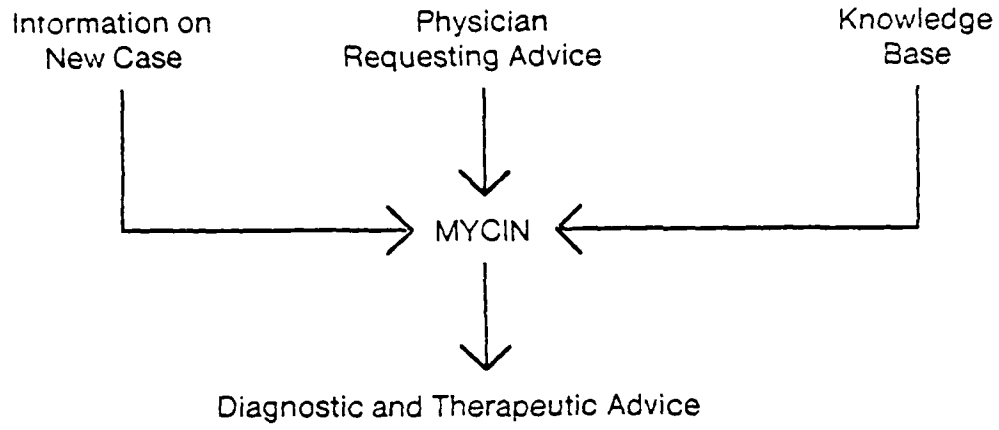
86. Yu, V.L., Fagan, L., Wraith, S.M., Clancey, W.J., Scott, A.C., Hannigan, J., Blum, R., Buchanan, B.G., Cohen, S.N., Davis, R., Aikins, J.S., vanMelle, W., Shortliffe, E.H., and Axline, S. "Antimicrobial Selection for Meningitis by a Computerized Consultant - A Blinded Evaluation by Infectious Disease Experts". *Journal American Medical Association 241*, (1979).

APPENDIX - MYCIN EXAMPLE

1 Overview

MYCIN EXAMPLE

MYCIN [Shortliffe, 1974] [Davis, 1975] is an interactive system designed to offer expert-level medical advice about patients with acute infections. Its overall design is shown in the figure



MYCIN'S knowledge is stored as fact-triples with associated degrees of certainty and as conditional rules relating facts, as shown in the following figures.

Fact Triple:
The X of Y is Z (with CF certainty)

Rule:
IF $FACT_1$ and $FACT_2$ and .....& $FACT_n$
Then Conclude $FACT_0$

RULE300

[This rule applies to all cultures and suspected infections, and is tried in order to find out about the organisms (other than those seen on cultures or smears) which might be causing the infection]

If: 1) The infection which requires therapy is meningitis, and
2) The patient does have evidence of serious skin or soft tissue infection, and

```
      3) Organisms were not seen on the stain of the culture
      4) The type of the infection is bacterial
 Then:  There is evidence that the organisms (other than those seen on
    cultures or smears) which might be causing the infection is
    staphylococcus-coag-pos (.75) streptococcus-group-a (.5)

 Author: YU

 Literature: G. Karalazin "Sickle-cell anemia - CLINICAL MANIFESTATIONS
    IN 100 PATIENTS" AmJMedSci 269:51 1975.
E. Barret-Connor "Acute pulmpnary disease and sickle-cell anemia"
          ARRD 104:159 Aug 1971.
       M. Robinson "Pneumococcal meningitis in sickle-cell anemia"NEJM
          274:1006 1966.
```

2 Infectious Disease Consultation Session

```
[User's responses follow double asterisks]

   --------PATIENT-538--------
   1) Patient's name:
   ** PT538
   2) Age:
   ** 34 YEARS
   3) Sex:
   ** MAEL
=MALE
   4) Race:
   ** CAUCASIAN

Please supply one answer under each heading.  If your response is not
understood, you will be prompted by the standard questioning method.
(enter ? for normal prompting, * for questions that are not applicable).

Please enter the results of positive cultures using the following table.

      site        culture   date          date|         method|
                  number    collected     positive      of collection
                                                        [urine,sputum only]
 5)**DONE

Please enter the results of pending cultures in the following table.

      site        culture   date          stain|        method of|
                  number    collected     examined      collection
                                                        [urine,sputum only]
 6)**  CSF        1234      30-Jan-77     YES           *

 stain of  morph|
 smear
 [type DONE if no organism was seen on the smear]
```

7)**DONE

.
.
.

[Considering organisms (other than those isolated from cultures) which might be present...]

14) Has Pt538 recently had symptoms of persistent headache or other abnormal neurologic symptoms (dizziness, lethargy, etc.)?
** YES
15) Has Pt538 recently had objective evidence of abnormal neurologic signs (nuchal ridigity, coma, seizures, etc.) documented by physician observation or examination?
** YES

The CSF cultures will be considered to be associated with MENINGITIS.
16) Please give the date on which clinical evidence (symptoms, signs, or laboratory tests) of the meningitis first appeared.
** 29-Jan-77 22:15

.
.
.

19) Does Pt538 have any cutaneous lesions or rash on physical examinatior
** NO
20) Does Pt538 have evidence of ocular nerve dysfunction?
** NO
21) Is Pt538 a compromised host (e.g. alcoholic, sickle-cell-disease, immunosuppressed)?
** YES
22) Is Pt538 immunosuppressed (e.g. corticosteroid therapy, cytotoxic drug therapy, radiation therapy, leukemia, lymphoma)?
** NO
23) For how many days has Pt538 had abnormal neurologic signs?
** 7.2 HOURS
24) For how many days has Pt538 had neurologic symptoms?
** 1 DAYS
25) Has Pt538 had recent evidence of tender or enlarged salivary glanc
** NO
26) Has Pt538 been exposed to any contagious disease recently (e.g. meningococcal disease, mumps) ?
** NO

Please enter CSF findings in the following table

	CSF WBC	Percent PMN	Protein	CSF Glucose	Peripheral Glucose
27)**	100	99	280	5	UNKNOWN

28) Do you know the results of a recent CBC of Pt538?
** YES

Please enter CBC results in the following table

```
      WBC                          Percent          Percent|
      in [illegible]s              PMN              BANDS
29)** [illegible]                  UNKNOWN          UNKNOWN
  30) Has Pt538 had any injury or insult to, or defect in the central-
      nervous-system (excluding neurosurgery)?
  ** NO
  .
  .
  .

  35) Is meningitis 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 organisms (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)
  Author: Yu
     [RULE545]

   [back to question 35...]
   ** HOW 2.3

[i.e. HOW was it established that the type of the infection is bacterial?]

The following rules concluded about:
     the likelihood that the type of the infection is bacterial

          Rule          Yes       No

     [2.7]     RULE148          (.19)
     [2.8]     RULE500          (.51)
     [2.9]     RULE501          (.95)
     [3.0]     RULE502          (.97)
     [3.1]     RULE526          (.98)
     [3.2]     RULE504          (.97)
     [3.3]     RULE524          (.98)
```

Thus, is has been established that the type of the infection is bacterial.

[back to question 35...]
** HOW 2.5

[i.e. HOW was it established that the age of pt538 is greater than 10 days?]

In answer to question 2 you said that the age of pt538 is 34 years

Thus, it has been established that the age of pt538 is greater than 10 days.

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

.
.
.

39) Is Pt538 a burn patient?
** WHY

[i.e. WHY is it important to determine whether PATIENT-538 has been seriously burned?]

[4.0] 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
[4.1] the infection which requires therapy is meningitis, and
[4.2] organisms were not seen on the stain of this pending csf culture, and
[4.3] the type of the infection is bacterial
Therefore, if
[4.4] pt538 has been seriously burned
then
there is suggestive evidence (.5) that pseudomonas-aeruginosa is one of the organisms (other than those seen on cultures or smears) which might be causing the infection
Author: Yu
[RULE578]

[back to question 39...]
** WHY

[i.e. WHY is it important to determine the organisms (other than those seen on cultures or smears) which might be causing the infection?]

[5.0] ... in order to determine the therapeutic regimen of PATIENT-538.
If
[5.1] information has been gathered about organisms isolated from the patient, organisms noted on smears of the patient, negative cultures of the patient, suspected infections without microbiological evidence, current drugs of the patient and prior drugs of the patient, and
[5.2] an attempt has been made to deduce the organisms which require therapy, and
[5.3] You have given consideration to organisms (other than those

seen in cultures or smears) which might be present, and found at least one organism which requires therapy.

then

determine the best therapy recommendation from among the drugs likely to be effective against the significant organisms

[RULE092]

[back to question 39...]

** Y

I have found evidence (RULE545 RULE557) that treatment should cover for E.coli causing meningitis.

I have found evidence (RULE545 RULE557 RULE578) that treatment should cover for Pseudomonas-aeruginosa causing meningitis.

I have found evidence (RULE545 RULE557) that treatment should cover for Klebsiella-pneumoniae causing meningitis.

I have found evidence (RULE507) that treatment should cover for Diplococcus-pneumoniae causing meningitis.

40) Does Pt538 have focal neurological signs?

** UNKNOWN

[Determining which drugs are desirable for use against the E.coli...]

[Determining which drugs are desirable for use against the Pseudomonas-aeruginosa...]

[Determining which drugs are desirable for use against the Klebsiella-pneumoniae...]

[Determining which drugs are desirable for use against the Diplococcus-pneumoniae...]

My therapy recommendations will be designed to treat for organisms that either are very likely or, although less likely, would have a significant effect on therapy selection if they were present. It is important to cover for the following probable infection(s) and associated organism(s):

INFECTION-1 is MENINGITIS

+ <ITEM-1> E.COLI [from clinical evidence only]
+ <ITEM-2> PSEUDOMONAS-AERUGINOSA [from clinical evidence only]
+ <ITEM-3> KLEBSIELLA-PNEUMONIAE [from clinical evidence only]
+ <ITEM-4> DIPLOCOCCUS-PNEUMONIAE [from clinical evidence only]

41) Does Pt538 have a clinically significant allergic reaction to any antimicrobial agent?

** NO

[Considering AMPICILLIN-AND-GENTAMICIN for use against INFECTION-1...]

42) Do you have reason to suspect that Pt538 may have impaired renal function?

** NO

43) Pt538's weight in kilograms (or <number> POUNDS):

** 70

[REC-1] My preferred therapy recommendation is as follows:

In order to cover for Items <1 2 3 4>:
Give the following in combination:
1) AMPICILLIN
Dose: 3.5g (28.0 ml) q4h IV [calculated on basis of 50 mg/kg]

2) GENTAMICIN
Dose: 119 mg (3.0 ml, 80mg/2ml ampule) q8h IV [calculated on basis of 1.7 mg/kg] plus consider giving 5 mg q24h Intrathecal
Comments: Monitor serum concentrations

Since high concentrations of penicillins can inactivate aminoglycosides, do not mix these two antibiotics in the same IV bottle.

3 Alternatives to First Choice Therapy Are Available

Do you wish to see the next choice therapy?
** YES
Which, if any, of the above drugs do you wish eliminated from consideration?
** AMPICILLIN

[Considering GENTAMICIN for use against INFECTION-1...]
[Considering PENICILLIN for use against INFECTION-1...]

[REC-2] Next best therapy recommendation:
In order to cover for Items <1 2 3>:
Give: GENTAMICIN
Dose: 119 mg (3.0 ml, 80mg/2ml ampule) q8h IV [calculated on basis of 1.7 mg/kg] plus consider giving 5 mg q24h Intrathecal
Comments: Monitor serum concentrations

In order to cover for Item <4>:
Give: PENICILLIN
Dose: 3.5 million units (7.0 ml) q4h IV [calculated on basis of 50000 units/kg]

RECOMMENDATION-2 is equivalent to RECOMMENDATION-1.

Would you like to see details of this comparison?
** NO

Do you wish to see the next choice therapy?
** NO

4 Answers to User's Questions about this Consultation and About the Knowledge Base

Please ask any simple-English questions regarding the above consultation session or the overall knowledge of the consultation system. When discussing the consultation above, please refer to organisms, cultures, etc. by their numeric names (eg. ORGANISM-1). Questions which were asked during the consultation may be specified by the numbers that preceded them above, (e.g., 'WHY DID YOU ASK QUESTION 1?'). Control-O will abort type-out, or 'cancel' the current question. Type HELP for a list of the options available. Enter the word STOP when you have no further questions.

Type HELP for the list of question options.

** HELP

QA-OPTIONS:
(Type ctrl-O to abort printout)
ADVICE - prints suggestions for using the QA system effectively
QT - prints examples of question types recognized by the system
EQ <numbers> - prints an explanation of the specified question(s) from the consultation, e.g., EQ 8 13
LP <object type> - prints list of attributes associated with object type, e.g., LP *CULTURE*. (*Object types are* patient, infection, culture, organism, and drug.)
PR <numbers> - prints specified rule(s), e.g., PR 373 49 87 2
EXPLTHER - prints a short description of the therapy program
REC - prompts user for alternative therapy RECommendation and compares it to mycin's recommendation
RX <infection> <organism identities> - prints treatment lists for specified organisms causing the indicated infection, e.g., RX MENINGITIS ENTEROCOCCUS
JUSTIFICATION - when rules are printed out, include their justifications
LITERATURE - when rules are printed out, include their literature references
NOJUST - stop giving justification and/or literature with rules
TERSE - print rules in terse format
VERBOSE - turns off terse printing of rules
STOP - escape from QA-module
HELP - prints this list

** JUSTIFICATION

** LITERATURE

** 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). The last question asked before the conclusion was made was 35.
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 last question asked before the conclusion was made was 37.

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

Would you like a summary of ALL the parameters used by these rules? ** YES

Parameter (rules with this parameter)

the white count from the patient's peripheral CBC (in thousands) (557)
whether the results of a recent CBC of the patient are available (557)
whether the infection was acquired while the patient was hospitalized (545)
the age of the patient (545)
the relevant items from the history of the patient (545)
the type of the infection (545, 557)
whether organisms were seen on the stain of the culture (545, 557)
whether a smear of the culture was examined (545, 557)
the infection which requires therapy (545, 557)

** WHY DIDN'T YOU GIVE TETRACYCLINE FOR E.COLI IN REC-1?
Shall I describe the therapy algorithm for you?** YES
DESCRIPTION OF THERAPY SELECTION
[Type ^O to abort printout]

MYCIN is designed to provide optimal therapy recommendations. The program uses the following three steps:
The first step ranks the drugs for each organism that requires therapy. Rules for doing this take into account invitro (lab) sensitivities and the patient's response to prior and current therapies. In the absence of other information, default lab sensitivities are used. Preference is given to current therapies if certain criteria are satisfied.
Next, 1 or 2 drugs are selected to cover for the organisms which are most likely to be causing the infection. As many of the "less likely" organisms as possible are treated, but they are disregard altogether if 3 or more drugs are necessary. Throughout, preference is given to drugs that appear higher in the treatment lists. (These lists can be examined via the RX option.)
Finally, the set of drugs is tested for patient factors that preclude their administration, such as allergies, age, and renal function.
TETRACYCLINE was not prescribed for ITEM-1
TETRACYCLINE was not prescribed for ITEM-1 in RECOMMENDATION-1 because:

TETRACYCLINE was discounted for ITEM-1 because there is ... be that this e.coli is not sensitive to it.

Would you like to see some details?** NO

** HOW DO YOU USE THE TYPE OF THE INFECTION?
The following rules use:
the type of the infection

154, 578, 574, 572, 570, 565, 564, 559, 557, 553, 551, 549, 547, 545, 543, 541, 539, 537, 535, 533, 531, 529, 523, 520, 519, 517, 515, 513, 511, 507, 300, 289, 288, 287, 280, 174, 167

The parameter most frequently concluded by these rules is:
the organisms (other than those seen on cultures or smears) which might be causing the infection

Would you like a summary of ALL the parameters concluded by these rules?
** NO

Which of the rules do you wish to see?
** 280

RULE280

[This rule applies to all cultures, and is tried in order to find out about the organisms (other than those seen on cultures or smears) which might be causing the infection]

If: 1) The infection which requires therapy is meningitis, and
2) The type of the infection is fungal, and
3) Organisms were not seen on the stain of the culture, and
4) The patient is not a compromised host, and
5) The patient has been to an area that is endemic for coccidiomycoses, nd
6) The race of the patient is one of: black asian indian, and
7) The cryptococcal antigen in the csf was not positive
Then: There is suggestive evidence (.5) that cryptococcus is not one of the organisms (other than those seen on cultures or smears) which might be causing the infection
Author: YU
Justification: Dark-skinned races, especially Filipino, Asian, and Black (in that order) have an increased susceptibility to coccidiomycoses meningitis.
Literature: Stevens, D.A. et al. Miconazole in Coccidiomycosis. Am J Med, 60:191-202, Feb 1976.

** HOW DO YOU TREAT PNEUMOCOCCAL PNEUMONIA
The MYCIN system currently can treat the following infections: BACTEREMIA MENINGITIS CYSTITIS. Because the knowledge base is incomplete, all other infections will be treated like bacteremia.

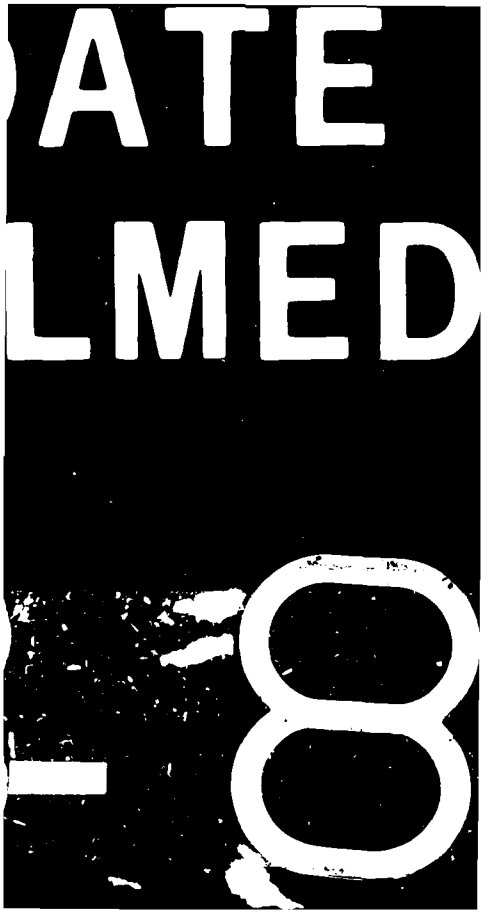
For treatment of DIPLOCOCCUS-PNEUMONIAE in BACTEREMIA
The drugs of choice (ranked by relative efficacy, assuming sensitivity)

are:
    1st choice:
PENICILLIN
AMPICILLIN
    2nd choice:
NAFCILLIN
CEPHALOTHIN
ERYTHROMYCIN
CLINDAMYCIN
    3rd choice:
TETRACYCLINE
CHLORAMPHENICOL
VANCOMYCIN
    Duration of therapy: 28 days

** WHAT DOSE OF ERYTHROMYCIN DO YOU RECOMMEND?
When ERYTHROMYCIN is given for BACTEREMIA, the recommended dosage is:
if age is > 12 yrs then 12 mg/kg q6h IV
    else if age is > .08 yrs then 5 mg/kg q6h IV
    else if age is > .02 yrs then 5 mg/kg q12h IV
[Different dosages are recommended for different infections.]

** HOW DO YOU TREAT CYSTITIS
In general, treatment for an infection depends on the causative organism.
Please include in your question the organism of interest.

** STOP


ATE
LMED
8