Systems That Know That They Don't Understand*

Avron Barr
Aldo Ventures
181 Bryant Street
Palo Alto, California 94301

There is an important underlying difference in perspective between Artificial Intelligence and Neuroscience: AI studies the way in which systems (computer programs) might *understand* such things as scenes, sentences, and problems. Neuroscience studies an organic system, the Brain, which is best viewed in terms of perpetually *coming to an understanding*, rather than having one.

The metaphor of computation itself is the new leverage that AI brings to the study of mind. However, since the field parted company with Cybernetics and Self-Organizing Systems research in the mid-1950s, work within AI has focussed primarily on computational models of "reasoning." Expert Systems is a hot area of AI research with important scientific and commercial impact. The prominence of expert systems research within AI is a natural consequence of the focus on logical reasoning.

A new, computational Theory of Understandings is sketched that places AI research on reasoning within a larger framework of mental activity. A computer system, called CONSENSYS, is described which, based on this framework, facilitates the process of "coming to an understanding." Viewing the function of Mind from the perspective of a system's coming to an understanding is a powerful idea. And if mankind has a problem in need of new ideas and new technology, coming to an understanding is it.

Systems that don't understand are the class of intelligent systems that includes nervous systems, gene pools, social systems, and ecosystems. Consider an example of a particular type of social system which is familiar to all of us: What is a *science*? It is an agreement about the understanding of a certain set of phenomena: namely, those *repeatedly observable* phenomena that constitute the science's corpus of expermimental results. This understanding includes not just the methodology, the current theory, and the supporting data: A science includes the anomalous results, the conflicting theories and methodologies, the current "hot spots," its own historical evolution, and even an understanding of its own goals and purpose.

In other words a science is an evolving, but never finished, interpretive system. And fundamental to science (unlike most human social endeavors) is its questioning of what it thinks it knows. A scientific theory or concept that offers no questions is not interesting and will be replaced by one that does. Scientific knowledge is an example of a system that doesn't understand, by its very definition. It is a system for coming to an understanding. This class of systems can be seen as a very important one indeed. For example, a gene pool can be viewed as a system for building a survival-oriented understanding of a never-to-be-understood environment. And even the

---

* Presented at *Cognitiva 85: Artificial Intelligence and Neuroscience,* Centre d'Etudes des Systemes et des Technologies Avancees, Paris, June, 1985. Revised in December 1985 for a colloquium at the University of Delaware.

everyday example of "understanding" another person or a conversation is easily seen to be an incompletable task.

The nervous system is an organic system of this class. In relating research in Artificial Intelligence to Neuroscience, one immediately sees that AI systems are not, yet, systems that don't understand. But they are a very important first-order approximation.

The Impact of Expert Systems

Artificial Intelligence is a subfield of Computer Science--the study of computing machines and their programs. AI researchers design computer systems that interact with their environment with some of the traits that characterize intelligence in humans. The subdisciplines of AI include work on robotics and vision, on responding to sentences in written and spoken language, on puzzle and problem solving, and on the tools and concepts of programming itself.

Nowadays, foremost among the subdisciplines of AI is work in an area called Expert Systems. These are programs that facilitate the extraction of heuristic knowledge from human "experts" in certain types of fields like medicine. The resulting programs can then be applied effectively to problems which the expert never saw, by people who are not themselves experts. The success of the field of Expert Systems is apparent--it is the hottest new commercial technology since genetic engineering and accounts for more than half of the current research funding for AI. This success deserves some analysis.

First of all, the work was a success in the eyes of the scientists involved in its conception because they accomplished what they said they could accomplish: demonstrating the possibility of "artifical intelligence" by building computer programs that could solve problems that were hard by any standards. For instance, they built programs that analyzed chemical data or interpreted patients' symptoms in cases that required the best human experts. These programs performed as well on very hard problems as did their obviously intelligent human couterparts [Lindsay, Buchanan, Feigenbaum, Lederberg].

Secondly, the commercial potential of these systems is indicated by the number of scientists who are starting companies in this area and, more dramatically, by the amount of venture capital that is backing them. The optimism is well founded--there are demonstrated examples of expert systems that make (or save) substantially more money than they cost. And the cost of building this kind of program is likely to go down rapidly as new generations of tools become available.

But there is a third aspect of the success of expert sytems that is even more important. The pioneering work on expert systems by Ted Shortliffe, Randy Davis, Ed Feigenbaum, Bruce Buchanan and their colleagues in the MYCIN group at Stanford was directly concerned with what they called "Transfer of Expertise" [Barr, Bennett, and Clancey]. They focussed on making MYCIN, a medical diagnosis system, able to explain its reasoning and justify its conclusions. The key scientific contribution of this work is that there are computational techniques (e.g., rule chaining) that allow the interactive development of programs which can not only solve problems but can also examine their own reasoning and make sense of it to people (by revealing the same rules that were originally acquired from experts and then used in the system's reasoning).

A word about how this effect was achieved: One can view the computer as an interpretive system

that "processes" a fixed repertoire of symbols (its instruction set). A Programming Environment, like FORTRAN or PASCAL, translates from a larger repertoire into the computer's native language. Every Programming Environment introduces new concepts; sometimes very general ones, and sometimes those quite specific to the languages users--e.g., the "cells" of a spreadsheet for business programming [Winograd]. It is the job of the Programming Environent to do all the bookkeeping so that the programmer can use those concepts naturally.

Expert systems have two parts. The Knowledge Base contains facts and heuristics about solving problems in some domain. It is a kind of program. The Programming Environement is called the Inference Engine and supports programming at what Allen Newell calls "the Knowledge Level." In the same way as a FORTRAN programmer writes code to be interpreted by the FORTRAN compiler (whose behavior he must "internalize"), an expert-systems programmer (called a Knowledge Engineer) interviews a problem-solver in some domain and then creates a coded Knowledge Base appropriate for interpretation by his Programming Environment. The difference between this "Fifth Generation" program and its FORTRAN precedecessor is that the Inference Engine does the necessary booking to draw inferences, explain alternatives, and justifications conclusions. A traditional program is a *procedural* description of what is to be done without explication of the knowledge underlying the alternatives.

Every technology has a natural "technological niche" (a phrase used by Egon Loebner to mark the parallels between the evolution of technology and the evolution of life forms). Expert systems, because their design is constrained not only by the demands of problem solving but also by the Transfer of Expertise, are directly involved in moving the technology of computation toward its niche. The eventual niche of computation lies in the interactions of people's knowledge--in systems that allow people to understand more together than we can understand alone. In other words, in the long run it is not important whether machines can "think" or not; it is only important that they help us think.

The Limitations of Expert Systems

Current research in AI laboratories that primarily focusses on expert systems is conceived in terms of issues like the following:

> *Learning*. How is it possible for a system to improve through its own experience, so that we needn't add knowledge from an expert one piece at a time (the "knowledge acquisition bottleneck", as it is called).
>
> *The two-expert problem*. How can the "knowledge" of two experts be used, e.g., in systems designed as decision aids for policy makers, where the experts' biases and other differences are of interest. Unfortunately, the differences between experts involve *experiences* and *values* that go beyond what easily fits into current "representations of knowledge" like rules and frames. This is the area from which my own research originated.
>
> *Teaching*: Exemplified by the work of Bill Clancey, the question is how to use the knowledge in a Expert System not only to do the original task of problem solving, but also as a resource for human learning. Clancey's work has shown that there are many things that the expert knows which, although they don't surface when he is solving problems, become central when he is teaching novices. For example, consider MYCIN, which can diagnose meningitis and bacteremia infections as

effectively as physicians who are experts in this area: MYCIN cannot begin to answer the question "What is a disease?"

Deep knowledge: This particularly important problem concerns using some model of the domain (e.g., the circuit diagrams of a computer) as a part of the knowlege base. There is no depth of knowledge in current expert systems.

Meta-knowlege: The ability of systems to know something about themselves and about what they know.

The bottom line is that expert systems as they were first developed, e.g., in MYCIN, serve a very limited range of applications: only certain types of knowledge (e.g., heuristic rules), for certain types of reasoning (e.g., diagnosis), in certain knowledge-related activities (e.g., consultation), obtainable from experienced people who are in demand by less experienced people who, nevertheless, share all the same terms, procedures, theories, goals, and values as the experts. And of course, much of the pressure for pushing out from this beachhead comes from our inability to build potentially profitable systems because they involve different types of knowledge, different knowledge-related activities, etc.

It is my belief that these limitations are not wrinkles that will be worked out as the AI research program develops. Rather, they reflect certain assumptions about the nature of Mind that underly not only Expert Systems research, but all work in AI as the field is currently understood. Working within these assumptions, while on the one hand making the dramatic scientific success possible, has at the same time created an impasse for further research.

The Origins of Artificial Intelligence

The intellectual currents of their times direct scientists to the study of certain phenomena as opposed to others. For the evolution of AI as a scientific discipline, the two most important forces in the intellectual environment of the 1930s and 1940s were *mathematical logic*, which had been under rapid development since the end of the 19th century, and new ideas about *computation*. The logical systems of Frege, Whitehead and Russell, Tarski, and others showed that some aspects of reasoning could be formalized in a relatively simple framework.

Ideas about the nature of computation, due to Church, Turing, and others, provided the link between the notion of formalization of reasoning and the computing machines about to be invented. What was essential in this work was the abstract conception of computation as *symbol processing.* Turing, who has been called the Father of AI, not only invented a universal, model of non-numerical computation, but also argued directly for the possibility that computating machines could behave in a way that would be perceived as intelligent.

What eventually connected these diverse ideas was, of course, the development of the computing machines themselves, conceived by Babbage, and guided in this century by Turing, von Neumann, and others. It was not long after the machines became available that people began to try to write programs to solve puzzles, play chess, and translate texts from one language to another--the first AI programs. What was it about computers that triggered the development of AI? The single attribute of the new machines that brought about the emergence of the new science was their inherent potential for *complexity*, encouraging the development of new and more direct ways of describing complex data structures and procedures with hundreds of steps.

As Pamela McCorduck notes in her entertaining historical study of AI, *Machines Who Think*, there has been a long-standing connection between the idea of complex mechanical devices and intelligence. Starting with the fabulously intricate clocks and mechanical automata of past centuries, people have made an intuitive link between the complexity of a machine's operation and some aspects of their own mental life. Modern computer systems are orders of magnitude more complex than anything man has built before, and AI systems are among the most complex computer programs.

The reason is that after all the levels of interpretation are peeled away, the computer performs its calculations following the step-by-step instructions it is given--the method must be specified *in complete detail!* Most computer scientists are concerned with designing new algorithms, new languages, and new machines for performing tasks like solving equations and alphabetizing lists--tasks that people perform using methods they can explicate. However, people cannot specify in detail how they decide which move to make in a chess game or how they determine that two sentences "mean the same thing."

The realization that the detailed steps of almost all intelligent human activity were unknown marked the beginning of Artificial Intelligence as a separate part of Computer Science. AI researchers investigate different types of computation and different ways of describing computation, in an effort not just to create intelligent artifacts, but also to understand what intelligence is. A basic tenet is that human intellectual capacity will be best described in the same terms as those they invent to describe their programs. However, they are just beginning to learn enough about those programs to know how to describe them scientifically--in terms of concepts that illuminate their nature and differentiate among fundamental categories. And they have started with the most obviously complex processes: deduction, parsing, search, etc.

Thus AI defined itself in terms of reasoning or problem solving as knowledgable activity in and of itself. Researchers dealt with knowledge as a static and manipulatable entitity, like statements in logic. They built their research on the implicit assumption that Mind was a process for acquiring and manipulating these objects. Within this framework, Expert Systems is the logical extreme of AI research. Expert systems are an optimal solution to the problem of reasoning with fixed, explicit goals; optimal in the sense that the goals describable to expert systems approximate the most meaningful problems people have that remain devoid of personal opinions, values, and goals.

As Newell and Simon, founders of the field of AI, point out in the Historical Epilog to their classic work *Human Problem Solving,* there were other strong intellectual currents that converged in the middle of this century in the popel who would found the science of Artificial Intelligence. The concepts of cybernetics and self-organizing systems of Wiener, McCulloch, and others deal with the macroscopic behavior of "logically simple" systems. Although the cyberneticians influenced AI, as they influenced many fields, with the broad applicability of their ideas, AI research turned away from self-organization. (There are exceptions: My own education in AI was strongly influenced by Terry Winograd and through him by the work of the cyberneticians Humberto Maturana and Gregory Bateson.)

A system is a (self-organizing) process in relation to its environment. When you remove the process from the environment, it is still a process, but it is not a system. In this sense, expert systems, and AI systems generally, are not really systems! They model the reasoning process, but conduct it outside of the context of human experience and purpose. These programs

take descriptions of problems and then solve them, whereas a Mind must cotemporaneously reduce its "goals" to descriptions of solvable problems, and learn more about the goals themselves while it tries to solve those (mythical) problems. To do this, a system must contain more than just a goal description and the appropriate "knowledge," it must have a record of it's own experience and the ability to reformulate its understanding of itself and its goals.

A Classification of Understandings

The introduction to this paper describes a science as a process of "coming to an understanding." The theory of understandings presented here tries to capture that type of process in computational terms. It is based on the fundamental notions of *understandings* and *phenomena*. An interpretive process is one which builds an understanding of the phenomena it experiences, just as a science builds an understanding of the experimental results. If that process is embedded in those phenomena, the way that the nervous system is embedded in experience, and if it seeks to maintain itself and achieve its goals, then the process is a system in the environment.

The theory concerns how understandings evolve through several qualitatively different stages: recognition of previously seen phenomena, classification (using concept formation), empirical associations (e.g., heuristic knowledge), etc. At the core of this computational theory is a "frame-like" data structure [Minsky], the UPLEX, which reflects a classification of qualitatively (vis a vis computation) different types of understandings. In particular, the UPLEX includes in the representation of an understanding the following aspects:

Experience Base: The phenomena about which this is an understanding.

Language: The features used for discriminating among phenomena, the terms used to express these cuts (defined extensionally over the experience base), and the syntax used to express statements in the Model. (See also [Wehyrauch].)

Model: The classification schemes, associations, rule-sets, theories, and good stories through which the phenomena can be made to "make sense."

Resources: The actions, methods, procedures, tests, and tricks available to the understander.

Anomalies: The caveats, dangers, and special cases known to be exceptions to the current model(s). E.g., cases that were treated according to prescribed procedures without success. (The point is that the understander may not be able to discriminate these case in its lexicon, but nevertheless remembers them.) Once again, as with all "experience" knowledge, all perceptions including "internal" ones (feelings) are included in the trace.

Multiplicity: The knowledge of other understandings and understanders. E.g., that there is another theory or another expert who has different values or another model that might be used if this one doesn't seem to be working.

History: The evolution of the UPLEX. E.g., successes, goof-ups, bug fixes, resolutions of previous anomalies and conflicts.

Hot Spots: The active tests, experiments, on-going studies, open questions, disputes, etc. which relate to the growth of this understanding. Hot spots focus attention on the types of experiences where computational

resources must be applied if the understanding is to grow.

Goal: An explicit representation of the goal(s) toward which this understanding has proven or might prove useful. Knowledge of the goal evolves with the understanding. Taking different roles or perspectives cause different understanders to build different understandings of the same phenomena. Since the goal is itself an understanding--one's understanding of what one thinks he's doing--it can be represented by a UPLEX and may be incomplete, experiential, or multiple.

Systems That Know That They Don't Understand

In summary, I believe that from the cybernetics perspective, one must view understanding as an activity of a system, and not as a property. In the words of my friend Bill Jayne, intelligence is a muddling process, not a modeling process. Furthermore, the goal of this process is not separable from the internal goals of the system, like survival. Work in AI to date has begun to develop a language for describing and differentiating such processes, including concepts like process, procedure, interpreter, bottom-up and top-down processing, objected-oriented programming, demons. And work on Transfer of Expertise has begun to focus on tools (programming environments) that interact with people in the understanding process.

I am developing a system called CONSENSYS, based on the UPLEX technology, which is tool for people who need to cooperatively solve a problem that is more important than their recognized differences. It is not a tool for reaching overall agreement or resolution of differences--only for identifying and setting aside those differences in order to cooperate for mutual benefit. In other words, CONSENSYS is a tool for coming to an understanding.

Finally, I would like to suggest some "products" that will be based on this next-generation technology. First, there will be better expert systems, more versatile in their capabilities, more understandable, more debuggable, less single-minded (multiple experts), and generally more useful. The next step, I believe, will be tools for organizational cooperation, e.g., tools for business and government planning. And finally, there will be political CONSENSYS. But of course, viewed from the perspective of Expert Systems research, the problem with politics is that everybody is an expert!

Readings

Barr, Avron, James Bennett, and William Clancey. 1979. *Transfer of expertise: A theme for AI research,* Stanford University, Heuristic Programming Project Working Paper No. HPP-79-11.

Barr, Avron, Paul R. Cohen and Edward A. Feigenbaum (Eds). 1981. *The Handbook of Artificial Intelligence,* Los Altos, California: William Kaufmann.

Barr, Avron. 1983. Artificial intelligence: Cognition as computation. In Fritz Machlup and Una Mansfield (Eds.), *The Study of Information: Interdisciplinary Messages,* New York: John Wiley, 237-262.

Bateson, Gregory. 1972. *Steps to an Ecology of Mind,* New York: Chandler.

Clancey, William. 1983. The epistemology of rule-based expert systems. In William Clancey and Edward Shortliffe (Eds.), *Readings in Medical Artificial Intelligence: The First Decade,* Reading, MA: Addison-Wesley.

Davis, Randall and Douglas Lenat (Eds.), *Knowledge-based Systems in Artificial Intelligence,* New York: McGraw Hill, 1982.

Feigenbaum, Edward A. 1977. The art of artificial intelligence, I: Themes and case studies of knowledge

engineering. *Fifth IJCAI Proceedings,* 1014-1029.

Kornfeld, W. A., and Carl Hewitt. 1981. *The scientific community metaphor,* AI Laboratory, MIT, AIM-641.

Lindsay, R., Bruce G. Buchanan, Edward A. Feigenbaum, Joshua Lederberg. 1980. *DENDRAL,* New York: McGraw Hill.

Loebner, Egon E. 1976. Subhistories of the light emitting diode. *IEEE Transactions on Electron Devices,* 23(7):675-699.

Loebner, Egon E., and H. Borden. 1969. Ecological niches for optoelectronic devices. *WESCON,* Vol. 13, Session #20, 1-8.

Marr, David. 1977. Artificial intelligence--a personal view. *Artificial Intelligence,* 9(1):1-13.

Maturana, Humberto. 1976. Biology of language: The epistemology of reality. In *Psychology and Biology of Language and Thought,* Ithaca: Cornell.

McCorduck, Pamela. 1979. *Machines Who Think,* San Francisco: Freeman.

McCulloch, Warren. 1964. The postulational foundations of experimental epistomology. *Embodiments of Mind,* Cambridge, Mass: MIT Press, 359-372.

McDermott, Drew. 1976. Artificial intelligence meets natural stupidity. *SIGART Newsletter,* 57:4-9.

Miller, Laurence. 1978. Has artificial intelligence contributed to an understanding of the human mind?: A critique of arguments for and against. *Cognitive Science,* 2(2):111-128.

Minsky, Marvin. 1975. A framework for representing knowledge. In Patrick Winston (Ed.), *The Psychology of Computer Vision,* New York: McGraw-Hill, 211-277.

Minsky, Marvin, and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry,* Cambridge, Mass.: MIT Press.

Newell, Allen. 1973. Artificial intelligence and the concept of mind. In Roger Schank and Kenneth Colby (Eds.), *Computer Models of Thought and Language,* San Francisco: Freeman, 1-60.

Newell, A. 1981. The knowledge level. *AI Magazine,* 2(2):1-20.

Newell, Allen, and Simon, Herbert A. 1972. *Human Problem Solving,* Englewood Cliffs, NJ: Prentice Hall.

Searle, J. R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences,* 3(3):411-457.

Shortliffe, Edward H. 1976. *Computer-based medical consultations: MYCIN,* New York: American Elsevier.

Simon, Herbert A. 1969. *The Sciences of the Artificial,* Cambridge, Mass.: MIT Press.

Torda, Clara. 1982. *Information Processing by the Central Nervous System and the Computer (A Comparison),* Berkeley: Walters.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind,* 59:433-460.

von Neumann, John. 1958. *The Computer and the Brain,* New Haven: Yale.

Weyhrauch, Richard W. 1978. *Prolegomena to a theory of mechanized formal reasoning.* Memo 315, AI Laboratory, Stanford University.

Wiener, Norbert. 1948. *Cybernetics: Or Control and Communication in the Animal and the Machine,* New York: Wiley.

Winograd, Terry. 1979. Beyond programming languages. *Communications of the ACM,* 22(7):391-401.

Winograd, Terry, and Fernando Flores. Forthcoming. *Understanding Computers and Cognition: A New Foundation for Design.* Norwood, NJ: Ablex.