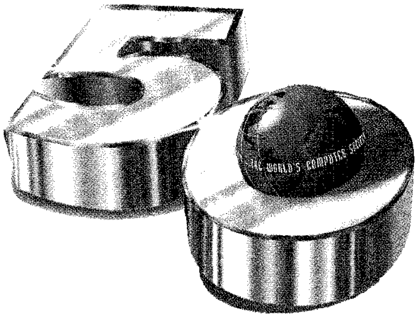


The Challenge of Artificial Intelligence

Raj Reddy
Carnegie Mellon University

Artificial intelligence is founded on the premise that all cognitive activity can be explained in terms of computation. This premise has a long and illustrious tradition in Western philosophy, starting with Aristotle and Plato, who believed that thought, like any other physical phenomenon, can be unraveled using scientific observation and logical inference. Gottfried Leibniz, who equated thought with calculation, set the stage for George Boole's treatise on propositional logic boldly titled "The Laws of Thought." The advent of computers and the progress made in symbolic computation led to a new branch of computer science envisioned in Alan Turing's "Computing Machinery and Intelligence."

Artificial intelligence was inaugurated as a formal discipline in 1956 at a Dartmouth College conference by John McCarthy, Marvin Minsky, Allen Newell, and Herbert Simon. AI's scientific goal is to understand the principles and mechanisms that account for intelligent action. The allied engineering goal is to design intelligent artifacts that can survive and operate in the physical world and solve problems of considerable scientific difficulty at high levels of competence.

The main challenge of AI is to create models and mechanisms of intelligent action. AI is primarily an empirical science, in which researchers use the classical hypothesis-and-test research paradigm to validate these models and mechanisms. The computer is the laboratory where AI experiments are conducted. Design, construction, testing, and validation of computer programs are the processes by which AI researchers verify hypotheses.

An intelligent system is characterized as one that can

- exhibit adaptive goal-oriented behavior,
- learn from experience,
- use vast amounts of knowledge,
- exhibit self-awareness,
- interact with humans using language and speech,
- tolerate error and ambiguity in communication, and
- respond in real time.

Table 1 lists some tasks AI researchers have explored. AI tasks run the gamut in knowledge content, data rates, and response times. Such diverse requirements for intelligent systems indicate that these systems place significant demands on the system architectures with respect to memory capacity, bandwidth, access times, and processor speed, and that there may be no simple unified architecture for creating AI (see also the "AI and computer science" sidebar.)

AI is a relatively young discipline, yet it has already led to general-purpose problem-solving methods and novel applications. Ultimately, AI's goals of creating models and mechanisms of intelligent action can be realized only in the broader context of computer science.

HUMAN AND OTHER FORMS OF INTELLIGENCE

Human intelligence has always had a mystical quality. Thus, the term

---

This article is based in part on material in the author's AAAI Presidential Address, August 1988, and on the author's Turing Award presentation, March 1995.

0018-9162/96/$5.00 © 1996 IEEE

"artificial intelligence" has generated strong emotions among some philosophers and physicists who are otherwise capable of rational, objective thinking. Whether or not computer-based intelligence equals human intelligence is debatable. However, if we start with an operational definition of intelligence such as, "a (mental) act or series of acts is intelligent if it accomplishes something that, if accomplished by a human being, would be called intelligence," computers have already exhibited intelligent behavior. A system that plays pretty good chess, keeps a car on the road, or diagnoses symptoms of a disease is exhibiting intelligent behavior under this definition. Herb Simon, founder of AI, says (personal communication),

> Computer intelligence has been a fact at least since 1956, when Logic Theorist, LT, found a proof that was better than the one found by Whitehead and Russell, or when the engineers at Westinghouse wrote a program that designed electric motors automatically.

Confusion arises when we attempt to equate human intelligence with artificial intelligence. In fact, some properties of human intelligence may not be exhibited in an AI system (often because the researchers have not yet studied that task). Occasionally, AI systems will perform beyond the reach of human intelligence. Ultimately, what AI accomplishes will depend more on the needs of society and funding rather than on philosophical considerations.

Two analogies will illustrate that AI can be both more and less than human intelligence. An electronic book provides the same information as a real book. However, one cannot lie in bed and read an electronic book, at least not yet. On the other hand, unlike a paper book, an electronic book permits processing, indexing, and searching of its contents. Thus, an electronic book has more functionality along some dimensions and less along others. A second example is the concept of a virtual shopping mall where you can try on some virtual clothing, place an order, and receive the real thing within 24 hours. This doesn't give you the thrill of trying on real clothes before you buy them; however, it does let you walk around a virtual mall in Paris, Milan, or Hong Kong, which could be expensive to do in person. Again, a virtual mall is both more and less than a real shopping mall.

Table 1. AI problem domains and their attributes.

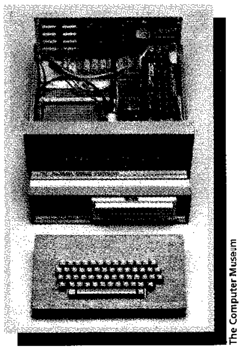
Problem domain	Knowledge content	Data rate	Response time
Puzzles	Poor	Low	Hours
Chess	Medium	Low	Minutes
Theorem proving	Medium	Low	Variable
Expert systems	Rich	Medium	Variable
Natural language	Rich	Medium	Real time
Motor processes	Rich	High	Real time
Speech	Rich	High	Real time
Vision	Rich	Very high	Real time

Certain human capabilities might be impossible for an AI system to accomplish, although AI capabilities will continue to change. However, clearly, some AI systems will have superhuman capabilities that would increase and extend specific human capabilities. Those possessing these tools will be regarded as having superhuman capabilities, just as was once true of people possessing other artifacts, such as the airplane. AI is all about creating artifacts that enhance humans' mental capabilities.

40 YEARS OF PROGRESS

Since its inception, AI has made steady progress (see the "Five Laws of Intelligent Action" sidebar). Space constraints of this article, however, permit only a brief sampling of the advances.

1975 The first PC, an Altair 8800, available as a kit, appears on the cover of Popular Electronics in January.



The Computer Museum

1975 Michael Jackson describes a method to treat a program's structure as a reflection of a problem's structure, a precursor to the Jackson System Development method.

1975 John Cocke works on the 801 project at IBM to develop a minicomputer with the yet-unnamed RISC architecture.

1975 IBM introduces the laser printer.

The George C. Page Museum, © LACMNH

1975 Frederick Brooks writes The Mythical Man-Month, which describes software development as "the mortal struggle of great beasts in the tar pits" and advises that adding more people to a late project only makes it later.

1975

AI and computer science

In a general sense, AI is just software. In building AI systems, which are frequently large and complex, designers must often develop the necessary algorithms and software tools to do so. For example, AI researchers were responsible for many early advances in computer science, such as search algorithms, list structures, pointers, virtual memory, dynamic memory allocation, garbage collection, and so on.

Algorithms and AI

Like complexity theorists, AI researchers are concerned with NP-complete problems. But unlike those theorists' interest in the complexity of a given problem, the focus of AI research is in finding algorithms that provide approximate solutions with no guarantee of optimality.

Besides finding solutions to exponential problems, AI algorithms must often satisfy one or more of the following characteristics of intelligent systems.

Goal-oriented behavior

Suppose you wanted to create an agent that can make a telephone call. This task requires converting the goal into subgoals, such as consulting the phone directory, dialing the number, and talking to the answering agent. Satisfying a goal requires translation of a desired outcome (a goal) into a sequence of actions, the so-called "what to how" problem. To solve such problems, an algorithm must create an agenda of goals and subgoals and use knowledge about operations and methods that can translate a desired goal into a sequence of actions.

Learn from experience

A unique AI system attribute is learning from experience. This implies that AI systems have algorithms for automatically modifying structure and function of a system based on experience. Thus, in the "get me X" phone task, if X is ambiguous and a human being helps resolve the ambiguity, the system must acquire and internalize that knowledge so that the system knows how to do it the next time without human assistance. Learning implies that the system is capable of acquiring, representing, and using the knowledge and engaging in a clarification dialog to resolve ambiguity.

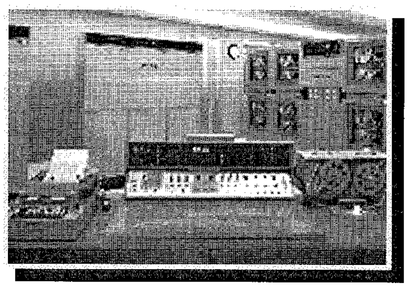
Chess

Chess is an AI problem par excellence. Chess is to AI what E. coli is to biology. Problems that arise in chess-playing programs (and the techniques for solving them) are relevant in solving other AI problems. The total number of moves in chess (over 10^{100}) is estimated to be larger than the number of atoms in the universe. Thus, attempts to solve the chess task based on a brute-force exhaustive approach invariably fail. Some minimal knowledge of chess strategy and tactics appears essential to achieve respectable performance.

In the mid-fifties, Allen Newell, Cliff Shaw, and Herb Simon developed the first operational chess program. In the sixties, the Greenblatt Chess Program won a game in a regular chess tournament, beating a Class C player. (The US Chess Federation's rating system ranges from a Class D player's 1,200 or more game points to a Grandmaster's 2,600 or more points.) In the seventies and eighties, a number of systems such as Northwestern, Belle, and HiTech made steady progress, ultimately achieving a Senior Master's rating.

The Deep Blue system developed at IBM was based on Deep Thought, the first system to achieve a Grandmaster's rating, developed at Carnegie Mellon University by C.B. Hsu and Thomas Anantharaman. At the 1996 ACM tournament, Deep Blue challenged Gary Kasparov, won the first game, and ultimately lost with a score of 4-2. This system, which uses brute-force search and, with limited knowledge, runs on a custom IBM SP-2 processor with 256 processing elements, and can analyze more than 100 million moves per second.

Another interesting system, Fritz 4.0, runs on a conventional PC and has a rating of more than 2,400 points. It beat Deep Thought II and other chess programs in a tour-



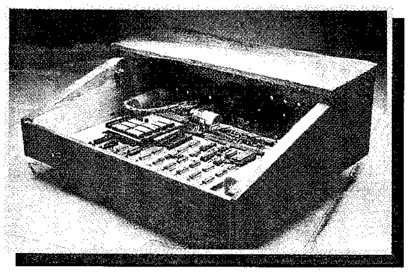
The Computer Museum

1976 The Cray-1 from Cray Research is the first supercomputer with a vectorial architecture.

1976 Gary Kildall develops the CP/M operating system for 8-bit PCs.

1976 OnTyme, the first commercial e-mail service, finds a limited market because the installed base of potential users is too small.

1976 IBM develops the ink-jet printer.



The Computer Museum

1976 Steve Jobs and Steve Wozniak design and build the Apple I, which consists mostly of a circuit board.

Vast knowledge

Effective intelligent system design often requires embodying the system with knowledge equivalent to what is possessed by a human being in solving similar problems. Even if a learning system can acquire and represent new knowledge, there is still the problem of scale. Using vast amounts of knowledge not only requires large memory capacity but creates the problem of selecting and applying the right knowledge for a given task and context. If someone asked, "Is the Pope sitting or standing right now," a system with a large database of facts might have to search terabytes of data before concluding that it does not know the answer. What is needed are algorithms that "know what they do not know," a currently unsolved problem in AI.

Self-awareness

AI systems need the capabilities to explain their behavior and to monitor, diagnose, and repair themselves in the presence of viruses. This requires internal mechanisms that can loosely be called self-awareness. On-line help manuals capable of answering "how-to and what-if" questions, and consistency checks, are some of the mechanisms that are needed for self-aware algorithms.

Language and speech

Because intelligent systems will be required to interact with humans and other intelligent systems, they must be capable of using language and speech. Unless an agent can conduct a clarification dialog, with a human master, in learning new knowledge and eliminating ambiguities, progress in the development of intelligent systems will be slow. Use of language and speech implies not only parsing and interpreting natural language, but handling ambiguity and nongrammaticality as well.

Error and ambiguity

Human communication is necessarily imprecise and sometimes inaccurate. Thus, AI systems must be able to tolerate error and ambiguity in communication. Rather than saying "does not compute," we must develop algorithms that can detect ambiguity and resolve it either through simultaneous parallel execution or by engaging in a clarification dialog.

nament in 1995. In 1994, it defeated Gary Kasparov, Nigel Short, and other Grandmasters in a blitz match.

Chess research has led to a number of search algorithms such as Alpha Beta, B* search, singular-extension-search and hash table representation for keeping previously examined moves. Many of these search concepts have since found their way into everyday applications and provided new insights into the nature of intelligence.

Speech

Speech recognition has always been one of AI's tough nuts. As a perceptual task, it represents a knowledge-intensive task domain with high data rates, requiring instantaneous response. These attributes make the speech recognition task several orders of magnitude more complex than tasks with low data rates and/or without time constraints. The speech task domain also embodies requirements of intelligent action: operate in real time; exploit vast amounts of knowledge; tolerate error and imprecision; use language; and learn from examples.

Speech, a natural medium of communication, offers a number of advantages in providing a natural, hands-free, eyes-free, location-independent input medium. At the same time, the many sources of variability substantially complicate the task. Sources include environmental noise, microphone characteristics, speech rate and loudness, speaker variability, and repetitions, restarts, and ungrammaticality of spontaneous, unrehearsed speech.

Harpy, Hearsay, and HWIM ("hear what I mean") systems, demonstrated in the seventies as the first large vocabulary (1,000+ words) connected-speech systems, used syntax and semantics as major knowledge sources for the first time. They also led to basic techniques such as blackboard models, network representations, Hidden Markov

Apple Computer, Inc.

1977 Steve Jobs and Steve Wozniak incorporate Apple Computer on January 3.

1977 The Apple II is announced in the spring and establishes the benchmark for personal computers.

1977 Several companies begin experimenting with fiber-optic cable.

1977 Bill Gates and Paul Allen found Microsoft, setting up shop first in Albuquerque, New Mexico.

Microsoft Archives

1977

Real time

Intelligent systems, such as an autonomous mobile vehicle, will need to respond in real time. Rather than continuing blissfully until completion, real-time algorithms must use iterative solutions in which partial, approximate results are available at all times. Thus the system always has an answer, but the quality of the answer gets better with time.

Software systems and AI

Large AI systems, especially those in daily use, share many of the problems of other software systems, that is, "not on time," "over budget," and "brittle." Not only are these systems usually large and complex, but because designers often can't fall back on what they learned from having developed similar systems before, it's difficult to formulate requirement specifications or testing procedures.

New concepts are, however, emerging within the context of AI that suggest new approaches and methodologies for all of software engineering. Integrated concept demonstrations of AI systems need components (modular reusable software) that learn, use knowledge, tolerate ambiguity, use language, and operate in real time. For a system to be operational in a reasonable amount of time, such systems must use software architectures and interfaces that plug and play together. This idea is similar to the old "blackboard" concept in AI of cooperating agents that can work together but that do not explicitly require each other.

Paradigm shift

Early AI systems were often designed to be fully autonomous and attempted to provide a replacement for human capability, such as a system for translating Russian to English. More recently, in an apparent paradigm shift, AI researchers are looking at 80/20 solutions. For example, rather than trying to create a fully automatic translation system, why not create a "translation assistant" that supports a human translator? If the system were able to do 80 percent of the task, leaving the remaining 20 percent to the human, such a system would significantly improve productivity. Once such a system became operational, the research focus would shift to partially automating the remaining 20 percent. Such 80/20 systems would incrementally approach human performance, while pro-

model-based learning, and beam search. During the eighties, several commercial systems with very large vocabularies were demonstrated by IBM and others, but the computational and accuracy constraints in these systems forced the speaker to pause between words while speaking.

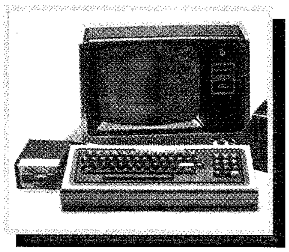
Advances in speech provide insights into the structure of intelligent systems, such as how to use incomplete, inaccurate, partial knowledge in problem solving. The Sphinx III system developed at Carnegie Mellon University can recognize continuous speech without having to retrain for each new speaker. The system operates in near real time on a Pentium Pro PC, using a 50,000-word vocabulary on a voice-mail dictation task. Sphinx III overcomes many limitations of speaker-dependent systems through careful modeling of speech knowledge, unsupervised learning, and effective use of large amounts of training data.

Vision

The goal of computer vision is to automatically interpret and understand image data and to construct 3D models from real-world scenes. Vision shares many of the problems and sources of variability observed in speech—only the task complexity is increased by two orders of magnitude as a result of higher data rates and larger amounts of data. Given the tasks that require vision—manipulation, mobility, and recognition—the performance of a vision module can be readily validated by whether or not a given task is accomplished.

Vision is an active process. This idea is not new in psychology, but in machine vision, the technology is not yet available to fully realize the concept. The implication of active vision is that control of the data acquisition process is part of the vision task. This requires enough memory and real-time processing to analyze spatio-temporal sequences.


The Computer Museum


The Computer Museum

***1977** PCs from Tandy and Commodore come with built-in monitors and thus require no television hookup.*


Digital Equipment Corporation

***1978** DEC introduces the VAX 11/780, a 32-bit computer that becomes popular for technical and scientific applications.*

1977-1978

viding a useful capability at all times. To be effective, such systems must have a smooth user interface, be self-aware, and know enough to say, "Please wait. I'll call my supervisor."

Two strategies

In designing large, complex systems, developers use two strategies: "get it right the first time" and "fail fast." The former is applied when designing a new model of a product or system that has been produced many times before, where there is a wealth of accumulated knowledge and experience to build on. In designing a system that has never been built before (usually the case with most AI systems), "get it right the first time" may not be practical. For example, NASA, which usually designs fail-safe systems, has recently adopted "fail fast" strategies in designing robots for planetary exploration. The first Dante system, designed at CMU, failed after taking a few steps. The second Dante lasted a week. Both time and cost expended on these experiments were at least an order of magnitude smaller than if NASA had insisted on a fail-safe system. In developing large and complex software systems, similar strategies may well be appropriate.

Computer architecture and AI

The Sphinx speech recognition system that was discussed in the main text achieves its real-time performance not just through the use of AI techniques such as representation, search, and learning but also through the use of efficient data structures and application-specific hardware architectures. At Carnegie Mellon in the eighties, Roberto Bisiani was able to reduce the time for recognition from about 10 minutes to under 5 seconds. How was this two-orders-of-magnitude improvement achieved?

- a speedup of 1.5 by replacing sparse arrays with link lists
- a speedup of 3.0 by redesigning the data structures to eliminate pointer chasing
- a speedup of 2.0 by redesigning the beam search algorithm to use dynamic thresholds and inserting best state at the top of the list
- a speedup of 2.5 by using faster processors
- a speedup of 1.6 by using a multiple memory architecture
- a speedup of 2.1 by using a multiprocessor architecture for parallel-search execution.

The first computer vision system was developed at MIT's Lincoln Laboratories by Larry Roberts in the early sixties. He created 3D models from gray-scale images of the "blocks world" task. Roberts used a bottom-up approach starting from edge detection, linear feature identification, 3D model matching, and object recognition. In the seventies and eighties, a number of fundamental results in vision were formulated: a solution for the inverse optics problem, shape from shading and color, shape from geometric properties and constraints, and shape from motion.

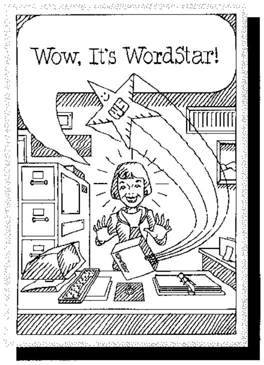
Thirty years of progress in vision can be measured by using the complexity of images analyzed: from "blocks world" to curved objects to multiple parts, and to more robust procedures that work with incomplete and noisy information. Two recent results highlight the dramatic progress that has been made since the late eighties:

- the ability to remove highlights from a color image to produce an "intrinsic body reflection" image using a "dichromatic reflection model" developed by Steve Shafer, and
- algorithms developed for estimating depth from a sequence of N images that is N times better than the stereo depth estimation.

Advances in range image processing and real-time depth recovery, using multiple cameras based on multi-baseline stereo theory developed by Takeo Kanade and others at the CMU Robotics Institute, are leading to new applications in virtual reality, telepresence, and entertainment.

Robotics

Work in robotics goes back to the early sixties when Henry Ernst developed a computer-controlled manipula-

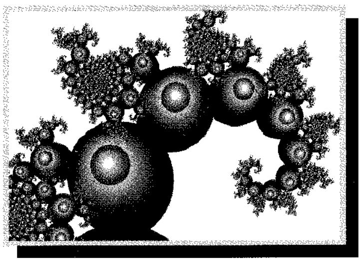


1978 Wordstar is introduced and goes on to become a widely used word processor with CP/M systems and later on DOS computers.

1978 Tom DeMarco's Structured Analysis and System Specification popularizes structured analysis.

1978 Ron Rivest, Adi Shamir, and Leonard Adelman propose the RSA cipher as a public-key cryptosystem for enciphering digital transmissions.

1978 Intel's first 16-bit processor, the 8086, debuts.


pjcarlsn@ix.netcom.com

1979 Benoit Mandelbrot continues his research into fractals by generating a Mandelbrot set, derived from $z(n+1) = z(n) * z(n) - (0)$.

1978 — 1979

All these speedups, though small and independent, are multiplicative (as conjectured by Reddy and Allen Newell in 1977), resulting in a speedup by a factor of 75, or 7,500 percent.

The main lesson here is that serious AI scientists of the future must also be competent in other fields of computer science: algorithm analysis, data structures and software, and computer architecture.

Neural architectures and AI

Recent interest in connectionist research has resulted in better understanding of the computational aspects of human intelligence. One of the human brain's intriguing characteristics is that the fan-in and fan-out of connections in neural cells is approximately 1,000. Most of the brain's volume is occupied by wires, even though the wire cross-sections are of submicron dimension. A hundred billion processing elements with 1,000 connections each represent a hundred trillion connections. As many as 1 percent of these wires are active in a 5-ms iteration involving a trillion computations every 5 ms. The brain thereby appears to perform 200 trillion operations per second, give or take a few orders of magnitude.

A scanning electron micrograph of a neuronal circuit grown in tissue culture on an M68k microprocessor by J. Trogadis and J.K. Stevens of the University of Toronto dramatically illustrates the differences between the human cell and electronic transistor architectures. Axon and dendritic structures of brain cells are seen to be much finer than the micron dimensions of the wires in M68k.

For tasks such as vision, language, and motor control, a brain is more powerful than 1,000 supercomputers. And yet, for simple tasks such as multiplication it is less powerful than a 4-bit microprocessor. This leads us to speculate that silicon-based intelligence, when it is achieved, may have different attributes. We need not build airplanes with flapping wings.

The brain processes and identifies a scene, such as the Washington Monument, in a few hundred milliseconds. This has led the connectionists to observe that whatever processing is occurring within the human brain must be happening in less than 100 clock cycles. AI scientists want to find answers to questions such as: What is the nature of representation and the nature of computation taking place within the human brain that leads to such performance, where no "recognition" task needs more than 100 steps?

tor at the MIT AI Labs. Since then, robotic manipulators have been able to perform with greater precision in increasingly complex task frameworks. Such robots are routinely used today in manufacturing environments, although acceptance has been slow in coming. A more exciting direction in robotics research has been the development of autonomous mobile systems. Robotic vehicles are challenging because such systems require the bringing together of disciplines as diverse as computer vision, advanced sensors, high-speed processors, planning, control, and learning, which results in vehicles that can navigate themselves on roads and cross country.

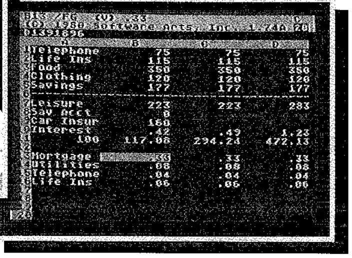
Early work in mobile vehicles resulted in the Stanford Cart, which took 15 minutes to map the environment, plan a path, and control the vehicle before it could move a meter. At CMU, a number of vehicles were built over a 15-year period starting with CMU IMP and Neptune, followed by a family of autonomous land vehicles called the Navlab Series.

Navlab I, developed in 1986, was the first self-contained test bed. It contained on-board generators, on-board sensors, on-board computers, and on-board graduate students. Navlab II, an army ambulance HMMWV (high mobility, multiwheeled vehicle), had many of the sensors from Navlab I in addition to pan/tilt cameras providing trinocular stereo vision. Computer-controlled motors turned the steering wheel and controlled brake and throttle. HMMWV was capable of 110 KPH in highway driving.

The more recent Navlab V, a commercial van from General Motors, was modified for autonomous steering along with appropriate sensing and control systems. This system was able to navigate correctly 98 percent of the

The Computer Museum

1979 The first electronic spreadsheet program, Don Bricklin's and Bob Franston's VisiCalc, is unveiled on May 11 and proves to be the "killer app" for early PCs.



```
B13 /FG (V) .33
(C) 1980 Software Arts, Inc. 1.740 20
01391896
          A           B          C          D
Telephone            75         75         75
Life Ins            115        115        115
Food                350        350        350
Clothing            120        120        120
Savings             177        177        177
-----------------------------------------------
Leisure             223        223        283
Sav Acct              0
Car Insur           160
Interest            .42        .49       1.23
       100       117.08     294.24     472.13

Mortgage            .33        .33        .33
Utilities           .08        .08        .08
Telephone           .04        .04        .04
Life Ins            .06        .06        .06
```

1979 Motorola introduces the 68000 chip, which will later support the Macintosh.

1979 Digital videodisks appear through the efforts of Sony and Philips.

© 1996 Motorola Museum of Electronics

1979 Cellular telephones are tested in Japan and Chicago.

1979

time from Washington, DC, to San Diego during the summer of 1995. When in doubt, the system warns the human driver to take over.

In the US, we have over six million automotive accidents every year, costing over $50 billion in repairs. These accidents result in over 40,000 fatalities, 96 percent of which are caused by driver error. In a driver-warning application, the Navlab V Vision Systems sound an alarm to the driver if the vehicle drifts out of lane, as might happen when a driver falls asleep or drives under the influence of alcohol or drugs.

The Automated Highway Systems project in the US hopes to develop automated cars, trucks, and buses driving on an instrumented highway. Such systems are expected to improve safety, decrease congestion, and improve mobility for the elderly and disabled.

Five laws of intelligent action

What fundamental principles govern AI? The past 40 years of sustained and systematic explorations have provided a number of insights into the nature of intelligent action.

1. Bounded rationality implies opportunistic search.

The first law says that "computational constraints" on human thinking lead people to be satisfied with a "good enough" solution rather than waiting for the optimal solution. This law is based on Herb Simon's Nobel Prize-winning research on decision making in organizations. When people must make decisions under conditions that overload human thinking capabilities, they use opportunistic strategies and tactics of "optimal least computation search" rather than "optimal shortest path search." Much of AI is the study of approximate algorithms of optimal least computation search. Silicon-based intelligence, given its differences in memory access time and bandwidth, may indeed use different strategies and tactics than human intelligence.

2. A physical symbol system is necessary and sufficient for intelligent action.

The second law of AI is the physical symbol system hypothesis—a physical symbol system is necessary and sufficient for intelligent action—stated by Allen Newell and Herb Simon in their Turing Award paper. A physical symbol system has the following properties:

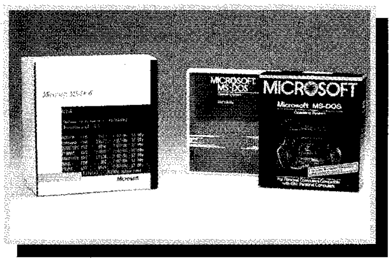
- Physical symbols are symbols that are realizable by engineered components.
- A physical symbol system is a set of these entities.
- A symbolic structure is an expression whose components are symbols.
- Operations on the expression include creation, modification, reproduction, and destruction of symbols.
- Expressions can be interpreted as plans of action.

Lisp, Prolog, the Turing machine, Post Productions—for that matter, any computer system—all have the mechanisms required of a physical symbol system. Although sufficient for intelligent action, is a physical symbol system necessary? Does the human brain have the mechanisms and properties of a physical symbol system? We in AI believe it does, but we cannot prove it. That's why this law is a hypothesis.

3. The magic number is 70,000 ± 20,000.

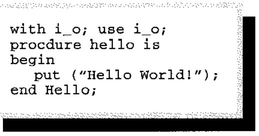
The third law is that an expert knows 70,000 "chunks" of information, give or take a binary order of magnitude. Why is it 70,000? Why not one thousand or one million? Experimental evidence in cognitive science leans toward 70,000 as a rough measure of the size of an expert's knowledge base. For example, it appears that chess masters recognize about 50,000 chunks. William Chase and Herb Simon quantified this number by constructing an experiment based on the ability of master-level, expert-level, and novice-level players to recreate chess positions they had seen for only a few seconds.

But is it true for experts in all domains? The evidence points to the magic number, 70,000 ± 20,000. We know that vocabularies of college graduates are about that size. As we gain



Microsoft Archives

1980 IBM selects PC-DOS from upstart Microsoft as the operating system for its new PC.

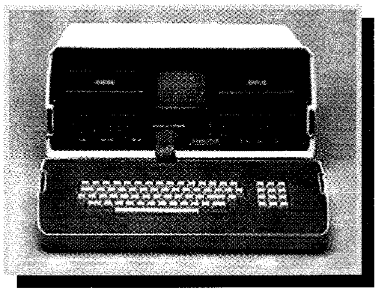


http://www.latec.edu/~acm/HelloWorld.shtml

```
with i_o; use i_o;
procdure hello is
begin
   put ("Hello World!");
end Hello;
```

1980 After a long development period, the Ada language emerges. Developed by the US Department of Defense, it is designed for process control and embedded applications.

1980 Wayne Ratliff develops dBase II, the first version of a PC database program. It goes on to enjoy wide market success.



1980 The Osborne 1 "portable" computer weighs 24 pounds and is the size of a small suitcase.

more experience in building expert systems, we find that the number of productions begins to grow toward tens of thousands if the system is to perform more than a very narrow range of tasks in a given domain. It has been observed that no human being reaches world-class expert status without at least a decade of intense full-time study and practice in the domain. Even the most talented don't reach expert levels of performance without immense effort. Each of us is an expert in speech, vision, motion, and language. Considering the time it took us to accumulate that much expertise, we seem to be left with only enough time to be an expert in only two or three other areas in a lifetime.

4. Search compensates for lack of knowledge.

The fourth law is that search compensates for lack of knowledge. When faced with a puzzle we have never seen before, we engage in trial-and-error behavior, usually until a solution is found. During the sixties and seventies, it was believed that masters-level performance in chess could not be achieved except by codifying and using knowledge of expert human chess players. We now know that Deep Blue can play at Grandmaster-level, even though its knowledge is nowhere comparable to a chess master. The key lesson here is that there may be more than one way to achieve expert behavior in a domain such as chess.

This law may be true for problem-solving tasks such as puzzles and games. But what about perceptual tasks? Language is one area where search appears to compensate for incomplete and inaccurate knowledge. For example, the verb "take" can mean many things: take a book, take a shower, take a bus, take a deep breath, take a measurement, and so on. In cases where words can have many meanings, the precise meaning can be clarified by the context and by exploring all the alternatives until the meaning is unambiguous. What this law tells us about the role of search is that we need not give up hope when faced with a situation in which all the known knowledge is yet to be acquired and codified.

5. Knowledge compensates for lack of search.

The fifth law, an important insight, was not clearly understood even as late as 1970; that is, knowledge reduces uncertainty and helps us constrain the exponential growth leading to the solution of many otherwise unsolvable problems. Knowledge is power. Indeed, "recognition" knowledge can eliminate the need for search altogether. This principle is essentially the converse of the previous principle. The solution of Rubik's cube illustrates the importance of experiential knowledge. The first time around, it is not uncommon for most people to take half an hour or more to solve this puzzle. With practice, however, the situation improves dramatically.

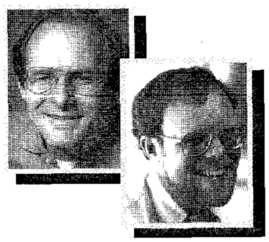
The speech task also provides some quantitative data about the importance of knowledge. The Sphinx system can be run with various knowledge sources turned off. If the syntactic knowledge source were removed, sentences of the form "Sleep roses dangerously young colorless" would be legal. Removing the syntactic knowledge source increases the error rate of Sphinx from 4 percent to 30 percent (about one out of three words would be incorrect). Removing the probabilistic knowledge about the frequency of word occurrence, however, only increases the error rate from 4 percent to only 6 percent

Expert systems

During the sixties, Edward Feigenbaum, Josh Lederberg, and others at Stanford University applied AI techniques to discover molecular structures from mass spectral data. The resulting systems, Dendral and its successor Mycin, gave birth to the field of expert systems.

This research led to the realization that general-purpose AI methods were incapable of delivering expert-level performance on problems that required domain-specific knowledge, such as those arising in medical diagnosis and chemical structure elucidation. Dendral laid the foundations for knowledge-based system development.

The main research issues in developing expert systems are the extraction of domain knowledge and the criteria

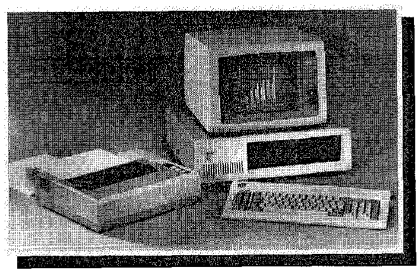


1980 David A. Patterson at UC Berkeley begins using the term "reduced-instruction set" and, with John Hennessy at Stanford, develops the concept.

1981 Barry Boehm devises Cocomo (Constructive Cost Model), a software cost-estimation model.

1981 Japan grabs a big piece of the chip market by producing chips with 64 Kbits of memory.

1981 Xerox introduces a commercial version of the Alto called the Xerox Star.



IBM Archives

1981 The open-architecture IBM PC is launched in August, signaling to corporate America that desktop computing is going mainstream.

1980-1981

Theoretical contributions of AI

Subbarao Kambhampati
Arizona State University

Theoretical AI is primarily the study of algorithms and representations that support efficient, approximate, continual, and resource-bounded reasoning. This significantly broadens the traditional computer science emphasis on correctness upon termination and worst-case space and time complexities.

The computational tasks arising in AI are critically linked to the types of representation languages in which reasoning is done. Mathematical logic has been the language of choice for representing declarative knowledge. Researchers have investigated a variety of logic subclasses that aid reasoning by limiting expressiveness. The area of deductive databases is a crossroads, where researchers from AI and databases consider how to represent and efficiently reason about logical knowledge.

The inherent uncertainty in most real-world knowledge has also spurred work on default logics and probabilistic representations, both of which support the incremental modifications of conclusions as evidence accumulates. Default (nonmonotonic) logics have been hailed as the first major extension of mathematical logic in a long time. Bayesian networks, which can compactly represent the dependency relationships among a set of random variables, have become the de facto representation standard for propositional uncertain knowledge. Although inference with these networks is still intractable in the worst case, many efficient approximation algorithms have been developed. Developing representations for first-order probabilistic knowledge is an active area of current research.

At the heart of AI enterprise lies the problem of helping an autonomous agent decide what to do next. Planning and reasoning about actions have thus driven many of the theoretical advances. Several representational restrictions for modeling action under uncertain and incomplete knowledge have been developed. Reasoning itself has moved from explicit simulations on world states to the more efficient action-centered representations that support direct manipulation of partially specified courses of action. This, in turn, allows plan synthesis through an efficient search of potential courses of action, with the sets being refined (narrowed) through goal-directed reasoning. The normative basis for action selection, traditionally

used for decision making.

Knowledge-based systems are the most visible contribution of AI to industrial applications. They apply a simple theoretical idea: Symbolic reasoning guided by heuristics over declaratively specified knowledge of a domain can result in impressive problem-solving ability.

GRAND CHALLENGES OF AI

A Grand Challenge is a seemingly reasonable problem that is exciting and challenging yet currently unsolvable. Solutions to AI's Grand Challenges will require major new insights and fundamental advances in computer science and artificial intelligence and, if they're successful, can be expected to have a major impact on society.

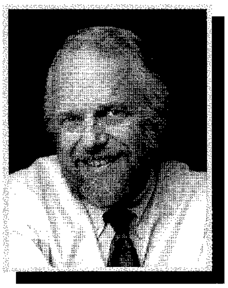
Each of these tasks requires long-term, stable funding at significant levels. Success is by no means guaranteed, and each problem represents a high-risk, high-payoff investment. However, even partial success can have spin-offs to industry and have a major impact on competitiveness.

TRANSLATING TELEPHONE. A translating telephone is a system in which a Japanese speaker can converse with, say, an English speaker in real time. This requires solutions to a number of currently unsolved problems: a speech recognition system capable of recognizing a large (possibly unlimited) vocabulary and spontaneous, unrehearsed, continuous speech; a natural-sounding speech synthesis preserving speaker characteristics; and a natural language translation system capable of dealing with ambiguity, nongrammaticality, and incomplete phrases.

ACCIDENT-AVOIDING CAR. An accident-avoiding car

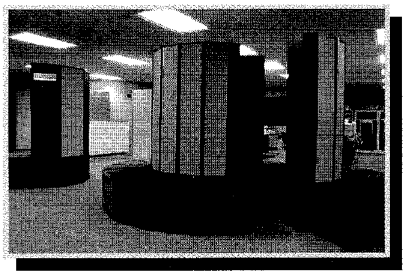
1982 Columbia Data Products produces the first IBM PC "clone." Compaq soon follows with its own version.

1982 Autodesk is founded and ships the first version of AutoCAD later that year.


Adobe Systems, Inc.

1982 John Warnock develops the PostScript page-description language and with Charles Geschke founds Adobe Systems.

1982 Time magazine names the computer as its "Man of the Year."


Cray Research

1982 The Cray X-MP (two Cray-1 computers linked in parallel) proves three times faster than a Cray-1.

provided by Bayesian decision theory, has been extended to consider the costs and benefits of deliberation.

Computational tasks considered in AI are complicated by the fact that the agent may have severe limitations on the computational resources available to it. Resource limitations are handled by a new breed of *any-space* or *any-time* algorithms. A variety of search algorithms have been developed that can work with arbitrarily low memories and guarantee optimal solutions upon termination. Although arbitrary time restrictions cannot be met given the mostly intractable computations, a reasonable requirement is that the algorithms in question be flexible (anytime). This means they can be interrupted at any time and they return results whose value monotonically increases with increased resources. Such anytime algorithms simplify the problem of deciding how many resources to expend on decision-making and how much on execution because they offer a regular relationship between computation time and value.

Many computational tasks arising in AI admit only approximate solutions. A canonical example is inductive learning, which requires inferring a function by looking at a finite set of its values. PAC (probably approximately correct) theory provides a normative basis for evaluating approximate algorithms. Here, an algorithm's performance is measured in terms of the likelihood that it makes less than a certain percentage of output errors. In addition to providing a formal foundation for machine learning, the PAC framework started the fertile field of *computational learning theory*, in which AI and computer science theorists look at the complexity of inductive tasks.

Given the worst-case intractability of many AI problems, researchers have naturally focused on understanding the properties of "average problems." An important recent insight in this area is that many worst-case intractable decision problems have a very narrow region of transition. In other words, they are easily decided in a positive way on one side of the region and easily decided in a negative way on the other side. The hard instances lie mostly in the transition area. An important open issue is the relation between these hard instances and the instances that arise naturally in the real world.

In summary, theoretical AI provides both an increasingly solid foundation for the development of intelligent autonomous agents and a medium of rich interactions with several disciplines of computer science.

***Subbarao Kambhampati** is an associate professor at Arizona State University, where he directs the Yochan group on planning and machine learning. Contact him at rao@asu.edu; http://rakaposhi.eas.asu.edu/yochan.html.*

equipped with an intelligent cruise control using sonar, laser, and vision sensors could eliminate 80 percent to 90 percent of the fatal accidents and cost less than 10 percent of the automobile's total cost. Such a device would require research in vision, sensor fusion, obstacle detection and avoidance, low cost/high speed (over a billion operations per second) digital signal processor chips, and the underlying software and algorithm design.

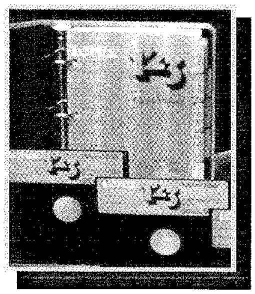
LEARNING SYSTEMS. Interest has long focused on systems that learn and discover from examples, observations, and books. Two long-term grand challenges for systems that acquire capability through learning are to read a chapter in a college freshman text (say, physics or accounting) and answer the questions at the end of the chapter, and learn to assemble an appliance (such as a food processor) from observing a person doing the same task. Both are extremely hard problems requiring advances in vision, language, problem-solving techniques, and learning theory. Both are essential to the demonstration of a self-organizing system that acquires capability through (possibly unsupervised) development.

SELF-REPLICATING SYSTEMS. What capabilities must exist for a factory to make a copy of itself? Self-replicating systems are of some practical interest in areas such as manufacturing in space. Rather than uplifting a whole factory, is it possible to have a small set of machine tools that can

1982 Japan launches its "fifth generation" computer project, focusing on artificial intelligence.

1982 Commercial e-mail service begins among 25 cities.

1982 In November, Compaq unveils an IBM-compatible portable PC.


Lotus

1983 By including graphics such as pie charts and bar graphs, Lotus 1-2-3 does for the IBM PC what VisiCalc did for the Apple II.

1983 A Josephson junction is developed on the basis of Brian Josephson's 1962 prediction, bringing higher speed and lower power dissipation to ICs.

1983 The IBM PC-XT heads for market success, while the PC Junior faces quick extinction.

1983 Completion of the TCP/IP switchover marks the creation of the global Internet.

1982 — 1983

Selected bibliography

Ballard, D.H., and C.M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, N.J., 1982.

A comprehensive review of research progress in computer vision.

Feigenbaum, E.A., P. McCorduck, and H.P. Nii, *The Rise of the Expert Company*, Times Books, New York, 1988.

An excellent summary of the contributions of expert systems to industrial applications.

Newell, A., and H.A. Simon, "Computer Science as Empirical Enquiry: Symbols and Search," *Comm. ACM*, Mar. 1976.

The seminal article proposing the physical symbol system hypothesis.

Rabiner, L., and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, N.J., 1993.

An excellent summary of research in speech recognition over the past 30 years.

Reddy, R., "To Dream the Possible Dream," *Comm. ACM*, May 1996, pp. 105-112.

This Turing Award presentation article elaborates on some of the topics discussed here, including AI's role in computer science.

Reddy, R., "Foundations and Grand Challenges of Artificial Intelligence," *AI Magazine*, Winter 1988, pp. 9-21.

This article, the 1988 AAAI Presidential Address, provides a comprehensive list of references for many of the topics discussed here. It also contains the scanning electron micrograph image J. Trogadis and J.K. Stevens produced, which clearly displays the relative dimensions of neuronal circuits versus VLSI circuits.

Simon, H.A., *Administrative Behavior*, Macmillan, New York, 1947.

The definitive work of Simon's research on human decision making in organizations and the principle of bounded rationality.

Thorpe, C.E., *Vision and Navigation: The Carnegie Mellon Navlab*, Kluwer Academic Publishers, Boston, 1990.

An up-to-date, comprehensive review of research in autonomous mobile systems.

produce, say, 95 percent of the parts needed for the factory using locally available raw materials and assemble it in situ? The solution to this problem on a planet like Mars involves many different disciplines including materials and energy technologies. Research problems in AI include knowledge capture for reverse engineering and replication, design for manufacturability, and robotics technologies for control, diagnosis, monitoring, and repair of machinery.

AI'S IMPACT ON SOCIETY

Mankind has dreamed for centuries of creating artificial but intelligent creatures that can remove the drudgery from everyday life. Hence AI's impact on society can be measured by answering two questions: Do the products that result from AI help society at large? and how much industry has been created as a consequence of AI?

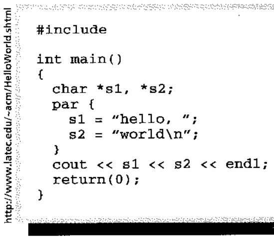
AI technologies have been used successfully by industry in tasks involving analysis (for example, machine diagnosis), synthesis (such as design and configuration), planning, simulation, and scheduling, all of which lead to significant economic gains. AI technologies are also helping chemists and biologists in studying complex phenomena, by searching through enormous databases, creating new molecular structures, and decoding the DNA sequences. Pilots and stock brokers use intelligent assistants in analysis, diagnosis, and planning at the expert level. Medical doctors can examine more hypothetical cases, hence preventing omission of not-so-obvious symptoms. Reasoning about images obtained from the earth-orbital satellite provides crucial information on our environment.

The best-known economic impact of AI is in the widespread use of expert systems. It is estimated that, worldwide, industry saves over a billion dollars each year through this technology. Speech analysis and speech generation

Apple Computer, Inc.

1983 Though not destined for commercial success, Apple's Lisa, launched in May, shows what can be done with a mouse, icons, and pulldown menus.

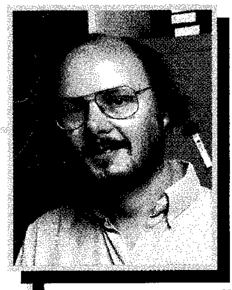
1983 Thinking Machines Corp. and Ncube are founded, providing a boost to parallel processing.



```
#include

int main()
{
  char *s1, *s2;
  par {
    s1 = "hello, ";
    s2 = "world\n";
  }
  cout << s1 << s2 << endl;
  return(0);
}
```

http://www.latec.edu/~acm/HelloWorld.shtml



Bell Laboratories

1983 At AT&T Bell Labs, Bjarne Stroustrup continues work on C++, an OO extension to C.

1983

products are beginning to reach the market. Image understanding programs, leading to image enhancement and recognition of objects, are in daily use in the government, health-care organizations, manufacturing, banking, and insurance. Vision and robotics are beginning to transform manufacturing. Planning and scheduling systems are in routine use in industry and in military settings. Research in AI is also making computers easier to use through the use of SILKy interfaces: interfaces that effectively utilize speech, image, language, and knowledge in user interactions.

Like any other science, AI has the potential to do a lot of good and some bad. The computer and communication technologies will make it possible for a rapid, inexpensive sharing of knowledge. For example, low-cost (perhaps $1,000 to $3,000) PCs capable of executing more than a billion instructions per second should be accessible by anyone in the world in the next decade. With such a system, AI researchers should be able to create a personalized, intelligent assistant to provide expert advice on day-to-day problems, make vast amounts of knowledge available in active form, and help ordinary mortals perform superhuman tasks. Such a system would help the illiterate farmer in Ethiopia as much as the scientist in Japan. To be usable by the farmer, however, such a system must use voice and vision for man-machine communication, and tolerate error and ambiguity in human interaction with machines. Besides providing assistance on day-to-day problems, such a system can also be used to provide education and entertainment on a personalized basis.

CREATING MECHANISMS FOR SHARING of knowledge, know-how, and literacy is the challenge. The great Chinese philosopher Kuan-Tzu once said: "If you give a fish to a man, you will feed him for a day. If you give him a fishing rod, you will feed him for life." We can go one step further: If we can provide him with the knowledge and the know-how for making that fishing rod, we can feed the whole village. Therein lies the promise—and the challenge—of AI. ▮

Acknowledgments

I am grateful to Herb Simon, Ed Feigenbaum, Jaime Carbonell, Chuck Thorpe, Tom Mitchell, Devika Subramanian, and the reviewers for input that helped improve t his article.

Raj Reddy *is dean of the School of Computer Science at Carnegie Mellon University and the Herbert A. Simon University Professor of Computer Science and Robotics. Previously, he taught at Stanford University and, earlier, served as an applied science representative for IBM in Australia. His research interests include the study of human-computer interaction and artificial intelligence. Current research projects include speech recognition and understanding systems, just-in-time learning technologies, and the Universal Digital Library project.*

Reddy is a fellow of IEEE, ASA, and AAAI, and he is a member of the National Academy of Engineering. He is an IBM Research Ralph Gomory Fellow and a recipient of Turing Award. Reddy was presented the Legion of Honor by French President François Mitterrand in 1984.

Contact Reddy at rr@cmu.edu.