NASA TM-89244

A Reproduced Copy

OF

NASATM-89244

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

Will Machines Ever Think?

Peter J. Denning

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report 86.12
May 16, 1986

Artificial Intelligence research has come under fire for failing to fulfill its promises. A growing number of AI researchers are reexamining the bases of AI research and are challenging the assumption that intelligent behavior can be fully explained as manipulation of symbols by algorithms. Three recent books -- *Mind over Machine* (H. Dreyfus and S. Dreyfus), *Understanding Computers and Cognition* (T. Winograd and F. Flores), and *Brains, Behavior, and Robots* (J. Albus) -- explore alternatives and open the door to new architectures that may be able to learn skills.

Work reported herein was supported in part by Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

This is a preprint of the column *The Science of Computing* for *American Scientist* 74, No 4, July-August, 1986.

Will Machines Ever Think?

Peter J. Denning

Research Institute for Advanced Computer Science

May 16, 1986

Can machines think? This question has fascinated inventors, philosophers, and logicians for centuries. In 1938, the British mathematician Alan Turing showed that a simple computational model, now called the Turing machine, was capable of universal computation: it could simulate the computation of any other Turing machine. This simulation is the basis of stored-program computers. It might seem then that stored-program computers would eventually be capable of simulating arbitrary human actions. For with information about the state of every neuron and the mechanics of neuron firing, it would seem possible to compute the next state of someone's every neuron and thus foretell that person's next action. Granted, in practice it may be impossible to acquire sufficient information for a precise prediction, but in principle it seems possible to program machines to act like people.

In 1950, Turing proposed a method of determining how close a machine might be to acting human. His method, now called the Turing test (*1*), envisages a human interrogator communicating with two entities by teletype: one a human, the other a computer. How long would it take the interrogator to determine which is which? How reliable might the determination be? Turing asserted that in fifty years' time it would be possible to make computers play the imitation game so well that 70% of interrogators would not make the right identification after five minutes of questioning. This line of reasoning underlies contemporary approaches to evaluating the skill of expert systems. The longer it takes a human expert to distinguish the actions of a program from the actions of a fellow expert, the more skillful the program is judged. This line of reasoning also leads some artificial intelligence (AI) programmers to be tolerant of programs that make mistakes, as long as the error rate is no worse than that of human experts.

Turing did well to replace the question, Can machines think? with the question, How well can a digital computer program imitate people? The terminology of the original question is too open to endless philosophical inquiry. What is a machine? What is thought? Turing offered a related, but not equivalent, question that could be resolved scientifically by experiment. An experiment to answer the original question is about as meaningful as an experiment to determine whether I have a soul.

How good is the substitute question? Will computer programs ever mimic human actions faithfully? It is important to realize that, while logic and science may help shed light on this question, it cannot be settled purely by argument or by experiment.

AI research has had many years' experience tussling with this seemingly more answerable question. In 1967, Marvin Minsky of MIT speculated that a computer with a properly organized set of one million facts about everyday life should be able to exhibit very great intelligence. Today's computers can easily store several times this amount of information and yet no one has succeeded in constructing an intelligent program. Around 1970, chess master David Levy challenged predictions that chess programs would soon play at master levels. He offered a prize to anyone who could write a computer program within ten years that would beat him; to this day, no one has constructed a program that can win more than one or two games out of ten against Levy. Around 1980, the Japanese began asserting that they would be able to construct many computer programs possessing expert levels of skill in selected domains. Will they succeed?

Intelligence has been a moving target for simulators. Most people distinguish between a complex mechanism and the thought process that created it. (Patent law makes this distinction as well.) Automatic pilots are expert fliers, but they are not pilots. Computer programs that search four moves ahead may be formidable chess players, but they are not masters. In 1966, Joseph Weizenbaum of MIT wrote a program called Eliza, a version of which simulated conversations of a psychologist (*2*). Eliza performed simple transformations on predefined keywords in the input text; in response to "I'm feeling *X*," it might type, "Why are you feeling *X*?" With Eliza, Weizenbaum demonstrated that it is possible to make convincing demonstrations of intelligence without giving the program any real understanding at all. As soon as someone finds a way of mechanizing a class of human actions, people tend to stop regarding those actions as intelligent; indeed, they often want the machines to take over the tedious routine!

It is disquieting to a growing number of AI researchers that the fundamental question of their field is so slippery. The elusiveness of answers has brought out the skeptics. David Parnas and John Shore have criticized the trial-and-error approach to programming expert systems; Hubert and Stuart Dreyfus have asserted that after 25 years AI has failed to live up to its promises, and there is no evidence that it ever will; Douglas Hofstadter says that the way many AI questions are posed is shallow. A growing number of AI researchers ask, Why is it so difficult to elicit intelligent behavior from machines? Is it poor engineering? Overly ambitious goals? Or is there something fundamental that puts the goals beyond our reach? The continuing debate may produce a new perspective on the role of machines in human society.

Three recent books shed intriguing new light on these old questions. The first, *Mind over Machine*, by Hubert and Stuart Dreyfus, analyzes the ability of machines to achieve human skills (*3*). The second, *Understanding Computers and Cognition*, by Terry Winograd and Fernando Flores, analyzes the possible relations between humans and machines in search of new foundations for computer design (*4*). The third, *Brains, Behavior, and Robots*, by James Albus, analyzes the characteristics of intelligent behavior and computing architectures required to simulate them (*5*). As you will see, the books shed light on each other as well.

The Dreyfuses set out to show that human beings have intelligence that machines cannot match. To support their argument, they analyze human skills, an important aspect of intelligence, distinguishing five levels: novice, advanced beginner, competent performer, proficient performer, and expert. The novice knows basic facts about a subject and context-independent rules for using those facts. The advanced beginner is able to recognize a few common situations and perform associated actions, and is able to take context into account. The competent performer experiences a high level of personal involvement in the subject; his behavior is strongly goal-oriented. He is able to act without conscious thought about the rules and to reason analytically about situations. The proficient performer is able to recall whole situations and apply them to the present without having to decompose them into componen··s. The expert is fully involved in the situation, ha· little awareness of the skill, and makes little conscious use of analytic reasoning; when exercising the skill, he operates in a larger world where he visualizes and manipulates whole objects and situations.

A good example that occurs to me to illustrate these points is the skill of writing. The novice depends on books detailing the rules of grammar, punctuation, and spelling; he finds writing correct sentences and paragraphs a tedious process requiring frequent reference to the rule books. The advanced beginner knows most of the major rules and can write sentences and paragraphs without having to look much up. The competent writer thinks more about the act of communicating than about the details of writing; he has learned rules of style and applies them automatically, and he is good at constructing and using outlines. The proficient writer feels completely at ease with writing; he seldom makes mistakes in grammar or style; he is able to analyze and edit his own writing; he is able to imitate the writing styles of others; he makes good use of figures of speech and other rhetorical devices; he is able to plan and execute large works. The expert writer lives in a world of ideas that he feels an urge to communicate. He is constantly formulating ways to say new things and new ways to say old things; he is able to communicate moods; he can imitate many styles of writing but he has his own style so distinctive that he can detect instantly when an editor has changed a single word of his work. The expert can formulate large works in his head, moving easily between general and detailed views.

In this hierarchy, manipulation of rules and symbols is characteristic of the lowest skill levels; the recall of highly abstract patterns is characteristic of the highest skill levels. The rules for the lowest levels can be written down unambiguously and used immediately by a novice; the skills of the highest levels are learned only through many repetitions. The Dreyfuses refer to the ability to perform at high levels as know-how, which they distinguish from the ability to manipulate symbols and rules. They argue that the stored-program computer, which is good at manipulations of rules and symbols but incapable of fast computations of whole patterns, is unlikely to pass much beyond the stage of bare competence. According to their account, entirely different forms of computation are involved in know-how, and the only existing device with some of the required properties of know-how is the hologram, where the entire image is encoded into every local region and where similarities can be determined by passing light through two holograms. Virtually nothing is known about the architecture or function of pattern computers, least of all whether they would have the human power to form images that are abstractions of other images.

The Dreyfuses conclude that stored-program computers cannot perform at high levels of skill because their internal structure is incapable of processing information in the same way human experts do. They warn that there is a lurking danger in research programs founded on the belief that today's computers can be smart: rather than waste the large investment, people will try to use expert systems anyway. Merely competent programs will be deployed where nothing less than full expert skill is required. They also say that expert systems used to teach or train may limit the progress of students to rule and symbol manipulation, whereas true expertise moves well beyond that, into know-how.

Like the Dreyfuses, Winograd and Flores are troubled by the inability of modern AI research to fulfill its promises. By analyzing the framework in which research is pursued, the nature of the questions being asked, and the approaches to these questions, they hope to open a new perspective.

According to their view, the basic questions of interest in computing are questions of design: What purposes do computers serve in the context of human practice? Design involves not just how computers operate but how they affect people. For example, a word processor looks different to different observers: to its manufacturer, it looks like a box of electronics; to its programmers, it looks like a package of software; to writers, it looks like a tool for effective communication. To all three, it is most visible when it breaks down.

Winograd and Flores find that our approach to design is strongly influenced by our tradition: the entire context of experience and ways of viewing the world, the common set of unspoken assumptions. Tradition is concealed by its gratuitousness. The Western scientific tradition -- scientific method -- holds that knowledge is objective; it values explicit theories that can be systematically used for predictions about parts of objective reality. Winograd and Flores call this the rationalistic tradition. The computer tradition, which derives from the rationalistic tradition, values information, representation, and decision-making; it also emphasizes analogies with the brain and intelligence, as well as other anthropomorphisms.

The rationalistic problem-solving process, held in high esteem by Western scientists, consists of three parts: give a precise statement of the situation in terms of objects and relationships among them; enumerate alternatives for solution; and evaluate each alternative until one of sufficiently low cost is found. Stored-program computers are ideally suited to this process: objects, relationships, and rules can be represented unambiguously as programming-language constructs; methods such as deduction and search can be used to enumerate alternatives; calculations and comparisons can be used to select a low-cost alternative. Much of current research in AI has a fundamental orientation that equates intelligence with rationalistic problem solving using heuristic procedures. If these are the only elements of human cognition, Winograd and Flores ask, then why are mindlike properties attributed to computers but not to other devices that provide or process information?

Considerable evidence is presented by Winograd and Flores to show that cognition is not based only on systematic manipulation of representations. The Dreyfuses' argument about skill levels and know-how illustrates the point. Rationalistic problem solving is at best a partial description of how people solve problems: people get thrown into situations and react. The world is constantly being interpreted by the organisms in it. The structure of a given organism plays a central role in determining the interpretations of which that organism is capable, and similar organisms evolve interpretations that have enough in common to allow communication among them. Here is where the crucial difference between a stored-program computer and a person shows up: a computer processes symbols without regard to their meaning; a person processes everything within a framework of interpretations. Stored-program computers have the wrong structure to compute like people.

Winograd and Flores call for a new orientation: let's ask not how to make computers behave more like people but how to design computers to help people do things more effectively. Some of the things computers are especially good at are executing algorithms, retrieving information, processing and filtering signals, assisting with communication and monitoring processes. Machines used in these ways can have a significant positive effect on society. Design processes that emphasize creative ways to use machines to help people are likely to succeed.

Winograd and Flores' book leaves an impression that searching for machines that think or understand is futile. The Dreyfuses are not so sure: although they feel that stored-program computers cannot achieve mindlike properties, they leave open the possibility that other architectures might.

Albus is much more confident that mindlike behavior can be elicited from machines. He is mostly interested in machines whose architecture resembles the human nervous system. (This is reminiscent of the argument of Winograd and Flores: structure determines function.) In 1950, Turing discussed two approaches to building machines that compete with people in intellectual fields. The first is to pursue very abstract activities, such as logical deduction or chess. The second is to provide the machine with sophisticated sensory organs and a mechanism for storing complex codes denoting moments of experience: by repeated exposure to sensory patterns, these machines could come to recognize familiar patterns and perform associated actions. Albus says that most AI research has followed the former path, perhaps because many of the early workers were mathematicians skilled in abstract reasoning. Virtually the entire effort in AI has been dedicated to modeling the abstract reasoning power of the thinking mind.

Albus suggests a way of following Turing's second approach. Rather than start at the top of the hierarchy of biological functions, he would start at the bottom, reproducing the control functions and behavioral patterns of insects, birds, mammals, and primates. He would use these as building blocks for the higher-level functions of the human being, ultimately coming to understand the mechanisms that give rise to intelligence and abstract thought in the human brain. As a step in this direction, he analyzes the components of the human nervous system and describes a Cerebellar Model Arithmetic Computer, on which he holds a patent, as an architecture capable of reproducing many of the nervous system's behaviors. Much of his book is devoted to showing how this computer could be taught and, after learning, how it could carry out functions of various nervous systems and react to complicated stimuli. He even shows how it could be taught to act as a stored-program computer.

I find the arguments of these writers refreshing and exciting. Rather than close the door on AI research, they reveal new doors. Some of their explorations will lead to new computers potentially capable of processing information more like human experts. Others will lead to better formulated design questions. But I doubt that the debate about whether computers can simulate thinking will ever end. Perhaps stored-program computers can't ever think, but maybe another kind of computer ....

References:

1. Douglas Hofstadter. 1985. "A Coffeehouse Conversation on Turing's Test." In *Metamagical Themas.* Basic Books.

2. Joseph Weizenbaum. 1976. *Computer Power and Human Reason.* Freeman Press.

3. Hubert Dreyfus and Stuart Dreyfus. 1986. *Mind over Machine.* Macmillan, The Free Press. Excerpts appear in "Why computers may never think like people." *Technology Review.* January.

4. Terry Winograd and Fernando Flores. 1986. *Understanding Computers and Cognition: A New Foundation for Design.* Ablex Publishing Corporation.

5. James Albus. 1981. *Brains, Behavior, and Robots.* BYTE Books of McGraw-Hill.

END

DATE

FILMED

DEC 1 1986