HPP-79-11 (Working Paper) March 1979

Transfer of Expertise: A Theme for AI Research

Avron Barr, James Bennett, and William Clancey

Heuristic Programming Project
Stanford University

Considering the many avenues and approaches taken by AI researchers in the first 25 years of work in the field, one is often hard-pressed to define the nature of AI or to identify the "shared scientific principles" that unify the enterprise. This is no less true of our research at the Heuristic Programming Project, which deals with the construction of "expert systems" in scientific and medical domains. This area of AI, which we call *knowledge engineering* (Feigenbaum, 1977), involves research on all of the following problems:

How do human experts solve problems?
How do they talk about what they know?
How can their knowledge be represented in a program?
How should programs use this knowledge to solve problems?
How should they explain these solutions to humans?

These broad questions involve psychology, epistemology, and human language, as well as computer science and AI. The actual research has dealt with diverse topics like developing capabilities for English dialogue with the systems and exploring the relative benefits of different representation schemes for the systems' knowledge base. These problems and their solutions are at times difficult to relate, but we feel that a number of underlying design principles are emerging. This paper is an attempt to present them.

A key idea in our current approach to building expert systems is that these programs should not only be able to apply the corpus of expert knowledge to specific problems, but they should also be able to interact with the users and experts just as humans do when they learn, explain, and teach what they know. We will show, as we review the major developments in the design of expert systems at HPP, that these *transfer of expertise* (TOE) capabilities were necessitated by "human engineering" considerations--the people who build and use our systems needed a variety of "assistance" and "explanation" facilities. However, we believe there is more to the idea of TOE than the implementation of needed user features: These social interactions--learning from experts, explaining one's reasoning, and teaching what one knows--are essential dimensions of human knowledge. They are as fundamental to the nature of intelligence as expert-level problem-solving, and they have changed our ideas about representation and about knowledge.

The Form of Expertise

Work at the HPP has, from the beginning, been concerned with constructing problem-solving systems that perform at the level of human experts in a specific scientific, medical, or engineering domain. At first, the only constraint on a system's design was its *performance* relative to human experts. To achieve high-level performance, expert systems use domain-specific heuristic knowledge, encoded primarily as production rules, to represent what the expert says he would do in specific situations. Originally, the rules were incorporated directly into the program by the system's designer after many hours of consultation with the human expert. These *knowledge acquisition* sessions between designer and expert served to identify and to organize what knowledge the expert brought to bear on particular problems.

Although production rules have not proven to be completely adequate for representing all aspects of the domain expertise, they are a very convenient formalism for encoding what experts say they would do in a given situation. In other words, the "situation-action" *form* of production rules comfortably captures the way people *talk about* their problem-solving behavior. On the other hand, the method for applying rules in these systems, for example backward-chaining, does not necessarily mimic the expert's own reasoning.

Early in the development of these systems, the problem of automating the knowledge acquisition process was considered. These investigations soon demonstrated that additional constraints on system architecture would be necessary: Experts were unable to remember the interrelations among rules in a large and complex knowledge base well enough to deal with all the ramifications that new rules might have on the system. Therefore, the relation between the system's representation of the problem-solving rules (as situation-action pairs) and the associations the experts actually had in mind had to be even closer than the basic similarity of *form* offered by the production formalism: The system had to be able to support additional knowledge *about the rules themselves*. Cast in the form of "rule models" and other annotations, this additional knowledge enabled the system to relate its knowledge to human users in a human way (Davis, 1976).

Explanation in Expert Systems

One of the important contributions to artificial intelligence of work on MYCIN and related systems has been the persistent emphasis on the systems' ability to *explain* the reasoning behind their decisions (Shortliffe, 1976). The purpose of the explanation is for the user to be able to validate the program's reasoning, and modify (or reject) the advice if he believes that some step in the decision process is not justified. In MYCIN and other production systems, the actual solution to a problem may involve perhaps a hundred production rules. The explanation facility must produce a "line of reasoning" that makes sense to humans, by identifying and summarizing the important steps.

As it turns out, both the user and the expert/designer need the explanation facility. While the normal user needs explanation as a justification for the solution that the system offers, the expert needs the explanation for another important reason: *debugging* the system's reasoning when its solution is incorrect. The systems we are building are becoming so complex that unless they can explain what they are doing in an understandable fashion, no one, not even their designers, would know what went into a problem's solution. The explanation, therefore, must be adequate either to show how the system got the right answer or to indicate where the system went astray.

Explanation plays a similar role in human cognition and discourse. A physican diagnosing a case uses a vast amount of knowledge: commonsense reasoning that he learned as a child, thousands of "book-learned" facts from college and medical school, and his clinical experience gleaned from hundreds of relevant cases. When asked to explain his conclusions, he simply cannot detail the way that all of this knowledge affected his decisions. Yet he is somehow able to explain what he has done in a manner that is sufficiently convincing. Should a second physician, for instance, disagree with the diagnosis, he may use the original explanation to identify a "bug" that led to his colleague's error. As a psychological phenomenon, explanation is not well-understood, but it is essential to systems that are designed to interact with human scientists and other professionals.

Tutoring as Explanation and Debugging

The field where human explanation has been most carefully considered is education. For example, the relationship between explanation and debugging is best exemplified in a "tutorial dialogue," where the expert's job is to listen to the student's explanation of his own reasoning until he figures out where the student went wrong. Our ideas about explanation and Transfer of Expertise came together when we began work at HPP on a new kind of program, a computer-based instructional system (Clancey, 1979). This program, called GUIDON, is intended to tutor a medical student in MYCIN's domain of expertise, the diagnosis of patients with infectious diseases.

In order to "reason" about a case with a student, GUIDON must explain MYCIN's line of reasoning in a way that the student can understand. Not only must the instructional system select the important diagnostic principles to point out to the student, but it must also take into account what the student already knows and what knowledge he needs in order to understand subsequent explanations. In other words, the system must know how to teach, just as a human tutor does, and this knowledge about teaching is knowledge about the transfer of expertise. This *transfer of expertise expertise* is primarily domain-independent and does not involve MYCIN's knowledge about infectious diseases--it refers only to the structure of the internal representation the system uses. In fact, we are developing a system, called EMYCIN, for building expert systems in other domains using MYCIN's representation and control schemes (van Melle, 1979). We have already constructed "MYCIN-like" systems in several other domains, including pulmonary dysfunction (Kunz et al., 1978), psychopharmacology (Heiser, 1978), and structural analysis (Bennett et al., 1978). Since its knowledge of tutoring is domain independent, we eventually expect the GUIDON tutor to offer instruction through these and other knowledge-based systems constructed with EMYCIN.

Thus, the ability to transfer expertise via dialogue, explanation, and question-answering requires knowledge about knowledge or *meta-level knowledge*--it is knowledge about how to transfer knowledge. The key breakthrough in the development of the TOE methodology was made by Davis (1976) in his work on automating knowledge acquisition and explanation in the TEIRESIAS system. He found that *explicitly* representing knowledge about the extent, origin, and structure of other entries in the knowledge base was essential to enabling the system to discuss what it knows and how it reasons. TOE expertise involves the use of at least four kinds of knowledge:

1. Knowledge about the internal representation and architecture of the system;

2. Knowledge about the structure of the rule set, i.e., the patterns and interactions among the domain rules;

3. Support knowledge to justify and explain individual rules and place them within an underlying mechanism model for the domain (i.e., in the case of MYCIN, the pathophysiological mechanism responsible for meningitis); and

4. Explicit discourse knowledge for guiding the interaction between the student and the tutor.

TEIRESIAS explored the expert's interaction with the system as he attempts, in the context of a specific case, to identify and correct a shortcoming in the knowledge base. The expert could indicate that some conclusions that MYCIN had reached about this case were incorrect and then request an explanation of MYCIN's reasoning that had led to those faulty conclusions. After identifying the error in MYCIN's rule base, he could then correct the situation by adding a new rule or modifying an existing rule.

A similar type of interaction occurs between the instructional program, GUIDON, and a medical student. Here, GUIDON uses its knowledge about how MYCIN would reason through a case to tutor the student regarding proper diagnostic procedures. Again, in the context of a specific case, GUIDON allows the student to venture hypotheses about various diagnostic points involved in the discussion. GUIDON then critiques the student's hypotheses with respect to the conclusions MYCIN drew and attempts to explain how MYCIN was able to arrive at these conclusions. Thus, GUIDON, coupled to MYCIN, takes on the role of the expert attempting to "debug" the student's reasoning.

Although there seem to be types of TOE expertise unique to tutoring (generating quiz questions) and to knowledge acquisition (acquiring new parameters and rules), there is a nontrivial overlap. The similarity between the dialogues of TEIRESIAS with a human expert and those of students with GUIDON suggests that there is a core set of TOE expertise that a single system might apply whenever engaged in a debugging session with either expert or student.

TOE Expertise and Meta-knowledge

This brings us back to the original question of underlying principles and unifying ideas in our research. We have argued that if we view these expert systems not as performance programs that solve problems in their domain, but rather as cognitive systems that are able to discuss their domain as humans do, then the issues involved in these TOE interactions of acquisition, explanation, and tutoring constrain all parts and levels of the system's design:

> The representation must not only be adequate for solving problems: it must also be accessed by "explanation" and "acquisition" processes that know what the system knows and how it reasons.
>
> The natural language interaction goes beyond "decoding" the user's input, becoming more like human dialogue where the "state of knowledge" and "goals" of the participants are the primary consideration in deciding what is said and what is meant.

The way that the system solves problems must be amenable to explanation for interactively acquiring new knowledge, justifying conclusions to users, and instructing students.

At our present stage of research at the Heuristic Programming Project, with the explicit representation of meta-knowledge and TOE expertise as new tools, we are building systems that take part in the human activity of Transfer of Expertise among experts, practitioners, and students in different kinds of domains. Our problems remain the same as they were before: We must find good ways to represent knowledge and meta-knowledge, to carry on a dialogue, and to solve problems in the domain. But the guiding principles of our approach and the underlying constraints on our solutions have subtly shifted: Our systems are no longer being designed solely to be expert problem solvers, using vast amounts of encoded knowledge. There are aspects of "knowing" that have so far remained unexplored in AI research. By participating in *human* transfer of expertise, these systems will involve more of the fabric of behavior that is the reason we *ascribe* knowledge and intelligence to people.

References

Barr, Avron. *Meta-knowledge and Cognition.* Memo HPP-79-6, Heuristic Programming Project, Stanford University, February, 1979.

Bennett, J. S., Creary, L., Englemore, R., and Melosh, R. *SACON: A Knowledge-based Consultant for Structural Analysis*, Memo HPP-78-23, Heuristic Programming Project, Stanford University, September, 1978.

Brown, J. S., Collins, A., and Harris, G. *Artificial intelligence and learning strategies.* In H. O'Neil (Ed.) **Learning Strategies.** New York: Academic Press, 1978.

Buchanan, Bruce G., and Feigenbaum, Edward A. DENDRAL and Meta-DENDRAL: Their applications dimesion. *Artificial Intelligence*, **11**, 1978, 5-24.

Clancey, William. *The Structure of a Case Method Dialogue.* **International Journal of Man-Machine Studies**, January, 1979.

Collins, Allan. *Processes for Acquiring Knowledge.* In R. C. Anderson, R. J. Spiro, and W. E. Montaque (Eds.), **Schooling and the Acquisition of Knowledge.** Hillsdale: Lawrence Erlbaum, 1977.

Davis, Randall. *Applications of meta-level knowledge to the construction, maintenance, and use of large knowledge bases* (doctoral thesis). Heuristic Programming Project, Memo HPP-76-7, Stanford University, July, 1976.

Feigenbaum, Edward A. The Art of Artificial Intelligence: I. Themes and case studies of knowledge engineering. *IJCAI-5 Proceedings*, 1977, 1014-1029.

Heiser, J. F., Brooks, R. E., Ballard, J. P. Progress Report: A Computerized Psychopharmacology Advisor. *Proceedings of the 11th Collegium Internationale Neuro-Psychopharmacologicum*, Vienna, 1978.

Kunz, Fallat, McClung, Osborn, Votteri, Nii, Aikins, Fagan, and Feigenbaum. *A physiological rule based system for interpreting pulmonary function test results.* Memo HPP-78-19, Heuristic Programming Project, Stanford University,November, 1978.

Shortliffe E. H., **Computer-based clinical therapeutics:** MYCIN, American Elsevier, 1976.

van Melle, William. *A domain-independent production-rule system for consultation programs.* Submitted to IJCAI-6, 1979.

Winograd, Terry. *A framework for understanding discourse.* In M. Just and P. Carpenter (Eds.), **Cognitive Processes in Comprehension**, Lawrence Erlbaum Associates, 1977.