Report 84-15
Stanford -- KSL

Scientific DataLink

Expert Systems.
Bruce G. Buchanan,
Jul 1984

Heuristic Programming Project
HPP-84-15

July 1984

EXPERT SYSTEMS

Bruce G. Buchanan
Department of Computer Science
Stanford University

EXPERT SYSTEMS

Bruce G. Buchanan

2 July 1984

Computer Science Department

Stanford University

(415) 497-0935

Table of Contents

EXPERT SYSTEMS

Bruce G. Buchanan[1]
Stanford University

1 AUTOMATED REASONING: LOGICAL AND HEURISTIC

The idea of automated reasoning is founded on the fact that computers are general-purpose symbol manipulation devices, and not mere numerical calculating machines. Symbolic inference since the time of Aristotle has involved the combination of symbolic expressions, but only in the last few decades has it been possible to build automated reasoning systems.

One line of attack on the problem of how to use computers for automated reasoning is the logical one: exploit the syntax and rules of deductive logic as expressed by Aristotle, Russell & Whitehead, or Church. Extend the formalism where necessary to represent useful concepts not easily expressed. But focus sharply on retaining the logical consistency that these formalisms provide. A primary research problem is finding computational methods that are efficient enough for this theorem-proving approach to be applied to reasoning problems of real-world complexity. This line of research is well represented on the editorial board of this journal.

But there is a second line of research on automated reasoning, which is the heuristic one: use whatever knowledge is available to make and guide plausible inferences, regardless of whether the knowledge is logically consistent or the inferences are deductively valid. This is the spirit in which most of the complex applications of artificial intelligence (AI) have been undertaken, and which motivated all of the recent work on expert systems. DENDRAL [5] is a good example, although there are many earlier ones. DENDRAL uses (in fact, needs) a large number of heuristics to solve a chemical structure problem; and the heuristics themselves can be inferred from analytic data by plausible inferences.

Polya [9] wrote about using heuristics in the "art of good guessing" in mathematics, and we see all around us examples of problems that require more than mathematical or strictly logical reasoning for their solution.

---

[1]This work was suppported, in part, by DARPA contract #N00039-83-C-0136, NSF grant #NSF-IST 83-12148, NASA grants #NCC 2-274 and #NAG 5261, and NIH grant #NIH-RR 00785-11. Several persons have given me assistance in formulating section 4 of this paper. I wish to thank in particular Ed Feigenbaum, Mike Genesereth, Yumi Iwasaki, and Earl Sacerdoti.

Many of the world's difficult problems are of this nature, such as troubleshooting electronic or mechanical equipment, medical diagnosis and therapy planning, and configuring orders for a computer into a complete description of a whole system. For problems such as these, AI offers new opportunities for developing computer-based solutions.

2 WHAT ARE EXPERT SYSTEMS?

Expert systems are first, and foremost, AI programs. AI is distinguished from other areas of computing in its attention to both symbolic (non-numeric) information and heuristic (non-algorithmic) methods for solving problems. While it certainly does not ignore mathematical expressions of knowledge and variables with numeric values, AI has a special place in computer science in dealing with symbolic inference. Similarly, AI methods often include logical inference but AI is also unique within computer science for focusing on heuristic reasoning.

Expert systems solve problems at the level of experts. Newell remarks [8] on this use of the term:

> "It is sometimes noted that the term *expert system* is a pun. It designates a system that is expert in some existing human art, and thus that operates at human scale -- not on some trifling, though perhaps illustrative task, not on some *toy* task, to use the somewhat pejorative term popular in the field. But it also designates a system that plays the role of a consultant, i.e., an expert who gives advice to someone who has a task. Such a dual picture cannot last long. The population of so-called expert systems is rapidly becoming mongrelized to include any system that is applied, has some vague connection with AI systems and has pretentions of success." (pp. xi-xii)

Expert systems are *flexible* in the sense that they can be changed and extended easily, and they are *understandable* in the sense that they can explain the contents of their own knowledge bases and their own lines of reasoning [4]. These features were recognized early to be especially important in medicine, where knowledge is changing rapidly and where practitioners have to understand the reasons for a program's decisions because they have to accept responsibility for following (or not following) those decisions. Thus some of the first expert systems were developed in medical domains.

In other domains, flexibility and understandability may not be of such direct importance as in medicine, but are still critical in the development of the knowledge bases. Knowledge bases for expert systems are now built iteratively -- usually through long interactions over many months between a human specialist, an expert, who understands the details of the domain and another specialist, known as a knowledge engineer, who understands the programming details of the system.

One of the key ideas in maintaining flexibility and understandability is the clean separation of elements of the

knowledge base from elements of the program that interpret the knowledge base. This simple principle is now well recognized, but in the early days of expert systems -- namely while the DENDRAL program was first being built -- this was a novel idea. Its simplicity, however, does not mean that it is often followed, or even easy to implement.

Note that the *source* of expertise has not been specified in this list of characteristics of expert systems. It is sometimes said that expert systems are just those whose knowledge *comes from* experts. It is obvious that human practitioners of a problem solving task constitute an important source of expertise. But it is also obvious, I believe, that future systems will be able to acquire new knowledge from other sources. Although the first generation of expert systems has been built with human experts as the exclusive source of expertise, there seems no particular reason to separate high performance AI systems into very different classes based on the source of their knowledge. In order for these systems to make their reasoning understandable, however, the concepts and inference rules they use must themselves be understandable. Although it is not necessary that the reasoning methods be the same as those of the human specialists, it is important that the concepts and inferential relations be natural enough to be modified and extended by them.

3 HISTORICAL BACKGROUND

Applications of computers to reasoning problems in medicine, science, engineering, law, and nearly every other discipline were foreseen in the early days of computers.[2] The first large applications of AI to tasks outside of game playing and puzzle solving were DENDRAL [5] and MACSYMA [7]. Their stories and the role of knowledge in their success are well-known [2]. It was at the time of their development in the mid 1960's that the difference between the two ways of looking at automated reasoning -- the logical and heuristic -- became apparent. Those two programs, as their expert system successors, depended on very specialized knowledge about their domains (chemistry and mathematics, respectively).

In medicine, CASNET, INTERNIST, PIP and MYCIN were built shortly after DENDRAL. Several other systems were constructed about the same time (late 1960's and early 1970's) that gained expertise from encoding knowledge from experts and textbooks, including the chemical synthesis planning programs SECS and SYNCHEM, the signal understanding program, HASP/SIAP, speech understanding programs of which HEARSAY-II is the best known, and the mineral exploration advisor, PROSPECTOR.

---

[2] Ramon Llull seems to have had similar ideas in the late 13th cen. in developing his reasoning machines to answer theological and other questions. Leibniz was considerably influenced by Llull in his desire to reduce argumentation in every discipline to a set of mechanical calculations. Llull's machines were actually built, but, as one might guess, depended to a large degree on the user to interpret a question "properly" and to interpret the output of the device. Leibniz' machines were not built. See [3] for more discussion.

MYCIN has come to be known as a prototypical expert system, even though it was never used routinely by physicians.[3] Its task of infectious disease diagnosis and therapy is relatively easy to understand. Most of all, it is easy to understand because of the clear separation of medical knowledge (encoded mostly in rules) from the interpreter of that knowledge. MYCIN also contains several parts that have come to be recognized as important elements of expert systems: an explanation module that explains the program's line of reasoning as well as the contents of its static knowledge base: an interface module that understands questions in a technical subset of English and constructs readable English answers; a (rudimentary) knowledge acquisition and maintenance package.

MYCIN's reasoning is primarily goal-oriented because this provides a strong focus on which facts are relevant and thus on which data need to be collected. This simple reasoning process has been augmented in a number of ways to make the program focus attention better and reason more efficiently. Forward-chaining was introduced to capture some definitional rules ("if the patient is male, then he is not pregnant"). The interpreter was given the ability to preview clauses in the premise of a rule to see if any conjunct was already known to be false before trying to establish the first conjunct as a subgoal. Most importantly, the rules were given associated measures of certainty and the interpreter was given a method for reasoning under uncertainty. These changes, and more, made MYCIN's reasoning less rigid than if it were pure backward-chaining with binary evaluation of premise clauses.

Rules are only one simple representation of knowledge, however. They were successfully used in DENDRAL and MYCIN, but the simplicity that makes them easy to work with also limits their expressive power. Not all expert systems are rule-based, nor should they be. Similarly, backward chaining is only one reasoning strategy. It facilitates construction of a coherent line of questions and of reasonable explanations, but it is not the best strategy for all problems.

A significant development in research on expert systems was the introduction of framework systems that provide an inference engine and syntax for knowledge but contain no problem-specific knowledge themselves. EMYCIN was designed and written in the mid-1970's by William van Melle as a test of the claim that MYCIN's inference engine was completely independent of the knowledge base. He developed generalizations of the tools in MYCIN, and developed new tools, that made EMYCIN a useful environment for building and interpreting knowledge bases for new problem areas. It is not a framework for building every kind of expert system. On the contrary, its utility is limited to a class of problems like MYCIN's: selecting plausible answers to a problem from a fixed set of alternatives by gathering and weighing evidence

---

[3] See [1] for a thorough discussion of MYCIN.

for the alternatives. By fixing the representation of knowledge and the modes of inference, however, framework systems allow builders of expert systems to start from a substantial base of programs and to concentrate on formulating the contents on the knowledge bases without having to design new data structures and programs that manipulate them. By now, several other framework systems have been built and used in both research and commercial settings.

It should be noted that expert systems are not the only applications of AI to important problems. Other important applications areas have been separated and named:

- Robotics: Vision & Manipulation
- Natural Language Understanding
- Automatic Programming

Programs in these areas share many of the characteristics of expert systems, but deal with areas of expertise requiring very specialized data structures and inference methods. In work on vision, for example, numerical methods for computational geometry constitute large parts of every vision program.

4 ARE THESE SYSTEMS "REALLY" EXPERT?

It has been popular to argue against the very idea of artificial intelligence by claiming that even though AI programs appear at times to behave intelligently they are not "really" intelligent. A variant of this argument has surfaced with respect to expert systems: that they only appear to behave expertly but are not "really" expert. Use of the term "really" is slippery, however, and hides the shifting criteria that are employed every time the behavior of the programs improves. There is no reason to take this criticism seriously.

There are three substantial criticisms of expert systems, however, which do need to be addressed. They are that expert systems are not going to pay off in the long[4] run because they lack:

- successively deeper layers of knowledge of their task areas to use when the shallow, compiled knowledge fails to reach a satisfactory answer,
- common sense to avoid errors due to reading the expressions of knowledge too literally or due to incomplete coverage of possibilities within the explicitly stated knowledge base, and
- ability to learn from experience.

The current generation of expert systems, does lack these three kinds of knowledge, but that is not to say that expert systems in the future will necessarily lack them. Nor does that lead directly to the conclusion that today's systems are not expert. Let us look at them separately.

---

[4]Some critics seem to imply that this line of work is too limited to pay off for anyone; others seem to argue that work on expert systems will not pay off for AI research in the long run.

Deeper Layers of Knowledge

Rule-based systems encourage encoding loose, empirical associations in the knowledge base to help expert systems reason about plausible solutions to problems. This is true also of frame-based systems. It can be true of the associations in logic-based systems, if the system builder deliberately uses implication in the sense of plausible, not deductive, inference. In MYCIN, many of the rules are empirical associations that lack a sound theoretical justification, often because medical scientists have not yet discovered the theory. Other rules are definitional, and thus encode a part of the exiting theory of medicine. Still other rules are surface-level encodings of associations between causes and effects, which skip the underlying, "deeper" layers of knowledge that explain the associations. In this sense, a rule may be "compiled" knowledge in that it accurately allows a system to reason from A to B but has skipped over the intermediate steps that persons sometimes go through, or appeal to, to justify B in the context of A.

One serious manifestation of this problem appears in the explanations that expert systems currently give of their line of reasoning. While a person can explain a phenomenon at successively deeper layers of detail, current systems show the individual elements of the knowledge base used to draw a conclusion without showing the "decompiled" forms that would justify those elements. Another related manifestation appears in the context of tutoring. A student trying to learn the contents of an expert system's knowledge base needs deeper layers of structure to help tie the elements of the knowledge base together.

MYCIN's rules, as with most current rule-based systems, were written and refined with a specific task in mind -- diagnosis and therapy in MYCIN's case. The knowledge base is not generally useful for other tasks but is engineered tightly for a single purpose. This is another sense in which a set of rules constitutes compiled knowledge and is a strong argument for a more declarative representation of knowledge than a set of rules provides.

MYCIN would be admittedly more knowledgeable if it had more knowledge, in particular, knowledge of the physiological and biochemical processes that justify many of its rules. Its explanations and tutorial dialogues could be smarter, and the deeper layers could make it easier to build and maintain the knowledge base. But it would not necessarily perform its tasks of diagnosis and therapy better in the kind of constrained context in which systems are now being designed.

Common Sense

McCarthy [6] has argued that MYCIN, and other expert systems, are bound to behave poorly at times because they lack common sense. He describes an interchange in which MYCIN looks stupid to him because it fails to

object to the possibility of amniocentesis for a male patient. Obviously, this could be remedied with the same kind of rule that prevents MYCIN from accepting the possibility of pregnancy for male patients. But McCarthy is pointing to a general fault that without common sense, there will always be failures of this simple kind. With common sense, people sometimes are misled when they fail to use it -- so just *having* it is not enough.)

McCarthy's point is well taken: if expert systems know more, particularly if they have more common sense, they will perform better. Current systems often exhibit the kind of brittleness that McCarthy points out because they make strong assumptions about the context in which they will be used, the types of users, the vocabulary, the "reasonableness" of other lines of reasoning, and so forth. They also tend to have rather sharp fall-off in performance at the boundaries of their knowledge. In common parlance, they "fall off knowledge cliffs" when we would expect an expert's performance to degrade gracefully at the boundaries of his or her knowledge.

But on the positive side, the context for which an expert system is designed to be used can limit the amount of common sense that is necessary in practice. For MYCIN, physicians using the system were assumed to have common sense enough not to tell MYCIN that a male patient has had amniocentesis -- or thousands of other things that would make MYCIN appear to be stupid. The users were assumed to want help enough to supply sensible information to MYCIN.

Learning From Experience

Current expert systems do <u>not</u> improve their own behavior based on experience; does that mean they are not "really" expert? This is a definitional question on which one may take a dogmatic stand[5] . However, if we use a performance-based definition of expertise and not a dispositional one, then we may be less dogmatic and say that a person, or a program, is an expert by virtue of excellent performance, regardless of how he, she or it gained his/her/its knowledge in the first place (and kept it current). For example, some of us, anyway, would prefer to have a medical problem diagnosed by a physician with 20 years' experience who knows 90% of what is relevant for our problem -- even if he or she has stopped learning -- instead of having the problem diagnosed by a recent medical school graduate who knows only 50% of what is relevant but who is learning rapidly. When the knowledge curves of the two cross, if we could measure them, then we would change physicians.

MYCIN, and other more current systems, do lack considerable knowledge that experts have. To argue that,

---

[5] As Schank did in the McNeil-Lehrer TV interview and in [10].

for this reason, they cannot *perform* at the level of experts is fallacious. Consider a similar argument[6] about chess playing programs:

- Because a program that plays master-level chess lacks knowledge that master chess players have, they cannot play master-level chess.

Focusing on performance puts the dispute on empirical grounds -- either a program does or does not do something as well as some set of people.[7]

5 FUTURE DIRECTIONS

Current AI research in many different areas can mean increased capabilities for expert systems. For example, the work of Forbus, deKleer, or Kuipers on qualitative reasoning could greatly enhance the reasoning power of systems over their present capabilities. Also, current work on meta-level reasoning can give programs a better sense of knowing their own limitations -- a form of common sense knowledge. Conversely, making progress on building more capable expert systems may provide insights on some of the fundamental issues. In particular, the issues of representing and using knowledge, inexact reasoning, models of interaction, explanation, knowledge acquisition, and validation of knowledge bases are important outside the bounds of expert systems, but become sharply focused in this work.

Simple, rule-based systems are now straightforward to build. They can be important for helping people solve problems for which expertise is in short supply, or is not well distributed, or is not available around the clock. Even though the problems are narrow, some assistance in solving them can improve overall productivity, efficiency, or quality of the broader system. More complex problems will require more complex knowledge structures and reasoning methods. They may well require knowledge of a qualitatively different kind. The next decade should prove to be a time of trying and testing many new ideas for extending the capabilities of expert systems.

Every limitation of an expert system presents opportunities for research, including the three areas of criticism listed above. One of the major benefits of focusing sharply on an application is that the limitations are difficult to ignore and proposed improvements have to pass the operational test of improving the performance of the expert system. For these reasons, and more, expert systems provide a rich environment for studying automated reasoning.

---

[6] Earl Sacerdoti pointed out this analogy.

[7] As we discovered in measuring the level of MYCIN's performance, it is not always *easy* to answer the empirical question, nor even to design a good study, but we understand what it means to do so.

References

[1] Buchanan, B.G., and E.H. Shortliffe.
Rule-Based Expert Systems: The MYCIN Experiments.
Addison-Wesley, New York, 1984.

[2] Feigenbaum, E. A., Buchanan, B. G. and Lederberg, J.
On Generality and Problem Solving: a Case Study using the DENDRAL Program.
Machine Intelligence 6:165-190, 1971.

[3] Gardner, Martin.
Logic Machines and Diagrams.
Mc-Graw-Hill, New York, 1958.

[4] Hayes-Roth, F., Waterman, D., and Lenat, D. (eds.).
Building expert systems.
Addison-Wesley, New York, 1983.

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

[6] McCarthy, J.
Some expert systems need common sense.
Annals of the New York Academy of Science, 1983 (forthcoming).
Invited presentation for the New York Academy of Sciences Science Week Symposium on Computer Culture, April 5-8, 1983.

[7] Moses, J.
Symbolic integration: The stormy decade.
Communications ACM 8:548-560, 1971.

[8] Newell, A.
Foreword.
In Buchanan, B.G. and Shortliffe, E.H. (editor), *Rule-Based Expert Systems*, pages xi-xvi. Addison-Wesley, Reading, MA, 1984.

[9] Polya, G.
Mathematics and Plausible Reasoning.
Princeton University Press, Princeton, N.J., 1954.

[10] Schank, Roger C.
The Current State of AI: One Man's Opinion.
AI Magazine IV (D Winter-Spring), 1983.