CHAPTER 1

DEFINING ARTIFICIAL INTELLIGENCE

1.1 BACKGROUND

Although most scientific disciplines, such as mathematics, physics, chemistry, and biology, are well defined, the field of artificial intelligence (AI) remains enigmatic. Indeed, Hofstadter (1985, p. 633) remarks, "The central problem of AI is the question: *What is the letter 'a'?* Donald Knuth, on hearing me make this claim once, appended, 'And what is the letter "i"?'—an amendment that I gladly accept." Despite nearly 50 years of research in the field, no widely accepted definition of artificial intelligence exists.

Artificial intelligence is sometimes defined as the manipulation of symbols for problem solving (e.g., Buchanan and Shortliffe, 1985, p. 3) or by tautological statements such as artificial intelligence is "the part of computer science concerned with developing intelligent computer programs" (Waterman, 1986, p. 10). Rich (1983, p. 1) offered, "Artificial intelligence (AI) is the study of how to make computers do things at which, at the moment, people are better." But this definition, if regarded statically, precludes the very existence of artificial intelligence. Once a computer program exceeds the capabilities of a human, the program is no longer in the domain of AI.

The majority of proffered definitions of artificial intelligence rely on comparisons to human behavior. Staugaard (1987, p. 23) attributes a definition to Marvin Minsky—"the science of making machines do things that would require intelligence if done by men"—and suggests that some define AI as the "mechanization, or duplication, of the human thought process." Charniak and McDermott (1985, p. 6) offer, "Artificial intelligence is the study of mental faculties through the use of computational models," while Schildt (1987, p. 11) claims, "An intelligent program is one that exhibits behavior similar to that of a human when confronted with a similar problem. It is not necessary that the program actually solve, or attempt to solve, the problem in the same way that a human would."

The question, "What is AI?" would become mere semantics if only the answers did not suggest or imply radically different avenues of research: "Some researchers simply want machines to do the various sorts of things that people call intelligent. Others hope to understand what enables people to do such things. Still other researchers want to simplify programming" (Minsky, 1991). Yet Minsky also laments, "Why can't we build, once and for all, machines that grow and improve themselves by learning from experience? Why can't we simply explain what we want, and then let our machines do experiments or read some books or go to school, the sorts of things that people do. Our machines today do no such things."

1.2 THE TURING TEST

Turing (1950) considered the question, "Can machines think?" Rather than define the terms *machines* or *think,* he proposed a test which begins with three people, a man (A), a woman (B), and an interrogator (C). The interrogator is to be separated from both A and B, say, in a closed room (Figure 1-1) but may

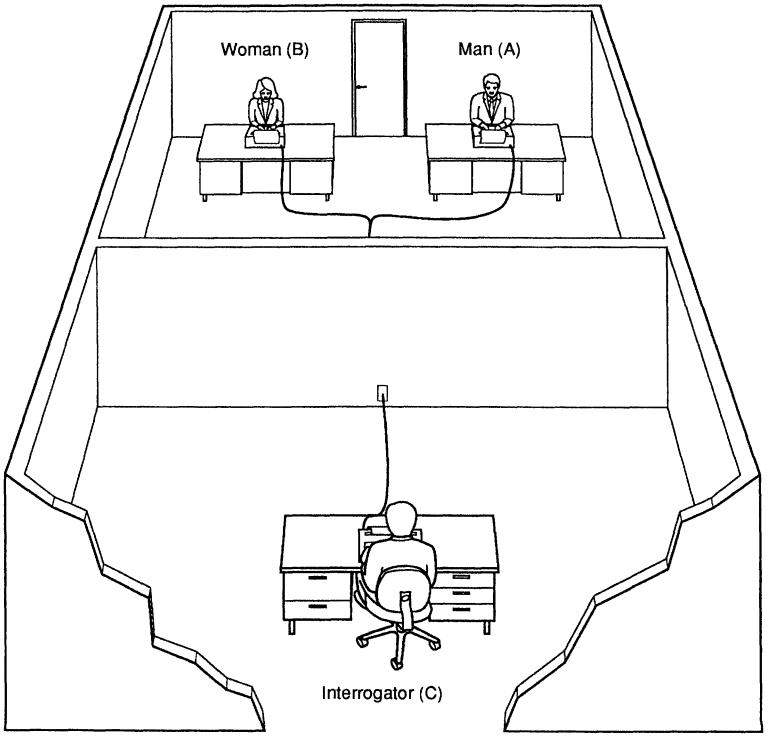

Figure 1-1 The Turing test. An interrogator (C) questions both a man (A) and a woman (B) and attempts to determine which is the woman.

ask questions of both A and B. The interrogator's objective is to determine which of A and B is the man and which is the woman. It is A's objective to cause C to make an incorrect identification. Turing (1950) provided the following example of a question posed to the man:

"C: Will X [C's name for A] please tell me the length of his or her hair?"

"A: My hair is shingled, and the longest strands are about nine inches long."

Player A may be deceitful, if he so desires. In contrast, the object for B is to help the interrogator. Turing suggested that the probable best strategy for her is to give truthful answers. In order that the pitch of the voice or other external clues may not aid in C's decision, a teleprinter was to be used for communication between the rooms.

Turing then replaced the original question, "Can machines think?" with the following: "We now ask the question, 'What will happen when a machine takes the part of A in this game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?" (Turing, 1950). This question separates the physical and intellectual capacities of humans. The form of interrogation prevents C from using sensory information regarding A's or B's physical characteristics. Presumably, if the interrogator were able to show no increased ability to decide between A and B when the machine was playing as opposed to when the man was playing, then the machine would be declared to have passed the test. Whether or not the machine should then be judged capable of thinking was left unanswered. Turing (1950) in fact dismissed the original question as being "too meaningless to deserve discussion."

Turing (1950) limited the possible machines to be the set of all digital computers. He indicated through considerable analysis that these machines are *universal,* that is, all computable processes can be executed by such machines. With respect to the suitability of the test itself, Turing (1950) thought the game might be weighted "too heavily against the machine. If the man were to try and pretend to be the machine he would clearly make a very poor showing." (Hofstadter, 1985, pp. 514–520, relates an amusing counterexample in which he was temporarily fooled in such a manner.)

Turing (1950) considered and rejected a number of objections to the plausibility of a "thinking machine," although somewhat remarkably he felt an argument supporting the existence of extrasensory perception in humans was the most compelling of all objections. The *Lady Lovelace* objection (Countess of Lovelace, 1842), referring to a memoir by the Countess of Lovelace on Babbage's Analytical Engine, is the most common of present refutations of a thinking machine. The argument asserts that a computer can only do what it is programmed to do and therefore will never be capable of generating anything new. Turing (1950) countered this argument by equating it with a statement that a machine can never take us by surprise. But

he noted that machines often act in unexpected ways because the entire determining set of initial conditions of the machine is generally unknown; therefore, an accurate prediction of all possible behavior of the mechanism is impossible.

Moreover, he suggested that a thinking machine should be a learning machine, capable of altering its own configuration through a series of rewards and punishments. Thus it could modify its own programming and generate unexpected behavior. He speculated that "in about fifty years' time it will be possible to programme computers, with a storage capacity of about 10^9 [bits], to make them play the imitation game so well that an average interrogator will not have more than a 70 per cent chance of making the right identification after five minutes of questioning" (Turing, 1950).

1.3 SIMULATION OF HUMAN EXPERTISE

The acceptance of the "Turing Test" focused attention on mimicking human behavior. At the time (1950), it was beyond any reasonable consideration that a computer could pass the Turing Test. Rather than focus on imitating human behavior in conversation, attention was turned to more limited domains of interest. Simple two-person games of strategy were selected. These games received attention for at least three reasons: (1) Their rules are static, known to both players, and easy to express in a computer program, (2) they are examples from a class of problems concerned with reasoning about actions, and (3) the ability of a game-playing computer can be measured against human experts.

The majority of research in game playing has been aimed at the development of heuristics that can be applied to two-person, zero-sum, nonrandom games of perfect information (Jackson, 1985). The term *zero-sum* indicates that any potential gain to one player will be reflected as a corresponding loss to the other player. The term *nonrandom* means that the allocation and positioning of resources in the game (e.g., pieces on a chess board) is purely deterministic. *Perfect information* indicates that both players have complete knowledge regarding the disposition of both players' resources (e.g., tic-tac-toe, not poker).

The general protocol was to examine an expert's decisions during a game so as to discover a consistent set of parameters or questions that are evaluated during his or her decision-making process. These conditions could then be formulated in an algorithm capable of generating behavior similar to that of the expert when faced with identical situations. It was believed that if a sufficient quantity or "coverage" of heuristics could be programmed into the computer, the sheer speed and infallible computational ability of the computer would enable it to match or even exceed the ability of the human expert.

1.3.1 Samuel's Checker Program

One of the earliest efforts along these lines was offered by Samuel (1959), who wrote a computer program that learned to play checkers. Checkers was chosen for several reasons: (1) There is no known algorithm that provides for a guaranteed win or draw, (2) the game is well defined with an obvious goal, (3) the rules are fixed and well known, (4) there are human experts who can be consulted and against which progress of the program can be tested, and (5) the activity is familiar to many people. The general procedure of the program was to look ahead a few moves at a time and evaluate the resulting board positions.

This evaluation was made with respect to several selected parameters. These parameters were then included in a linear polynomial with variable coefficients. The result of the polynomial indicated the worth of the prospective board under evaluation. The most critical and obvious parameter was the inability for one side or the other to move. This can occur only once in a game and was tested separately. Another clearly important consideration was the relative piece advantage. Kings were given 50 percent more weight than regular pieces. Samuel (1959) tried two alternative methods for including additional parameters. Initially, Samuel himself chose these terms, but he later allowed the program to make a subset selection from a large list of possible parameters.

To determine a move, the game tree of possible new boards was searched. A minimax procedure was used to discover the best move. The *ply,* or number of levels to be searched in the tree, was initially set at three unless the next move was a jump, the last move was a jump, or an exchange offer was possible. The analysis proceeded backward from the evaluated board position through the tree of possible moves, with the assumption that at each move the opponent would always attempt to minimize the machine's score while the machine would act to maximize its score. Under these conditions the search was continued until these circumstances were no longer encountered or until a maximum of 20 levels had been searched.

After initial experiments in which the selected polynomial had four terms (piece advantage, denial of occupancy, mobility, and a hybrid term that combined control of the center and piece advancement), the program was allowed to select a subset of 16 parameters from a list of 38 chosen parameters. Samuel allowed the computer to compete against itself; one version, Alpha, constantly modified the coefficients and parameters of its polynomial, and the other version, Beta, remained fixed (i.e., it was replaced by Alpha after a loss). A record of the correlation existing between the signs of the individual term contributions in the initial scoring polynomial and the sign of the change between the scores was maintained, along with the number of times that each particular term with a nonzero value was used. The coefficient for the polynomial term

of Alpha with the then-largest correlation coefficient was set at a prescribed maximum value with proportionate values determined for all the remaining coefficients. Samuel noted some possible instabilities with this modification technique and developed heuristic solutions to overcome these problems. Term replacement was made when a particular parameter had the lowest correlation eight times. Upon reaching this arbitrary limit, it was placed at the bottom of the reserve list and the first parameter in the reserve list was inserted into the scoring polynomial.

After a series of 28 games, Samuel described the program as being a better-than-average player. "A detailed analysis of these results indicated that the learning procedure did work and that the rate of learning was surprisingly high, but that the learning was quite erratic and none too stable" (Samuel, 1959). Refinements were made to the method of altering the scoring polynomial to help prevent this problem.

In 1962, at the request of Edward Feigenbaum and Julian Feldman, Samuel arranged for a match between his program and Robert W. Nealy, a purported former Connecticut checkers champion. Samuel's program defeated Nealy, who commented (cited in Samuel, 1963):

> Our game . . . did have its points. Up to the 31st move, all of our play had been previously published, except where I evaded "the book" several times in a vain effort to throw the computer's timing off. At the 32–27 [a specific move] loser and onwards, all the play is original with us, so far as I have been able to find. It is very interesting to me to note that computer had to make several star moves in order to get the win, and that I had several opportunities to draw otherwise. That is why I kept the game going. The machine, therefore, played a perfect ending without one misstep. In the matter of the end game, I have not had such competition from any human being since 1954, when I lost my last game.

The moves of the game appear in Samuel (1963).

In retrospect, perhaps more acclaim was given to this result than was deserved. Schaeffer (1996, p. 94) indicated that Nealy was in fact not a former Connecticut state champion at the time of the match against Samuel's program, although he did earn that title in 1966, four years later. Moreover, Nealy did not enter the U.S. Championship checkers tournament, and thus the strength of his play at the national level was based more on opinion than on record. Schaeffer (1996, pp. 94–95) reviewed the sequence of moves from the Nealy match, and with the aid of *Chinook* (a current world champion artificial intelligence checkers program designed by Schaeffer and his colleagues), indicated that Nealy made several blunders during the game and that Samuel's checkers program also did not capitalize on possible opportunities. In sum, the glowing description that Nealy gave of Samuel's program's endgame is well accepted in the literature but is an overstatement of the program's ability. Schaeffer (1996, p. 97) also reported that, in 1966, Samuel's program was

played against the two persons vying for the world championship. Four games were played against each opponent, with Samuel's program losing all eight matches.

1.3.2 Chess Programs

Researchers in artificial intelligence have also been concerned with developing chess programs. Initial considerations of making machines play chess date to Charles Babbage (1792–1871). Babbage had described the Analytical Engine, a theoretic mechanical device that was a digital computer, although not electronic. This machine was never built (but an earlier design, the Difference Engine, was in fact successfully constructed only as recently as 1991; Swade, 1993). Babbage recognized that, in principle, his Analytical Engine was capable of playing games such as checkers and chess by looking forward to possible alternative outcomes based on current potential moves.

Shannon (1950) was one of the first researchers to propose a computer program to play chess. He, like Samuel later, chose to have an evaluation function such that a program could assess the relative worth of different configurations of pieces on the board. The notion of an evaluation function has been an integral component of every chess program ever since. The suggested parameters included material advantage, pawn formation, positions of pieces, mobility, commitments, attacks, and options (cited in Levy and Newborn, 1991, pp. 27–28). Shannon noted that the best move can be found in at least two ways, although the methods may be combined: (1) Search to a given number of moves ahead and then use a minimax algorithm or (2) selectively search different branches of the game tree to different levels (i.e., moves ahead). The second method offers the advantage of preventing the machine from wasting time searching down branches in which one or more bad moves must be made. This method, later termed the *alpha-beta algorithm,* has been incorporated in many current chess playing programs.

Turing (1953) is credited with writing the first algorithm for automatic chess play. He never completed programming the procedure on a computer but was able to play at least one game by hand simulation. Turing's evaluation function included parameters of mobility, piece safety, castling, pawn position, and checks and mate threats. The one recorded game (cited in Levy and Newborn, 1991, pp. 35–38) used a search depth of two ply and then continued the search down prospective branches until "dead" positions (e.g., mate or the capture of an undefended piece) were reached. In this game, the algorithm was played against "a weak human opponent" (Levy and Newborn, 1991, p. 35) and subsequently lost. Turing attributed the weakness of the program to its "caricature of his own play" (cited in Levy and Newborn, 1991, p. 38).

The first documented working chess program was created in 1956 at Los Alamos. An unconfirmed account of a running program in the Soviet Union

was reported earlier by *Pravda* (Levy and Newborn, 1991, p. 39). Bernstein et al. (1958) described their computer program, which played a fair opening game but weak middle game because the program only searched to a depth of four ply. Newell et al. (1958) were the first to use the alpha-beta algorithm (Shannon, 1950). Greenblatt et al. (1967) are credited with creating the first program, called MACHACK VI, to beat a human in tournament play. The program was made an honorary member of the United States Chess Federation (USCF), receiving their rating of 1640. MACHACK VI used a search of at least nine ply.

In 1978, CHESS 4.7, a revised version of a program originally written by Atkin, Gorlen, and Slate of Northwestern University, defeated David Levy, Scottish chess champion, in a tournament game. Levy was "attempting to beat the program at its own game" and returned in the next match to a "no-nonsense approach," presumably to win (Levy and Newborn, 1991, pp. 98, 100). BELLE, written by Thompson and Condon, was the first program that qualified, in 1983, for the title of U.S. Master.

In the 1980s, efforts were directed at making application-specific hardware capable of searching large numbers of possible boards and quickly calculating appropriate evaluations. Berliner created HITECH, a 64-processor system. Hsu produced an even more powerful chip and its resident program, now known as DEEP THOUGHT, quickly outperformed HITECH. DEEP THOUGHT was able to search to a level of 10 ply and became the first program to defeat a world-class grand master, Bent Larsen. In 1989, DEEP THOUGHT, then rated at 2745, played a four-game match against David Levy. Levy admits, "It was the first time that [I] had ever played a program rated higher than [I] was at [my] best" (Levy and Newborn, 1991, p. 127), and correctly predicted that the machine would win 4–0. In 1990, Anatoly Karpov, the former world champion, lost a game to a MEPHISTO chess computer while giving a simultaneous exhibition against 24 opponents.

The pinnacle of beating a human world champion in match play finally was achieved in May 1997 when IBM's Deep Blue, the successor to DEEP THOUGHT, defeated Garry Kasparov, scoring two wins, one loss, and three draws. The previous year, Kasparov had defeated Deep Blue, scoring three wins, one loss, and two draws. The computer horsepower behind Deep Blue included 32 parallel processors and 512 custom chess ASICs which allowed a search of 200 million chess positions per second (Hoan, cited in Clark, 1997). Although the event received wide media attention and speculation that computers had become "smarter than humans," surprisingly little attention was given to the event in scientific literature. McCarthy (1997) offered that Deep Blue was really "a measure of our limited understanding of the principle of artificial intelligence (AI) . . . . this level of play requires many millions of times as much computing as a human chess player does." Indeed, there was no automatic learning involved in Deep Blue. A. Joseph Hoan Jr., a member of

the team that developed Deep Blue, remarked (in Clark, 1997): "we spent the whole year with chess grand master, Joel Benjamin, basically letting him beat up Deep Blue—making it make mistakes and fixing all those mistakes. That process may sound a little clunky, but we never found a good way to make automatic tuning work." Between games, adjustments were made to Deep Blue based on Kasparov's play, but these again were made by the humans who developed Deep Blue, not by the program itself.

Judging by the nearly linear improvement in the USCF rating of chess programs since the 1960s (Levy and Newborn, 1991, p. 6), the efforts of researchers to program computers to play chess must be regarded as highly successful. But there is a legitimate question as to whether or not these programs are rightly described as intelligent. Schank (1984, p. 30) commented, "The moment people succeeded in writing good chess programs, they began to wonder whether or not they had really created a piece of Artificial Intelligence. The programs played chess well because they could make complex calculations with extraordinary speed, not because they knew the kinds of things that human chess masters know about chess." Simply making machines do things that people would describe as requiring intelligence is insufficient (cf. Staugaard, 1987, p. 23). "Such programs did not embody intelligence and did not contribute to the quest for intelligent machines. A person isn't intelligent because he or she is a chess master; rather, that person is able to master the game of chess because he or she is intelligent" (Schank, 1984, p. 30).

1.3.3 Expert Systems

The focus of artificial intelligence narrowed considerably from the early 1960s through the mid-1980s (Waterman, 1986, p. 4). Initially, the desire was to create general problem-solving programs (Newell and Simon, 1963), but when preliminary attempts were unsuccessful, attention was turned to the discovery of efficient search mechanisms that could process complex data structures. The focus grew even more myopic in that research was aimed at applying these specific search algorithms (formerly termed *heuristic programming*) to very narrowly defined problems. Human experts were interrogated about their knowledge in their particular field of expertise, and this knowledge was then represented in a form that supported reasoning activities on a computer. Such an *expert system* could offer potential advantages over human expertise: It is "permanent, consistent, easy to transfer and document, and cheaper" (Waterman, 1986, p. xvii). Nor does it suffer from human frailties such as aging, sickness, or fatigue.

The programming languages often used in these applications are LISP (McCarthy et al., 1962) and Prolog (invented by Colmerauer, 1972, as cited in Covington et al., 1988, p. 2). To answer questions that are posed to the system, an *inference engine* (a program) is used to search a knowledge base.

Knowledge is most frequently represented using first-order predicate calculus, production rules, semantic networks, and frames. For example, a knowledge base might include the facts: Larry is a parent of Gary and David. This might be represented in Prolog as

1. parent(larry, gary), and
2. parent(larry, david).

(Versions of Prolog often reserve the use of capitals for variables.) If one were to be able to interrogate about whether or not a person was a child of Larry, the additional facts

1. child(gary, larry), and
2. child(david, larry),

or the rule

1. child(X, Y) :-
   parent(Y, X).

could be included (:- denotes "if"). The computer has no intrinsic understanding of the relationships "parent" or "child." It simply has codings (termed *functors*) that relate "gary," "david," and "larry."

With the existence of such a knowledge base, it becomes possible to query the system about the relationship between two people. For example, if one wanted to know whether or not "david" was the parent of "gary," one could enter

```
? - parent(david, gary).
```

The inference engine would then search the knowledge base of rules and facts and fail to validate "parent(david, gary)" and therefore would reply, "no." More general questions could be asked, such as

```
? - parent(larry, X).
```

where X is a variable. The inference engine would then search the knowledge base, attempting to match the variable to any name it could find (in this case, either "gary" or "david").

Although these examples are extremely simple, it is not difficult to imagine more complex relationships programmed in a knowledge base. The elements in the knowledge base need not be facts, but may be conjectures with degrees of confidence assigned by the human expert. The knowledge base may contain conditional statements (production rules), such as, "IF *premise* THEN *conclusion*" or "IF *condition* WITH *certainty greater than x* THEN *action*." Through the successive inclusion of broad-ranging truths to very specific knowledge about a limited domain, a versatile knowledge base and query system can be created.

DENDRAL, a chemistry program that processed mass spectral and nuclear magnetic response data to provide information regarding the molecular structure of unknown compounds, was one of the first such systems. The program was started in the mid-1960s and was subsequently refined and extended by several researchers (e.g., Feigenbaum et al., 1971; Lindsay et al., 1980). MYCIN (Shortliffe, 1974; Adams, 1976; Buchanan and Shortliffe, 1985), a program to diagnose bacterial infections in hospital patients, was an outgrowth of the "knowledge-based" DENDRAL project. Other examples of well-known knowledge-based systems can be found in Bennett and Hollander (1981), Barr and Feigenbaum (1981), and Lenat (1983).

The expert system PROSPECTOR was developed by the Stanford Research Institute to aid exploration geologists in the search for ore deposits (Duda et al., 1978). Work on the system continued until 1983 (Waterman, 1986, p. 49). Nine different mineral experts contributed to the database; it contains over 1,000 rules and a taxonomy of geological terms with more than 1,000 entries. The following sequence represents an example of PROSPECTOR receiving information from a geologist:

"1: THERE ARE DIKES
(Dike) (5)
2: THERE ARE CRETACEOUS DIORITES
(Cretaceous diorites) (5)
3: THERE IS PROBABLY SYENODIORITE
(Monzonite) (3)
4: THERE MIGHT BE SOME QUARTZ MONZONITE
(Quartz-monzonite) (2)"

(Waterman, 1986, p. 51). The values in parentheses represent the degree of certainty associated with each statement (−5 indicates complete certainty of absence while +5 indicates complete certainty of existence). The nouns in parentheses represent the internally stored name for the substance described. Through subsequent questioning of the human expert by the expert system, the program is able to offer a conjecture such as

My certainty in (PCDA) [Type-A porphyry copper deposit] is now: 1.683

followed by a detailed listing of the rules and facts that were used to come to this conclusion.

In 1980, PROSPECTOR was used to analyze a test drilling site near Mount Tolman in eastern Washington that had been partially explored. PROSPECTOR processed information regarding the geological, geophysical, and geochemical data describing the region and predicted the existence of molybdenum in a particular location (Campbell et al., 1982). "Subsequent drilling by a mining company confirmed the prediction as to where ore-grade

molybdenum mineralization would be found and where it would not be found" (Waterman, 1986, p. 58).

Waterman (1986, pp. 162–199) described the construction of an expert system and discussed some potential problems. For example, the number of rules required for a given application may grow very large. "PUFF, an expert system that interprets data from pulmonary function tests, had to have its number of rules increased from 100 to 400 just to get a 10 percent increase in performance" (Waterman, 1986, p. 182). PUFF required five person-years to construct. There are no general techniques for assessing a system's completeness or consistency. Nevertheless, despite the procedural difficulties associated with constructing expert systems, useful programs have been created to address a wide range of problems in various domains including medicine, law, agriculture, military sciences, geology, and others (Waterman, 1986, p. 205). Building expert systems has now become routine (Barr et al., 1989, p. 181).

1.3.4 A Criticism of the Expert Systems or Knowledge-Based Approach

There is some question whether or not the research in expert or knowledge-based systems truly advances the field of artificial intelligence. Dreyfus and Dreyfus (1984, 1986) claimed that when a beginner is first introduced to a new task, such as driving a car, he or she is taught specific rules to follow (e.g., maintain a two-second separation between yourself and the car in front of you). But as the beginner gains experience, less objective cues are used. "He listens to the engine as well as looks at his speedometer for cues about when to shift gears. He observes the demeanor as well as the position and speed of pedestrians to anticipate their behavior. And he learns to distinguish a distracted or drunk driver from an alert one" (Dreyfus and Dreyfus, 1984). It is difficult to believe that the now-expert driver is relying on rules in making these classifications. "Engine sounds cannot be adequately captured by words, and no list of facts about a pedestrian at a crosswalk can enable a driver to predict his behavior as well as can the experience of observing people crossing streets under a variety of conditions" (Dreyfus and Dreyfus, 1984). They asserted that when a human expert is interrogated by a "knowledge-engineer" to assimilate rules for an expert system, the expert is "forced to regress to the level of a beginner and recite rules he no longer uses . . . there is no reason to believe that a heuristically programmed computer accurately replicates human thinking" (Dreyfus and Dreyfus, 1984).

Other problems occur in the design of expert systems. Human experts, when forced to verbalize rules for their behavior, may not offer a consistent set of explanations. There may be inherent contradictions in their rules. In addition, different experts will differ on the rules that should be employed. The question of how to handle these inconsistencies remains open and is often handled in an ad hoc manner by knowledge-engineers who do not have ex-

pertise in the field of application. Simply finding an expert can be troublesome, for most often there is no objective measure of "expertness." And even when an expert is found, there is always the chance that the expert will simply be wrong. History is replete with incorrect expertise (e.g., a geocentric solar system, and see Cerf and Navasky, 1984). Moreover, expert systems often generate preprogrammed behavior. Such behavior can be *brittle,* in the sense that it is well optimized for its specific environment, but incapable of adapting to any changes in the environment.

Consider the hunting wasp, *Sphex flavipennis.* When the female wasp must lay its eggs, its builds a burrow and hunts out a cricket, which it paralyzes with three injections of venom (Gould and Gould, 1985). The wasp then drags the cricket into the burrow, lays the eggs next to the cricket, seals the burrow, and flies away. When the eggs hatch, the grubs feed on the paralyzed cricket. Initially, this behavior appears sophisticated, logical, and thoughtful (Wooldridge, 1968, p. 70). But upon further examination, limitations of the wasp's behavior can be demonstrated. Before entering the burrow with the cricket, the wasp carefully positions its paralyzed prey with its antennae just touching the opening of the burrow and "then scoots inside, 'inspects' its quarters, emerges, and drags the captive inside" (Gould and Gould, 1985). As noted by French naturalist Jean Henri Fabré, if the cricket is moved just slightly while the wasp is busy in its burrow, upon emerging the wasp will replace the cricket at the entrance and again go inside to inspect the burrow. "No matter how many times Fabré moved the cricket, and no matter how slightly, the wasp would never break out of the pattern. No amount of experience could teach it that its behavior was wrongheaded: its genetic inheritance had left it incapable of learning that lesson" (Gould and Gould, 1985).

The wasp's instinctive program is essentially a rule-based system that is crucial in propagating the species, unless the weather happens to be a bit breezy. "The insect, which astounds us, which terrifies us with its extraordinary intelligence, surprises us, the next moment, with its stupidity, when confronted with some simple fact that happens to lie outside its ordinary practice" (Fabré, cited in Gould and Gould, 1985). Genetically hard-coded behavior is inherently brittle. Similarly, an expert system chess program might do very well, but if the rules of the game were changed, even slightly, by perhaps allowing the king to move up to two squares at a time, the expert system might no longer be expert. This brittleness of domain-specific programs has been recognized for many years (Samuel, 1959).

1.3.5 Fuzzy Systems

Another procedural problem associated with the construction of a knowledge base is that when humans describe complex environments, they do not typically speak in absolutes. Linguistic descriptors of real-world circumstances are not precise but rather are "fuzzy." For example, when one describes the opti-

mum behavior of an investor interested in making money in the stock market, the adage is "buy low, sell high." But how low is "low"? And how high is "high"? It is unreasonable to suggest that if the price of the stock climbs to a certain precise value in dollars per share, then it is high; yet if it were only $0.01 lower, it would not be high. Useful descriptions need not be of a binary or crisp nature.

Zadeh (1965) introduced the notion of "fuzzy sets." Rather than describing elements as being either in a given set or not, membership in the set was viewed as a matter of degree ranging over the interval [0, 1]. A membership of 0.0 indicates that the element absolutely is not a member of the set, and a membership of 1.0 indicates that the element absolutely is a member of the set. Intermediate values indicate degrees of membership. The choice of the appropriate membership function to describe elements of a set is left to the researcher.

Negoita and Ralescu (1987, p. 79) note that descriptive phrases such as "numbers approximately equal to 10" and "young children" are not tractable by methods of classic set theory or probability theory. There is an undecidability about the membership or nonmembership in a collection of such objects, and there is nothing random about the concepts in question. A classic set can be represented precisely as a binary valued function f_A: $X \rightarrow \{0, 1\}$, the *characteristic function,* defined as

$$f_A(x) = \begin{cases} 1, \text{if } x \in A; \\ 0, \text{otherwise.} \end{cases}$$

The collection of all subsets of X (the power set of X) is denoted

$$P(X) = \{A | A \text{ is a subset of } X\}.$$

In contrast, a fuzzy subset of X is represented by a *membership function:*

$$u : X \rightarrow [0, 1].$$

The collection of all fuzzy subsets of X (the fuzzy power set) is denoted by $F(X)$:

$$F(X) = \{u | u : X \rightarrow [0, 1]\}.$$

It is natural to enquire as to the effect of operations such as union and intersection on such fuzzy sets. If u and v are fuzzy sets, then

$$(u \text{ or } v)(x) = \max[u(x), v(x)]$$
$$(u \text{ and } v)(x) = \min[u(x), v(x)].$$

Other forms of these operators have been developed (Yager, 1980; Dubois and Prade, 1982; Kandel, 1986, pp. 143–149). One form of the complement of a fuzzy set, $u : X \rightarrow [0, 1]$ is

$$u^c(x) = 1 - u(x).$$

Other properties of fuzzy set operations, such as commutativity, associativity, and distributivity, as well as other operators such as addition, multiplication, and so forth, may be found in Negoita and Ralescu (1987, pp. 81–93).

It is not difficult to imagine a *fuzzy system* that relates fuzzy sets in much the same manner as a knowledge-based system. The range of implementation and reasoning methodologies is much richer in fuzzy logic. The rules are simply fuzzy rules describing memberships in given sets rather than absolutes. Such systems have been constructed and Bezdek and Pal (1992) give a comprehensive review of efforts in fuzzy systems from 1965.

1.3.6 Perspective on Methods Employing Specific Heuristics

Human experts can only give dispositions or rules (fuzzy or precise) for problems in their domain of expertise. There is a potential difficulty when such a system is required to address problems for which there are no human experts. Schank (1984, p. 34) states, definitively, "Expert systems are horribly misnamed, since there is very little about them that is expert . . . while potentially useful, [they] are not a theoretical advance in our goal of creating an intelligent machine." He continues:

> The central problem in AI has little to do with expert systems, faster programs, or bigger memories. The ultimate AI breakthrough would be the creation of a machine that can learn or otherwise change as a result of its own experiences . . . like most AI terms, the words "expert system" are loaded with a great deal more implied intelligence than is warranted by their actual level of sophistication. . . . Expert systems are not innovative in the way the real experts are; nor can they reflect on their own decision-making processes.

This generalization may be too broad. Certainly, both expert and fuzzy systems can be made very flexible. They can generate new functional rules that were not explicitly stated in the original knowledge base. They can be programmed to ask for more information from human experts if they are unable to reach any definite (or suitably fuzzy) conclusion in the face of current information. Yet one may legitimately question whether the observed "intelligence" of the system should really be attributed to the system, or merely to the programmer who implemented knowledge into a fixed program. Philosophically, there appears to be little difference between such a hard-wired system and a simple calculator. Neither is intrinsically intelligent.

The widespread acceptance of the Turing Test has both focused and constrained research in artificial intelligence in two regards: (1) the imitation of human behavior and (2) the evaluation of artificial intelligence solely on the basis of behavioral response. But, "Ideally, the test of an effective understanding system is not the realism of the output it produces, but rather the validity of the method by which that output is produced" (Schank, 1984, p. 53).

Hofstadter (1985, p. 525) admitted to being an "unabashed pusher of the validity of the Turing Test as a way of operationally defining what it would be for a machine to genuinely think." But he also cogently wrote while playing devil's advocate: "I'm not any happier with the Turing Test as a test for thinking machines than I am with the Imitation Game [the Turing Test] as a test for femininity" (Hofstadter, 1985, p. 495). Certainly, even if a man could imitate a woman, perfectly, he would still be a man. Imitations are just that: imitations. It is important to define and program mechanisms that generate intelligent behavior. Artificial intelligence should not seek to merely solve problems, but should rather seek to solve the problem of how to solve problems.

1.4 NEURAL NETWORKS

A human may be described as an intelligent problem-solving machine. Singh (1966, p. 1) suggested that "the search for synthetic intelligence must begin with an inquiry into the origin of natural intelligence, that is, into the working of our own brain, its sole creator at present." The idea of constructing an artificial brain or *neural network* has been proposed many times (e.g., McCulloch and Pitts, 1943; Rosenblatt, 1957, 1962; Samuel, 1959; Block, 1963; and others).

The brain is an immensely complex network of neurons, synapses, axons, dendrites, and so forth (Figure 1-2), a "mammoth automatic telephone ex-

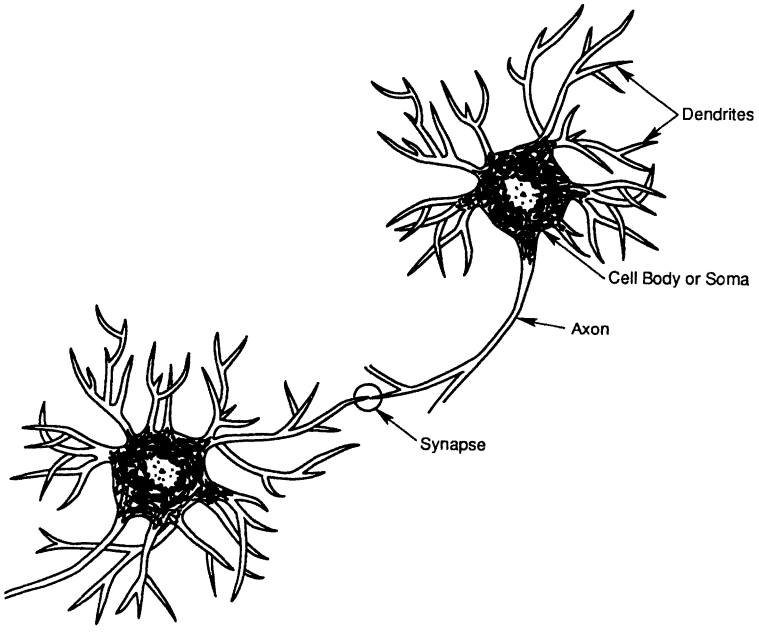

Figure 1-2 The basic structure of a neuron.

change" (Singh, 1966, p. 129). Through detailed modeling of these elements, a simulated network that is capable of diverse behaviors may be constructed. The human brain comprises at least 2×10^{10} neurons, each possessing about 10,000 synapses distributed over each dendritic tree with an average number of synapses on the axon of one neuron again being about 10,000 (Block, 1963; Palm, 1982, p. 10). Modeling the precise structure of this connection scheme would appear beyond the capabilities of foreseeable methods. Fortunately, this may not be necessary.

Rather than deduce specific replications of the human brain, models may be employed. Among the first such artificial neural network designs was the *perceptron* (Rosenblatt, 1957, 1958, 1960, 1962). A perceptron (Figure 1-3) consists of three types of units: sensory units, associator units, and response units. A stimulus will activate some sensory units. These sensory units in turn activate, with varying time delays and connection strengths, the associator units. These activations may be positive (excitatory) or negative (inhibitory). If the weighted sum of the activations at an associator unit exceeds a given threshold, the associator unit activates and sends a pulse, again weighted by a connection strength, onto the response units. There is obvious analogous behavior of units and neurons, of connections and axons and dendrites. The characteristics of the stimulus-response (input-output) of the perceptron describe its behavior.

Earlier work by Hebb (1949) indicated that neural networks could learn to recognize patterns by weakening and strengthening the connections between neurons. Rosenblatt (1957, 1960, 1962) and others (e.g., Keller, 1961; Kesler, 1961; Block, 1962; Block et al., 1962) studied the effects of changing the connection strengths in a perceptron by various rules (Rumelhart and McClelland, 1986, p. 155). Block (1962) indicated that when the perceptron was employed on some simple pattern recognition problems, the behavior of the machine degraded gradually with the removal of association units. That is, the perceptrons

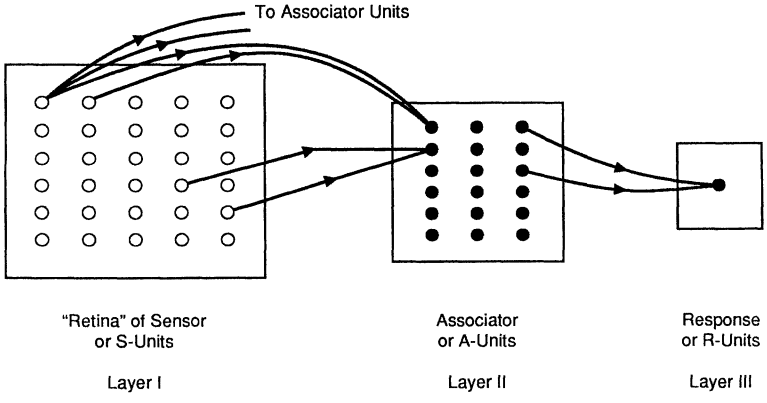

Figure 1-3 Rosenblatt's perceptron model.

were robust, not brittle. Rosenblatt (1962, p. 28) admitted that his perceptrons were "extreme simplifications of the central nervous system, in which some properties are exaggerated and others suppressed." But he also noted that the strength of the perceptron approach lay in the ability to analyze the model.

Minsky and Papert (1969) studied the computational limits of perceptrons with one layer of modifiable connections. They demonstrated that such processing units were not able to calculate mathematical functions such as parity or the topological function of connectedness without using an absurdly large number of predicates (Rumelhart and McClelland, 1986, p. 111). But these limitations do not apply to networks of perceptrons that consist of multiple layers of perceptrons, nor does their analysis address networks with recurrent feedback connections (rather than simple feedforward connections, although any perceptron with feedback can be approximated by an equivalent but larger feedforward network). Nevertheless, Minsky and Papert (1969, pp. 231–232) speculated that the study of multilayered perceptrons would be "sterile" in the absence of an algorithm to usefully adjust the connections of such architectures.

Minsky and Papert (1969, p. 4) offered, "We have agreed to use the name 'perceptron' in recognition of the pioneer work of Frank Rosenblatt," but Block (1970) noted that "they study a severely limited class of machines from a viewpoint quite alien to Rosenblatt's." While Block (1970) recognized the mathematical prowess of Minsky and Papert, he also replied, "Work on the four-layer *Perceptrons* has been difficult, but the results suggest that such systems may be rich in behavioral possibilities." Block (1970) admitted the inability of simple perceptrons to perform functions such as parity checking and connectedness, but remarked, "Human beings cannot perceive the parity of large sets . . . nor connectedness." The recognition of more common objects such as faces was viewed as a more appropriate test.

Block questioned prophetically, "Will the formulations or methods developed in the book have a serious influence on future research in pattern recognition, threshold logic, psychology, or biology; or will this book prove to be only a monument to the mathematical virtuosity of Minsky and Papert? We shall have to wait for a few years to find out" (Block, 1970).

Minsky and Papert's speculation that efforts with multilayered perceptrons would be sterile served in part to restrict funding and thus research efforts in neural networks during the 1970s and early 1980s. The criticisms by Minsky and Papert (1969) were passionate and persuasive. Papert (1988) admitted, "Yes, there was *some* hostility in the energy behind the research reported in *Perceptrons,* and there is *some* degree of annoyance at the way the new [resurgence in neural network research] has developed; part of our drive came, as we quite plainly acknowledged in our book, from the fact that funding and research energy were being dissipated on what still appear to me . . . to be misleading attempts to use connectionist methods in practical applications." Subsequent to Minsky and Papert (1969), neural network research was continued by Grossberg (1976, 1982), Amari (1967, 1971, 1972, and many oth-

ers), Kohonen (1984), and others, but to a lesser degree than that conducted on knowledge-based systems. A resurgence of interest grew in the mid-1980s following further research by Hopfield (1982), Hopfield and Tank (1985), Rumelhart and McClelland (1986), Mead (1989), and others (Simpson, 1990, pp. 136–145, provides a concise review of efforts in neural networks; also, see Hecht-Nielsen, 1990; Haykin, 1994).

It is now well known that multiple layers of perceptrons with variable connection strengths, bias terms, and nonlinear sigmoid functions can approximate arbitrary measurable mapping functions. In fact, universal function approximators can be constructed with a single hidden layer of squashing units and an output layer of linear units (Cybenko, 1989; Hornik et al., 1989; Barron, 1993). The application of such structures to pattern recognition problems is now routine. But as noted, such structures are simply mapping functions. Functions are not intrinsically intelligent. The crucial problem then becomes training such functions to yield the appropriate stimulus-response. There are several proposed methods to discover a suitable set of weights and bias terms, given an overall topological structure and a set of previously classified patterns (e.g., Werbos, 1974; Hopfield, 1982; Rumelhart and McClelland, 1986; Arbib and Hanson, 1990; Levine, 1991).

Currently, the method most often employed (i.e., back propagation) is based on a simple gradient descent search of the error response surface determined by the set of weights and biases. But this method is likely to discover suboptimal solutions because the response surface is a general nonlinear function and may possess many local optima. Moreover, a gradient descent algorithm is inherently no more intelligent than any other deterministic algorithm. Conceptually, the back propagation routine is little different from the addition of two integers.

There does not appear to be any evidence to suggest that biologic neural networks alter connection strengths via methods similar to gradient-based search. If the bottom-up approach to neural networks is to lead to artificially intelligent machines, the intelligence must come in the search for the appropriate structure and parameters of the particular mapping function. "Learning models which cannot adaptively cope with unpredictable changes in a complex environment have an unpromising future as models of mind and brain" (Grossberg, 1987). Models of self-organizing networks (e.g., Grossberg, 1987, and others) appear more broadly useful as explanations of biologic nervous systems.

1.5 DEFINITION OF INTELLIGENCE

If one word were to be used to describe research in artificial intelligence, that word might be *fragmented*. Opinions as to the cause of this scattered effort are varied. Atmar (1976) remarked:

> Perhaps the major problem is our viewpoint. Intelligence is generally regarded as a uniquely human quality. And yet we, as humans, do not understand ourselves, our capabilities, or our origins of thought. In our rush to catalogue and emulate our own staggering array of behavioral responses, it is only logical to suspect that investigations into the primal causative factors of intelligence have been passed over in order to more rapidly obtain the immediate consequences of intelligence.

Minsky (1991), on the other hand, assigns blame to attempts at unifying theories of intelligence: "There is no one best way to represent knowledge or to solve problems, and the limitations of current machine intelligence largely stem from seeking unified theories or trying to repair the deficiencies of theoretically neat but conceptually impoverished ideological positions."

A prerequisite to embarking upon research in artificial intelligence should be a definition of the term *intelligence*. As noted above (Atmar, 1976), many definitions of intelligence have relied on this property being uniquely human (e.g., Singh, 1966, p. 1) and often reflect a highly anthropocentric view. Schank (1984, p. 49) stated:

> No question in AI has been the subject of more intense debate than that of assessing machine intelligence and understanding. People unconsciously feel that calling something other than humans intelligent denigrates humans and reduces their vision of themselves as the center of the universe. Dolphins are intelligent. Whales are intelligent. Apes are intelligent. Even dogs and cats are intelligent.

Certainly other living systems can be described as being intelligent, without ascribing specific intelligence to any individual member of the system. Any proposed definition of intelligence should not rely on comparisons to individual organisms.

In contrast to Staugaard's comments (1987, p. 23), Minsky has offered the following definition of intelligence (1985, p. 71): "Intelligence . . . means . . . the ability to solve hard problems." But how hard does a problem have to be? Who is to decide which problem is hard? All problems are hard until you know how to solve them, at which point they become easy. Finding the slope of a polynomial at any specific point is very difficult, unless you are familiar with derivatives, in which case it is trivial. Such a definition would appear problematic.

For an organism, or any system, to be intelligent, it must make decisions. Any decision may be described as the selection of how to allocate the available resources. And an intelligent system must face a range of decisions, for if there were only one possible decision, there would really be no decision at all. Moreover, decision making requires a goal. Without the existence of a goal, decision making is pointless. The intelligence of such a decision-making entity becomes a meaningless quality.

This argument begs the question, "Where do goals come from?" Consider biologically reproducing organisms. They exist in a finite arena; as a conse-

quence, there is competition for the available resources. Natural selection is inevitable in any system of self-replicating organisms that fill the available resource space. Selection stochastically eliminates those variants that do not acquire sufficient resources. Thus, while evolution as a process is purposeless, the first purposeful goal imbued into all living systems is survival. Those variants that do not exhibit behaviors that meet this goal are stochastically culled. The genetically preprogrammed behaviors of the survivors (and thus the goal of survival) are reinforced in every generation through intense competition.

Such a notion has been suggested many times. For example, Carne (1965, p. 3) remarked, "Perhaps the basic attribute of an intelligent organism is its capability to learn to perform various functions within a changing environment so as to survive and to prosper." Atmar (1976) offered, "Intelligence is that property in which the organism senses, reacts to, learns from, and subsequently adapts its behavior to its present environment in order to better promote its own survival."

Note that any automaton whose behavior (i.e., stimulus-response pairs that depend on the state of the organism) is completely prewired (e.g., a simple hand-held calculator or the hunting wasp described previously) cannot learn anything. Nor can it make decisions. Such systems should not be viewed as intelligent. But this should not be taken as a contradiction to the statement that "the genetically preprogrammed behaviors of the survivors (and thus the goal of survival) are passed along to future progeny." Behaviors in all biota, individuals or populations, are dependent on underlying genetic programs. In some cases, these programs mandate specific behaviors; in others, they create nervous systems capable of adapting behavior of the organism based on its experiences.

But the definition of intelligence should not be restricted to biological organisms. Intelligence is a property of purpose-driven decision makers. It applies equally well to humans, colonies of ants, robots, social groups, and so forth. Thus, more generally, following Fogel (1964; Fogel et al., 1966, p. 2), intelligence may be defined as *the capability of a system to adapt its behavior to meet its goals in a range of environments*. For species, survival is a necessary goal in any given environment; for a machine, both goals and environments may be imbued by the machine's creators.

1.6 INTELLIGENCE, THE SCIENTIFIC METHOD, AND EVOLUTION

Ornstein (1965) argued that all learning processes are adaptive. The most important aspect of such learning processes is the "development of implicit or explicit techniques to accurately estimate the probabilities of future events." Similar notions have been offered by Atmar (1976, 1979). When faced with a

changing environment, the adaptation of behavior becomes little more than a shot in the dark if the system is incapable of predicting future events. Ornstein (1965) suggested that as predicting future events is the "forte of science," it is sensible to examine the scientific method for useful cues in the search for effective learning techniques.

The scientific method (Figure 1-4) is an iterative process that facilitates the gaining of new knowledge about the underlying processes of an observable environment. Unknown aspects of the environment are estimated. Data are collected in the form of previous observations or known results and combined with newly acquired measurements. After the removal of known erroneous data, a class of models of the environment that is consistent with the data is generalized. This process is necessarily inductive. The class of models is then generally reduced by parametrization, a deductive process. The specific hypothesized model (or models) is then tested in its ability to predict future aspects of the environment. Models that prove worthy are modified, extended, or combined to form new hypotheses that carry on a "heredity of reasonableness" (Fogel, 1964). This process is iterated until a sufficient level of credibility is achieved. "As the hypotheses correspond more and more closely with the logic of the environment they provide an 'understanding' that is demonstrated in terms of improved goal-seeking behavior in the face of that environment" (Fogel et al., 1966, p. 111). It appears reasonable to seek methods to mechanize the scientific method in an algorithmic formulation so that a machine may carry out the procedure and similarly gain knowledge about its environment and adapt its behavior to meet goals.

The scientific method can be used to describe a process of human investigation of the universe, or of learning processes in general. Atmar (1976), following Weiner (1961), proposed that there are "three distinct organization

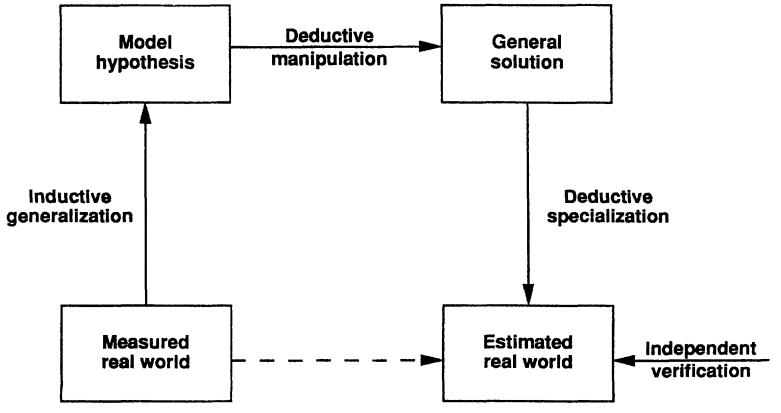

Figure 1-4 The scientific method.

forms of intelligence: phylogenetic, ontogenetic and sociogenetic, which are equivalent to one another in process, each containing a quantum unit of mutability and a reservoir of learned behavior." Individuals of most species are capable of learning ontogenetically (self-arising within the individual). The minimum unit of mutability is the proclivity of a neuron to fire. The reservoir of learned behavior becomes the entire "collection of engrams reflecting the sum of knowledge the organism possesses about its environment" (Atmar, 1976). Sociogenetic learning (arising within the group) is the basis for a society to acquire knowledge and communicate (Wilson, 1971; Atmar, 1976). The quantum unit of mutability is the "idea," while "culture" is the reservoir of learned behavior.

But phylogenetic learning (arising from within the lineage) is certainly the most ancient, and the most commonly exhibited, form of intelligence. The quantum unit of mutability is the nucleotide base pair, and the reservoir of learned behavior is the genome of the species. The recognition of evolution as an intelligent learning process is a recurring idea (Cannon, 1932; Turing, 1950; Fogel, 1962; and others). Fogel et al. (1966, p. 112) developed a correspondence between natural evolution and the scientific method. In nature, individual organisms serve as hypotheses concerning the logical properties of their environment. Their behavior is an inductive inference concerning some as yet unknown aspects of that environment. Validity is demonstrated by their survival. Over successive generations, organisms become successively better predictors of their surroundings.

Minsky (1985, p. 71) disagreed, claiming that evolution is not intelligent "because people also use the word 'intelligence' to emphasize swiftness and efficiency. Evolution's time rate is so slow that we don't see it as intelligent, even though it finally produces wonderful things we ourselves cannot yet make." But time does not enter into the definition of intelligence offered above, and it need not. Atmar (1976) admits that "the learning period [for evolution] may be tortuously long by human standards, but it is real, finite, and continuous." And evolution can proceed quite rapidly (e.g., viruses). Units of time are a human creation. The simulation of the evolutionary process on a computer need not take billions of years. Successive generations can be interated very quickly. Metaphorically, the videotape of evolution can be played at fast forward with no alteration of the basic algorithm. Arguments attacking the speed of the process (as opposed to the rate of learning) are without merit.

There is obvious value in plasticity. Organisms that can adapt to changes in the environment at a greater rate than through direct physical modifications will tend to outcompete less mutable organisms. Thus the evolutionary benefit of ontogenetic learning is obvious. Sociogenetic learning is even more powerful as the communicative group possesses even greater plasticity in behavior, a more durable memory, and a greater range of possible mutability (Atmar, 1976). But both ontogenetic learning and sociogenetic learning are

"tricks" of phylogenetic learning, invented through the randomly driven search of alternative methods of minimizing behavioral surprise to the evolving species.

If the evolutionary process is accepted as being fundamentally analogous to the scientific method, then so must the belief that this process can be mechanized and programmed on a computing machine. Evolution, like all other natural processes, is a mechanical procedure operating on and within the laws of physics and chemistry (Fogel, 1964; Fogel et al., 1966; Wooldridge, 1968, p. 25; Atmar, 1976, 1979, 1991; Mayr, 1988, pp. 148–159). If the scientific method is captured in an algorithm, then so must induction be captured as an intrinsic part of that algorithm.

Fogel et al. (1966, p. 122) noted that induction had been presumed to require creativity and imagination, but through the simulation of evolution, induction can be reduced to a routine procedure. Such notions have a propensity to generate pointed responses. Lindsay (1968) wrote in criticism of Fogel et al. (1966), "The penultimate indignity is a chapter in which the scientific method is described as an evolutionary process and hence mechanizable with the procedures described." People unconsciously feel that calling something not human "intelligent" denigrates humans (Schank, 1984, p. 49; cf. Pelletier, 1978, pp. 240–241). Yet wishing something away does not make it so. Wooldridge (1968, p. 129) stated:

> In particular, it must not be imagined that reduction of the processes of intelligence to small-step mechanical operations is incompatible with the apparently spontaneous appearance of new and original ideas to which we apply such terms as "inspiration," "insight," or "creativity." To be sure, there is no way for the physical methods . . . to produce full-blown thoughts or ideas from out of the blue. But it will be recalled that there is a solution for this problem. The solution is to deny that such spontaneity really exists. The argument is that this is an example of our being led astray by attributing too much reality to our subjective feelings—that the explanation of the apparent freedom of thought is the incompleteness of our consciousness and our resulting lack of awareness of the tortuous . . . nature of our thought processes.

Hofstadter (1985, p. 529) went further:

> Having creativity is an automatic consequence of having the proper representation of concepts in a mind. It is not something you add on afterward. It is built into the way concepts are. To spell this out more concretely: If you have succeeded in making an accurate model of *concepts*, you have thereby also succeeded in making a model of the creative process, and even of consciousness.

Creativity and imagination are part of the invention of evolution just as are eyes, opposable thumbs, telephones, and calculators.

The process of evolution can be described as four essential processes: self-reproduction, mutation, competition, and selection (Mayr, 1988; Hoffman,

1989; and many others). The self-reproduction of germline DNA and RNA systems is well known. In a positively entropic universe (as dictated by the second law of thermodynamics), the property of mutability is guaranteed; error in information transcription is inevitable. A finite arena guarantees the existence of competition. Selection becomes the natural consequence of the excess of organisms that have filled the available resource space (Atmar, 1979). The implication of these very simple rules is that evolution is a procedure that can be simulated and used to generate creativity and imagination mechanically.

1.7 EVOLVING ARTIFICIAL INTELLIGENCE

It is natural to conclude that by simulating the evolutionary learning process on a computer, the machine can become intelligent, that it can adapt its behavior to meet goals in a range of environments. "Intelligence is a basic property of life" (Atmar, 1976). It has occurred at the earliest instance of natural selection and has pervaded all subsequent living organisms. In many ways, life is intelligence, and the processes cannot be easily separated.

Numerous opinions about the proper goal for artificial intelligence research have been expressed. But intuitively, intelligence must be the same process in living organisms as it is in machines. The "artificial" is not nearly as important as the "intelligence." "Artificial Intelligence is the study of intelligent behavior. Its ultimate goal is a theory of intelligence that accounts for the behavior of naturally occurring intelligent entities and that guides the creation of artificial entities capable of intelligent behavior" (Genesereth and Nilsson, 1987). Evolutionary processes account for such intelligent behavior and can be simulated and used for the creation of intelligent machines.

REFERENCES

Adams, J. B. (1976). "A Probability Model of Medical Reasoning and the MYCIN Model," *Mathematical Biosciences,* Vol. 32, pp. 177–186.

Amari, S.-I. (1967). "A Theory of Adaptive Pattern Classifiers," *IEEE Trans. of Elec. Comp.,* Vol. EC-16, pp. 299–307.

Amari, S.-I. (1971). "Characteristics of Randomly Connected Threshold-Element Networks and Network Systems," *Proc. of the IEEE,* Vol. 59:1, pp. 35–47.

Amari, S.-I. (1972). "Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements," *IEEE Trans. Comp.,* Vol. C-21, pp. 1197–1206.

Arbib, M. A., and A. R. Hanson (1990). *Vision, Brain, and Cooperative Computation.* Cambridge, MA: MIT Press.

Atmar, J. W. (1976). "Speculation on the Evolution of Intelligence and Its Possible Realization in Machine Form." Sc.D. diss., New Mexico State University, Las Cruces.

Atmar, J. W. (1979). "The Inevitability of Evolutionary Invention." Unpublished manuscript.

Atmar, W. (1991). "On the Role of Males," *Animal Behaviour,* Vol. 41, pp. 195–205.

Barr, A., and E. A. Feigenbaum (1981). *The Handbook of Artificial Intelligence,* Vol. 1. San Mateo, CA: William Kaufmann.

Barr, A., P. R. Cohen, and E. A. Feigenbaum (1989). *The Handbook of Artificial Intelligence,* Vol. 4. Reading, MA: Addison-Wesley.

Barron, A. R. (1993). "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Trans. Info. Theory,* Vol. 39:3, pp. 930–945.

Bennett, J. S., and C. R. Hollander (1981). "DART: An Expert System for Computer Fault Diagnosis," *Proc. of IJCAI-81*, pp. 843–845.

Bernstein, A., V. De, M. Roberts, T. Arbuckle, and M. A. Belsky (1958). "A Chess Playing Program for the IBM 704," *Proc. West Jut. Comp. Conf.,* Vol. 13, pp. 157–159.

Bezdek, J. C., and S. K. Pal (1992). *Fuzzy Models for Pattern Recognition: Models that Search for Structures in Data.* Piscataway, NJ: IEEE Press.

Block, H. D. (1962). "The Perceptron: A Model for Brain Functioning," *Rev. Mod. Phys.,* Vol. 34, pp. 123–125.

Block, H. D. (1963). "Adaptive Neural Networks as Brain Models," *Proc. of Symp. Applied Mathematics,* Vol. 15, pp. 59–72.

Block, H. D. (1970). "A Review of 'Perceptrons: An Introduction to Computational Geometry,' " *Information and Control,* Vol. 17:5, pp. 501–522.

Block, H. D., B. W. Knight, and F. Rosenblatt (1962). "Analysis of a Four Layer Series Coupled Perceptron," *Rev. Mod. Phys.,* Vol. 34, pp. 135–142.

Buchanan, B. G., and E. H. Shortliffe (1985). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley.

Campbell, A. N., V. F. Hollister, R. O. Duda, and P. E. Hart (1982). "Recognition of a Hidden Mineral Deposit by an Artificial Intelligence Program," *Science,* Vol. 217, pp. 927–929.

Cannon, W. D. (1932). *The Wisdom of the Body*. New York: Norton and Company.

Carne, E. B. (1965). *Artificial Intelligence Techniques*. Washington, DC: Spartan Books.

Cerf, C., and V. Navasky (1984). *The Experts Speak: The Definitive Compendium of Authoritative Misinformation.* New York: Pantheon Books.

Charniak, E., and D. V. McDermott (1985). *Introduction to Artificial Intelligence.* Reading, MA: Addison-Wesley.

Clark, D. (1997). "Deep Thoughts on Deep Blue," *IEEE Expert,* Vol. 12:4, p. 31.

Countess of Lovelace (1842). "Translator's Notes to an Article on Babbage's Analytical Engine." In *Scientific Memoirs,* Vol. 3, edited by R. Taylor, pp. 691–731.

Covington, M. A., D. Nute, and A. Vellino (1988). *Prolog Programming in Depth.* Glenview, IL: Scott, Foresman.

Cybenko, G. (1989). "Approximations by Superpositions of a Sigmoidal Function," *Math. Contr. Signals, Syst.,* Vol. 2, pp. 303–314.

Dreyfus, H., and S. Dreyfus (1984). "Mindless Machines: Computers Don't Think Like Experts, and Never Will," *The Sciences,* November/December, pp. 18–22.

Dreyfus, H., and S. Dreyfus (1986). "Why Computers May Never Think Like People," *Tech. Review,* January, pp. 42–61.

Dubois, D., and H. Prade (1982). "A Class of Fuzzy Measures Based on Triangular Norms—A General Framework for the Combination of Uncertain Information," *Int. J. of General Systems,* Vol. 8:1.

Duda, R., P. E. Hart, N. J. Nilsson, P. Barrett, J. G. Gaschnig, K. Konolige, R. Reboh, and J. Slocum (1978). "Development of the PROSPECTOR Consultation System for Mineral Exploration," SRI Report, Stanford Research Institute, Menlo Park, CA.

Feigenbaum, E. A., B. G. Buchanan, and J. Lederberg (1971). "On Generality and Problem Solving: A Case Study Involving the DENDRAL Program." In *Machine Intelligence 6,* edited by B. Meltzer and D. Michie. New York: American Elsevier, pp. 165–190.

Fogel, L. J. (1962). "Autonomous Automata," *Industrial Research,* Vol. 4, pp. 14–19.

Fogel, L. J. (1964). "On the Organization of Intellect." Ph.D. diss., UCLA.

Fogel, L. J., A. J. Owens, and M. J. Walsh (1966). *Artificial Intelligence through Simulated Evolution.* New York: John Wiley.

Genesereth, M. R., and N. J. Nilsson (1987). *Logical Foundations of Artificial Intelligence.* Los Altos, CA: Morgan Kaufmann.

Gould, J. L., and C. G. Gould (1985). "An Animal's Education: How Comes the Mind to be Furnished?" *The Sciences,* Vol. 25:4, pp. 24–31.

Greenblatt, R., D. Eastlake, and S. Crocker (1967). "The Greenblatt Chess Program," *FJCC,* Vol. 31, pp. 801–810.

Grossberg, S. (1976). "Adaptive Pattern Classification and Universal Recoding: Part I. Parallel Development and Coding of Neural Feature Detectors," *Biological Cybernetics,* Vol. 23, pp. 121–134.

Grossberg, S. (1982). *Studies of Mind and Brain.* Dordrecht, Holland: Reidel.

Grossberg, S. (1987). "Competitive Learning: From Interactive Activation to Adaptive Resonance," *Cognitive Science,* Vol. 11, pp. 23–63.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation.* New York: Macmillan.

Hebb, D. O. (1949). *The Organization of Behavior.* New York: John Wiley.

Hecht-Nielsen, R. (1990). *Neurocomputing.* Reading, MA: Addison-Wesley.

Hoffman, A. (1989). *Arguments on Evolution: A Paleontologist's Perspective.* New York: Oxford Univ. Press.

Hofstadter, D. R. (1985). *Metamagical Themas: Questing for the Essence of Mind and Pattern.* New York: Basic Books.

Hopfield, J. J. (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. Nat. Acad. of Sciences,* Vol. 79, pp. 2554–2558.

Hopfield, J. J., and D. Tank (1985). " 'Neural' Computation of Decision in Optimization Problems," *Biological Cybernetics,* Vol. 52, pp. 141–152.

Hornik, K., M. Stinchcombe, and H. White (1989). "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks,* Vol. 2, pp. 359–366.

Jackson, P. C. (1985). *Introduction to Artificial Intelligence,* 2nd ed. New York: Dover.

Kandel, A. (1986). *Fuzzy Expert Systems*. Boca Raton, FL: CRC Press.

Keller, H. B. (1961). "Finite Automata, Pattern Recognition and Perceptrons," *J. Assoc. Comput. Mach.,* Vol. 8, pp. 1–20.

Kesler, C. (1961). "Preliminary Experiments on Perceptron Applications to Bubble Chamber Event Recognition." Cognitive Systems Research Program, Rep. No. 1, Cornell University, Ithaca, NY.

Kohonen, T. (1984). *Self-Organization and Associative Memory*. Berlin: Springer-Verlag.

Lenat, D. B. (1983). "EURISKO: A Program that Learns New Heuristics and Domain Concepts," *Artificial Intelligence,* Vol. 21, pp. 61–98.

Levine, D. S. (1991). *Introduction to Neural and Cognitive Modeling*. Hillsdale, NJ: Lawrence Erlbaum.

Levy, D. N. L., and M. Newborn (1991). *How Computers Play Chess*. New York: Computer Science Press.

Lindsay, R. K. (1968). "Artificial Evolution of Intelligence," *Contemp. Psych.,* Vol. 13:3, pp. 113–116.

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

Mayr, E. (1988). *Toward a New Philosophy of Biology: Observations of an Evolutionist*. Cambridge, MA: Belknap Press.

McCarthy, J., P. J. Abrahams, D. J. Edwards, P. T. Hart, and M. I. Levin (1962). *LISP 1.5 Programmer's Manual*. Cambridge, MA: MIT Press.

McCarthy, J. (1997). "AI as Sport," *Science,* Vol. 276, pp. 1518–1519.

McCulloch, W. S., and W. Pitts (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bull. Math. Biophysics,* Vol. 5, pp. 115–133.

Mead, C. (1989). *Analog VLSI and Neural Systems.* Reading, MA: Addison-Wesley.

Minsky, M. L. (1985). *The Society of Mind.* New York: Simon and Schuster.

Minsky, M. L. (1991). "Logical Versus Analogical or Symbolic versus Connectionist or Neat versus Scruffy," *AI Magazine,* Vol. 12:2, pp. 35–51.

Minsky, M. L., and S. Papert (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Negoita, C. V., and D. Ralescu (1987). *Simulation, Knowledge-Based Computing, and Fuzzy Statistics*. New York: Van Nostrand Reinhold.

Newell, A., J. C. Shaw, and H. A. Simon (1958). "Chess Playing Programs and the Problem of Complexity," *IBM J. of Res. & Dev.,* Vol. 4:2, pp. 320–325.

Newell, A., and H. A. Simon (1963). "GPS: A Program that Simulates Human Thought." In *Computers and Thought,* edited by E. A. Feigenbaum and J. Feldman. New York: McGraw-Hill, pp. 279–293.

Ornstein, L. (1965). "Computer Learning and the Scientific Method: A Proposed

Solution to the Information Theoretical Problem of Meaning," *J. of the Mt. Sinai Hosp.,* Vol. 32:4, pp. 437–494.

Palm, G. (1982). *Neural Assemblies: An Alternative Approach to Artificial Intelligence.* Berlin: Springer-Verlag.

Papert, S. (1988). "One AI or Many?" In *The Artificial Intelligence Debate: False Starts, Real Foundations,* edited by S. R. Braubard. Cambridge, MA: MIT Press.

Pelletier, K. R. (1978). *Toward a Science of Consciousness.* New York: Dell.

Rich, E. (1983). *Artificial Intelligence.* New York: McGraw-Hill.

Rosenblatt, F. (1957). "The Perceptron, a Perceiving and Recognizing Automaton." Project PARA, Cornell Aeronautical Lab. Rep., No. 85-640–1, Buffalo, NY.

Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychol. Rev.,* Vol. 65, p. 386.

Rosenblatt, F. (1960). "Perceptron Simulation Experiments," *Proc. IRE,* Vol. 48, pp. 301–309.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington, DC: Spartan Books.

Rumelhart, D. E. and J. L. McClelland (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition,* Vol. 1. Cambridge, MA: MIT Press.

Samuel, A. L. (1959). "Some Studies in Machine Learning Using the Game of Checkers," *IBM J. of Res. and Dev.,* Vol. 3:3, pp. 210–229.

Samuel, A. L. (1963). "Some Studies in Machine Learning Using the Game of Checkers." In *Computers and Thought,* edited by E. A. Feigenbaum and J. Feldman. New York: McGraw-Hill, pp. 71–105.

Schaeffer, J. (1996). *One Jump Ahead: Challenging Human Supremacy in Checkers.* Berlin: Springer.

Schank, R. C. (1984). *The Cognitive Computer: On Language, Learning, and Artificial Intelligence.* Reading. MA: Addison-Wesley.

Schildt, H. (1987). *Artificial Intelligence Using C.* New York: McGraw-Hill.

Shannon, C. E. (1950). "Programming a Computer for Playing Chess," *Philosophical Magazine,* Vol. 41, pp. 256–275.

Shortliffe, E. H. (1974). "MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection." Ph.D. diss., Stanford University.

Simpson, P. K. (1990). *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations.* New York: Pergamon Press.

Singh, J. (1966). *Great Ideas in Information Theory, Language and Cybernetics.* New York: Dover.

Staugaard, A. C. (1987). *Robotics and AI: An Introduction to Applied Machine Intelligence.* Englewood Cliffs, NJ: Prentice Hall.

Swade, D. D. (1993). "Redeeming Charles Babbage's Mechanical Computer," *Scientific American,* Vol. 268, No. 2, February, pp. 86–91.

Turing, A. M. (1950). "Computing Machinery and Intelligence," *Mind,* Vol. 59, pp. 433–460.

Turing, A. M. (1953). "Digital Computers Applied to Games." In *Faster than Thought,* edited by B. V. Bowden. London: Pittman, pp. 286–310.

Waterman, D. A. (1986). *A Guide to Expert Systems*. Reading, MA: Addison-Wesley.

Weiner, N. (1961). *Cybernetics,* Part 2. Cambridge, MA: MIT Press.

Werbos, P. (1974). "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences." Ph.D. diss., Harvard University.

Wilson, E. O. (1971). *The Insect Societies*. Cambridge, MA: Belknap Press.

Wooldridge, D. E. (1968). *The Mechanical Man: The Physical Basis of Intelligent Life*. New York: McGraw-Hill.

Yager, R. R. (1980). "A Measurement-Informational Discussion of Fuzzy Union and Intersection," *IEEE Trans. Syst. Man Cyber.,* Vol. 10:1, pp. 51–53.

Zadeh, L. (1965). "Fuzzy Sets," *Information and Control,* Vol. 8, pp. 338–353.