EXPERT SYSTEMS

An expert system is a program that can offer advice on a particular subject, much as a human expert might. Such a program's knowledge of the subject is stored in the form of hundreds or thousands of rules of the form:

IF (something is the case)
THEN (the following is true)

These rules are derived through conversation with human experts. In general, the rules are just very explicit statements of the kind of reasoning which human experts go through.

Expert systems in this exhibit can bargain with you for a box of strawberries, offer advice on choosing a wine, diagnose a disease, compose a piece of music and draw a picture.

Several hundred expert systems are currently used in industry. They offer advice and improve accuracy and efficiency in tasks ranging from canning soup to planning space missions to planting crops. Descriptions of several industrial expert systems are presented on this wall.

Cooker Advisor

The canning industry produces billions of containers yearly. Despite an overwhelming volume, a high level of quality control is needed to make sure that canned goods remain unspoiled. The Campbell Soup Company, with help from Texas Instruments, has developed an expert system to diagnose malfunctions that can occur in the cookers, which are used to sterilize the food in the cans.

The program Campbell and Texas Instruments chose uses a backward chaining control structure. This means the system begins with a known goal, then follows a series of commands designed to lead to that conclusion. An analogy would be catching some criminals in the act, then setting up a logical chain of events or motives leading up to the crime to prove their guilt. The backward chaining format allows for a relatively small set of rules and is most common on systems that sell for less than $1000.

An example of a cooker advisor rule translated into English:

IF the cooker's symptom is temperature–deviation,

and the problem temperature is T30–inter mediate–cooling–spray,

and the input and output air signals for TIC–30 are correct,

and the valve on TCV–30 is not open,

THEN the problem with the cooker is that TCV–30 is not working properly. Check the instrumentation and the air signal.

Dipmeter Advisor

Geologists use a combination of core drillings, seismic tests, and other methods to probe into the earth as they look for gas and oil. A dipmeter measures the resistivity and, with analysis, the tilt of rock layers intersected by a borehole. These clues can tell geologists where oil and gas may be located underground.

A system developed by Schlumberger Well Services can interpret dipmeter readings much like a geologist. The system follows a forward–chaining program. This means the system begins with data, then uses a series of narrowing questions to determine the nature of the rock strata. Essentially, the user gives the system information, then asks, ”What can you make of this?” Here is a sample rule from the system:

IF there exists a normal fault with class unknown,
and there exists a red pattern with length greater than 50 feet,
with bottom above the top of the fault,
with azimuth perpendicular to the faultstrike,
THEN the fault is a late fault with direction

to downthrown block equal to the azimuth of the red pattern.

The Dipmeter Advisor printout gives a variety of readings that can help a geologist determine what kind of rock, oil or gas deposits may exist. The series of "tadpole" marks indicate the angle and direction of the dip, or tilt from horizontal, of the rock crossed by the dipmeter tool. The Wolff Plot shows both the dip and the strike, or orientation, of the strata. The True Vertical Depth Stick Plot shows a cross–section of the rock layers if you were to look from inside the borehole. The numbers down the left margin show the depth of the wellbore in feet, and the jagged shaded area indicates the rock types at each depth.

Oil was discovered below the "Growth Fault" identified by the Dipmeter Advisor at 9194 feet. This is an example of a "structural trap" which seals the upper surface of a hydrocarbon reservoir. Since oil rises through porous rock, the trap prevents oil from escaping towards the earth's surface.

Wind Shear

Sudden downbursts of air, commonly found around thunderstorms or mountain regions, can be disastrous to aircraft during landing and takeoff. In the past decade this wind shear has been found directly responsible for numerous air accidents, resulting in hundreds of injuries and deaths. Since wind shear often occurs suddenly and without visible warning, it poses a serious threat to safe air travel.

Federal Aviation Consulting Services, with help from NASA, is developing an expert system that helps air controllers identify likely wind shear conditions. The system combines an airport's weather history with flight data and hourly updates of weather conditions to predict when a "low," "moderate" or "high" wind shear condition exists.

When the likelihood for wind shear is moderate or high, the computer can sound a warning telling pilots not to land or to delay takeoff and landing until the danger passes.

The system will not replace a human flight controller, but rather serves as a backup to improve accuracy and safety.

Planting

Expert systems are used even in an industry that seems far removed from computers. The U.S. Department of Agriculture has developed the ”PLANTING” system to counsel farmers on what farm machinery to purchase, how to manage land wisely, and which planting methods will help avoid soil erosion.

The user inputs soil and planting conditions to receive a listing of machinery that will work. The system can also estimate the erosion danger caused by different types of farm machinery. PLANTING will warn farmers if the erosion hazard exceeds tolerance levels for that area.

This sample rule from PLANTING is used to select from a choice of machines:

```
IF        (1)SOIL AND RESIDUE CUTTING is:
             bubble coulter or narrow fluted coulter
      and (2)ROW PREPARATION is:  not used
      and (3)DEPTH CONTROL is:  side gauge
             wheels
      and (4)SOIL OPENING FOR SEED
             PLACEMENT is:  double disks or
             runner opener
      and (5)SEED Embedding is:  not used
      and (6)SEED COVERING is:  not used
      and (7)SEED SLOT CLOSURE is:  dual
             angled semi-pneumatic press wheels
             or dual angled cast or steel presswheels
THEN      John Deere 7100 - Probability=5/10
      and [PLANTER FLAG] is given the value 1.
```

NAVEX

The danger of manned space flights has been made clear by NASA's space shuttle program. In an effort to minimize the level of hazard, NASA is using expert systems to reduce some of the responsibilities ordinarily borne by crew members.

The Navigational Expert System, NAVEX, monitors the space shuttle's high speed navigation control console. The console plays an important part during the shuttle's return to Earth. Without NAVEX, monitoring the control console requires three operators to check the position, velocity and accuracy of data. The operators must continuously check over 100 radar related parameters and up to 50 status lights. With NAVEX, one operator alone can handle the control console faster, with more accuracy and more data than three could before.

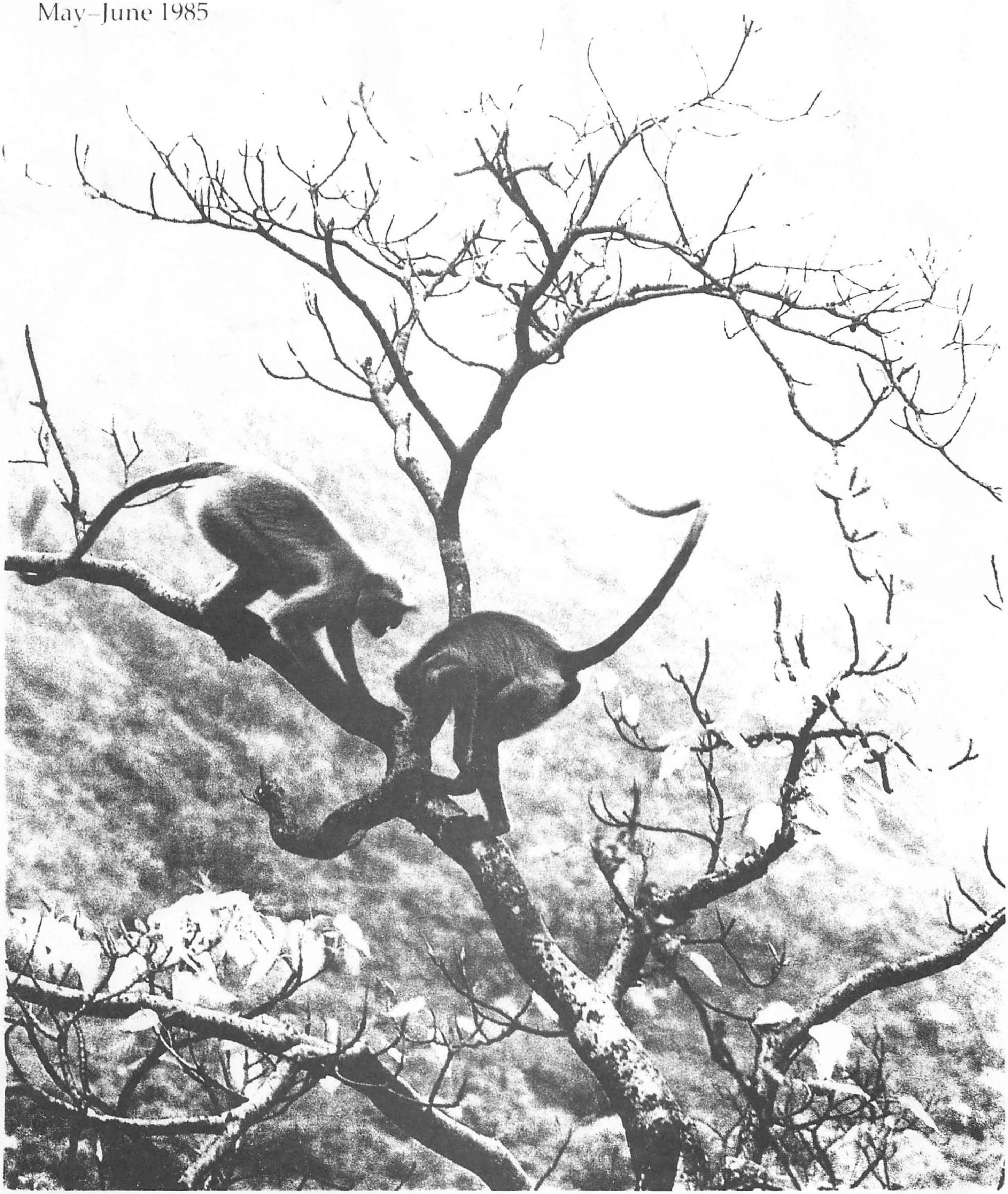
An example rule from NAVEX:

IF a C–Band radar station has a bias problem,

THEN exclude that station from this cycle's processing, unless there are no other stations to process.

American Scientist

May–June 1985



Sigma Xi, The Scientific Research Society, has been since 1886 the honor society of scientists. Its 125,000 members in North America and abroad belong to over 500 chapters on university campuses and in government and industrial laboratories. Election to Sigma Xi, made on the recommendation of members, recognizes research achievement and carries with it the opportunity to cooperate with other scientists in the activities of individual chapters.

American Scientist

Volume 73, No. 3, May–June 1985 ISSN 0003-0996

Published by Sigma Xi, The Scientific Research Society

Contents

Cover

Juvenile langurs play in a tree on Mount Abu, India. Young primates develop a versatile repertoire of behavior in play, and the primate order as a whole is characterized by adaptation through complex behavior. See the article by Alison Jolly on p. 230. (Photograph courtesy of Jim Moore/Anthro-Photo.)

What Is Artificial Intelligence Anyway?

Raymond Kurzweil

As the techniques of computing grow more sophisticated, machines are beginning to appear intelligent—but can they actually think?

Fans of Norbert Wiener will be familiar with the following German children's song:

> Knowest thou how many stars
>   stand in the blue tent of heaven?
> Knowest thou how many clouds
>   pass far over the whole world?
> The Lord God hath counted them,
>   that not one of the whole great
>   number be lacking (*1*).

Wiener was fond of reciting this song partly because it combined two of his favorite topics—astronomy and meteorology—but also because he enjoyed pointing out that experts were capable, in the mid-twentieth century, of accomplishing those functions. From Wiener's writings, one senses some pride that scientists could perform at least one of God's chores. We might, in turn, take pride that we have now turned over the job of keeping track of our stars and clouds to computers, which are probably doing a more thorough job than the scientists of Wiener's day, although perhaps not yet up to God's standards. We might also ask the question that if computers can now take over one of God's many chores, what about human chores? Indeed, what about our most prized chore—that of thinking?

Machines that think, or as a recent book title put it, machines who think, have been a topic of reflection and debate since Charles Babbage designed his "Analytical Engine" in the early nineteenth century and ruminated on the subject with his friend the Lady Ada Lovelace, daughter of Lord Byron (2). In those days there were few backers for such ambitious projects, and those few tended to be skeptical. In the last several years, however, the subject of thinking machines, or to put it more conservatively, machines that at least appear intelligent, has become of consuming interest to a widespread audience, including Wall Street and Madison Avenue.

Along with the increased attention has come increased controversy. Experts still debate about what constitutes an intelligent machine, and they still disagree about what constitutes the field of study that has been called artificial intelligence (or simply AI) since John McCarthy gave it that name at a Dartmouth conference in 1956. The uncertainty has always been there, but now there is more at stake. AI has been compared to bioengineering in having brought an academic discipline together with venture capital and public attention. Yet, while bioengineering has its own controversies, it lacks this one. Articles and talks do not start with the question, What is bioengineering anyway? If it has to do with splicing genes, no one doubts that it is bioengineering.

The controversy surrounding AI is evidenced by a certain amount of discord within the field. Academic researchers have accused some industrial AI development of being shallow and giving too high a priority to short-term commercial goals rather than long-term research. Industrial researchers have accused academic AI of providing superficial demonstrations, of not developing robust systems that really work. Companies have accused their competitors of not using real AI techniques, and so on.

Unfortunately, we are not likely to find published any time soon the definitive work that will lay this confusion to rest, once and for all. Since that will not be the result of this article, either, I will refrain from making such an assertion. But an examination of issues related to artificial intelligence may nonetheless help us to identify those questions that cannot be answered, as well as those that are worth answering. In the latter category, I would include an understanding of what the discipline of artificial intelligence is now capable of achieving, what it will be capable of achieving, and how we can best reach those goals.

I will approach these questions and the central question—What is artificial intelligence anyway?—from three perspectives: the past, the present, and the future.

AI past

Rather than recite the usual litany of early experiments, I will begin by examining instead some of the intellectual roots of the AI movement, which I feel will be more revealing. One is tempted to go back to mechanical calculating engines, which were proposed as early as the seventeenth century by Pascal and Leibniz. Perhaps the

Raymond Kurzweil is an inventor whose work centers on practical applications of pattern-recognition technology. In 1976 he introduced the world's first optical scanning system—the Kurzweil Reading Machine for the Blind—which scans printed documents in any typeface and converts them to synthetic speech. In 1984 one of his companies began manufacturing the Kurzweil 250 electronic keyboard—a digital instrument that uses AI techniques to recreate the complex sounds of the grand piano and other orchestral instruments. Kurzweil is currently leading a team in the development of a voice-activated word processor with a 10,000–15,000 word vocabulary. This article is adapted from Kurzweil's keynote address to the annual IEEE *International Conference on Computer Design in October 1984. Address: Kurzweil Applied Intelligence, 411 Waverley Oaks Road, Waltham, MA 02154.*

most famous was Babbage's analytical engine of 1833, the first machine designed to employ a stored program, which was to be read in from punched cards. Although the machine was never built, programs were written for it, and intense discussions by Babbage and Lovelace have been preserved which reflect on such issues as self-modifying code, programs that would play chess or compose music, and how (or whether) automated intelligence might be related to human thought.

While fascinating in retrospect, these early discussions did not immediately produce the ferment from which computers and artificial intelligence were to spring. These emerged instead from some of the more powerful intellectual movements of the early twentieth century. One of them was the philosophical movement called logical positivism, which strove to examine epistemology with the same rigor that was then coming into fashion in the world of mathematics. Epistemology is the study of the origins, methods, and limits of knowledge—what we can know and how we can know it. Although computers as such were not on the minds of the early logical positivists, it is not surprising that an attempt to define the nature of knowledge with mathematical precision would have relevance to the subsequent emergence of computers.

One work that was influential in the development of logical positivism was the *Tractatus Logico-Philosophicus* by Ludwig Wittgenstein, published in 1921 (*3*). The book is interesting both in what it says and in its own internal structure. The treatise contains only seven primary formal statements, numbered 1 through 7. To help us along, Wittgenstein also includes several levels of modifying statements to clarify the primary statements. For example, statements 1.1 and 1.2 modify statement 1. In turn, 1.1.1 is provided to help explain 1.1, and so on. The reader has the choice of reading the book from left to right, from the top down, or even in reverse. The modular structure would, I am sure, please proponents of good programming style.

Wittgenstein starts with statement 1, "The world is all that is the case," and ends with statement 7, "What we cannot speak about we must pass over in silence." As you might gather from these starting and finishing points, it is an ambitious work. The statements in between deal with a formal concept of language—what can be said—and make a connection between what can be said and what can be thought or known.

He states:

4.0.0.3.1 All philosophy is a "critique of language."
5.6 The limits of my language mean the limits of my world.
5.6.1 We cannot think what we cannot say.

In Wittgenstein's world, there are certain elementary facts, there are propositions about relations between elementary facts, and there are certain allowable transformations on such propositions that yield composite propositions. His model of human thought is that we can receive sense impressions which comprise elementary facts. We can then transform these elementary facts and derive relationships among them according to certain allowable logical processes. Any thought outside this scheme is either false or nonsensical.

While I have oversimplified the theory, I have done so to bring out two important points. Both have a bearing on the intellectual roots of artificial intelligence. The first—and Wittgenstein makes this point in several different ways—is that what we can think is what computes. He makes a direct link between human thought and a formal process that can be described only as computation. To reorder Wittgenstein's statements, we cannot think what we cannot say; we cannot say, or at least we ought not say, what is meaningless in the language we are speaking; and statements in any language are indeed meaningless unless they can be derived from a formal sequence of computation-like transformations on a database of elementary propositions.

This description of human thought as a formal sequence of computation would be restated two decades later in the Church-Turing thesis, which I will discuss shortly. It is not a thesis that everyone familiar with it necessarily accepts, and it remains controversial in the philosophical literature. Wittgenstein himself ended up rejecting it, and in his later works had a lot to say about subjects that he had argued in the *Tractatus* should be passed over in silence.

The second point made in the *Tractatus* which would have significance later to computational theorists was that thought is embedded in language. It is also interesting to note that language as conceived in the *Tractatus* has more of the quality of the programming language LISP or even PROLOG than it does of Wittgenstein's native German.

Although Wittgenstein ended up rejecting many of the ideas expressed in his first major work, his thoughts were elaborated into the formal theory of logical positivism by such philosophers as Alfred Ayer and Bertrand Russell. Russell also extended this new formalism in epistemology to the world of mathematics in the seminal work he wrote with Alfred North Whitehead, the *Principia Mathematica*, which represented the founding of modern set theory (*4*).

From the foundations of set theory came, in turn, a startling new theory by the mathematician Alan Turing, in 1937, which would lead to modern computational theory. In effect, Turing restated the assertion originally made by Wittgenstein, that true thought is computation, only this time he stated the idea explicitly as a formal definition of computation rather than of human thought.

One purpose of Turing's pioneering work was to address a problem that the *Principia* had failed to answer—the so-called twenty-third problem of the great German mathematician David Hilbert. This question, briefly stated, is the question of whether it is possible to devise a method that can establish the truth or falsity of any statement in a certain language of logic called the predicate calculus (*5*). In examining what Hilbert meant by the word "method," Turing came up with a formal definition of method as algorithm. He also devised an enduring concept of an algorithm as a program that could run on what has become known as a Turing machine.

The Turing machine has persisted as our primary theoretical model of computation because of its combination of simplicity and power. Its simplicity derives from its very short list of capabilities. It can read a tape and determine its next operation, based on whether it reads a zero or a one; it can move the tape left or right; it can write a zero or a one on the tape; it can jump to another command; and it can halt. As for its power, Turing was able to show that this extremely simple machine can compute anything that any machine can compute, no matter how complex.

There are two reasons why the Turing machine created the stir that it did. First is this astonishing combination of simplicity and power. Second, Turing discovered something unexpected—that there are well-defined problems for which we can prove that an answer exists but for which we can also prove that the answer can never be found. These are the so-called unsolvable problems.

The most famous of these, the Busy Beaver problem, was discovered by Tibor Rado (*6*). It may be stated as follows. Given a positive integer *N*, we construct all the Turing machines that have *N* states, which is to say *N* distinct internal configurations (this will always be a finite number); eliminate those that get into infinite loops; and then select the machine that writes the largest number of ones on its tape. The number of ones that this Turing machine writes is called the Busy Beaver of *N*. Rado showed that there is no algorithm, that is, no Turing machine, that can compute this function for all *N*s. The crux of the problem is sorting out those *N*-state Turing machines that get into infinite loops. If we program what is called a universal Turing machine to simulate all the *N*-state Turing machines, the simulator itself goes into an infinite loop. The Busy Beaver function can be computed for some *N*s, and it is also, interestingly, an unsolvable problem to separate those *N*s for which we can determine Busy Beaver of *N* from those for which we cannot.

Aside from its interest as an example of an unsolvable problem, the Busy Beaver function is also interesting in that it can be considered to be itself an intelligent function. More precisely stated, it is a function that requires increasing intelligence to compute for increasing arguments. As we increase *N*, the complexity of the processes needed to compute Busy Beaver of *N* increases. With *N* equal to 6, we are dealing with addition, and Busy Beaver of 6 equals 35. At 7 the Busy Beaver learns to multiply, and Busy Beaver of 7 equals 22,961. At 8 it learns to exponentiate, and the number of ones that our eighth Busy Beaver writes on its tape is approximately 10^{43}. By the time we get to 10, we are dealing with a process more complex than exponentiation, and to represent Busy Beaver of 10 we need an exotic notation in which we have a stack of exponents the height of which is determined by another stack of exponents, the height of which is determined by another stack of exponents, and so on. For the twelfth Busy Beaver, we need an even more exotic notation. It is likely that human intelligence is surpassed well before the Busy Beaver gets to 100.

Turing showed that there are as many unsolvable problems as solvable ones, the number of each being the lowest order of infinity, the so-called countable infinity.

Working independently during the 1930s, three mathematicians—Kurt Gödel, Alan Turing, and Alonzo Church—each showed that the answer to Hilbert's twenty-third question, originally posed in the year 1900, is no. There is no method or algorithm that can determine the truth or falsity of any logical statement in the predicate calculus, nor can we even sort out those statements that can be proved from those that cannot be. The work of these three mathematicians created reverberations still being felt today. Gödel's Incompleteness Theorem, for example, which showed that all formal systems of sufficient power are capable of generating propositions that cannot be decided at all, has been called the most important in all mathematics (*7*). Gödel, Turing, and Church's work represented the first formal proofs that there are definite limits to what logic, mathematics, and computation can do. These discoveries strongly contradict Wittgenstein's statement 6.5: "If a question can be framed, it can be answered."

In addition to finding some profound limits to the powers of computation, Church and Turing also advanced, independently, an assertion which has become known as the Church-Turing thesis—that if a problem that could be presented to a Turing machine is not solvable by a Turing machine, then it is also not solvable by human thought. Others have restated the Church-Turing thesis to propose an essential equivalence between what a human can think or know and what is computable. The Church-Turing thesis can be viewed as a restatement in somewhat more precise terms of one of Wittgenstein's primary theses in the *Tractatus*.

It should be pointed out that although the existence of unsolvable problems is a mathematical certainty, the Church-Turing thesis is not a mathematical proposition at all. It is a statement which in various disguises is at the heart of some of our most profound philosophical debates. It has both a negative and a positive side to it. The negative side is that problems which cannot be solved through any theoretical means of computation also cannot be solved by human thought. Accepting this thesis means that questions exist for which answers can be shown to exist but which can never be found. The positive side is that if humans can solve a problem or engage in some intelligent activity, then machines can ultimately be constructed to perform in the same way. This is a central thesis of the artificial intelligence movement—that machines can be made to perform intelligent functions, that intelligence is not the exclusive province of human thought. We can thus arrive at one possible definition of artificial intelligence—that AI represents attempts to provide practical demonstrations of the Church-Turing thesis.

In its strongest formulation, the Church-Turing thesis addresses issues of determinism and free will. Free will, which has been described as purposeful activity that is neither determined nor random, would appear to contradict the Church-Turing thesis. Nonetheless, the truth of the thesis is ultimately a matter of personal belief, and examples of intelligent behavior by machines are likely to influence one's belief in at least the positive side of the question. Lady Lovelace's skepticism regarding the possibility of intelligent machines was no doubt related to the limitations of the mechanical computer with whirling gears and levers that was proposed to her. Today it is possible to imagine building machines whose hardware rivals the complexity of the human brain. As our algorithms grow more sophisticated and machines at least appear to be more intelligent and more purposeful, discussions of the Church-Turing thesis will become more practical than the highly theoretical debate of Church and Turing's time.

Up through the late 1940s, the link between thought and computation was necessarily theoretical.

With the development of electronic computers during World War II, the discussion turned quickly to the reality of what might be done with the available hardware. The early AI efforts were enthusiastic, productive, and guilty, perhaps, of just a bit of "blue skying." For example, Herbert Simon and Allen Newell, in a paper published in 1958, say the following: "There are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in a visible future—the range of problems they can handle will be coextensive with the range to which the human mind has been applied" (*8*).

The paper goes on to predict that within ten years (that is, by 1968) a digital computer will be the world chess champion. Eight years later, in 1965, Simon wrote in another article that "machines will be capable, within twenty years, of doing any work that a man can do" (*9*).

Now I do not mean to pick on Simon. He has contributed as much as anyone to the substantial progress that has in fact been made, and he is far from alone in making such unfulfilled promises. My point is only that the AI field started with a romantic energy that enabled it to achieve some impressive intellectual accomplishments but at the same time caused a credibility problem, from which, to some extent, it still suffers.

The romanticism of early (as well as some current) work in AI is also reflected in a strong tendency to use anthropomorphic terms to describe its techniques. Any discussion of artificial intelligence is likely to include references to experts, expert managers, demons, communication through blackboards, learning, logical inference processes, and knowledge sources. Personally, I have mixed feelings about such terminology. On the one hand, I enjoy using it as much as anyone. It can also be argued that such terms are reasonably descriptive of the methods being labeled. They are certainly far more relevant to what they are describing than truth, charm, and strangeness are to several recently discovered phenomena in particle physics. The negative side of this tendency to anthropomorphize is the accusation, sometimes justified, that such terminology is vague and that its primary purpose is to make certain techniques appear more complex and mysterious than they really are. It can be pointed out, for example, that a logical inference process, as used in recent expert systems, is no more complicated an operation than the comparatively dull-sounding fast Fourier transform.

There have been attempts along the way to quell this perceived overenthusiasm (see, for example, ref. *10*). Perhaps, in those early years, an unbridled enthusiasm was more important than restraint. I believe there is a consensus today, however, that establishing realistic standards as well as expectations is essential if the field's enormous potential is to be realized. The real accomplishments of AI technology are now substantial enough that the field no longer needs to live almost entirely in the future.

AI present

As we shift our attention to the present, I would prefer again to avoid a roundup of the usual suspects—not that a recitation of the triumphs and frustrations of the AI world of today is of no interest, but it is material available elsewhere. Instead, I will try to share with you a personal view of the state of the art. To do this, I will cite six assertions that have been made in recent months, and state whether or not I agree with each. Because AI researchers such as myself are often accused of being vague, I will respond to each assertion with a short and precise answer. Having done that, I will succumb to the temptations of my training and also give a long and vague answer.

The first assertion is that "AI is LISP." My short answer is no.

LISP is a computer-programming language concerned with the evaluation of symbolic expressions. These expressions are comprised primarily of lists and functions. Lists in turn are defined as ordered sequences of items, which may be numbers, symbols, expressions, or other lists. By defining hierarchies of lists, very complex data structures may be created; these structures can be interactive.

Before I explain my negative short answer, let me speak in praise of LISP. First of all, LISP is an elegant and satisfying language which, reflecting its mathematical roots, gains a great deal of power through recursion. Recursion, as many will remember from elementary number theory, allows us to make an infinite number of assertions by proving a proposition true in one case, for $n = 0$, and then deriving the proposition for $n + 1$ from the proposition for n. Much of mathematics relies on recursion, and this technique allows simple LISP programs to perform relatively powerful transformations. Second, LISP is one of the few high-level languages that allows self-modifying code, which is another powerful concept in the theory of computation. Third, LISP allows highly flexible information structures that are amenable to representing concepts more complex than numbers. LISP has also changed over the years more than most languages. Many of these changes have added significantly to its sophistication and capability.

There are two reasons why I answered no. First, I feel there is an excessive reliance and emphasis on LISP. A recent market intelligence report for the venture capital community, in advising prospective investors, declared that if the software they were considering investing in was not written in LISP, then it was almost certainly not "true artificial intelligence." While most serious AI researchers, even ardent supporters of LISP, would find fault with that statement, it is not an uncommon sentiment.

My second reason is probably more salient and has to do with some of LISP's drawbacks. The same market report stated that LISP used to be inefficient and expensive but that this is no longer true. It is certainly true that the advent of lower-cost microcoding has brought down the cost of LISP machines. They have come down from high tens of thousands of dollars to low tens of thousands, and soon will be in the high thousands. As I will note shortly, however, the future of AI will depend heavily on parallel processing—the linking of many processors to perform many operations simultaneously—rather than step-by-step computing. Products that link a hundred small machines in parallel are practical and can be manufactured at a reasonable cost. No one would propose, however, manufacturing a product that included a hundred efficient LISP machines. Parallel processing was not a consideration when LISP was designed, and LISP is not particularly well suited for it. In my opinion, parallel processing will be more important to the future of AI than list processing.

There is no question that LISP will continue to be an important language in AI research and that it will continue to evolve. I do not believe, however, that it will continue to play the dominant role that it has to date.

The second assertion I would like to examine is: "AI is AI techniques." My short answer again is no.

The assertion appears to be a tautology, the type of meaningless statement that Wittgenstein advises us to pass over in silence. If we define some terms, however, it will begin to have meaning. By AI, I mean the art of creating machines that perform tasks considered to require intelligence when performed by humans. This is the definition that Minsky gave to AI in the mid-sixties. I have seen it repeated at least twenty times since, generally without attribution. By AI techniques, I mean what people generally refer to when they say "AI techniques"—that is, those techniques that the AI journals will accept articles on. It is the same set of techniques that have been worked on in AI research centers over the past five years, but not those that were researched prior to that time.

One problem with the second assertion, as my definition probably makes clear, is that the concept of AI techniques is somewhat arbitrary and vague. I suggest that we refer instead to inference engine techniques or concept associative techniques or high-level feature extraction techniques or expert management techniques rather than use the vague term "AI techniques." We would, at least, have a better idea of what is being referred to.

There is, however, a more important problem with the assertion. Regardless of any confusion about the meaning of terms, it has been my experience that the bulk of the technology applied successfully to AI problems is not concerned with such AI techniques at all, but rather with what we call domain-related techniques. For example, my colleagues and I are working on a machine that will recognize human speech with virtually no restrictions on vocabulary. While there are so-called AI techniques involved, such as expert management, directed search, context analysis, and others, the bulk of our technology is specific to the domain of inquiry. We draw it from linguistics, speech science, psychoacoustics, signal processing, information theory, human factors, computer architectures, very large scale integrated circuit design, and other fields.

Another of Minsky's early definitions of artificial intelligence was "a grab bag of tricks." I could not agree more. Each specific task that we associate with an intelligent machine will require a different set of techniques, with methods derived primarily from our understanding of each problem. We hear every now and then about "generalized perception algorithms" that can recognize any type of pattern, whether it be manifested in speech, printed characters, land terrain maps, or fingerprints. It turns out that such claims are absolutely correct—such algorithms do, in fact, recognize every type of pattern. Only they do all these tasks very poorly. To perform any of them well, with satisfactory rates of accuracy, requires a great deal of knowledge deeply embedded in the algorithms and specific to the domain of inquiry, so much so that this aspect of the technology far outweighs the generic AI techniques.

The third assertion is: "AI is parallel processing." My short answer is yes.

I answer yes not because an AI implementation must be parallel by definition, any more than it must use LISP, but because I believe that parallel architectures are the wave of the future, particularly for the more complex systems which we tend to consider as artificial intelligence. The brain, as we know, more than makes up for the inherently slower speed of nerve cells as compared to silicon with almost total parallelism, and no doubt with its superior algorithms.

Parallel processing can represent a much more effective use of resources than serial processing. Our complete reliance on serial processing up to now can be compared to a society in which work can be done only by one person at a time.

Another reason for using parallel processing is that applying sufficient brute force to a problem is often necessary to achieve the desired result. For example, the typical organization of an expert system consists of three components—a massive database relevant to the area of expertise; a set of rules as to how the database is to be searched, manipulated, and transformed; and a logical inference processor that can apply these rules to this organized base of knowledge. The bottleneck in such systems is not the memory cost of storing the information and rules but the real-time requirements of the inference operations. This can be overcome at reasonable cost through massively parallel architectures, which the next generation of expert systems will undoubtedly rely on.

An example from the beginning of AI concerns game playing. The essential algorithm for playing games like chess is fairly simple and well known: generate a tree branch, push down, and prune. Interestingly, during the thirty years that game-playing machines have been around, we have found that the most effective means of improving performance has been the brute force approach. Applying greater computation power has brought more improvement than has algorithmic sophistication. It is no coincidence that the best computer chess is now played by the supercomputers. The computer that does end up winning the world chess championship will undoubtedly be a highly parallel computer.

One approach to parallel processing is to wire a lot of microcomputers together and provide ways for them to communicate and coordinate their activities. Another, even more powerful, approach is to provide an array of what I would call subprocessors on a single chip that are dedicated to some particular task. A subprocessor is not a programmable computer but a dedicated circuit designed to implement a desired algorithm. Examples of the functions such chips could perform would include signal processing, logical inference processing, time warping, pattern matching, and image processing, as well as implementing the algorithms specific to some particular system. Using custom very large scale integration, it is possible to put a dozen or more subprocessors on a single chip. A product with 100 such chips thus provides the equivalent computing power of over 1,000 microprocessors with a foundry cost of less than $1,000.

This computing power is, of course, dedicated to a particular set of algorithms, which is why I consider system architecture to be as important as the algorithms themselves. A typical product architecture used to be a single computer with appropriate peripherals and software. Increasingly, we will see algorithms distributed instead throughout a complex and diversified parallel architecture.

The fourth assertion is: "AI is interdisciplinary." My short answer is yes.

This assertion may at first seem obvious. I feel, however, that the extent to which work in AI is necessarily interdisciplinary, and the challenge that this presents, is not fully appreciated. As I mentioned earlier, in my experience the domain-related techniques constitute a larger share of the technology used in successful AI systems than do the generic AI techniques. This has important implications for the type of effort required.

One implication is that a major part of the challenge in solving an AI problem involves organization and communication. Research teams spanning a half dozen or more distinct disciplines do not come together easily, and once assembled, there is the problem that the team members speak different technical languages.

One of the first persons actually to recognize this as a potential stumbling block to the development of intelligent machines was Norbert Wiener. His book *Cybernetics*, published in 1948, was a remarkably comprehensive look at the future of computing. In it he points out that no one since Leibniz has had "a full command of all the intellectual activity of his day." He goes on to state the inevitable result: "There are fields of scientific work ... which have been explored from the different sides of pure mathematics, statistics, electrical engineering and neurophysiology, in which every single notion receives a separate name from each group, and in which important work has been triplicated or quadruplicated, while still other important work is delayed by the unavailability in one field of results that may have already become classical in the next field" (*11*).

The fifth assertion is: "AI models human intelligence." The short answer is no.

It would be beneficial if it could, but we just do not know enough at present to gain a great deal from modeling human cognition. There are exceptions. In our work on speech recognition we have taken advantage of what is known about the auditory processing of the cochlea and the auditory nerve. We find that our speech recognition algorithms perform better if we attempt to model human auditory front-end processing. I still answered no, however, because this front-end processing represents only a small part of the overall processing involved in recognizing speech, whether by machine or by a person.

One of the reasons we know something about auditory front-end processing is that the cochleas and auditory nerves of test animals are relatively accessible. The bulk of cognitive processing takes place, however, out of reach of our probes. Not only are most brain cells deeply embedded in the brain, but we now realize that most of the processing takes place within the cells themselves through a series of complex chemical transformations. Progress in understanding the brain will continue, but this is not our most promising source of ideas for new AI algorithms.

My last assertion is the reverse of the previous one: "We can learn about human cognition from AI algorithms." My short answer is yes.

Before we can run experiments on humans to test alternative theories, we need to have theories to test. One of the best sources for those theories is techniques we have found to work in machines. The fact that an algorithm works in a machine does not prove that the same technique is used in the brain, but it does prove that this is one way that the brain could work, and provides a potential theory that could be subjected to neurophysiological testing.

AI future

If we shift our focus to the future, it is important for us to understand precisely what it is that we are on the verge of. Perhaps the best place to start would be to state what I do not expect to see in the near future. We are still far from Simon's second prediction, of being able to replicate by machine the vast range of human intellectual capability. It is doubtful, for example, that by the end of this century computers will be able to watch a movie and write a coherent review.

What we are on the threshold of is nonetheless of major significance. We are gaining the ability to apply sharply focused machine intelligence, or perhaps I should say narrowly focused intelligence, to a wide range of problems. I would like to emphasize the word "narrow," because the computers that will be created over the next decade are not going to be wide-ranging intellectuals. Instead, we will see a proliferation of systems with well-defined areas of expertise—systems that have a mastery of our knowledge about a particular class of diseases, or that have an ability to develop certain types of financial investment strategies, or that can help guide a complex negotiation.

Despite my emphasis on the word narrow, it is important not to underestimate the significance of automating this kind of expertise. Look at the impact computers are having today on almost all areas of endeavor, despite the fact that virtually all are idiot savants. Adding some well-focused intelligence to our computers' already well-demonstrated superiority in mastering vast amounts of information and conducting repetitive operations at high speeds without tiring will be a powerful combination. Our goal, in fact, should not be to copy human intelligence in the next generation of computers, but rather to concentrate on the unique strengths of machine intelligence, which for the foreseeable future will be quite different from the strengths of human intelligence.

At least as important as the emerging expert systems are the efforts to develop intelligent computer interfaces in such areas as speech recognition, the understanding of natural languages, and computer vision. The primary users of the next generation of intelligent machines are not intended to be computer experts but everyone, including children. There are extensive efforts, for example, to add some measure of intelligence to computer-assisted instruction systems. The next generation of teaching machines will attempt to evaluate students to determine their strengths and weaknesses, as well as their interests, and will provide instruction that is both entertaining and pedagogically sound for each individual.

To engage in some predicting of my own, I believe that by the end of this century AI will be as ubiquitous as personal computers are today. The majority of software will be intelligent, at least by today's standards. It will interact with users through intelligent front and back ends; highly organized databases will be embedded within the software, as well as available through telecommunications; and high-speed parallel computation engines will be able to manipulate the information as needed.

If we consider what it is that makes certain behavior appear to be intelligent, one observation we might make is that it combines two attributes. First, the behavior appears to be reasonably appropriate to the situation. Second, it is not totally predictable. Computers are now on the verge of satisfying both these conditions. By being tied into vast yet rapidly changing databases and by applying highly parallel processing to generate the needed inferences and deductions, the answers generated by such systems will indeed combine appropriateness with a measure of unpredictability, giving at least the impression of intelligent behavior.

To what extent we will consider such systems actually to be intelligent is hard to say. Our standards as to what constitutes artificial intelligence are constantly changing. In the fifties there was a great deal of excitement as computers began to play chess and checkers, albeit at a crude level. That was considered AI at the time, yet today we do not consider the $20 pocket-sized chess-playing machines, which play a much better game, to be examples of artificial intelligence. There was excitement in the late fifties and early sixties when AI programs were able to prove theorems and solve calculus problems. Today there are far more powerful packages to manipulate equations that are considered quite useful, but are not generally pointed to as examples of AI. Apparently, as we understand a process well enough, we begin to consider it just a rote technique and not an example of intelligence. As Minsky pointed out recently, if a superior being were to analyze human behavior and understand in great detail how we operate, it might not consider us to be very intelligent either (*12*).

Some observers have actually suggested that artificial intelligence is inherently on the moving edge of technical feasibility, that it should be defined as those computer science problems we have not yet solved.

References

1. Norbert Wiener. 1961. *Cybernetics; or, Control and Communication in the Animal and the Machine*, 2nd ed., p. 30. MIT Press.
2. Charles Babbage. 1973. *Babbage's Calculating Engines.* Charles Babbage Research Institute Reprint Series for the History of Computers, vol. 2. Los Angeles: Tomash Publishers.
3. Ludwig Wittgenstein. 1961. *Tractatus Logico-Philosophicus.* New trans. D. F. Pears and B. F. McGuinness. Routledge and Kegan Paul.
4. Alfred North Whitehead and Bertrand Russell. 1957. *Principia Mathematica*, vol. 1. Cambridge Univ. Press.
5. Douglas M. Campbell and John C. Higgins, eds. N. d. *Mathematics: People, Problems and Results.* Wadsworth International.
6. John M. Hopcroft. 1984. Turing machines. *Sci. Am.* 250 (May): 86–98.
7. Douglas R. Hofstadter. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books.
8. Herbert A. Simon and Allen Newell. 1958. Heuristic problem solving: The next advance in operations research. *Oper. Res.* 6: 7–8.
9. Herbert A. Simon. 1965. *The Shape of Automation for Men and Management*, p. 96. Harper & Row.
10. Marvin Minsky and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry.* MIT Press.
11. Ref. *1*, p. 2.
12. Marvin Minsky. 1984. Artificial intelligence. *A. I. Reporter*, April, p. 6.

ARTIFICIAL INTELLIGENCE ENTERS THE MAINSTREAM

Computer users who have been waiting for products based on the much-touted technology of artificial intelligence (AI) have so far seen only a trickle of initial offerings. But AI, which has been confined largely to R&D laboratories, is finally poised to move forcefully from the development to the delivery stage. Propelling this transition will be the mainstream computer vendors, who view AI as a pervasive technology that can be merged with existing products to make them more powerful and easier to use—in effect, "smarter."

AI is fundamentally a software technology that can be adapted to run on any type of computer—whether micro, mini, or mainframe—in conjunction with most conventional software. The manufacturers of these computers have therefore become active in developing and disseminating AI technology, not only to garner a portion of what promises to be a multibillion-dollar business, but also to ensure that their users have access to its benefits.

The definition and value of the AI market will become increasingly difficult to gauge, because the technology will often be buried within more conventional products. For example, a credit authorization package could incorporate an AI module to handle marginal cases now referred to human operators for resolution. An electronic mail system might add a component that intelligently sorts messages by priority and content. Process control software could "understand" the nature and operation of the equipment it directs. And much software could exploit the power of speech recognition and natural-language interfaces to permit users to interact with the computer in ordinary spoken English.

by Dwight B. Davis

This integration is at odds with the common perception that AI is synonymous with special-purpose computers called LISP machines and with "expert system" software (which emulates the specialized knowledge and reasoning ability of human experts). LISP machines and expert systems have typically operated in a world of their own, and the field of AI has tended to be associated more with trying to produce "thinking" computers than with enhancing traditional applications.

But developers' current focus on practical uses represents a bona fide metamorphosis from the early years of AI, says Richard P. Ten Dyke, assistant for business analysis, products, and technology at IBM's Information Systems & Storage Group (White Plains, N.Y.). Ten Dyke, who first explored AI at IBM in the early 1960s, notes that "instead of trying to recreate the human mind, the goal of AI has become much more focused on making computers more productive."

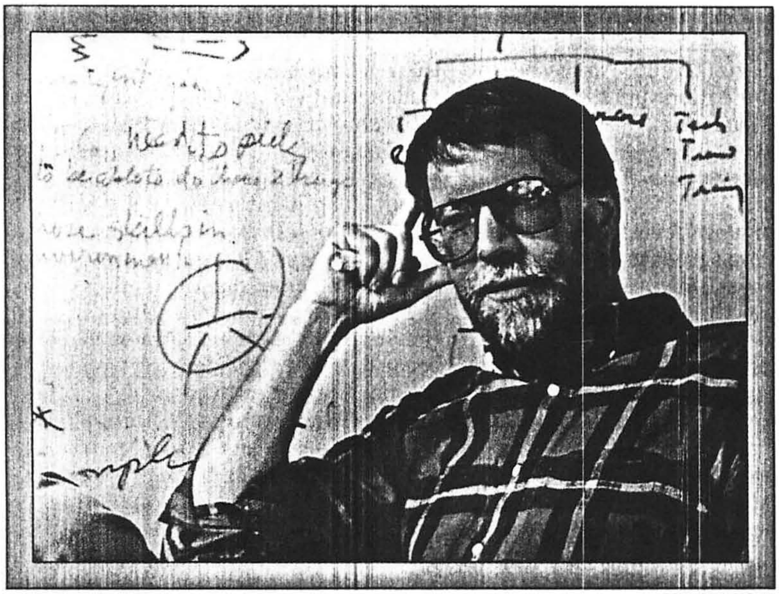
Computer vendors' role. To participate in the commercialization of AI, the major computer companies are pursuing several tacks: developing in-house AI projects to streamline their own operations and to gain experience with the technology; planning new generations of hardware products that run symbolic AI programs as well as conventional numeric applications; remarketing AI-based software programs developed by third parties; and, in some cases, planning AI software of their own.

The most renowned of the in-house projects is an expert system called XCON (short for "expert configurer"), developed jointly by researchers at Carnegie-Mellon University and Digital Equipment Corp. (Maynard, Mass.). DEC's VAX computer line consists of hundreds of components that can be configured in thousands of ways; XCON suggests the best configuration on the basis of each customer's requirements. By reducing the number of false orders for unneeded components, "XCON is saving us roughly $18–20 million per year in manufacturing costs," says Joel Magid, senior product manager of DEC's AI Technology Group.

The success of in-house AI development activities has often helped convince computer vendors' top management that AI will become an important part of the products they market to the outside world. Xerox's Palo Alto Research Center (PARC)—for years a leading AI research institution—has kept the company abreast of the technology and provided the basis of Xerox's LISP machine product line. But AI did not become one of Xerox's main strategic goals until in-house projects began to streamline the company's own internal operations and to assist in product development.

In one case, the use of AI support software—soon to be marketed as a product called Trillium—enabled Xerox designers to efficiently collaborate on the development of simple interfaces for a complex line of copiers. Thanks to that project and others, "we find that the company at large has a better understanding of what the AI game is all about, and is becoming more interested in aggressively supporting it," says John Seely Brown, manager of PARC's Intelligent Systems Laboratory.

Major computer vendors are helping to push "smart" systems out of the lab and into the marketplace



BRIAN TRAMONTANA

Xerox's Brown: "Along with streamlining our own operations, we hope to use AI to add intelligence to office automation."

Similarly, in-house AI activities at IBM helped convince the company's management to form an AI Projects Office about a year ago to coordinate the development of commercial AI products. IBM's primary AI focus will be on its traditional market: the management information systems community working with the company's mainframe computers. Its first products have been expert system development packages for mainframes that run the VM or MVS operating systems. Outside software suppliers, on the other hand, are targeting IBM's new PC/RT engineering workstation as the vehicle of choice for their AI products. In any case, stresses Ten Dyke, the AI products must mesh smoothly with existing software and databases. "Most of the people using AI applications in the commercial environment are going to want access to corporate data that already exist on their current systems," he says. "It would be silly to have to reintroduce those data into a stand-alone system to perform AI operations."

Customers of computer manufacturers that use AI internally may realize benefits beyond those of AI products alone. If the vendor succeeds in using AI to improve the efficiency of its operations, it may reduce the manufacturing time and cost of its products. In addition, such vendors will be able to pass their AI experience on to customers. Indeed, some observers believe the main role for hardware manufacturers in popularizing AI may be as service providers that introduce AI to their user bases and help them assimilate the technology. "It will take the computer companies to educate the world about artificial intelligence; it won't take them to develop the technology," says Alexander D. Jacobson, president of Inference (Los Angeles), the independent supplier of a leading expert system development tool.

The LISP environment. In-house AI expertise at mainstream computer firms may have another important benefit: helping the computer manufacturers learn how to build machines that better run AI programs in conjunction with conventional software. It's no mystery how to build hardware dedicated to AI software alone; several companies have such computers on the market already. But they were all built specifically to run LISP—the programming language that has become the lingua franca of AI scientists in the United States. They tend to be expensive and special-purpose.

The four symbolic processing vendors—Symbolics (Cambridge, Mass.), LISP Machine Inc. (Andover, Mass.), Xerox, and Texas Instruments (Dallas)—are currently trying to broaden their products' usefulness in two ways. They are making their equipment compatible with industry standards such as the UNIX operating system and the IBM personal computing and networking architectures, and they are introducing new low-cost models of their symbolic processors to serve as delivery vehicles for products developed on their more powerful machines.

Xerox was the first to introduce inexpensive LISP machines, and still has the low end of the market much to itself. Its 1185 workstation costs as little as $9995 and can interface to the IBM PC. Market leader Symbolics has jumped into the delivery game as well, albeit with a higher-priced product. In quantities of 10, its new 3610AE machine costs $39,600—relatively expensive, but nevertheless a considerable drop from the $100,000–$200,000 price of its top-of-the-line 3670 family of development products.

The conventional computer vendors have been introducing LISP compilers on their machines as a first step toward being able to support AI programs. This step has been simplified by the establishment of a standard version of LISP called Common LISP (many "dialects" of LISP had evolved over the years) and by the formation of companies such as Lucid (Palo Alto, Cal.), Gold Hill Computers (Cambridge), and Franz (Alameda, Cal.), which have written LISP compilers for most of the popular computers. One of the last holdouts is Xerox, but it plans to support Common LISP on its machines soon. IBM also has yet to introduce such a compiler, but Ten Dyke says the company believes that "Common LISP is a requirement for the future."

By running LISP compilers, the general-purpose computers have begun to compete with LISP machines. "The fallacy is that a conventional computer system cannot run LISP as well as a LISP machine does," says Nelson Hazeltine, director of systems environment architecture and advanced software technologies at NCR (Dayton, Ohio). "We've found that there's no performance difference." The LISP machine manufacturers contest such claims, however, asserting that benchmark tests, which measure a computer's speed in running a LISP program, fail to measure the full value of their products. "LISP machines give you a total LISP environment, containing all the features that improve your productivity and your ability to generate code reliably and quickly," says Wally Rhines, president of TI's Data Systems Group. "Those qualities are hard to measure in standard benchmarks."

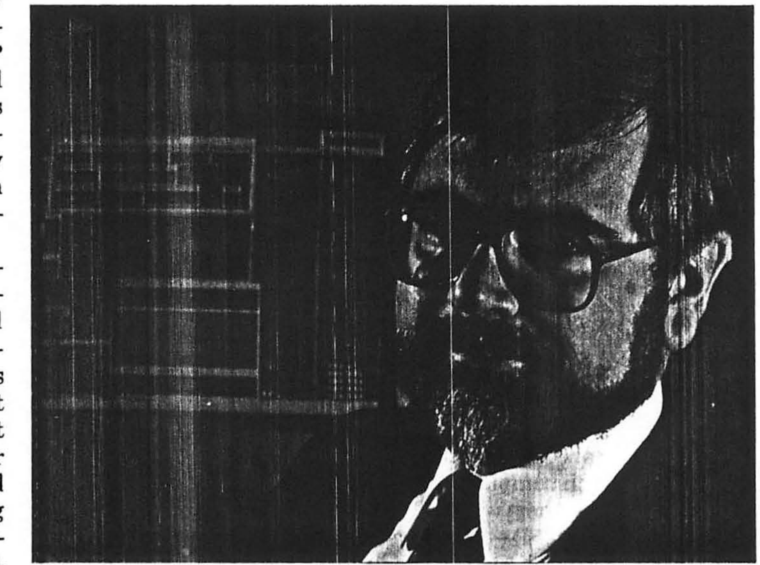
Increasingly, however, conventional computers do have access to such LISP environments. Several years ago, a few start-up companies began introducing software designed specifically for the construction of expert systems, which by then had become the focus of most AI activity. These software packages, known collectively as expert system "tool kits" or "shells," were written in LISP and initially designed to run on LISP machines. But in recognition of market opportunities, the tool kit vendors have more recently been rewriting their products to run with the LISP compilers on conventional machines. Some speed is lost—LISP machine implementations are generally more efficient—but a broad new spectrum of users is gained.

Many in the industry believe the widespread dissemination of powerful programming environments will be one of AI's most lasting legacies. At large corporations, most software is developed or customized in-house, and even with programming tools such as fourth-generation languages and COBOL generators (HIGH TECHNOLOGY, April 1986, p. 38), the job can be long and arduous. "By the time the software is developed, the user's requirements have often changed," says Inference's Jacobson.

The AI programming environments speed this process, permitting the rapid prototyping of new software systems. And the development benefits are not limited to AI applications. "You can use this AI programming methodology to tackle effectively a broad class of problems, producing solutions that are not in their own right classified as intelligent," says Brown at Xerox.

To further expand the audience for their products, and to speed up running time, the leading tool kit vendors are introducing new versions of their products written in C, the language of the Unix operating system. C is not really a symbolic processing language, but with certain extensions it can be used to write AI programs that perform the same functions as their LISP counterparts. "The optimal solution," says Larry K. Geisel, president of the tool kit vendor Carnegie Group, "may be to develop AI applications in LISP and port them to C for execution."

Hybrid systems. Even as developers move to fast-running C as a language for AI delivery, the pressure remains great to develop more powerful general-purpose computers capable of blending AI smoothly into their software repertoire. "You want machines that can do symbolic computing, but not at the expense of conventional computing," says Ira Goldstein, director of the Distributed Computing Center at Hewlett-Packard Laboratories (Palo Alto).



Larry Walker, director of Sperry's Knowledge Systems Center, views AI as both a mandatory technology and an unparalleled opportunity.

Goldstein says that HP's forthcoming Spectrum line is a step in the right direction. The new computers will have a large address space—a key requirement for knowledge-intensive AI programs—as well as a large number of storage registers, which can hold functions commonly used by the LISP language. Perhaps most important, the Spectrum computers are designed to support coprocessors of different types. Many observers believe that the hybrid machine of the future will have both general-purpose and LISP microprocessors working together to run (respectively) the numeric and. symbolic portions of mixed applications.

Texas Instruments is a leading advocate of supporting a general-purpose microprocessor with a LISP chip—not surprising, since the company expects to produce the industry's first such chip this fall. The chip, which will contain approximately 60% of the circuits found in TI's full-blown Explorer LISP machine, is being developed as part of a $6 million DARPA contract to produce a "Compact LISP Machine" no larger than a shoebox. Once it has met its contractual obligations, TI can use the LISP chip in whatever way it wishes, bundling it within its own computers or selling it on the open market. "We might keep it internal for a while; we might also sell it to the workstation vendors," says George Heilmeier, senior VP and chief technical officer.

The vendors of engineering workstations are a likely market for such a chip because two of the leading vendors—Apollo Computer (Cambridge, Mass.) and Sun Microsystems (Mountain View, Cal.)—have already signed marketing and development agreements with Texas Instruments in order to link their products to TI's Explorer machines. Under these agreements, the workstation vendors will work with TI to integrate Explorers into the workstation networks, where they will serve as development systems and as "knowledge servers," running AI applications that users can access over the networks. In these initial links between symbolic and numeric processing, the "hybrid" environment is being produced by coupling full-blown computers of each type via a network, not through the use of coprocessor chips within a single machine. "We're looking at AI as a systems approach, rather than as a solution-in-a-box approach," says Paul Armstrong, project manager of AI and data management at Apollo.

Armstrong agrees, however, that LISP chips might eventually work their way into workstations. Already, workstations are emerging in many circles as the preferred type of AI computer. They cost considerably less than most LISP machines, incorporate the same kind of powerful graphics displays, and can provide respectable performance, thanks to the power of general-purpose microprocessors such as the Motorola 68020 and the Intel 80386. The entry of IBM, with its PC/RT, and DEC, with its MicroVAX II, into the workstation market has attracted even more interest from the AI community. Most expert system tool kit vendors, for example, have already ported their products to these machines or plan to do so soon. DEC even markets a version of its MicroVAX II bundled with Common LISP and a graphics interface as the "AI VAXstation."

A sampling of AI activity at computer companies

Apollo

AI products: Common LISP; TI's Explorer LISP machines with links to Apollo's Domain workstations; various expert system tool kits.
Organization: Created an AI project manager position a year and a half ago. The company's internal AI work consists primarily of ensuring that third-party AI products are compatible with its own.
Comments: Positioning its Domain workstations as AI delivery vehicles and medium-power development machines; expects the Explorer line to handle the development and running of extremely complex software. Considers AI a critical technology, as important as graphics.

Data General

AI products: Common LISP; negotiating marketing agreements with expert system tool kit vendors.
Organization: Created an AI Business Unit, preparing its first strategic recommendations. Has several internal AI projects under way.
Comments: Expects the AI terminals of choice to be IBM PC/ATs and compatibles in commercial markets, engineering workstations in technical markets. Plans to be "pragmatic," providing its customers with the AI they want, regardless of source.

Digital Equipment

AI products: Common LISP, OPS-5, and PROLOG languages; third-party expert system tool kits; AI VAXstation.
Organization: Has AI technical development and marketing groups. More than 300 employees are working on 40-plus internal AI projects. Member of the Microelectronics and Computer Technology Corp. (MCC) consortium.
Comments: Has a long AI history, including close collaboration with leading universities. Expects to develop its own AI software for horizontal applications such as office automation. Plans to leverage its experience to serve as an AI consultant to its customers.

Hewlett-Packard

AI products: Common LISP on HP 9000 workstations; third-party expert system tool kits.
Organization: HP Labs focuses on AI research with a 3–7-year leadtime; the R&D labs within each operating division work on projects with faster payoffs. Developing several AI packages for internal use.
Comments: Will rely on third-party suppliers for AI, except in areas such as instrumentation, where the company may develop its own AI products. Believes the software productivity benefits of AI will be extremely important throughout the company.

IBM

AI products: Expert System Environment/MVS and ESE/VM tool kit products for its mainframes; remarkets the Intellect natural-language processing interface.
Organization: Established AI Project Office about a year ago. Has various AI R&D projects scattered throughout the company. Its Yorktown research labs have conducted AI studies for more than 20 years.
Comments: Considers AI to be one of several important technologies, but not a revolutionary one. Expects AI to find application across its product lines, which is why the company formed the corporate-level Projects Office rather than an independent business unit (as was done for the PC).

NCR

AI products: Common LISP; plans to resell third-party expert system tool kits.
Organization: AI falls under the Advanced Software Technologies group, which acts as an information clearinghouse and helps fund AI research within various company business units. Has over 20 in-house projects under way. Member of MCC.
Comments: Believes both AI development and delivery should be done on general-purpose computers to achieve the best integration of the technology. Aiming to make products useful to software engineers.

Prime

AI products: Plans Common LISP release; evaluating third-party expert system tool kits.
Organization: No central AI group. Research is dispersed throughout the company. Funds some AI research at two British universities.
Comments: Doesn't plan to pioneer AI; rather, will work with outside suppliers to integrate the technology with its products. A key area of development will be knowledge bases for CAD/CAM systems.

Sperry

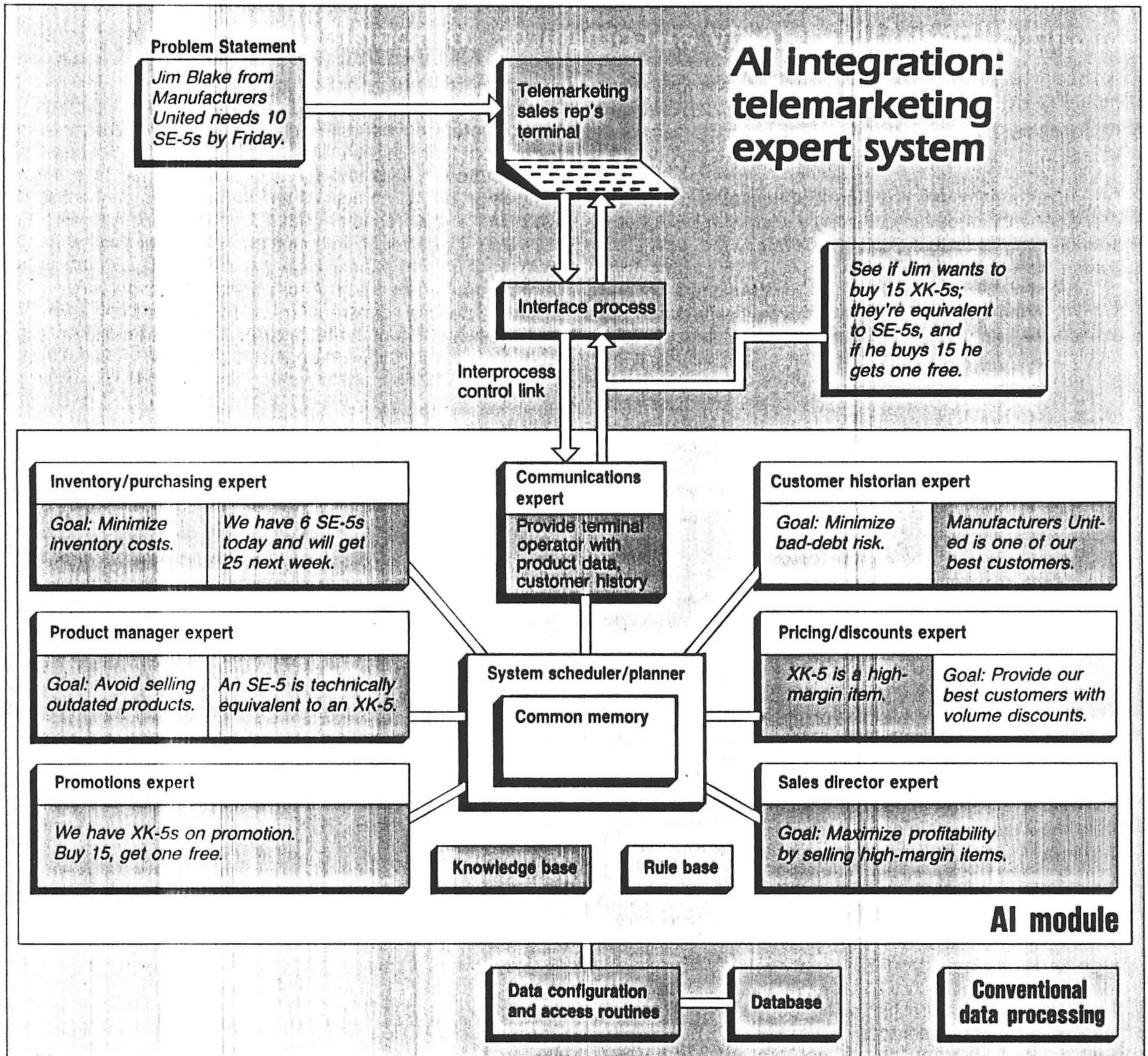
AI products: Sells a bundled AI development package consisting of TI's Explorer LISP machines and IntelliCorp's KEE expert system tool kit.
Organization: AI under the control of the 20-person Knowledge Systems Center. More than 200 people working on 50-plus internal projects. Spent about $30 million on AI R&D last year. Member of MCC.
Comments: Views AI as strategic technology of the 1990s, and has embraced it as the best way to distinguish itself from competitors. Plans to develop its own horizontal application products and to serve as a consultant to customers.

Texas Instruments

AI products: Explorer LISP machines; Personal Consultant Plus expert system tool kit; remarkets third-party tool kits; developing LISP chip.
Organization: AI R&D has been conducted at its Central Research Labs since 1978. Most products marketed through the Data Systems Group. Developing and operating numerous internal AI systems.
Comments: Hopes to use AI to leverage itself into the forefront of mini/microcomputer vendors. Considers AI second only to VLSI in strategic importance. Has worked harder than any other computer company to identify itself with the technology; stages nationwide AI satellite symposia.

Xerox

AI products: The 1100 series of LISP machines running Interlisp D; Common LISP due soon; NoteCards-D, an "idea-processing" software package.
Organization: Advanced R&D performed at its Palo Alto Research Center, including the study of both human and artificial intelligence. Working to ensure close cooperation with the company's AI systems business unit.
Comments: Performing what is probably the most theoretical AI research among all the computer companies. Focusing its efforts on document processing, including software that facilitates the "social intelligence" involved in collaborations. Plans to reverse the "PARC syndrome," under which some innovations have never reached the market.



AI integration: telemarketing expert system

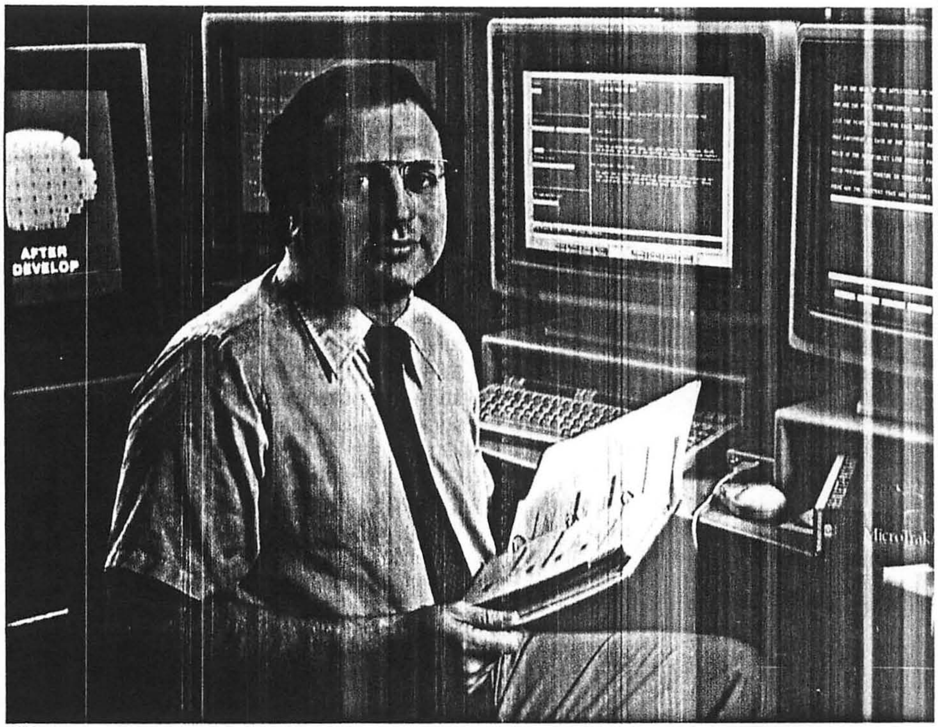
Problem Statement
Jim Blake from Manufacturers United needs 10 SE-5s by Friday.
Telemarketing sales rep's terminal
Interface process
Interprocess control link
See if Jim wants to buy 15 XK-5s; they're equivalent to SE-5s, and if he buys 15 he gets one free.
Inventory/purchasing expert
Goal: Minimize inventory costs.
We have 6 SE-5s today and will get 25 next week.
Communications expert
Provide terminal operator with product data, customer history
Customer historian expert
Goal: Minimize bad-debt risk.
Manufacturers United is one of our best customers.
Product manager expert
Goal: Avoid selling outdated products.
An SE-5 is technically equivalent to an XK-5.
System scheduler/planner
Common memory
Pricing/discounts expert
XK-5 is a high-margin item.
Goal: Provide our best customers with volume discounts.
Promotions expert
We have XK-5s on promotion. Buy 15, get one free.
Knowledge base
Rule base
Sales director expert
Goal: Maximize profitability by selling high-margin items.
AI module
Data configuration and access routines
Database
Conventional data processing

The TOLAS-Telestream system, developed by Carnegie Group for CSI-Transcomm (Pittsburgh), shows how AI components can be integrated with an existing data processing application. The AI module contains six expert systems that perform functions previously handled by human specialists, plus a communications expert to manage the exchange of information. When a customer calls in an order, it enters the system as a "problem statement." Each expert system, under the arbitration of a system scheduler/planner, tries to solve the problem with its own special knowledge. Most have goals, which are based on rules from the rule base, knowledge from the knowledge base, and facts stored in the conventional database. For example, realizing the goal of not selling to credit-risk customers requires that the caller's rating be determined, on the basis of conventional data such as annual sales. Once the AI module has arrived at a solution that meets both the customer's needs and the vendor's goals, the solution is sent to the terminal screen to be relayed to the customer.

The growing power of the general-purpose microcomputers embedded in such workstations has some manufacturers questioning the need for specialized LISP hardware. If conventional chips get fast enough, the argument goes, they will be able to adequately run both numeric and symbolic processing tasks at a price far below that of special-purpose devices. Digital Equipment's Magid points out that the MicroVAX II, which costs less than $50,000, is roughly the same speed as a VAX-11/780, which sold for more than $300,000 in 1978. "I think in a short period of time you will see the MicroVAX as a $10,000-$15,000 machine," he says. "There's a point at which the LISP machine argument holds, and a point at which it breaks."

But some maintain that the argument will hold indefinitely—that there will always be a role for LISP machines in developing and running large and complex AI programs. "DEC is saying that its standard microprocessors are getting to be powerful enough that you can do significant AI work on them. No one would quarrel with that," says Xerox's Brown. "But you might still get another fivefold improvement by going to specialized architectures." Such an improvement could be critical, says Inference's Jacobson, because "artificial intelligence is an extremely 'cycle-hungry' [i.e., power-demanding] technology. There is never going to be a chip powerful enough that you can afford to throw away cycles."

CAROL LEE



RON KIMBALL

HP's Goldstein: "Where we have special needs that aren't ideally served by third-party vendors, we may choose to build our own AI products."

DEC's Magid: "AI alone won't solve everybody's problems, but it will be integrated with other technologies to produce useful new products."

LISP machine evolution

LISP (short for "list programming") became popular in the AI community because it was designed to process—and form associations between—symbols such as words. In this it is different from most other programming languages, which are suited to performing primarily numeric calculations. Most languages can be used to do some sort of symbolic processing, but LISP and PROLOG (short for "programming in logic") are far more efficient at this task than numeric-oriented languages such as Cobol.

Even with a symbolic programming language, however, AI researchers were pushing the limits of software and hardware technology from the beginning. AI programs are notorious for requiring huge amounts of memory and machine cycles, especially during development. To speed the development process, researchers gradually built up an "environment" of programming tools such as program debuggers and sophisticated graphics interfaces. The resulting LISP environment far surpassed those of the numeric-type programming languages.

Still, software tools on their own weren't enough of a solution for AI development, and in the mid-1970s projects were independently begun at MIT and Xerox's Palo Alto Research Center to build hardware specifically matched to LISP and its development environment. The MIT project—funded by grants from the Defense Advanced Research Projects Agency (DARPA)—resulted in a machine called CADR (after a LISP primitive function), which eventually formed the basis of the first commercial LISP machines, introduced by Symbolics and LMI in the early 1980s. TI also sells a LISP machine whose roots can be traced to the CADR. Xerox, pursuing its own design program, also introduced its first LISP machine in the early 1980s.

The availability of commercial computers designed specifically to exploit the LISP development environment brought artificial intelligence a step closer to the consumer market, but the early machines were expensive and targeted only toward laboratory researchers. New, less expensive models have been introduced, however, in an attempt by the LISP machine vendors to garner a piece of the lucrative end-user delivery market.

Software skepticism. No one doubts that the major computer vendors will play an important role in disseminating AI products and building suitable hardware to run hybrid applications. But when vendors such as TI and DEC discuss plans to develop commercial AI software, rather than simply resell the products of third parties, skeptics begin to materialize. "The hardware companies have never done well in software," says Jacobson, who thinks that situation is unlikely to change. Even "system software" such as operating systems, language compilers, and programming tools is developed largely by outside parties, notes Larry R. Harris, president of Artificial Intelligence Corp. (Waltham, Mass.).

Some of the major computer firms actually concur with these opinions. "You shouldn't expect to see fundamental AI research coming out of Prime," says Richard H. Mott, director of AI within the company's CAD/CAM and Workstations Group (Natick, Mass.). "We will strike suitable relationships with external AI companies, so there will be no need for us to recreate products." Similarly, Data General (Westboro, Mass.) is formulating agreements to remarket various expert system tool kits on its computers, says Peter Jessel, director of the company's AI Business Unit. "It doesn't take a hell of a lot to be a major player in the game, other than the commitment to sign the appropriate third-party vendors and to start installing their AI products," he says. In

BUSINESS OUTLOOK

Expert systems and AI hardware reach commercial markets

As mainstream computer vendors begin to enter the artificial intelligence (AI) arena, the technology's pioneers—relatively small companies, in general—are following a number of strategies to survive and grow. In essence, they are taking steps to make products more appealing to traditional computer users in a wide variety of business environments.

AI products fall into several market segments, including specialized hardware, programming languages, expert systems, natural-language software, voice recognition, and artificial vision systems. These segments should make up a total market of $1 billion this year, rising to $4.2 billion in 1990, according to DM Data (Scottsdale, Ariz.), a market research firm that follows the industry. Two of the most active segments are AI hardware and expert systems, which account for 49% and 13% of the current market.

Symbolics (Concord, Mass.) dominates the production of symbolic processing computers—hardware optimized for creating AI software based on the LISP programming language. With 58% of the market for such dedicated computers, Symbolics is trailed by LISP Machine Inc. (Andover, Mass.), Texas Instruments (Dallas), and Xerox (Pasadena, Cal.), which share another 34% of the LISP machine market almost equally. LISP machines themselves, however, constitute only $200 million of the current $510 million market for all computers used in AI processing. The rest of this market consists of general-purpose machines that run AI software written in languages more widely used than LISP.

To expand their commercial opportunities, these first-generation AI companies have been adapting their products to conventional computing systems and traditional end users. LISP machine manufacturers, for instance, are developing networking links between symbolic processing computers and standard minis and mainframes. They are also offering low-cost machines that can compete with relatively inexpensive engineering workstations for running AI software.

At the same time, these firms are opening up niches for their products in specialized fields outside AI. For example, Symbolics is developing marketing alliances with companies that will package Symbolics' LISP machines with software that may or may not be related to AI. The first such company to sign up has been Icad (Woburn, Mass.), which produces modeling software used by contract engineering companies for the custom design of large industrial machines; Icad considers a LISP operating environment the most appropriate for its software. "By making such use of applications-oriented software firms, Symbolics can move its product out of the initial round of hard-core AI customers," says Carol Weiszmann, editor of *AI Markets* (Natick, Mass.).

> ***"To reach mainstream markets, AI vendors must make their products more accessible to end users and relevant to the computing environments in which they operate."***
>
> ***Carol Weiszmann, Editor***
> **AI Markets**

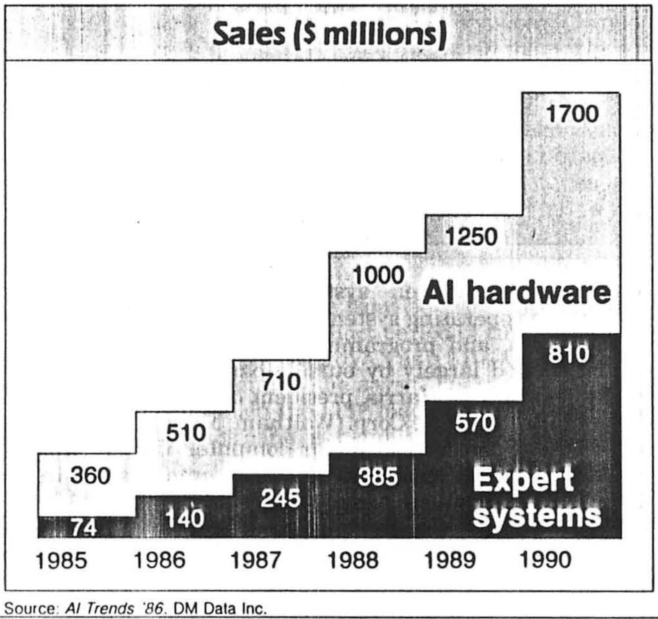
Expert systems—programs that codify and manipulate the knowledge of human specialists—have traditionally been developed in-house, and from the ground up, by large companies for their own use. But expert system development can be simplified and speeded up by a wider range of users with the aid of "shell" or "tool kit" systems, commercially available products that contain generalized rules of logic and support software.

The leading tool kit vendors are IntelliCorp (Mountain View, Cal.), Teknowledge (Palo Alto, Cal.), Carnegie Group (Pittsburgh), and Inference (Los Angeles); together, these firms control over two-thirds of the $70 million 1986 market for commercial tool kits. An additional $70 million is earned by companies performing government expert system contract work or offering custom services.

"Most AI development systems available are oriented to the several thousand computer scientists who have worked in LISP environments," says Larry K. Geisel, president of Carnegie. He believes that tool kits must be offered on two other levels to reach a wider market. One level consists of novice AI programmers for whom existing or somewhat redesigned tool kit software written in the C language is more appropriate. C is less powerful but more flexible and familiar than LISP, and can be used on Unix-based workstations, which run C faster than LISP.

Geisel says that the second level comprises the much larger group of conventional programmers who have had little or no exposure to AI. To tap this market, Carnegie is working on C-based shells embedded with inference strategies (such as when to use forward-chain reasoning) and other design considerations that AI programmers would prefer to handle themselves. Such shells also contain some of the knowledge needed to solve particular classes of problems, such as simulations and product configurations. "With this kind of product," says Geisel, "companies won't have to retrain whole cadres of programmers to use AI."

—Dennis Livingston



Source: *AI Trends '86*, DM Data Inc.

Japanese firms pursue their own AI agenda

Much of the current artificial intelligence activity at U.S. computer manufacturers can be directly linked to Japan's 1982 announcement of its Fifth Generation project. The 10-year project, directed by the Institute for New Generation Computer Technology (ICOT), has set ambitious goals to develop AI software and related hardware. These goals include the creation of knowledge bases that hold up to 100 billion bytes of information and 20,000 expert system rules.

ICOT is organized under the powerful Ministry of International Trade and Industry (MITI) and supported by eight Japanese companies—including Fujitsu, NEC, and Hitachi, the three largest computer manufacturers. It exercises an important influence on the direction of Japan's AI research and has provoked responses throughout the computer community in the U.S., most visibly in the formation of the Microelectronics and Computer Technology consortium (Austin, Tex.). Nevertheless, ICOT by no means controls all the Japanese AI research. Some corporate R&D projects in Japan are actually pursuing alternatives to the Fifth Generation's path. For example, while ICOT has embraced PROLOG as the foundation language for its AI efforts, many researchers at NEC are working with LISP.

In general, however, Japanese firms support ICOT without being overly reliant upon it. "We cooperate with ICOT, and if their work proves feasible we will acquire the technology," says Makoto Amamiua, director of information sciences at the NTT Research Laboratories. "But we don't give ICOT access to our proprietary research."

If any one theme is held in common by both ICOT and the computer companies, it is that AI software should be developed in close coordination with new types of supporting hardware. This pairing goes beyond the matching of LISP with LISP machines in the U.S., although there are some similarities. The Japanese are trying to build hardware "inference engines" to speed the operation of their AI software, as well as database management machines to handle huge storehouses of facts.

In each case, researchers are developing parallel computers—in which many processors tackle a problem simultaneously—to provide performance several orders of magnitude beyond today's machines. "Parallel architectures have been shown to be very good for logic-programming languages such as PROLOG," says Yoshihisa Ogawa, chief of ICOT's Research Planning Section. When PROLOG searches for information, it follows several branches of a treelike hierarchy at the same time; a parallel computer could closely match this software mechanism.

Each of the Japanese computer companies claims to be working on the whole software/hardware milieu. Mitsubishi Electric, for example, is investigating highly parallel architectures for superspeed machines, and is bringing to market a sequential inference machine. The latter, similar in concept to a LISP machine, was developed in conjunction with ICOT to run PROLOG. The company's software efforts, meanwhile, include projects in intelligent database management systems, natural-language processing—including Japanese/English translation systems—and knowledge representation studies.

Through their software projects, the companies intend to disprove the common belief that the Japanese, while good engineers and hardware architects, are poor software developers. For example, Computer Service Corp. has formed an entire branch, the CSK Research Institute (CRI), to perform basic AI research and to develop and market AI software products. CRI has also shown a willingness to market existing products; one of its first was the Knowledge Engineering Environment expert system tool kit from IntelliCorp (Mountain View, Cal.).

"Over 300 companies are working with us in one capacity or another," says Koji Yada, president of CRI, "which shows how many Japanese firms are interested in artificial intelligence." CRI is developing software for existing computers, such as the Digital Equipment VAX line, as well as for the computers coming from the ICOT project. CRI may eventually offer some packaged software, but off-the-shelf applications "are not the method used in Japan," says Yada. "We will work jointly with our customers to develop custom artificial intelligence packages."

the end, Jessel believes, customers will commit to companies like Data General, DEC, and IBM not so much because of their AI expertise as because they represent full-service vendors that have proved themselves capable of meeting their customers' needs.

To help the computer companies satisfy customers, independent firms known as value-added resellers (VARs) have traditionally bridged the gap between the manufacturers' products and the users' application requirements. But VARs specializing in AI have been slow to develop. "Vendors like TI and DEC really have to cultivate those firms in order to spread AI technology," says Kenneth R. Soneclar, VP of research at market research firm New Science Associates (South Norwalk, Conn.).

Still, mainstream computer companies are under increasing pressure to diversify into application software. Profit margins have dropped as computers have become more like commodities, says Susan Messenheimer, president of AIM Publications (Natick, Mass.), which publishes the *Artificial Intelligence Markets* newsletter. "The large companies are not just going to hand over the very lucrative AI software market to anyone else," she says.

Sperry, for example, is spending $30 million on AI R&D a year and has more than 200 people working on over 50 internal projects. It is remarketing an AI development package consisting of TI's Explorer LISP machine and IntelliCorp's KEE expert system tool kit, but it plans to produce its own AI-based horizontal application products as well.

Texas Instruments has produced expert system development tools, and may do packaged applications in the future, according to Heilmeier. But the most important AI software area that TI plans to exploit is what he calls "late-binding" applications, "in which most of the work is generic to several different applications, and there's customization at the final step that makes them unique to a specific customer," analogous to gate arrays in the integrated circuit domain.

Regardless of whether the computer companies actually make modifiable software products, such packages are considered crucial. This is especially true for large corporate customers, which often rely heavily on their own data processing departments or on third-party software houses to tailor applications to their employees' needs. In this customization process it is extremely important to get both end users and computer companies involved. "If you fail to do this," says IBM's Ten Dyke, "you run a very high risk of offering them something that does not solve their problems." □

Dwight B. Davis, a senior editor of HIGH TECHNOLOGY, *is coauthor of* Artificial Intelligence Enters the Marketplace *(Bantam Books, 1986).*

For further information see **RESOURCES,** *p. 68.*

EXPERT SYSTEMS STRATEGIES

The Monthly Newsletter for Managers and Developers of Expert Systems, from Cutter Information Corp.

Editor: Paul Harmon

Vol. 2, No. 8

August, 1986

AAAI-86

This issue marks our first anniversary, and for it we have produced a special expanded issue. You will find included an inventory of all of the fielded expert systems applications in commercial use that we have been able to document, which should help answer two questions: (1) What expert systems are actually being used? and (2) What sorts of tasks are expert systems being used to accomplish?

The second article lists all of the expert systems-building tools that we know about to provide you with a good checklist of the current tool vendors.

Finally, we preview two new tools and one new hardware product that should attract a lot of attention at this year's convention. There may be other tools that will be equally exciting, but we expect these products will be among the most talked about. Paperback Software's VP-Expert and Gold Hill Computer's Acorn will make major contributions to the continuing evolution of the expert systems market by offering developers more power for less money. Gold Hill's 386 HummingBoard will also contribute to the market's evolution by accelerating the trend toward the development and delivery of commercial expert systems on PC-based hardware. Detailed reviews of each tool will appear in future issues.

We anticipate an exciting convention in Philadelphia and hope this special issue will add to your enjoyment by providing you with a concise summary of the current expert systems applications and tools.

APPLICATIONS

INVENTORY AND ANALYSIS OF EXISTING EXPERT SYSTEMS

The February 1986 issue of Expert Systems Strategies included a list of all the expert systems that we believed were being applied in business or industrial settings. That list consisted of 49 systems. Since then, we have continued to accumulate new information and our list has now grown to 138 fielded commercial systems. Other people have developed much larger lists by including systems that have been developed but not fielded (i.e., a research system), or by including commercial systems that are still in some stage of development or testing. A larger list can also be compiled if one includes military systems. We decided to omit those since it is very difficult to get accurate information about them.

The actual number of fielded expert systems is probably between two and three times our total. Many vendors and user companies have indicated that they have other systems that they do not want us to include, to keep their competitors from learning about them. Casey Branscomb, of Branscomb Associates, conducted a survey of the Fortune 500 companies earlier this year. In her survey she promised respondents anonymity, and she came up with a list of about 140 systems, only 20% of which were fielded. Our list contains a large number of small and mid-sized systems, but it probably ignores just as many. Small systems are being developed with PC-based tools by managers who simply don't bother to report their development or use to anyone, and it's hard for us to learn about such systems.

An example of this is provided by Ed Mahler of Du Pont. Du Pont's AI group is encouraging end-users to develop small systems to solve problems that would otherwise have been solved by documentation, memos, and job aids. When a user comes to Mahler's group for help, they work with him or her to define the nature of the task and then, depending on the problem, provide the user with 1st Class, Insight 2+, or an internally developed small tool that does forward chaining. Using this small-systems approach, Du Pont has already fielded about 20 systems and is adding 4 to 5 new systems each month. We have only listed one of Du Pont's systems. Mahler promises that he'll work up a list someday soon, but meantime he is too busy fielding systems that save Du Pont money to take the time to write news releases. We expect that Mahler's situation will become more common in the coming months and that, when we attempt lists like this in future issues of the newsletter, we will be forced to simply list companies and their numbers of systems rather than identifying each individual small and mid-sized system.

As you look at Tables 1-A through 1-L, you will notice that we list each system by name. In the few cases where developers did not name their systems, we have provided a name and have used quotation marks to indicate that it is our creation. In the second column we indicate if the system is being used inside a company (internal use) or being offered for sale to the public. In the third column we provide a brief description of the task the system performs. Next, we list the company that owns the system, and, if the system was developed by a third party, we list the developer's name in parentheses. Whenever possible, we identify our contact at the company owning the system and that person's phone number. Finally, we list the tool or language that was used to develop the system. (In Table 3 we indicate how many systems were developed with each particular tool.) When we compiled our list in February, about 25% had been developed in a language while 75% were developed by means of expert systems-building tools. Our current list suggests that expert systems-building tools are now responsible for about 85% of the fielded systems, a proportion that we expect to increase in the future.

In Table 2, the systems listed on Table 1 are divided into groups according to the size of the application and the general domain in which the system is used. A quick glance at this table should convince you that small to mid-sized applications in the areas of manufacturing, equipment maintenance, and computers continue to provide most of the successes. Obviously, small to mid-sized systems are quicker to develop and easier to field, and this explains their dominance at this time. We expect this overall pattern to continue for some time, however. The number of small systems will continue to grow rapidly. Mid-sized systems will probably increase significantly in the coming year, and the number of large fielded systems will increase slowly but steadily in the next two years.

This list should help answer questions about the viability of expert systems technology. Systems are being fielded and are helping companies solve problems at a rapidly increasing rate.

Potentially Comprehensible by Public

415-326-6200

SRI Int'l Menlo Park, CA → Deborah Anderson ← left message

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
AALPS	•	Helps plan optimal loading of equipment on an aircraft	U.S. Army (Quintus Computer Systems) ~~(415-494-3612)~~ 965-7700	Quintus Prolog left message
ACE	•	Troubleshoots cable systems and recommends maintenance	Greg Versonder Bell Labs (AT&T)	OPS4
ACUMEN 335	o	Assess psychological characteristics in terms of management theory	Human Factors Advanced Technology Group	
"Aircraft Finish Advisor"	•	Assists in selection of aircraft finishing specifications	McDonnell Aircraft Chung-Ta Tsai 314-234-5394	Insight 2
AIRLINE SEAT ADVISOR	o	Allots discount fares to flights	Sperry (Intellicorp) 415-965-5627 (Ray Carhart) ~~[illegible]~~	KEE Jack Dynis ← left message on machine Piedmont 523 1700
"Ash Mixer"	•	Control mixing of radioactive ash with concrete for disposal	Dupont P.D. Soper	EXSYS
"Autonomous Vehicle Control (Robot)"	o	Autonomous vehicle control	Oakridge National Labs (LMI) - perpetually busy (Rod Khanna 617-686-8382)	PICON
BDS	•	Troubleshoots communications hardware problems	Lockheed	LES (Lockheed's tool written in PL/1)
BEACON	•	Configures Burroughs Computer hardware	Burroughs Co.	Quintus Prolog
"Brush Designer"	•	Assists in design of brushes for DC motors	General Motors/Delco Products Div.	S.1
"Business Classifier"	•	Classifies incoming business at Norknions Corp. Center	L. Johnson Creighton Companies Inc.	EXSYS

o = For sale • = For internal use

Table 1-A. Expert Systems in Use Today

	Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
✓	Can Am Treaty	o	Interprets legal and financial differences between US and Canadian practice	Raymond Chabot (Montreal) Pierre Lessard 514-878-2691	GURU
	Capital Asset Process Advisor	•	Assists in developing procedures for moving capital equipment	IBM Dick Ten Dyke 914-696-4435	E.S. Environment /VM
✗	CATS (DELTA)	o	Troubleshoots problems with diesel-electric locomotives	General Electric 518-438-6500 (sLeet) ~~Ethan Kahn - info.~~	Developed in Lisp Implemented in Forth
	CBT Advisor	•	Helps determine suitability of org instructional unit for Computer Based Training delivery	Courseware Inc. Greg Kearsley	
1 ✗	"Chemical Plant Control"	•	Chemical plant control optimization	Exxon (LMI) (Rod Khanna 617-686-8382)	PICON
✓	"Chemical Process Simulator"	•	Simulations of complex chemical processes	Eastman Kodak (LMI) (Rod Khanna 617-686-8382)	PICON
✓	Chest Pain	o	Helps ER staff diagnose chest pains	Evlin Kinney 305-672-5084	First Class (GC Lisp)
✗	CHINA	•	Front-end that helps highway engineers create queries for a Fortran system that designs highway noise barriers	Federal Highway Adm.	FranzLisp (GENIE)
✓	CLASS	•	Determines secrecy status of sensitive information	Dustin Huntington CTID/AL 846-3304	DOE (EXSYS)
	COCOMO1	o	Assists in planning and scheduling software development projects	Level Five Research 305-729-9046	Insight 2
✗ ✗	COMAX	o	Advises farmers on irrigation, fertilization and when to harvest	USDA-Mississippi 601-359-3642	→ [illegible] →
✓	COMPASS	•	Troubleshoots problems with telephone switching equipment.	GTE Dr. Sheri K. Goyal 617-466-2940	KEE

o = For sale • = For internal use

601-359-1395
Frank Steppens
Miss. Central Data Processing Authority

Table 1-B. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
✓ COMPONENT IMPACT ANALYSIS SYSTEM	•	Advises nuclear power plant operators on value and switch settings	Argonne National Labs (Quintus Computer Systems) (415-494-3612)	C and Quintus Prolog
CONAD	•	Checks orders and configures NCR computers	Nixdorf (Germany)	Twaice (Nixdorf's Mprolog-based tool)
✗ "Cooker Advisor"	•	Troubleshoots electrostatic soup "cookers"	Campbell Soup (Texas Instruments) Neil McGlone 250-4133 (~~Sue Metzler~~ 512-250-7302)	Personal Consultant - left message to call back
✗ Crew Control Mock-Up Expert System	•	Advises on reconfiguring space station or simulation for anticipated problems	NASA (Lockheed) (Kent Lennington 713-333-6466)	ART - left message to call back - said that pictures were probably not taken; will get back in touch
CSF Advisor	o	Guide development of cost estimate for moving DPS equipment	IBM Dick Ten Dyke 914-696-4435	E.S. Environment /VM
CV Filter	•	Screens resumes of job applicants	David Imberg Helix Expert Systems-London England Tel 01-583-9391	Helix Expert Edge
✓ "Corrosion Expert"	•	Assists in design of steam generators by recommend alloy to avoid corrosion	Westinghouse Electric Co. (Texas Instruments) (Sue Metzler 512-250-7302)	Personal Consultant
✓ DART	•	Diagnoses computer faults	IBM Michael Geneserth	EMYCIN
"Data Classifier"	•	Classifies sensitive information	Investware Corporation Walter Cooke	EXSYS
Data Protection Act Advisor	o	Classifies sensitive data	Helix Laura Baxter	Helix Expert Edge
"Diagnoser"	•	Diagnoses problems in manufacturing equipment	Ross Laboratories Regina Shanklin 614-229-7070	EXSYS
✗ Diagnostics I	o	Assists in DSM-111 psychiatric diagnoses	Expert Ware Inc	EXSYS

o = For sale • = For internal use

Table 1-C. Expert Systems in Use Today

Eliah - Kahn

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
✓ DIA 8100	•	Diagnoses routine failures of data processing equipment	Traveler's Insurance 229-8200 (Teknowledge) Applicon in Bedford, MA	M.1 → 713-928-4000
X x Dipmeter Advisor	•	Analyzes data from oil well logging instrument	Schlumberger Bay Area 408-998-0123	STROBE (Schlumberger's large,hybrid tool) David Kitson (will call back) 713-928-4680
✓ Dispatcher	•	Selects, transports and delivers parts for assembly while maintaining inventory records	DEC	VAX OPS5 Gordon Pire - engineer
✓ Doppler Diagnosis	o	Helps train doctors to use non-invasive echo effect equipment	Dr. Evlin Kinney 305-672-5084	Basic
✓ DRAGON	•	Configures orders for ICL's Series 39 Computers	ICL (British) 703-893-5915 (Systems Designers Software) (617-935-8009)	Envisage
✓ Drug Interaction Advisor	•	Detects harmful drug interactions	Miami VAMC Evlin L. Kinney 305-672-5084	EXSYS
X x "Dyke Maintanance"	•	Advises on repairs and maintenance of dykes.	Rykwaterstaat Adviesdienst (Systems Designers Software) 2 call ~~617-935-8009~~ 499-2000	Sage 703-820-2700 Larry Parsons - not in - left mess. ① ② left message again
EDDS	•	Advises clerical personnel regarding what information in databases can be disseminated upon request	Environmental Protection Agency	
ESPm	•	Analyzes computer maintenance logs to identify future faults	NCR Vince Santurbane 513-445-4169	S.1
Expert Agriculture Information System	o	Helps library users find references	National Agriculture Library USDA Samuel T. Waters 301-344-3780	First Class
✓ Fan Vibrator Advisor	•	Troubleshoots industrial fan problems	Stone & Webster Engineering Gavin Finn 617-589-1567	EXSYS
✓ Financial Advisor	o	Assists managers in the analysis of capital investment proposals	Palladian Software 617-661-7171	Lisp

o = For sale • = For internal use

Table 1-D. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
Genesis	o	Assists in recombinant DNA research	Intelligenetics	Units
Chavez? GEOX	•	Identifies minerals from hyper-spectral-image data from satellites	NASA – Pat Chavez, I Argonne 6025277221	Lisp Georgia 818-3545011 → PIO at JPL
"Herbicide Advisor"	•	Assists in identifying new herbicides based on structural properties of chemicals	Shell	
"Herbicide Selector"	•	Selects herbicides for use in pipelines	British Gas (Systems Designers Software) (617-935-8009)	Envisage
HI Class	•	Assists in solving circuit board assembly problems and provides on-line instruction	Hughes Aircraft	C
Hotline Helper	•	Helps hotline workers give advice to callers with hardware malfunctions	Texas Instruments Sue Metzler 512-250-7302	Personal Consultant
HP4760AI Electrocardiograph	o	Aids physicians in interpreting electrocardiograph readings	Hewlett-Packard/McMinnville Div.	Lisp
IDEA	•	Troubleshoots telephone switching equipment	Pacific Bell John Gerard 415-823-1961	Exsys
IMP	•	Troubleshoots Epitaxial Reactor Machinery	Texas Instruments Cindy Griffen 512-250-7984	Personal Consultant
Information Engineering Workbench	o	Assists in design of Management Information Systems	Knowledgeware Kim Frazier 313-971-5363	
Ingot	o	Assists in financial forecasting	Schonfeld & Associates 312-869-5556	Fortran
"Intelligent Building Management"	o	Intelligent building management	Johnson Controls (Rod Khanna 617-686-8382)	PICON

o = For sale • = For internal use

Table 1-E. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
Intelligent Peripheral Troubleshooter	•	Troubleshoots computer peripheral equipment	Hewlett Packard	
Intelligent Software Configurator	•	Help configure software for DPS 6 computers	Honeywell Barabara Braden 617-552-6351	Lisp and Loops
ISS Three	o	Manages computer capacity	International Systems Services Corporation 212-972-4400	
✓ "Legal Expert"	•	Legal advice on estate planning and investment	More For Less Hawaii David Bernstein 808-545-7117	GURU
Letter of Credit Advisor	o	Assists in processing foreign letters of credit	Bank of America (U.K.) (Helix-Fiona Bell 01-583-9391)	Expert Edge
?✓ Macsyma	o	Helps scientists and engineers with mathematics problems	Symbolics Inc. 617-576-2600	Lisp
Matchware	o	Assists accountants in selecting appropriate software	Matchware Computer Services Gary Forsyt 317-841-8100	
"Material Handling"	o	Materials handling and scheduling of discreet manufacturing areas	EXON (LMI) (Rod Khanna 617-686-8382)	PICON
Med Ex	o	Intelligent interface to medical information for the layman	Perceptronics 213-641-2660	
Mentor	o	Diagnoses preventive maintenance need for large central air conditioning systems	Honeywell Roy Joy 612-541-6807	G C Lisp
✓ Mercury	•	Recommends interest swap deals	Shearson American Express Bruce Gras	
Metals Analyst	o	Identifys commercially used metals and alloys	General Electric Corp. Tom Anthony 518-387-6160	Exsys

o = For sale • = For internal use

Table 1-F. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
"Micro"	o	Analyzes blood serum protein readings on instrument printouts	Helena Laboratories (Rutgers Uni.)	EXPERT
Micro Genie	o	Performs DNA and RNA analysis	Beckman 800 DNA-ACGT (Sci Soft Inc)	Pascal
More	o	Identifies potential buyers from mailing lists	Persoft Inc 617-935-0095	COBOL
"Mortgage Advisors"	•	Rates new mortgage applications	Nederlandsche Medderstandsbank (Systems Designers Software) 617-935-8009	Sage
Mudman	•	Diagnoses problems with "mud" used in oil well drilling and recommends new composition	N.L. Baroid Co.	OPS5
NAVEX	•	Monitors controls on space shuttle flights	NASA-Bob Brown 713-483-4751 (Inference Corp.) (Leiann Lee 503-245-0905) Julie McHenry	ART Jill Pembroke - will call back
Network Diagnostician	•	Troubleshoots telephone switching equipment	Standard Telephone and Cables (Systems Designers Software) 617-935-8009	Envisage
OCEAN	•	Checks orders and configures NCR computers	NCR (Teknowledge)	S.1
Oleophilic Advisor	•	Assists lithography R&D group in choosing new materials	Rockwell (Teknowledge)	M.1
Oncocin	o	Advises physicians on the treatment of patients receiving chemotherapy	Stanford Medical Center	Lisp
PAGE1	•	Troubleshoots non-impact Page Printing System	Honeywell Barbara Braden 617-552-6351	Lisp and Loops
Paradox	o	Assists in management of relational databases	Ansa Software	

o = For sale • = For internal use

Table 1-G. Expert Systems in Use Today

justification

John Morrison

ARS - USDA
PO Box 6112
Temple, Texas 76503-6112

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
X Peanut/Pest	o	Recommends irrigation and pest control for peanut farmers	USDA (Georgia) Sarah Parker (817)774-1201	Exsys - due to USDA she couldn't send this one but she is sending one called "Planting" probably. She will call back tomorrow
X X Performance Mentor	o	Guides managers in shaping the performance of subordinates	AI Mentor Inc 415-969-4500	Exsys will look into it and get back to us - not sure if they have pictures
✓ Permaid	o	Troubleshoots large disc drives	Honeywell Dave Rolston 602-862-6925	Loops on Xerox 1109
✓ Photolithography Advisor	•	Troubleshoots photolithographic problems in chip fabrication	Hewlett-Packard	HP-RL (Hewlett-Packard's proprietary tool)
Pine	•	Guide for entering info to retain database about problems and fixes	IBM Dick Ten Dyke 914-696-4435	E.S. Environment /VM
✓ X Pipeline Advisor	•	Helps determine flow characteristics within pipes	British Hydronomic Research Assn. (Systems Designers Software) 617-935-8009	Poplog
✓ X "Plant Safety Advisor"	o	Advises managers on appropriate safety procedures	Stone & Webster Engineering Gavin Finn 617-589-1567	MAIDService
PRESS	•	Debugs operating systems software	Honeywell Dave Rolston 602-862-6925	OPS5 and Mac Lisp
"Process Optimization for Energy Savings"	o	Process control	TEXACO (LMI) (Rod Khanna 617-686-8382)	PICON
PTE Analyst	o	Advises lawyers on prohibited transaction exceptions	Computer Law Systems Jill Swenson 612-941-3801	Personal Consultant Plus
X X Purdue Grain Market Advisor	o	Helps farmers determine the best way to market the grain they produce	Purdue University Ohio?	Personal Consultant
Planpower	o	Assists financial planners	First Financial Planner Services Jeanne Bromley 203-520-8106 (Applied Expert Systems) (617-492-7322)	Lisp

o = For sale • = For internal use

Table 1-H. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
"Power Plant Management"	o	Power plant management	Leeds and Northrup (LMI) (Rod Khanna 617-686-8382)	PICON
✗✗✓ Puff	o	Interprets output of a pulmonary function instrument (medical)	Pacific Medical Center	Developed in Emycin Implemented in Basic
Pump Pro	o	Preventive maintenance and troubleshooting for centrifugal pumps	Stone & Webster Engineering Gavin Finn 617-589-1567	MAIDService
QMF Advisor	•	Query management facility for use by project service representative to guide through problem solving session with client	IBM Dick Ten Dyke 914-696-4435	E.S. Environment /VM
✓ "Repair Diagnosis"	•	Diagnoses transmission faults in passenger cars	Renault (Cap Gemini Sogeti [Paris])	
The Requirements Analyst	o	Helps accountants choose software appropriate to their needs	Computer Training Services 301-468-4800	Lotus 1-2-3
✗ × Resource Apportionment Aid	•	Assists Air Force officers in assigning air units to missions and targets	USAF (Perceptronics) -Dr. Weltman - 818 - Woodland Hills 884-~~[illegible]~~ 7470 (John Semming 213-~~641-2660~~) ~~[illegible]~~	left message (1) (2) left message
Rotating Equipment Diagnostic System	•	Troubleshoots various rotating equipment	Stone & Webster Engineering Gavin Finn 617-589-1567	Exsys
× RPMS	•	Helps NASA officers in Resource Planning and management	NASA	OPS5 → Ask JP
Schooner	•	Troubleshoots asynchronous datacom links	Hewlett-Packard	
✓ "Site Layout Advisor"	•	Configures machine-room floor for computer systems and peripherals	Hitachi	
✓ SNAP	•	Assists new Infomart vistors to determine computer needs	Infomart - Dallas TX (Boeing Computer Serv.)	Personal Consultant

o = For sale • = For internal use

Table 1-I. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
"Source Rock Advisor"	•	Assists geologist in the evaluation of oil potential of rock	Phillips	M.1 → Coll
Spin Pro	o	Helps scientist perform ultracentrifugation operations	Beckman Instruments	G C Lisp
"Styrene"	o	Simulate manufacture of styrene	Badger (LMI) (Rod Khanna 617-686-8382)	PICON
Storage System Test	•	Manages final manufacturing test for storage systems	IBM Dick Ten Dyke 914-696-4435	E.S. Dev. Env/VM
"Switch Diagnoser"	•	Troubleshoots telephone switching equipment	Bell Communications Research	S1
Syscon	o	Configures DPS 90 Mainframe	Honeywell Dave Rolston 602-862-6925	OPS5 and Mac Lisp
TIMM-Tuner	o	Assists system manager in tuning a VAX/VMS operating system	General Research	TIMM
TITAN	•	Assists technicians in troubleshooting TI's 990 minicomputer	Radian Charles Riese 512-454-4797	RuleMaster
TOGA	o	Diagnoses faults in large utility transformers based on gas chromatographic analysis of insulating oil	Radian Charles Riese 512-454-4797	RuleMaster
TQMSTUNE	•	Assists in tuning Triple Quadrapole Mass Spectrometers	Lawrence Livermore Labs Carla Wong 415-422-0435	KEE
"Train Breaking Advisor"	o	Controls locomotive braking for accuracy and comfort	Hitachi	
"Train Travel Advisor"	•	Helps travel agents plan train itineraries	Thomas Cook Travel Agent	ESI Prolog-2

o = For sale • = For internal use

Table 1-J. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
Troubleshooting Aid for F6502	o	Assists technicians in diagnosing and repairing a F6502 instrument	Tektronix	Detekr (Tektronix's electronics troubleshooting tool)
TURBOMATIC	o	Aids in diagnosis of vibration problems in large turbomachinery	Radian Charles Riese 512-454-4797	RuleMaster
Unit Committment Advisor	o	Helps schedule power plant fire up and shut-down to meet demand	IBM ~~Dick Ten Dyke~~ ~~914-696-4495~~	MAIDService left message
"Warehouse Planner"	o	Assists in automated warehousing	Hitachi	
Water Permit Review System	•	Helps review for water use permits applications	EPA (Software Architecture and Engineering Inc.-Jack Collins) (703-276-7910)	KES
WAVES	o	Helps geologists choose appropriate process for siesmic data	Teknowledge	KS 300
Weld Defect Diagnosis System	•	Helps Determine cause of welding defects	Stone & Webster Engineering Gavin Finn 617-589-1567	Exsys
Weld Procedures Selection System	•	Advises welders on procedures materials and electrolite type	Stone & Webster Engineering Gavin Finn 617-589-1567	Exsys
Weld Selector (B&W)	•	Helps welding engineers choose proper weld procedures	Babcock and Wilcox (Intellicorp 415-965-5500)	KEE
Weld Selector (AWI)	o	Helps welding engineers choose proper weld procedures	American Welding Institute (Colorado School of Mines)	Personal Consultant
Welder Qualification Test Selection System	•	Helps managers choose appropriate qualification tests for welders	Stone & Webster Engineering Gavin Finn 617-589-1567	Exsys
Wheat Counselor	o	Aids in the diagnosis of wheat diseases	ICI [Imperial Chemicals]	

o = For sale • = For internal use

Table 1-K. Expert Systems in Use Today

Name of System	o •	Specific Domain and Task	Developer or Owner (Developer)	Development Software
✓ "Wire Editor"	•	Processes unformatted money orders	Citibank	
✓ XCON	•	Configures orders for DEC computers	DEC -Bob Abramson (Carnegie-Mellon University)	VAX OPS5
✓ XSITE	•	Prepares site layout plan for customer's computer room	DEC	VAX OPS5
"X-ray Diffractometer Asst."	•	Assists in identifying crystallographic structures of metals	Westinghouse Electric Co.	Personal Consultant
Yes/MVS	•	Controls MVS operating system and advises human manager	IBM	OPS5

o = For sale • = For internal use

Table 1-L. Expert Systems in Use Today

66 selected

Table 2-A EXPERT SYSTEM SIZE AND DOMAIN	Size of Expert System		
Domain	Small (50-500 rules)	Mid-Sized (500-2000 rules)	Large (2000+ rules)
Management	Acumen Performance Mentor Welder Qualification Test Selection System	COCOMO1 Plant Safety Advisor RPMS Warehouse planner	Resource Apportion-ment Aid
Finance	Letter of Credit Advisor Mercury "Mortgage Advisor"	Ingot Legal Expert More	Financial Advisor Planpower
Office Automation	Business Classifier CLASS CV Filter Data Classifier Data Protection Act Advisor	EDDS Water Permit Review System Wire Editor	
Manufacturing (Planning/Scheduling)	Ash Mixer Cooker Advisor Weld Procedures Selection System	Aircraft Finish Advisor AWI Weld Selector Brush Designer Conponent Impact Analysis System CHINA Dispatcher HiClass Oleophilic Advisor Photolithography Advisor Unit Commitment Advisor	Babcock Weld Selector
Equipment Maintainance (Troubleshooting)	Diagnoser Fan Vibration Advisor Herbicide Advisor Rotating Equipment Diagnostic System Troubleshooting Aid for F6502	BDS CATS Corrosion Expert Hotline Helper IMP Mentor Network Diagnostician PAGE1 Pump Pro Pipeline Advisor Repair Diagnosis Switch Diagnoser Turbomatic TOGA	ACE Compass

Table 2-B EXPERT SYSTEM SIZE AND DOMAIN Domain	Size of Expert System		
	Small (50-500 rules)	Mid-Sized (500-2000 rules)	Large (2000+ rules)
Computers	Information Engineering Workbench Matchware Paradox PRESS QMF Advisor Schooner SNAP Syscon Timm-Tuner	Capital Asset Procedures Advisor CSF Advisor DART DIA 8100 ESPm Intelligent Software Configurator Intel Peripheral Troubleshooter ISS Three OCEAN PINE Site Layout Advisor Storage System Test Advisor Titan	BEACON CONAD Permaid XCON XSITE Yes MVS
Oil/Geology		GEOX Mudman Source Rock Advisor WAVES	Dipmeter Advisor
Transportation		AALPS Train Braking Advisor Train Travel Advisor	Airline Seat Advisor NAVEX
Agriculture	COMAX Expert Agr iculture Information System Peanut/Pest	Herbicide Advisor Purdue Grain Market Advisor Wheat Counselor	
Science & Medicine	Chest Pain Drug Interaction Advisor Metals Analyst Micro Puff	HP4760AI MedEx Micro Genie Spin Pro X-Ray Diffractometer Assistant	Genesis Macsyma Oncocin TQMSTUNE

Table 2-C

EXPERT SYSTEM SIZE AND DOMAIN

Domain	Size of Expert System		
	Small (50-500 rules)	Mid-Sized (500-2000 rules)	Large (2000+ rules)
Training	CBT Advisor Doppler Diagnosis		
Misc.	Diagnostics I	"Dyke Maintanance" PTE Analyst	Crew Control Mock-Up Expert System

TOOLS

EXPERT SYSTEMS-BUILDING TOOLS

Tables 3A-3C list all of the expert systems-building tools that we know about as this newsletter is completed. We have undoubtedly missed several small tools, but we probably have all of the large and mid-sized tools and we certainly have all of the tools that are being actively advertised in the trade press.

As you examine the charts you will notice that we indicate the name of the tool (and the vendor), the price of the tool (a range if the tool is offered at different prices on different hardware), the approximate number the vendor claimed to have sold as of January 1986, the number of fielded systems that have been built using that particular tool, and the hardware on which the tool runs. Vendors are constantly expanding the range of hardware options available, so our data may be a little out of date. Likewise, vendors will be introducing new tools at AAAI and may be changing prices or adding other hardware options.

If you are at AAAI, we suggest that you use this chart as a checklist and update it during your visits to vendor's exhibits.

We have divided the tools into eight categories. Large hybrid tools represent the most complex expert systems development environments currently available. Until recently these tools were only available in Lisp and only ran on Lisp machines, VAX's, and Unix workstations configured for Lisp. Mid-sized hybrid tools is a new category of tools that run on personal computers and provide environments with a number of knowledge representation paradigms. This category is new with this listing, but we expect it to grow rapidly in the next two years as PC hardware improves and people find that they can build and run large, multi-paradigm systems on PCs.

Large rule-based tools tend to offer context trees and run on mainframes, VAXs, Lisp machines, or Unix workstations. Mid-sized rule-based tools also generally offer context trees, but they run on personal computers. Small rule-based tools run on personal computers but lack context trees, Emycin-like confidence factors, and some of the other editing features available in the mid-sized tools.

Inductive tools generate rules from examples. These weak but friendly matrix-based tools come in two sizes: large inductive tools, which run on mainframes and PCs, and small inductive tools that run only on personal computers.

Finally, we have listed domain specific tools. These tools are specifically designed only to develop expert systems for a particular domain. We expect this category to expand rapidly in coming years.

In previous lists we included a category for logic-based tools. We have dropped the category since most logic-based tools have been improved to the point where the developer does not not need to be aware of the logic but can simply focus on developing rules. Hence tools that were formerly listed as logic-based tools are now listed under one of the rule-based categories.

We do not intend to suggest that tools in any one group are better or worse than those in another. The appropriate tool must be determined by the nature of the problem that is to be solved. Some problems demand small inductive tools, some others require mid-sized rule-based tools, and other problems call for large or mid-sized hybrid tools. It's frustrating to attempt a complex problem with a weak, inflexible tool, but it's equally foolish to attempt to master the complexities of a large tool in order to solve a problem more appropriate for one of the simple PC-based tools to solve.

You should also be careful not to draw any conclusions based on price. Many good tools are priced low and some of the tools on this list are outrageously overpriced.

Also keep in mind that a tool is much more than just an inference engine and a knowledge representation paradigm. It includes a developer interface, a user interface, and utilities to pass data to or from other programs or databases. Depending on the hardware on which it runs and the language in which it is written, a tool may or may not be easy to field, and it may or may not run fast enough for your users. Moreover, you buy more than a tool when you give your money to a vendor. You obtain some package of training, support, and consulting services and some agreement on future updates. Some vendors have special expertise in particular problem domains or in fielding systems on particular hardware, and that can also be an important consideration. All of these things need to be taken into account when you decide to purchase a tool; they are not adequately reflected in Table 3, where we simply divide the tools on the basis of their knowledge representation strategy.

Table 4 is a simplified version of the chart that we use when examining a tool. If you are considering a purchase, we suggest you use some version of this chart when talking with potential vendors.

Addresses and telephone numbers of vendors in Table 3 are listed on pages 22-23.

Table 3-A
CURRENT EXPERT SYSTEM BUILDING TOOLS

	TOOL (Vendor)	Price of Tool	Approx. Number sold by Jan. 86	Fielded Systems	Hardware			
					Mini/ Mainframe	Workstation (Unix)	Lisp Machine Workstation	PC
LARGE HYBRID TOOLS	ART (Inference Corp)	$60-80K	300	2		Sun-3 Lisp	• Lisp	
	KEE (Intellicorp)	$60K	600	5			• Lisp	
	Knowledge Craft (Carnegie Group)	$50K	50				• Lisp	
	LOOPS (Xerox)	$300	NA	3			Xerox D Machines Lisp	
MID-SIZE HYBRID TOOLS	Acorn (Gold Hill Computer)	$5K	NA					IBM-AT Lisp
	Personal Consultant Plus (Texas Instruments)	$2,950	NA	1				TI, IBM Scheme
	Intelligence/Compiler	$990	NA					IBM Pascal
LARGE RULE-BASED TOOLS	ENVISAGE (Systems Designers Software)	$15-40K	25	1	VAX Pascal	MicroVAX Pascal		
	ES Environment/VM & /MVS (IBM)	$60K	NA	5	IBM Pascal			
	IKE (LISP Machine Inc.)	$15K	NA				LMI Lisp	
	KES/VS (Control Data)	$11-71K	NA	1	Cyber Lisp			
	Knowledge Workbench (Silogic, Inc.)	$1.5-12K	6		SUN, MicroVax Prolog			
	OPS5e (Verac, Inc.)	$3K	NA				• Lisp	
	OPS83 (Production Systems Technologies)	$2-25K	60		VAX C	MicroVAX Apollo, Sun C		IBM-AT AT&T C
	S.1 (Teknowledge)	$25-45K	65	4	IBM/VM C	Sun C		IBM-AT C

Table. 3-B

CURRENT EXPERT SYSTEM BUILDING TOOLS

	TOOL (Vendor)	Price of Tool	Approx. Number sold by Jan. 86	Fielded Systems	Hardware			
					Mini/ Mainframe	Workstation (Unix)	Lisp Machine Workstation	PC
LARGE RULE-BASED TOOLS	TWAICE	$64K	NA	1	IBM/VM VAX Prolog			
	VAX OPS5 (DEC)	$7.5K	NA	8	VAX Bliss	MicroVAX Bliss		
MID-SIZED RULE-BASED TOOLS	(Aion) Aion Dev. System /PC & /MVS	$5K	NA		IBM/ MVS			IBM Pascal
	GURU (Micro Data Base Systems)	$3K	NA	1				IBM C
	HUMBLE (Xerox Corp.)	$300- 3K			Tektronix Smalltalk		Xerox Smalltalk	IBM-AT MAC Smalltalk
	KES II (Software A & E)	$4K	65					IBM C
	M.1 (Teknowledge)	$5K	408	3				IBM C
	Nexpert (Neuron Data)	$5K	100					Mac Assembler
	Personal Consultant (Texas Instruments)	$950	1200	7			Explorer Scheme	TI, IBM IQ Lisp
SMALL RULE-BASED TOOLS	Advisor (Ultimate Media)	$99.50- 295	100					IBM,Apple Commodore Assembler
	Apes (Programming Logic Systems)	$1-6K	NA		VAX Prolog			IBM Prolog
	Arity E.S. Dev. Package (Arity Corp.)	$295	NA					IBM-AT Prolog
	DUCK (Smart Systems Technologies)	$6K	30		IBM,VAX Lisp			IBM Lisp
	ESP Advisor (Expert Systems Int'l.)	$895- 6500	NA		VAX Prolog			IBM Prolog
	EST (Mind Path Product Corp.)	$495	NA					IBM TI PC

Table 3-C
CURRENT EXPERT SYSTEM BUILDING TOOLS

	TOOL (Vendor)	Price of Tool	Approx. Number sold by Jan. 86	Fielded Systems	Hardware			
					Mini/ Mainframe	Workstation (Unix)	Lisp Machine Workstation	PC
SMALL RULE-BASED TOOLS	ExperOPS5 (ExperTelligence)	$525	NA					Mac Lisp
	Expert Edge (Human Edge)	$795	NA	3				IBM C
	EXSYS (Exsys Inc.)	$395-900	1K+	16	VAX C			IBM C
	Insight 1.2 (Level Five Research)	$95	1K					IBM Pascal
	Insight 2+ (Level Five Research)	$485	300	2				IBM Pascal
	Knowol and Knowol+ (Intelligent Machine Co)	$39.95-99.95	NA					IBM
	MacKIT (Knowledge System Environments)	$500	NA					Mac Fortran
	MicroExpert (McGraw-Hill)	$49.95-59.95	NA					IBM,CP/M Pascal
	SAGE (Systems Designers Software)	$995	300	2				IBM Pascal
	TOPSI (Dynamic Master Systems)	$125-375	400					IBM,CP/M C
	(Paperback Software)	$195	NA					IBM
	Wisdom XS (Software Intelligence Laboratory)	$750-13K	NA					IBM C
	Xi (Expertech)		NA					IBM
	XPER (Abacus Software)	$59.95	NA					Commodore C
	XSYS (California Intelligence)	$995	40					DOS IQ Lisp

Table. 3-D CURRENT EXPERT SYSTEM BUILDING TOOLS

	TOOL (Vendor)	Price of Tool	Approx. Number sold by Jan.86	Fielded Systems	Hardware			
					Mini/ Mainframe	Workstation (Unix)	Lisp Machine Workstation	PC
LARGE INDUCTIVE TOOLS	EX-TRAN 7 (Intelligent Terminals)	$3.5-47K	NA		IBM,VAX Fortran	• Fortran		IBM Fortran
	RuleMaster (Radian Corp.)	$1-17K	60	3	VAX C	• C		IBM C
	TIMM (General Research)	$5-39.5K	30	1	IBM Fortran			IBM Fortran
SMALL INDUCTIVE TOOLS	Expert-Ease (Human Edge)	$695	NA					IBM Pascal
	1st Class (Programs in Motion)	$495	250	2				IBM Pascal
	KDS (KDS Corp)	$400	150					IBM Assembly
	Super Expert (Softsync, Inc.)	$199-1,200	NA					IBM
DOMAIN SPECIFIC TOOLS	IN-ATE, LISP (Automated Reasoning Corp.)	$15K	NA				LMI Symbolics	
	IN-ATE, MICRO (Automated Reasoning Corp.)	$2.5K	NA					Mac C
	PICON (Lisp Machine Inc.)	$60K	NA	8			LMI Lisp	

TABLE 4 — VENDOR LIST

Acorn
Gold Hill Computers
163 Harvard Street
Cambridge, MA 02139
(800)242-LISP or
(617)492-2071 (in MA)
Contact: Jane Dusza

Advisor
Ultimate Media, Inc.
275 Magnolia Ave.
Larkspur, CA 94939
(415)924-3644
Contact: William Maulton

ExperOPS5
ExperTelligence
559 San Ysidro Road
Santa Barbara, CA 93108
(805)969-7874
Contact: Jim Giles

Expert Ease & Expert Edge
Human Edge Software Corp.
2445 Faber Place
Palo Alto, CA 94303
(415)493-1593
Contact: Jim Chapman

KEE
IntelliCorp
1975 El Camino Real West
Mountain View, CA 94040-2216
(415)965-5633
Contact: Sue Brown

Knowledge Craft
Carnegie Group
650 Commerce Court
Station Square
Pittsburg, PA 15219
(412)642-6900
Contact: Michael Chambers

Personal Consultant & Personal Consultant Plus
Texas Instruments
12501 Research Blvd. MS 2244
P.O. Box 2909
Austin, TX 78769
(512)250-6785

PICON
Lisp Machine Inc.
6033 West Century Blvd.
Suite 900
Los Angeles, CA 90045
(617)682-0500
Contact: Rod Khanna

Table 4 — VENDOR LIST (Cont'd.)

Aion Development System/MVS & Aion Development System/PC
Aion Corp.
101 University Ave.
Palo Alto, CA 94301
(415)924-9595
Contact: Joel Voelz

Apes
Programming Logic Systems
31 Crescent Drive
Milford, CT 06460
(203)877-7988
Contact: Roberta Hanson

Arity Expert System Development Package
Arity Corporation
385 Baker Ave.
Concord, MA 01742
(617)371-1243
Contact: Meredith Bartlett

ART
Inference Corp.
5300 West Century Blvd., 5th Floor
Los Angeles, CA 90045
(213)417-7997
Contact: Donald Gammon,
VP Sales

DUCK
Smart Systems Technologies
7700 Leesburg Pike
Falls Church, VA 22043
(703)448-8562
Contact: Doug Berry

ENVISAGE & SAGE
Systems Designers Software Inc.
444 Washington St., Suite 407
Woburn, MA 01801
(617)935-8009
Contact: Bruce Holt

ES Environment/VM & ES Environment/MVS
IBM
P.O. Box 10
Princeton, NJ 08540
(201)329-7000
Contact: W.S. Redfield, Jr.

ESP Advisor
Expert Systems International
1700 Walnut Street
Philadelphia, PA 19103
(215)735-8510
Contact: Susan Strange

EST (Expert Systems Toolkit)
Mind Path Products
12750 Merit Drive, L.B. 77
Dallas, Texas 75251
(214)770-5435
Contact: Alex Tsakiris

EX-TRAN 7
Intelligent Terminals Ltd.
c/o Jeffrey Perrone & Assoc.
3685 17th Street
San Francisco, CA 94114
(415)431-9562
Contact: Jeffrey Perrone

EXSYS
Exsys Inc.
P.O. Box 75157, Contract Sta. 14
Albuquerque, NM 87194
(505)836-6676
Contact: Dustin Huntington

1st Class
Programs in Motion
10 Sycamore Rd.
Wayland, MA 01778
(617)655-6005
Contact: William Hapgood

GURU
Micro Data Base Systems, Inc.
P.O. Box 248
Lafayette, IN 47902
(317)447-1122
Contact: Kevin Castleberry

HUMBLE
Xerox Special Information Systems
250 N. Halstead
P.O. Box 7018
Pasadena, CA 91109
(818)351-2351
Contact: Dan E. Dody

IKE
Lisp Machine Inc.
6 Tech Drive
Andover, MA 01810
(617)689-3554

Insight 1 & Insight 2+
Level Five Research
503 Fifth Ave.
Indialantic, FL 32903
(305)729-9046
Contact: Cornelius Willis

Intelligence/Compiler
IntelligenceWare Inc.
9800 S. Sepulveda Blvd.,
Suite 730
Los Angeles, CA 90045
(213)417-8896
Contact: Kamran Parsaye

KES and KES II
Software Architecture & Engineering
1500 Wilson Blvd., Suite 800
Arlington, VA 22209
(703)276-7910
Contact: Andrew Copland

KES/VS
Control Data
P.O. Box 0 HQW096
Minneapolis, MN 55440
(612)853-6137
Contact: Joanne Henry

KDS
KDS Corporation
934 Hunter Road
Wilmette, IL 60091
(312)256-4201
Contact: William J. Wallace

Knowledge Workbench
Silogic, Inc.
9841 Airport Blvd., Suite 600
Los Angeles, CA 90045
(213)337-7477
Contact: Jim Boates

Knowol & Knowol+
Intelligent Machine Co.
1907 Red Oak Circle
Newport Richey, FL 33553
(813)844-3262
Contact: Jeffrey Ferris

Lisp In-Ate & Micro In-Ate
Automated Reasoning Corp.
290 W. 12th Street, Suite 1-D
New York, NY 10014
(212)206-6331
Contact: Richard Cantone

LOOPS
Xerox Corp.
250 N. Halstead St.
P.O. Box 7018
Pasadena, CA 91109
(818)351-2351 (Ext. 1603)
Contact: Deborah Kelfer

M.1 & S.1
Teknowledge, Inc.
525 University Ave.
Palo Alto, CA 94301
(415)327-6600
Contact: Dina Barr

MacKIT
Knowledge Systems Environments
201 South York Road
Dillsburg, PA 17019
(717)766-4496
Contact: Ed Beauregard

MicroExpert
McGraw-Hill Book Company
P.O. Box 400
Hightstown, NJ 08520
(609)426-5750
Contact: Debbie Innie

Nexpert
Neuron Data
444 High St.
Palo Alto, CA 94301
(415)321-4488
Contact: Patrick Perez

OPS5e
Verac, Inc.
P.O. Box 26669, Dept. 418
San Diego, CA 92126-0669
(619)457-5550
Contact: Sally Tumia

OPS83
Production Systems Technologies, Inc.
642 Gettysburg Street
Pittsburg, PA 15206
(412)362-3117
Contact: Diana Connan

RuleMaster
Radian Corp.
8501 Mo-Pac Blvd.
P.O. Box 9948
Austin, TX 78766
(512)454-4797
Contact: Ben Finkel

Super Expert
Softsync, Inc
162 Madison Ave.
New York, NY 10016
(212)685-2080
Contact: Nigel Searle

TIMM
General Research
7655 Old Springhouse Road
McLean, VA 22102
(703)893-5915
Contact: Wanda Rappaport

TOPSI
Dynamic Master Systems, Inc.
P.O. Box 566456
Atlanta, GA 30356
(404)565-0771
Contact: David Smith

TWAICE
Logicware, Inc.
(Representing Nixdorf Computer)
1000 Finch Ave. West, Suite 600
Downsview, Ontario M3J 2V5
Canada
(416)665-0022
Contact: Michael Anthony

VAX OPS5
Digital Equipment Corp. (DEC)
77 Reed Road (HL02-3/E09)
Hudson, MA 01749-2895
(617)568-4000
Contact: Juanita Thiel

VP Expert
Paperback Software Int.
2830 Ninth Street
Berkeley, CA 94710
(415)644-2116

Wisdom XS
SIL, Inc.
1593 Locust Ave.
Bohemia, NY 11716
(516)589-1676
Contact: Connie Chun

XPER
Abacus Software
P.O. Box 7211
Grand Rapids, MI 48510
(616)241-5510
Contact: Abacus Software

XSYS
California Intelligence
912 Powell Street #8
San Francisco, CA 94108
(415)391-4846
Contact: Ray Winestock

Table 4. A Checklist for Evaluating an Expert System-Building Tool

Tool: ______________________ Version: __________

Contact: ______________________ Date: __________

Power and Flexibility of Knowledge Representation and Inference and Control Techniques	
Developer Interface: Ease of use and power of editing utilities.	
User Interface: Ease and flexibility of user interface development utilities	
System Interface: Ability of tool to send and obtain data from other programs and databases. Hardware the tool runs on.	
Training and Support: Documentation, courses, consulting availability and vendor experience with particular domains and hardware.	
Runtime speed	
Cost: Initial, multiple copies, runtime copies, training, support and updates.	

NEW PRODUCTS

VP-EXPERT

VP-Expert is a very high-quality tool, combining powerful new features with a variety of features currently found only in mid-size tools costing from $995 to $5,000. And it's priced at $195!

VP-Expert, which is being introduced at AAAI by Adam Osborne's Paperback Software, is a rule-based system with an inductive front end. Several of the current inductive tools, such as 1st Class, claim to combine rules and induction, but are in fact inductive tools that simply allow you to create matrices containing a single "example" (which functions as a rule). However, since these tools still apply an inductive algorithm to the matrices, one is left without the full power of a rule-based system. VP-Expert is a rule-based tool that provides the developer with the option of inputting examples, which the system will convert into rules (eliminating unnecessary rules or clauses in the process). In essence, however, VP-Expert is a rule-based tool with all of the power and flexibility one expects from a rule-based tool.

VP-Expert is written in C and is very fast. It can read or write directly to dBase II and III, Lotus 1-2-3, VP-INFO, and VP-PLANNER. The developer, simply by using a single command, can create a system that will search multiple records or whole ranges of cells. (Most tools, such as GURU, either limit you to accessing a single cell or record, or force you to resort to a serious programming effort to do more complex searches.) Moreover, calls to external programs are accomplished very quickly with VP-Expert.

VP-Expert represents facts as attribute-value pairs. Like M.1, however, VP-Expert supports multi-valued attributes and thus allows the developer to insert "variables" in rules via subscribed attributes (e.g., TIME[1], TIME[2] . . .). This feature is especially useful for processing multiple spreadsheet cells or multiple database records and would be much harder to do via multiple instantiation in most structured rule systems.

VP-Expert allows the developer to create special "database rules" -- generalized rules that can replace several similar rules. These rules are similar to the "variable rules" of M.1. Where M.1 requires the developer to create a "lookup table" within the knowledge base, however, VP-Expert allows these values to be derived from database records. "Database rules" are inherently multiply instantiated, since they are applied to EVERY record in a database.

Most small and mid-sized rule-based tools either lack confidence factors or implement them using standard probabilities, which makes them undesirable if you are serious about using confidence factors. VP-Expert provides the full implementation of the EMYCIN confidence factor schema that is otherwise only found in the expensive mid-size tools. VP-Expert also supports floating-point math expressions in the premises or conclusions of rules and also supports a variety of trigonometric functions.

Backward chaining is the primary control strategy in VP-Expert, but commands in rule conclusions can be given to force limited forward chaining.

To create a knowledge base in VP-Expert, one uses an internal text editor. If desired, however, developers can also use other text editors, like Wordstar, to create or edit a knowledge-base file.

The basic rule syntax used in VP-Expert is similar to M.1. The developer writes rules and adds text or questions by associating them with attributes and adding them to the end of the knowledge base. In our experience, this is still the most elegant and efficient syntax to use to quickly develop a small to mid-size knowledge base.

VP-Expert uses menus and color to provide one of the best interfaces available on a PC-based tool. Windows can be opened during a consultation to let the user see what rules are being tried and what conclusions have been reached. Full trace facilities are provided. In addition, VP-Expert provides a very nice color graphic depiction of how rules relate to each other.

VP-Expert supports "How" and "Why" explanations and will allow the user to ask "What if" questions after a consultation to see how the results would have changed if different responses had been given.

VP-Expert is one of the best tools available for the PC. For its price, it is clearly the most cost-effective PC tool on the market. When you consider that it will be promoted by a marketing pro like Adam Osborne, you realize that the expert systems-building tools market is ready to enter a new phase in which tools will be mass-marketed to corporate MIS departments. VP-Expert will be one of the tools that will open that market up. We expect it to set the standard in functionality, user-friendliness, and cost against which all of the other PC tools will soon be measured.

VP-Expert guarantees that current vendors who want to continue compete in the expanding small to mid-size tool market must offer a lot more power and functionality and much better user interfaces for a lot less money.

(The address of Paperback Software is listed with Table 3.)

ACORN

As VP-Expert redefines the market for small to mid-size tools, Gold Hill Computer's new tool, Acorn, promises to redefine the high end of the market. Acorn is a hybrid tool offering most of the features found in ART, KEE, or Knowledge Craft, including rules, frames, contexts, and the various sophisticated Lisp-based editing features that serious knowledge engineers expect. Unlike its competitors, which cost over $50,000 and only run on machines that cost about the same, Acorn runs on an IBM PC AT and is priced at $5,000!

Acorn is designed to be used at two different levels. At the higher level, "the developer interface," the knowledge engineer interacts with menu-driven interfaces to develop end-user systems. At the lower level, the knowledge engineer has access to the GCLISP 286 DEVELOPER package in which the higher-level package is written and thus has the ability to modify the system in any desired manner.

ACORN provides a frame-like knowledge representation network, integrated forward and backward chaining, and certainty factors. It supports object-oriented programming, rule-based programming, and action-oriented programming. Acorn also provides a graphics module, interface building facilities, power screen generation, and various browsers. It also includes explanation facilities, hooks to standard PC packages, and mouse support.

In addition, Acorn has an on-line help system that provides sophisticated user assistance by developing a model of the user and tailoring help messages. For example, Acorn maintains a history of the user's help queries which provides easy reference to information previously examined. The tool is also context sensitive, providing informative messages about the current data and situation at hand. These facilities, combined with the on-line tutorial developed by San Marco Associates in cooperation with Patrick Winston, should provide new knowledge engineers with one of the easiest ways to learn and systematically upgrade their ability to handle the most complex problems.

Gold Hill used its knowledge of the Intel chip to provide considerable programming power at a very respectable speed. For example, Acorn addresses up to 15 megabytes of memory, compared to other MS-DOS tools which are limited to 640K.

Acorn's open architecture provides a flexible framework for tackling a wide variety of problems. Corporate users can integrate C functions into Acorn systems. In addition, ACORN provides access data stored in dBase II and III and Lotus 1-2-3.

Gold Hill Computers established its reputation by offering the best version of Common Lisp available on the PC. It guaranteed its success by packaging that product with a superior on-line tutorial that has made it a favorite with schools and individuals trying to learn Lisp quickly and efficiently. Gold Hill's strength lies in knowing how to get Lisp to work on Intel chips. Acorn represents a very logical step for Gold Hill. It has developed a large, hybrid tool and has used its knowledge of the Intel architecture to get that system to run on an AT. It has also developed a tutorial to teach developers to use the tool to develop large applications, hoping to capture the high end of the emerging MIS market for mid-size to large expert systems development, in the same way that it captured the PC Lisp market a couple of years ago. Our first look at Acorn suggests that Gold Hill just may be able to do it.

Acorn is being demonstrated at AAAI. It will be available in the first quarter of 1987 for $5,000. Individuals who start now with the GCLISP 286 DEVELOPER will receive full credit toward the purchase of Acorn when it becomes available. A training program and consulting will be available for an additional fee.

(For more information contact Jane Dusza. The address and telephone number of Gold Hill Computers is listed with the other tool vendors on page 23.)

THE 386 HUMMINGBOARD

At the same time that Gold Hill Computer is introducing a PC-based hybrid expert systems building tool, it is also announcing a 386-based plug-in board with memory for a IBM PC XT or AT. This board was jointly developed by Gold Hill Computers and AI Architects of Cambridge, Mass. Hosted in any IBM PC XT or AT, the 386 HummingBoard serves as a coprocessor that is capable of executing GCLISP 5 times faster than an IBM AT. The board is based on the 32-byte Intel 80386 chip running at 16MHz and is specifically tailored to run large Lisp applications quickly. The directly addressable on-board memory is expandable to 24 megabytes via 1Mbyte DRAMS. The 32-byte memory data paths and 2K-line-high speed cache memory yield a processor/memory cycle time characteristic of superminis.

A complete interface for shared memory with the base processor, in accordance with the Intel-Lotus-Microsoft extended memory specifications (EMS) and optional support for the Intel floating point processor (80287 or 80387) is also provided.

In effect, Gold Hill Computer is providing a very serious alternative to Lisp machines and Unix workstations for corporations that want to develop and deliver large expert systems applications on existing IBM PC machines. By combining their existing PCs with the 386 HummingBoard and Gold Hill's GCLISP 386 DEVELOPER, they can have a very respectable Lisp-configured machine. If we imagine a version of Acorn running on the Hummingboard and the possibility of using Gold Hill's run-time utility, it is apparent that Gold Hill is aiming at nothing less than offering developers the power of the current large hybrid tools, which run on Lisp machines, on a PC for a fraction of the cost!

The 386 HummingBoard will be available in the fourth quarter of 1986; the price has yet to be set. For further information, contact Judy Bolger at (800)242-LISP.

CALENDAR

The following AI events will be of special interest to people involved in managing or developing expert systems.

NORTH AMERICA

1986

SEPT 21-25: ***Third Int'l. Conference on Logic Programming.*** Salt Lake City, UT. Sponsor: IEEE-CS. Contact: Gary Lindstrom, Dept. of Comp. Sci., Engineering Bldg., Univ. of Utah, 3100 Merrill, Salt Lake City, UT 84112; (801)581-8224.

SEPT 29-OCT 2: ***ACM Conference on Object-Oriented Programming Systems Languages and Applications.*** Portland, OR. Sponsor: ACM. Contact: Daniel Ingalls, MS 22-Y, Apple Computer, 20525 Mariani Ave., Cupertino, CA 95014.

• OCT 7-8: ***Workshop on High Level Tools for Knowledge-Based Systems.*** (By invitation.) Sponsors: AAAI, Ohio State Univ., and DARPA. Contact: B. Chandrasekaran, Ohio State Univ.-LAIR, 2036 Neil Ave., Columbus, OH, 43210-1277.

• OCT 13-14: ***Second Kansas Conference on Knowledge-Based Software Development.*** Manhattan, KS. Sponsor: Kansas State Univ. Contact: David A. Gustafson, Comp. Sci. Dept., Nichols Hall, Kansas State Univ., Manhattan, KS 66506; (913) 532-6350.

• OCT 14-17: ***Second Annual Conference on Aerospace Applications of A.I.*** Dayton, OH. Contact: Jack Schira, Vitro Corp., 5100 Springfield Pike, Suite 119, Dayton, OH 45431. (513)254-5010.

OCT 20-24: ***Second Annual Conference on Expert Systems in Government.*** McLean, VA. Sponsors: IEEE, Mitre Corp. Contact: Dr. Kamal Karna, IEEE Computer Society, 1730 Mass. Ave., NW, Washington, DC 20036-1903; (202)371-0101.

OCT 23-25: ***Int'l. Symposium on Methodologies for Intelligent Systems.*** Knoxville, TN. Sponsors: ACM SIGART, Univ. of Tenn., Oak Ridge Nat'l. Labs. Contact: Zbigniew W. Ras, Univ. of Tenn., Dept. of Comp. Sci., 8 Ayres Hall, Knoxville, TN 37996-1301; (615)974-8928.

NOV 2-6: ***Fall Joint Computer Conference of the ACM and IEEE Computer Society. (Theme: Exploring the Knowledge-Based Society).*** Sponsors: ACM, IEEE. Contact: Dr. Stanley Winkler, FJCC°86, IEEE Computer Society, 1730 Mass. Ave., NW, Washington, DC 20036-1903; (202)371-0101.

NOV 3-7: ***Knowledge Acquisition for Knowledge-Based Systems Workshop.*** Banff, Canada. Sponsor: AAAI. Contact: John Boose, Boeing AI Center, Boeing Computer Services, M/S 7A-03, P.O. Box 24346, Seattle, WA 98124; (206)763-5811.

• NOV 24-25: ***Conference on A.I./Expert Systems.*** Boston, MA. Sponsor: Suffolk Univ. Contact: Warren G. Briggs, Suffolk Univ., Boston, MA, 02108; (617)723-2349.

1987

• FEB 2-6: ***Third Int'l. Conference on Data Engineering.*** (Special Emphasis on Knowledge-based designs.) Sponsor IEEE. Contact: Third Int'l Conference on Data Eng., c/o IEEE, 1730 Mass. Ave., NW, Washington, DC 20036-1903; (202)371-0101.

• FEB 23-27: ***Third Conference on Artificial Intelligence Applications.*** Orlando, FL. Sponsor: IEEE. Contact: A.I. Conference, 1730 Mass. Ave., NW, Washington, DC 20036-1903; (202)371-0101.

• APRIL 22-24: ***AI'87 (AI & Advanced Computer Technology Conference & Exhibition).*** Long Beach, CA. Contact: Jim Hay, Tower Conference Management Co., 331 W. Wesley St., Wheaton, IL 60187; (312)668-8100.

• JUNE - : ***WestEx '87 (Expert Systems Conference).*** Anaheim, CA. Sponsor: IEEE. Contact: Keith Warn, Rockwell International, 3370 Miraloma Ave. M/S 0B38, Anaheim, CA 92803; (714)779-3174.

INTERNATIONAL

1986

SEPT 23-25: ***AI Europa'86 (The AI and Advanced Computer Technology Conference and Exhibition in Europe).*** Wiesbaden, West Germany. Sponsor: TCM Expositions Ltd. Contact: Tower Conference Mangagement Co., 331 W. Wesley St., Wheaton, IL 60187; Phone: (312)668-8100. (In Europe: (44) 0428-724660; Telex: 859438 TOWER.)

SEPT 29-OCT 2: ***AI BIOMED 86 (First Int'l. Conference on A.I. in Biology and Medicine).*** Montpellier, France. Contact: Organizing Sec., AI BIOMED 86, Centre de Recherche en Informatique de Montpellier, 860 rue de Saint Priest, 34100 Montpellier Cedex, France. Phone: 67 63 0460.

SEPT 30 -OCT 2: ***Second Annual Int'l. Expert Systems Conference and Exhibition.*** London, England. Sponsor: Learned Information. Contact: E.S. Conf., Learned Info. Ltd., Besselsleigh Rd., Abingdon, Oxford OX13 6LG, U.K.; Phone: (0865) 730275.

• NOV 18-20: ***First Australian AI Congress.*** Melbourne, Australia. Contact: Stephen Moore, Director, 1AAIC86. 37-43 Alexander Street, Crows Nest, NSW 2065, Australia. Phone: (02) 439-5133.

NOV 26-28: ***Third Int'l. Exhibit and Symposium on AI and Productivity.*** Paris, France. Contact: Assoc. Francaise d'Intelligence Artificielle et des Systemes de Simulation, 211, Rue St-Honore, 75001, Paris, France.

1987

FEB 17-20: ***Expert Systems in Computer-Aided Design.*** Sydney, Australia. Sponsor: Int'l. Federation for Info. Processing. Contact: Prof. John S. Gero, Dept. of Architectural Sci., Univ. of Sydney, NSW 2006, Australia. Phone: 61-2-908 2942.

• APRIL 6-10: ***AISB 87 Conference.*** Edinburgh, U.K. Sponsor: Soc. for Study of A.I. & Simulation of Behavior. Contact: Chris Mellish, AISB 87 Conference, Arts D Building, University of Sussex, Falmer, Brighton, BN1 9QN, Great Britain.

AUG 23-28: ***Int'l. Joint Conference on A.I. (IJACI-87).*** Milan, Italy. Sponsor: Int'l. Joint Committee on A.I. Contact: Professor Marco Somalvico, Politecnico di Milano, Piazza Leonardo de Vinci 32, 20133 Milano, Italy. Phone: 2 236-7241.

Publisher: Karen Fine Coburn
Editor: Paul Harmon

Advertising Manager: Tomlin P. Coggeshall
Subscription Manager: Charles Gibbs
Production Coordinator: Barbara Shackelford

Editorial Office: EXPERT SYSTEMS STRATEGIES, 151 Collingwood, San Francisco, CA 94114; (415) 861-1660.
Circulation Office: EXPERT SYSTEMS STRATEGIES, 1100 Massachusetts Avenue, Arlington, MA 02174, U.S.A. Phone: (617) 648-8700. Telex: 650 100 9891 MCI UW.
Subscriptions: $247 per year for single subscription (U.S. and Canada); $198 per year for each additional subscription entered at the same time; $307 foreign subscription (U.S. dollars only; air mail). Published monthly by CUTTER INFORMATION CORP. ISSN 0887-221X