Expert Systems: A View of the Field

John Durkin, University of Akron

REPORTS OF THE DECLINE OF EXPERT SYSTEMS ARE GREATLY EXAGGERATED. SURVEY RESULTS INDICATE IMPRESSIVE GROWTH, AS RESEARCHERS DEVELOP SYSTEMS TO TACKLE DIFFICULT BUT COMMERCIALLY REWARDING PROBLEMS.

DURING THE MID-SIXTIES, RESEARCHERS at Stanford set into motion a series of events that would give birth to a new industry in the field of computer science. Their aim was to develop a computer program (later called Dendral) that could elucidate the structure of complex molecules from mass spectrograms at a performance level rivaling that of human experts. Their approach, which was unique at the time, was to encode knowledge obtained from an expert chemist and use it as the driving force of the program. The project's success highlighted for the first time that an intelligent computer program could be developed when the emphasis was on what the program knew about the problem, rather than on clever search algorithms. The era of the expert system had begun.

Having been fortunate to be present at the birth of expert systems, I was curious to learn how well the technology had matured. The birth began with optimism and hope. Visions of powerful knowledge-processing programs fulfilling numerous difficult human decision-making tasks were abundant in the early days. As the initial systems emerged, flexing their power, these visions appeared to be approaching reality. Later reports also pointed to successes that fortified this belief. However, some critics have recently pointed out the technology's shortcomings in producing meaningful results, or the recent decline in the field. So what is the truth? Where do we stand?

To answer these questions, I surveyed the field's literature[1] to find answers to more detailed questions: How many systems have been developed? Where is the development taking place? How is the technology being used? Has the effort produced worthwhile results? The answers reveal that expert systems have come far, yet still have enormous potential.

An overview of the results

Most of the findings came from an extensive review of magazine articles, conference proceedings, and books. I also obtained many findings from information provided by software vendors, which included reports on systems developed using their software. My survey uncovered approximately 2,500 developed systems. Although the survey was extensive, it was by no means comprehensive. I believe the number uncovered represents about 20% of all developed expert systems—12,500 systems! This number might be small compared to developments in the more traditional computer sciences, such as database management, but it represents an increase that marks the field's success. I'll review this point in detail later.

Application areas. An expert system is inherently a tool to assist human decision making. We apply it to knowledge-intensive tasks that require expertise. Therefore, wherever we find humans doing such activities as diagnosing a system, designing a structure, or tutoring a student, we have also found a home for the technology.

To obtain some sense of where the action was taking place, I categorized (as best as possible) each example from the survey into an application area. I found developed expert systems for a wide range of fields. Figure 1 shows the major application areas that naturally developed—the breadth of the applications is remarkable. The figure also shows the number of developed systems for each area.

Table 1. Types of problems solved by expert systems.

Problem type	Description
Control	Governing system behavior to meet specifications
Design	Configuring objects under constraint
Diagnosis	Inferring system malfunctions from observables
Instruction	Diagnosing, debugging, and repairing student behavior
Interpretation	Inferring a situation description from data
Monitoring	Comparing observations to expectations
Planning	Designing actions
Prediction	Inferring the likely consequences of given situations
Prescription	Recommending solutions to system malfunctions
Selection	Identifying the best choice from a list of possibilities
Simulation	Modeling the interaction between system components

Types of applications. Human experts perform a generic set of tasks when solving problems such as diagnosis or planning.[2] Regardless of the application area, given the type of problem, the expert collects and reasons with information similarly. Expert systems likewise accomplish generic tasks on the basis of the problem type (see Table 1).

Figure 2 shows the percentage of applications for each problem type in Table 1. Many applications employ more than one activity. For example, a diagnostic system might first interpret the available data, and later prescribe a remedy for the recognized fault.

As Figure 2 illustrates, the predominant role of expert systems has been diagnosis. One reason for this result is that this is the role most experts play. Fields such as medicine, engineering, and manufacturing have many individuals who help diagnose problems. Another reason for the large percentage of diagnostic systems is their relative ease of development. Most diagnostic problems have a finite list of possible solutions and a limited amount of information needed to reach a solution. These bounds provide an environment that is conducive to effective system design.

The large percentage can also be traced to the practical considerations of introducing a new technology into an organization. Most organizations prefer to take a low-risk position when considering a new technology. So, they prefer projects that require the minimum resources and have the maximum likelihood of success. Because diagnostic systems are relatively easy to build, they are attractive to firms venturing into the field.

The drop-off from the large number of diagnostic applications to that of some other problem types is dramatic. Two reasons help explain this result. First, tasks such as design and planning are difficult to implement in an expert system because their steps vary greatly between application areas and it is often hard to precisely define these steps. Second, tasks such as instruction, control, and simulation, although they are excellent areas for expert system applications, are relatively new ventures.

Platform and software choices. During the seventies, most expert systems were developed on powerful workstations, using languages such as Lisp, Prolog, and the Official Production System (OPS). This left the challenge of developing systems in the hands of the select few who could afford the platforms and had the patience to learn the complexities of the available languages.

During the eighties, we witnessed the proliferation of PCs and the introduction of easy-to-use expert system development shells. These shells were built to run on platforms ranging from PCs to mainframes. The opportunity to develop expert systems was now in the hands of many individuals from all disciplines.

Figure 3 shows the percentage of applications from the survey that were developed on

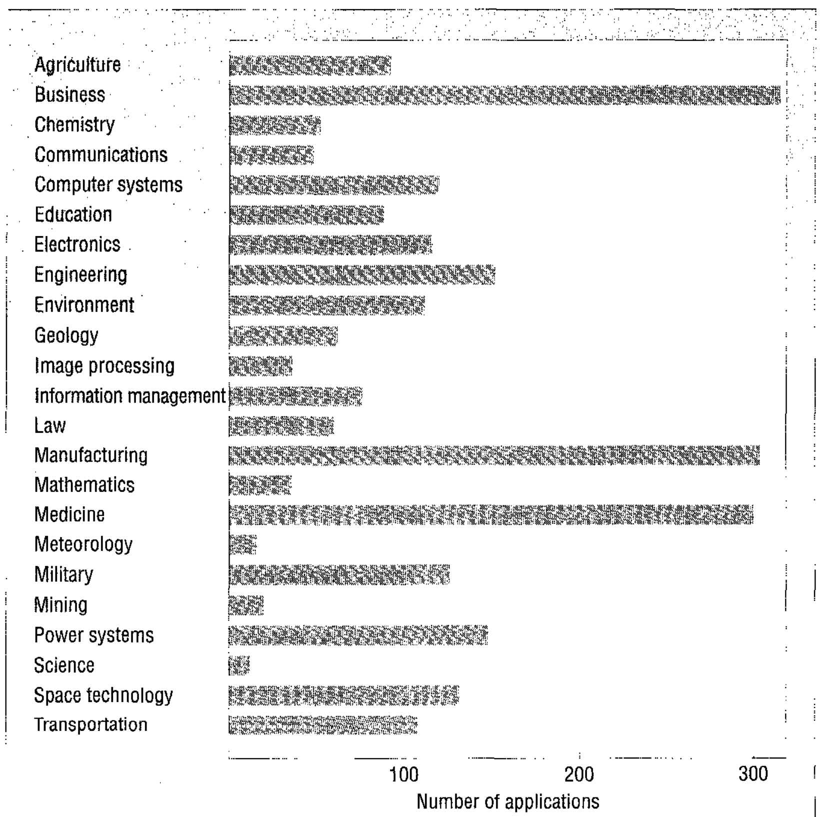

Figure 1. The number of developed expert systems in various application areas.

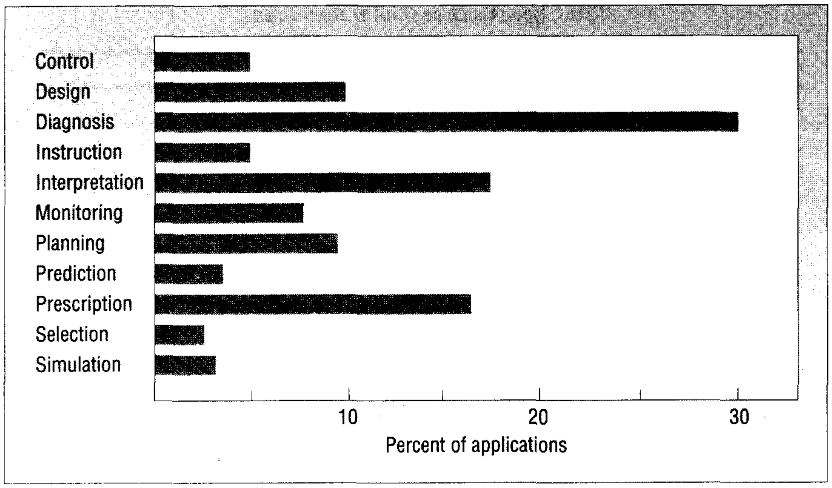

Figure 2. The percentage of expert system applications by problem type.

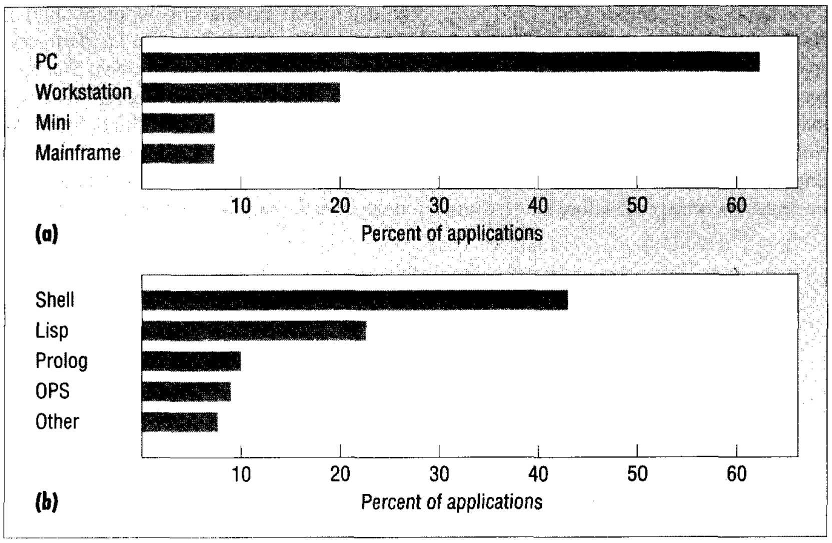

Figure 3. The percent of expert system applications developed (a) on different platforms; (b) with different software.

different platforms and with different software tools. In Figure 3b, "Other" includes the languages C, Pascal, Loops, Fortran, Smalltalk, and Basic. As the graphs illustrate, the vast majority of systems have been developed on a PC with the aid of a shell.

This result also shows that the technology has better aligned itself with the needs of the business and industrial sectors. Typically, managers do not want to purchase special-purpose AI workstations requiring Lisp programmers. Such steps are costly, inhibit the growth of systems in the organization, and often result in stand-alone products that are difficult to integrate with existing computer facilities. Managers want easy-to-use expert system development software that they can easily integrate into existing hardware and software (we'll discuss this integration in more detail later). Today's powerful but easy-to-use shells meet the needs of most managers and have helped spur expert system development.

The state of the field

Let's now turn our attention to the important trends in the field to gauge its health. We'll look at the rate of developed systems since the early eighties, review the technology's major application areas, study the state of the expert system tool market, and see how well the technology has been integrated into conventional software environments. We'll also examine expert systems' relationship with object-oriented programming and their difficult road to acceptance in the corporate world.

Rate of application. One way we can judge the acceptance of any new technology is to track its rate of application. If it has value, we would expect to see an increase. Conversely, if the trend is downward or stagnant, we should be concerned.

During the sixties and seventies, the world of expert systems rested in the hands of researchers whose main focus was on ways to better represent and control knowledge represented symbolically in a computer. The number of systems developed during this period was modest; however, their contribution was significant. Classic systems such as Mycin, Prospector, and Xcon demonstrated the technology's capability. In particular, Xcon and Prospector provided our first glimpse of the technology's commercial potential. Xcon saved up to an estimated $20 million per year, and Prospector aided the discovery of a $100-million molybdenum deposit.

These well-publicized accomplishments were magnets that attracted many organizations looking to capitalize on the technology to the AI field. During the eighties, over two-thirds of the Fortune 1000 companies applied the technology to daily business activities. This expanded interest led to a dramatic increase in the number of developed expert systems during recent years (see Figure 4).

Is this development rate significant? Nathan Rosenberg, an economic historian specializing in new technologies, believes that the pace of the insertion of expert system technology has been rapid by historical measures.[3]

Application trends. Figure 1 showed that expert systems' primary use has been for business, manufacturing, and medicine. Figure 5 shows the number of developed expert systems per year in these areas.

In the early eighties, medical expert system applications dominated the scene. This is primarily due to the diagnostic nature of these applications and the relative ease of developing such systems. However, as we moved toward the mid-eighties, and the low-hanging fruit was picked, it was time to reach for more difficult problems. It was also time to develop systems that benefited the commercial sectors. Unfortunately, initial attempts frequently met with limited success.

In hindsight, three primary reasons help explain this result.

First, early applications of expert systems in industry often over-challenged the technology, leading to poor results. Fascinated by the notion of machines that could think, many designers tried to build systems to solve problems that were beyond even the best experts. The thinking was "Well, we can't solve this problem, so let's try throwing AI at it." Second, other designers often took on a project whose scope was so broad that completing it in a reasonable time frame was impossible. Third, some designers developed remarkably intelligent systems; however, the development was often divorced from an understanding of the client's need to integrate the system into existing hardware and software. The result was a powerful finished product that was left to collect dust on a shelf.

With the few successes being produced during this period, coupled with earlier glowing promises of the technology, critics crept out of the bushes and quickly pounced on the situation. Journal and conference papers, newsletters, and the national media were swift to point to the shortcomings. For example, *Forbes* asked, "What happened to those 'expert' systems that were supposed to transform the world of business forever?"[4] Expert system designers began to realize that finding a place for the technology can be as tedious as matching the glass slipper to Cinderella's foot.

The turning point came in the mid-eighties when designers began to focus on very narrow, well-defined, and sometimes even mundane tasks. They also took the time to look at where the technology would be embedded. Although the systems developed from this focus might have seemed unimpressive to the AI researcher viewing the scene from the ivory tower, they were well received by managers in industry because they produced commercially worthwhile results. For example, consider banking—specifically the granting of loans.

Money lending is one of the main activities of all banks. A bank must decide whether or not to grant a loan by judging the customer's ability to repay the loan according to lending guidelines established by the bank. This task might be repeated often for different customers, requiring select bank personnel who must repeat the same routine steps when processing the loans.

Evalog, a French bank, faced this situation when processing large loans to companies and decided to investigate the application of expert systems to ease the workload. The system they developed, called EvEnt, judges a company's overall credibility and performance when making its recommendation. It considers such factors as the company's financial structure, size, and management performance to evaluate the risk in lending the company funds. EvEnt decreased the cost to process a loan tenfold, helped Evalog process more loan applications, and helped minimize the bank's risk exposure.

Similar successes caused a dramatic swing toward commercially viable systems, and applications for business and manufacturing began to pick up steam. Medical applications continued to grow, but not at a similar rate. We'll explore this point further by performing a simple test of the survey data.

First, let's divide the application areas into two categories: commercial and scientific. The first category includes systems that produce economically beneficial results for organizations in business, manufacturing, power systems, and transportation. The second category includes systems that produce primarily scientific results for chemistry, geology, image processing, and space technology. Next, let's form a ratio of systems developed in the commercial category to those developed in the scientific category and plot this ratio per year. Figure 6 shows the results.

In the early eighties this ratio remained around one-to-one; that is, as many commercial as scientific applications. The mid-to-later eighties showed a two-to-one ratio, while the early nineties showed a dramatic increase toward commercial applications. So, what can we learn from this trend?

During the seventies AI was a cult activity—almost a religion. Researchers centered on producing intelligent general-purpose reasoning machines. The fascination of achieving this academic challenge drove their efforts. By the eighties, when the fuel for the advancement of the technology came from sectors that demanded a return on their investment, researchers began to realize that this is not a religious experience but an economic one. The trend in Figure 6—from laboratory to industrial applications—is one measure we

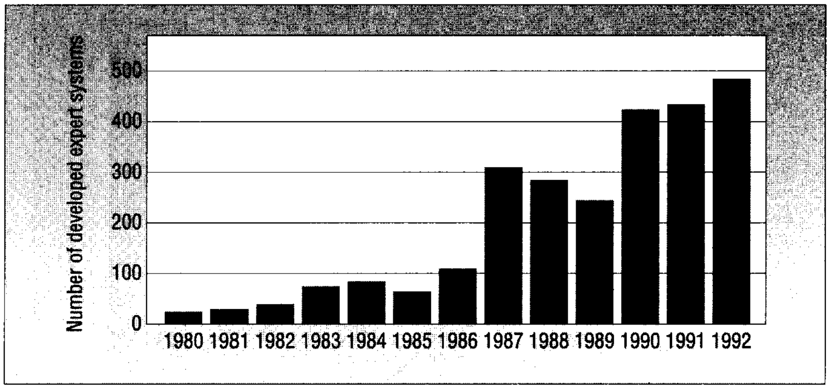

Figure 4. The number of expert systems developed from 1980 to 1992.

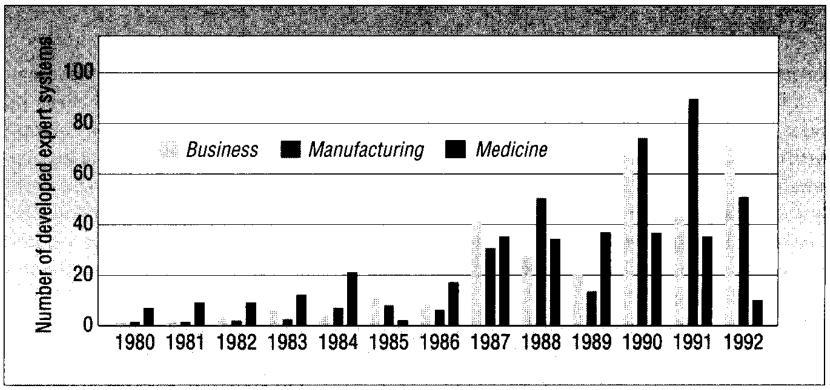

Figure 5. The number of expert systems developed per year for business, manufacturing, and medicine.

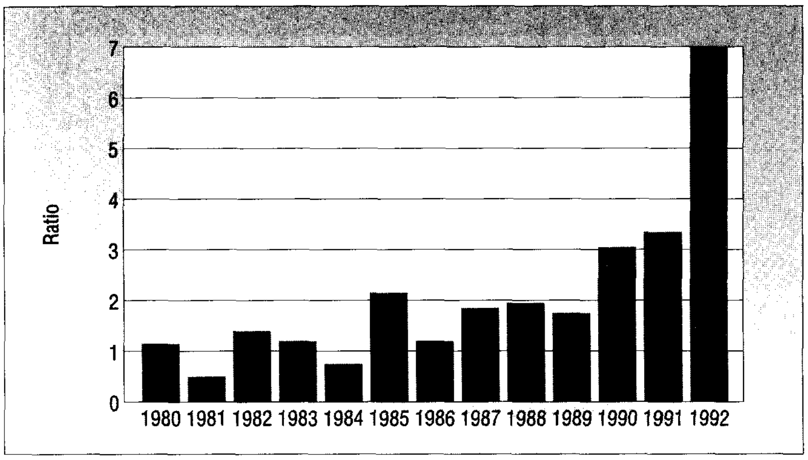

Figure 6. The ratio per year of developed commercial expert systems to developed scientific expert systems.

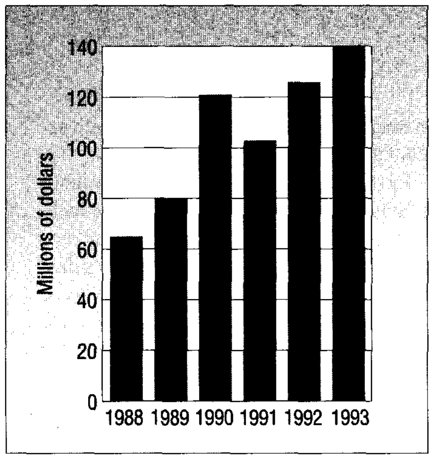

Figure 7. Sales of expert system software tools per year.

can use to judge the technology's value.

A recent confirmation of this trend occurred at the Innovative Applications of Artificial Intelligence conference in Washington, D.C., in July 1993. Of the 16 expert systems that received awards, 10 were related to manufacturing. These 16 systems promised a projected savings of $60 million to $150 million per year.

Our field has moved to the point where we are recognized more for our tangible contributions than our academic ones. This situation is no different than that of our predecessors in traditional programming who took the computer technology of the 1940s, developed by academia and the military, and formed it into a valuable tool for the commercial sectors.

We are now accountable for our actions and would be well advised to consider the words of Ron Roberts, a business-applications advisor with Federal Express. Laying aside the philosophical discussions, jargon, and hype of AI, he asks, "Does it improve my operation, and if so, to what extent relative to the required investment?" The trends seem to indicate that expert system developers are responding to this question.

Health of software vendors. In recent years, some critics have asserted that the expert system software development industry is declining, indicating that AI is on a downswing. They often point to companies that have folded their tents, describe the poor performance of some companies, or discuss the apparent change of heart of some of the leading AI vendors. How, then, can we explain the disparity between this perceived failure of AI and the expanding list of successful intelligent applications?

To begin with, judging a market by the health of its vendors can be misleading. Every new market sees an initial surge in suppliers looking to capitalize on the opportunity, with the eventual shakeout leaving survivors who have learned to revise their products to meet the market's real needs. This is simple survival economics—innovate or evaporate. In the AI arena, the remaining vendors have become an adaptive breed. They have listened to industry's demands for affordable, easy-to-use products that easily integrate into existing computer facilities.

Consider the merger of Aion and AICorp into Trinzic. This marriage was first viewed as a consolidation of two AI giants who decided that it was easier to work together than to battle each other at every turn. However, they were actually positioning themselves better in the marketplace by supplying products with wider appeal. Without mentioning AI jargon, but maintaining AI capability in their product, they offer their clients powerful tools that can be embedded in present facilities to enhance operations. Although companies like this can be perceived as turning away from their AI roots, they can also be viewed as tuning into the market needs to broaden their customer base.

What about the frequent cry of critics that AI software sales have declined recently, implying a weakening of the field? Let's look at the history of tool sales to see if this criticism has any substance.

Until the mid-eighties, sales of expert system development software were primarily to universities, the US military, and a few research groups in major companies. Users of this software were principally researchers who were exploring the technology's potential. Because these tools were mainly for research, rather than for solving practical commercial problems, the number of tool sales was modest.

This situation changed dramatically in the later eighties as the commercial value of deployed expert systems was realized. Figure 7 shows sales figures for expert system development tools during recent years as reported in the monthly newsletter *Intelligent Software Strategies*,[5,6] one of the best publications for monitoring the state of the field. Total sales have grown an average of approximately 16% per year since 1988. According to personal correspondence I received from Timothy Biebelhausen, associate vice president for investments at Kemper Securities, many economists argue that a healthy growth rate for any industry is around 10% per year. Clearly, there is no evidence to support the belief of a decline in AI based on software sales.

There has also been an interesting trend in the types of AI tools being marketed. Most of the tools offered during the eighties were advertised for designing systems for a wide range of application areas. Recently, vendors have begun to market tools to develop systems that serve specific problems and domains, ranging from diagnosis and scheduling applications to applications that support real-time intelligent control in manufacturing sectors. According to a report by Paul Harmon,[5] these types of tools represent the largest growing market for knowledge-based tools. He further reports that the "sales of domain- and problem-specific tools in 1992 totaled approximately $43 million, nearly double the 1991 figure of $21 million."

Integration. Expert systems technology originated in academia, where research was relatively isolated from the realities of business data processing. Systems were developed on specialized AI platforms using software geared to the development of intelligent programs. End products, for all their prowess, often could not run on computers used by the commercial sectors or talk to conventional software. As such, the developed systems served discussions at AI conferences well, but not the world of business.

Today, expert systems are built to merge with the mainstream of information processing and are well attuned to the requirements for integration posed by real business applications. They can interact with databases, spreadsheets, and other conventional programs. Consider Medchec, an expert system that helps audit medical insurance claims.[7]

Medchec, developed at Lockheed, detects fraud and overcharging by searching through billing data in a database to find such items as inconsistencies and overbilling. The system reviews the approximately 800 transactions conducted each day and requires about 25 seconds per claim. The system saves the company approximately $1,500 per day.

Many other examples exist that show how an expert system embedded in established computer facilities can work hand-in-hand with existing programs to enhance operations. The trend toward integration has given rise to an interesting phenomenon: the technology is helping us but we are not even aware of it. Ironically, this situation has fueled the critics who ask where the technology has gone—the technology has become a victim of its own success.

Object-oriented technology. AI owes much to the developments in object-oriented programming. To some extent this debt has been repaid; many ideas pioneered by AI researchers have found their way back into OOP. To appreciate OOP's contributions to AI, consider the history.

The ideas behind OOP date back to the forties.[8] These ideas, however, were not put into practice until the introduction of the Simula_97 programming language.[9] Simula, a superset of Algol, was designed for describing a wide class of discrete event simulations and implementing them for simulations. Simula objects represent data and an operation on the data. These objects communicate with each other through messages to determine their next action. Although primitive by today's standards, Simula provided the first insight into the value of OOP.

The form of OOP we are now accustomed to seeing took shape in the seventies with the development of Smalltalk at the Xerox Palo Alto Research Center. Although Smalltalk is used to develop expert systems, its real value is that it offers a user-friendly programming environment. Much of the early work using Smalltalk focused on the design of user interfaces. Xerox used Smalltalk to develop the interface for the Xerox Star computer, and later Apple used a derivative of the Star to design the Macintosh's interface.

What made Smalltalk easy to use, and conceptually appealing, was the extensive use of techniques commonly found today in OOP languages: class/object representations, inheritance, message-passing, and encapsulation, to name a few. Researchers at Xerox found that these techniques enabled a programmer to easily perceive an object system's structure and operation, and to use this understanding to efficiently develop an interface, or for that matter, an entire functioning program.

TODAY, EXPERT SYSTEMS ARE BUILT TO MERGE WITH THE MAINSTREAM OF INFORMATION PROCESSING AND ARE WELL ATTUNED TO THE REQUIREMENTS FOR INTEGRATION POSED BY REAL BUSINESS APPLICATIONS.

OOP's intuitive approach was the key to Smalltalk's success. Programming solutions frequently followed the methods that humans use to address everyday problems. (Indeed, these techniques can be traced through the history of science as far back as Linnaeus (1707–1778), or even further back to the writings of Plato.[10]) The appeal of an object-oriented environment is evident in Alan Kay's work with Smalltalk. Kay found that it was easier to teach programming to children than to computer professionals.[11] Traditional programmers have been "programmed" in the ways of traditional programming, so they have difficulty changing to the ways of OOP. Being uncontaminated by the procedural computer world, children find OOP a natural extension of the way they solve problems daily.

Given Smalltalk's intuitive programming environment, coupled with AI researchers' interests in computers representing and reasoning with knowledge similarly to humans, it was only natural for these researchers to adopt object-oriented techniques. This trend was most noticeable during the eighties.

One of the most important events during the eighties that spurred the interest in AI was the marketing of expert system development shells. Most of the early shells were rule-based. However, given the appeal of object-oriented systems, as demonstrated by Smalltalk's success, the demand pushed vendors to offer tools with object-oriented techniques. These tools, commonly called *frame-based development programs* (but sometimes called *hybrid tools*), usually combine object-oriented techniques with rule-based programming. New procedural languages with object-oriented techniques also surfaced, such as Objective C, C++, Pascal Object, Modula-2, and Lisp extensions such as Scoops, Flavors, Loops, and the Common Lisp Object System (CLOS).

Armed with powerful object-oriented shells and languages, expert system developers took aim at problems that were often out of the reach of rule-based approaches. A review of systems developed during the later eighties and early nineties clearly shows a swing toward object-oriented techniques.[1] This trend was due partly to the availability of relatively inexpensive frame-based shells that ran on a variety of platforms. Two of the earliest frame-based shells, the Knowledge Engineering Environment from IntelliCorp and the Automated Reasoning Tool from Inference, offered AI researchers powerful tools, but were costly and ran on mainframes or workstations, preventing their widespread use. In the mid-eighties, vendors began marketing cheaper object tools, many of which ran on a PC. This situation led to the accelerated development of frame-based expert systems. Most important, it open the door at most universities for teaching OOP techniques to the next generation of AI researchers.

Vigorous development of object-oriented knowledge-based systems continues. Most corporations—including many in the Fortune 500—are focusing on client-server and object-oriented problems. These organizations have come to recognize the technology's value, and have used it improve operations. They have also come to recognize AI in general, and OOP in specific, as a standard way of doing business. Whereas many of these companies first ventured into AI by forming a dedicated group of AI specialists, most of these specialists now work in the more traditional programming departments, where they routinely carry on their trade of knowledge-based programming.

A look at the recent marketing approach of vendors of AI object-oriented tools is also revealing. As any good business would do, these vendors have kept a finger in the air to sense the direction of their clients' interests. They found that although terms such as "AI" and "expert systems" might have fallen out of favor in some circles, their clients still wanted the object-oriented capability of their products. To go with the flow, these vendors began to advertise their products as "intelligent application tools." AI capability was still there, but the idea of AI faded into the background.

This presents an interesting situation: companies using AI but not promoting it, and vendors marketing products with AI capabilities but not advertising it. Although abandoning the AI label, both have created a new infrastructure on which to build the knowledge-based technology that should flourish in the latter part of the nineties. The irony: even if the spotlight is no longer on AI, AI's contributions will continue to positively affect future information processing, only under other labels.

Technology culture. New technology often has difficulty gaining a foothold in an organization. One reason is that individuals might view it with a cautious eye. They might be skeptical of its value and, in extreme cases, resent or even fear its introduction into the workplace. This attitude can prevent its early or even eventual acceptance. If the technology can pass this initial inspection and demonstrate that it is a potential ally for enhancing operations, then the chances of its long-term survival will improve and much of the early concerns will fade from memory. A good example of this is the PC as a word processor. As these devices were being installed in offices, secretaries treated them like the plague and sought safety by firmly grabbing hold of their typewriters. When they gave the technology a chance, they quickly saw its benefits and retired their typewriters to collect dust on the shelf.

During the past decade, expert systems technology has been emerging from its R&D phase and is steadily taking a position in the corporate world. However, given its innovative nature—particularly with its ties to the AI label—it receives very close initial inspection. In many cases, it is accepted on probation while the company studies its impact on both operations and the corporate culture.

One of the earliest and best examples of the rough road that an expert system project might travel is John McDermott's experience when developing R1, later called Xcon. Xcon was an expert system to help Digital Equipment Corporation personnel configure VAX systems. An expert system approach to computer configuration represented an entirely different style, one that some management personnel met with skepticism. McDermott reported,

THE IRONY: EVEN IF THE SPOTLIGHT IS NO LONGER ON AI, AI'S CONTRIBUTIONS WILL CONTINUE TO POSITIVELY AFFECT FUTURE INFORMATION PROCESSING, ONLY UNDER OTHER LABELS.

> The only reason we were asked to build the expert system was that the problem was bothering some people enough so that they were willing to try anything.[12]

Xcon became a huge success, partly because McDermott was sensitive to the concerns of individuals in DEC about the new technology. He proceeded in a low-profile manner: never promising more than could be delivered and providing frequent demonstrations that illustrated the system's value. Following successful demonstrations, he found that the skepticism was replaced not with belief, but with caution and hope. He was also fortunate to be working closely with technical personnel who were receptive to a new technology and were willing to accept any small failures along the way. Xcon's achievement also caught the attention of others in DEC, which led to the development of other expert system applications. A similar story unfolded at DuPont, whose venture into the field began with the development of a few small expert systems, leading later to the design of over 200 systems.

Future successful deployments of expert systems will probably follow the course of the past. Companies new to the technology will seek potentially cost-effective places for the introduction of expert systems that have some benefit but low risk. Learning about the technology's applicability is of primary importance to these organizations. Companies that have already successfully tested the water will likely expand their program with additional applications, much in the same way as DEC and DuPont.

SINCE THE FIRST SUCCESS STORIES produced by computer programs fueled by knowledge, much has been expected and much has been delivered. We have witnessed a technology mature from a laboratory curiosity into a valuable tool for assisting human decision making. We have seen it applied over a wide range of applications: from helping mine managers plan mining activities, to helping farmers avoid pest infestations, to controlling life-support systems aboard a space station. We have also seen the birth of a new industry—one based on knowledge programming: AI vendors, AI consultants, AI journals and conferences, AI special interest groups, AI research groups in companies and governments, AI courses in universities, AI government funding, and AI development in major corporations. The credit for this success story goes to two sources: the people and the technology.

From the time the developers of Dendral first met with the expert chemist, through the time Prospector discovered a valuable mineral deposit, the early developers of these systems labored to make this new and sometimes fragile technology work. We often take for granted today the techniques first pioneered by these individuals. We need to also recognize the later practitioners who molded the technology into an effective tool for addressing real-world problems.

In this article I've focused on the technology's success in the commercial sectors. Many developers have leveraged the technology to increase productivity and profits through better business decisions. During the past decade, the technology has won many converts in these sectors. For example, Ron Roberts of Federal Express believes that AI may be the most significant influence in how businesses use automation since the invention and introduction of the computer.

References

1. J. Durkin, *Expert Systems: Catalog of Applications*, Intelligent Computer Systems, Inc., Akron, Ohio, 1993.
2. F. Hayes-Roth, D.A. Waterman, and D.B. Lenat, *Building Expert Systems*, Addison-Wesley, Reading, Mass., 1983.
3. E. Feigenbaum, P. McCorduck, and H. Nii, *The Rise of the Expert Company*, Times Books, New York, 1988.
4. R. Simon, "The Morning After," *Forbes*, Vol. 140, No. 8, Oct. 19, 1987, pp. 164–168.
5. P. Harmon, "The Market for Expert Systems-Building Tools," *Intelligent Software Strategies*, Vol. 9, No. 2, Feb. 1993, pp. 3–13. The publisher is Cutter Information Corp., 37 Broadway, Ste. 1, Arlington, MA 02174-5500.
6. P. Harmon, "The Size of the Commercial AI Market in the US," *Intelligent Software Strategies*, Vol. 10, No. 1, Jan. 1994, pp. 1–6. See reference 4 for the publisher's address.
7. B.L. Bivens, "Expert Systems in Auditing and Aerospace Engineering," in *Intelligent Systems in Business: Integrating the Technology*, J. Richardson and M. DeFries, eds., Ablex Publishing, Norwood, N.J., 1990, pp. 127–128.
8. K. Nygaard, "Basic Concepts in Object-Oriented Programming," *Sigplan Notices*, Vol. 21, No. 10, Oct. 1986, p. 117.
9. O. Dahl and K. Nygaard, *Simula—A Language for Programming and Description of Discrete Event Systems, Introduction and Users Manual*, Norwegian Computing Center, Oslo, Norway, 1967.
10. P. Wegner, "Classification in Object-Oriented Systems," *Sigplan Notices*, Vol. 21, No. 10, Oct. 1986, pp. 173–182.
11. A. Kay, "Microelectronics and the Personal Computer," *Scientific Am.*, Vol. 237, No. 3, 1977, pp. 230–244.
12. J. McDermott, "R1: The Formative Years," *AI Magazine*, Vol. 2, No. 2, Summer 1981, pp. 21–29.

John Durkin is a professor of electrical engineering at the University of Akron, where he established and has led the Program of Research for Innovative Applications of Knowledge-Based Systems. He is also a research associate with Intelligent Computer Systems, Inc., Akron, Ohio, an R&D firm specializing in AI applications. He received his BS in electrical engineering from Pennsylvania State University in 1970, and his MS and PhD in electrical engineering from the University of Pittsburgh in 1973 and 1983. He can be contacted at the Dept. of Electrical Engineering, College of Engineering, Univ. of Akron, Akron, OH 44325-3904; jdurkin@dax.cc.uakron.edu.