Expert Systems: A Technology Before Its Time

by Avron Barr and Shirley Tessler

The commercial rollout of expert systems has not been what we envisioned back in 1980, but things are not as bad as they seem. True, investors lost big money in expert systems start-up companies. And, yes, many people's careers, both AI research folk and corporate technologists, took a serious detour. And the trade press has indeed used the words “another AI” as a metaphor to describe other overly-hyped new technologies. Yet, people are still selling, using and benefiting from commercial applications, including new types of applications we hadn't imagined. The current press coverage is positive and relatively well-informed. Furthermore, as standard-issue PC's become more powerful and more networked, the potential platform for knowledge-enabled applications, which once required high-end workstations and avant-garde systems integration, is mushrooming. And the mystery and magic of AI's mind-machine dream is still quite alive in stubborn old-timers and, more importantly, among the new students of computing, engineering and business.

Expert Systems in the 1980's

Much has been written about what happened in the commercialization of expert systems in the 1980s. (We'll use the term “expert systems” to refer exclusively to the underlying technology (see below) and “knowledge systems” or “knowledge-based systems” to refer to the applications of all sorts built using this technology.) We think that expert systems technology was commercialized before its time. Its initial roll-out was overly-influenced by its AI-laboratory origins, and rushed by the venture capitalists' equity market-driven timetable. Its introduction was further stymied by an unrelated but paralyzing revolution which was underway in corporate information systems (IS) departments. AI made the cover of Newsweek and then failed to mature within the 5-year time frame that constitutes the “long term” in contemporary American business.

1 This article appeared in AI Expert, June, 1995.
3 See V. Vasallos and S. Venkatasubramanian, 1995, Technology transfer in the software industry: the case of AI, Stanford Computer Industry Project, unpublished working paper, for a current bibliography and a case study of software technology transfer; and Edward A. Feigenbaum, Tiger in a cage Address to the American Association for Artificial Intelligence Conference, 1993, for an informed opinion about expert systems technology, specifically.
4 As early as the Machines That Think cover article, Newsweek, June 30, 1980.
What exactly is expert systems technology? We define it as a set of software techniques for embedding substantial amounts of knowledge into programs explicitly. All programs contain knowledge by which the developer informs the behavior of the program. Usually this knowledge is embedded implicitly in the program's code. With expert systems, the knowledge is made explicit by putting it in a separate part of the code that acts like a memo from the programmer to the program. This approach makes it more convenient to encode, test and maintain large amounts of knowledge. The distinctive elements of software development technology that constitute "expert systems" include:

- Programming tools for coding, editing and managing a large repository of special data structures which are used to represent the program's knowledge;
- Software routines for using those data structures to inform the behavior of the program, allowing the program to use the right knowledge from the repository, in the right way, at the right time.
- Utilities for debugging the program's knowledge-based behavior, i.e., conveniently determining during runtime exactly what knowledge (or lack of knowledge) caused the program to behave erroneously;
- Methods for systematically capturing (via interviews, for example), organizing and then encoding workplace knowledge; and
- Methods for changing workplace practices to incorporate the new classes of knowledge-based programs (advisors, schedulers, filters, agents, etc.).

Expert systems technology has matured in some of these areas since it left the AI laboratories in 1980, but it still has a long way to go before these essential tools and methods meet the needs of a large class of software and content developers.

What's Taking So Long?

Somewhere between the early commercial success with first-generation applications, the incessant media and marketing hype, the hot capital markets interested in IPO-ing anything that moved, and our blissful ignorance of the software technology transfer process, there developed a universal expectation that expert systems technology would have an enormous commercial impact in short order. It didn't, and “AI” got a bad rep and began to experience a noticeable “winter” in commercial interest and activity.

---

5 As Michael Schrage points out, this is more an organizational principle for software design than an actual technology per se. Senile dementia for artificial intelligentsia? Computerworld, May 9, 1994, 28:19, p39.

6 Prospector, Dipmeter Advisor and XCON are all examples of first generation expert systems, built with high-priced knowledge engineers on expensive Lisp machines, to emulate expert human problem solvers in critical industrial situations.
Historically, most technological innovations don’t come out of university research laboratories, for good reason. Usually innovations start in the organizations that are using the technology every day. They take the form of minor improvements in tools and methods and accidental discoveries of new uses of the technology. Even when a laboratory-born technology is adopted successfully, it is most often because it is a “better mousetrap,” a better way to do something the market is already doing. Relational databases, for example, took only a decade to mature after commercial introduction because, although they required investment in training and new tools, they fit into the world of corporate database applications in a more-or-less “plug-compatible” way. Expert systems technology was many things, but none of them were plug-compatible with any existing automation activity.

The time was, after all, the mid-1980s, before the days of Business Process Reengineering (BPR), the current buzzword of management consultants. In BPR, before building new business systems, one first examines the existing process, rethinking it in the light of new possibilities created by automation and the changing business environment. While BPR is a child of the ‘90s, we realized 10 years ago that to get adequate returns from technologies like expert systems that were being used to automate or support decision making, those business processes often had to be radically altered. “Organizational change,” the last thing a marketing guy wants on his products’ requirements specs, was often the essential ingredient in the implementation of a useful knowledge system. The time frame was all wrong. Expert systems technology was not going to be an overnight success.

Major technologies rarely are overnight successes. Nathan Rosenberg, a Stanford University economist, points out many examples of technologies, like steam engines and lasers, that took several decades between their first commercial introduction and true market acceptance. Sometimes, the technologies themselves need to evolve further; sometimes they need other supporting or complementing technologies and institutions to emerge, or both.7

Side bar:

Technologies find their own way and take their own time about it. Take the laser, for example. Invented at Bell Labs in 1958, the laser was originally envisioned as a communications technology, but was first introduced commercially in weapons systems and then in medical instruments. It eventually found applications in everything from metallurgy to rock concerts until, with the advent of fiber optics, it became obvious that the major niche for the laser would be in communications, back where it started.

Just as the laser required the invention of fiber optics to realize its potential in communications, expert systems technology required powerful workstations on most desks, a ubiquitous network, rapid prototyping methodology, and a widespread understanding of business process reengineering in order to succeed. Unfortunately, in the early 1980's, these supporting technologies were all 10 years in the future.

But it wasn't just the time frame that was wrong. It was bad timing altogether. During this critical time period for the first commercialization of an AI software technology, the place where all the corporate technology decisions were being made was under siege. In the mid-1980s, corporate IS departments were under attack for being expensive, unproductive and unresponsive. There were widespread layoffs in IS shops across many industries, and the advanced technology and innovative applications groups were often the first to go. Even those who weren't laid off were paralyzed by the repeated cycle of dread and reorganization that occurred every few months and lasted for years. By the early 1990's, R&D groups within IS were all but eliminated in many companies. During this period, there was just no place to introduce new technology into organizations undergoing this IS shakeup, in industries like insurance, consumer products, manufacturing, petrochemicals and banking.

Beware! Scientists Wearing Business Suits Escape from Laboratory

The inventors of expert systems and those of us who left the laboratory to make our fortunes commercializing the technology, had a bad case of “inventor’s myopia.” We knew exactly what the technology was good for — and we were wrong. As is so often the case when technologists bring their work out into the world, we believed that expert systems techniques would be used basically for the same purposes for which they had been developed in the lab: intelligent problem solving in our case, like a human “expert,” hence the name. We could not distinguish the technology from the array of applications that had been explored during its development in the lab. Technologists rarely are able to take an objective view of their inventions. Usually it is the end users, over a period of many years, who evolve the technology. They figure out what parts of it are useful, what else is needed, and other ways to use the technology that the inventors never even envisioned.

And we were more than a little bit arrogant. We insisted, in the beginning, that LISP and LISP machines, for example, were essential parts of expert systems technology — they were not. We thought the systems we could build would solve the most important
business problems — they did not. We thought the existing information systems technology, mainframes and COBOL, was simple and backwards — it was neither. And we assumed that our working environment, powerful workstations with graphical user interfaces all networked together, was commonplace — we were ten years too early.

Basically, we thought we had on our hands a business, a big business, or at least a “solution” to business problems, but all we had was a technology. And moving a technology into widespread commercial use requires much more than finally getting the documentation written for the LISP code.

The State of Knowledge Systems Today

So, what’s become of this attempt to commercialize expert systems technology? Well, to quote Paul Harmon, “Trying to figure out the market for commercial AI is like trying to figure out the size of the market for illegal drugs. No one wants to admit they’re engaged in selling or using AI products.” Nevertheless, Harmon estimates the market for expert-systems tools in 1993 at $350 million in the US, which would put it at perhaps $500 million worldwide. A Department of Commerce report puts the annual worldwide market for all AI products at $900 million, and estimates that 70-80% of the Fortune 500 companies use AI to varying degrees. As with illegal drugs, it is apparent that somebody’s doing it!

Much of the activity in applying expert systems technology has indeed gone underground. The need to find new buyers for their software development products have forced most vendors to “reposition” themselves and their tools: client server, object-oriented, middleware or whatever. Those corporate AI groups that survived downsizing have similarly found that repositioning, and broadening their focus, was strategically advisable. Many newer applications of the fundamental technology don’t look much like the systems we developed in the lab: these systems now sport labels such as document generators and smart forms. And finally, the technology has been embedded in applications, like word processors, that have no apparent link to the AI origins of the expert systems components. For example, the latest version of Microsoft (Redmond, WA) Word, uses explicitly embedded knowledge to help the user correct a wide range of typos and writing flaws automatically.

As you look around, in fact, there is activity in developing several distinct kinds of knowledge systems, each pulling the underlying expert systems technology in a different direction.

The classical, problem-solving system. The classical knowledge system follows the course first envisioned by the inventors. As its name implies, these systems apply their

---


knowledge to a problem-solving task like diagnosis or scheduling. Expert-level performance is based on large amounts of task-specific knowledge, like knowledge about tax law or about aircraft design constraints. That’s why expert systems technology is needed, to conveniently get all that expert-level knowledge into the program, debug it, and maintain it over time. The value of the resulting knowledge system is that it solves hard problems well, or quickly, or at odd hours when human experts are not available.

If you know what you are doing and the economics are right, a knowledge system of this sort can be a powerful competitive weapon with known costs and, these days, little risk. In some industries, such as insurance, banking, and securities and commodities trading, many central business activities like underwriting, credit analysis, fraud detection and computerized trading are now commonly supported by knowledge systems.

So why doesn’t every company have thousands of these systems in place by now? Many industry observers say that the difficulty of knowledge acquisition (eliciting and encoding human knowledge in a timely fashion) accounts for the lack of widespread use. While we recognize that knowledge acquisition is still an immature technology, we believe that the real reason for the slow pace of adoption is that situations where classical, problem-solving applications are appropriate are fairly rare. Sometimes having a human expert solve these problems just makes better business sense.

In many working situations, solving the “problem” doesn’t really require an expert — people just don’t have the knowledge they need at the time they need it, or they can’t find the information conveniently. For example, keeping track of thousands of overlapping and confusing regulations is a superhuman task which can easily be automated using the same technology, expert systems, to retrieve the relevant rules for each situation. Same technology, new application. That’s the way technologies evolve, when users find new ways to apply the technology that the inventors didn’t envision but that fit into the way people think about their work.

**Automating the distribution of knowledge — help desks.** Corporations already attend, as best they can, to the problems they perceive to be critical, including problems about knowledge. They spend a great deal of money, for example, on training programs, procedures manuals (safety, accounting, etc.), policy guidelines (HR, purchasing authority, etc.), sales bulletins, regulatory advisories, product documentation, and on-line information systems. The purpose of all of these very expensive efforts is to make knowledge available to those who need it — knowledge that has been accumulated in one part of the organization and is needed to do jobs in other departments. The best example of how expert systems technology can be applied to the task of corporate knowledge distribution can be found at what we generically call the “help desk.” Tech support hotlines, customer service call centers, corporate network administration, and emergency response centers are all examples of help desks: departments whose job is to offer information in response to specific queries.
Expert Systems: A Technology Before Its Time

Expert systems technology is evolving rapidly at the help desk. Help desk systems push the technology in the direction of communication and knowledge sharing. The shared knowledge need not be high-level expertise, just the knowledge that people need to know to do their jobs competently. These systems are not designed to improve peak performance; rather they are used to raise minimum performance levels for everyone.

The real bottleneck at most of these customer service operations is not troubleshooting itself. The real problem is avoiding reinventing the solution every time someone calls with the same symptoms but gets a different representative on the phone. The economic problem requires a technology for sharing knowledge: a system that need answer only one question: “has anyone seen anything like this before?” It also requires a dynamic solution, since the knowledge is always changing and can’t be “engineered” into a static repository.

For example, Compaq has hundreds of technical support representative (TSR’s) answering customers’ questions on the phone. These TSR’s could never possibly receive, during their tenure in the tech support organization, enough training on all of Compaq’s product lines and all of the variations of users’ problems with them to master everything that the company has published in its documentation. And they need to know even more than that — new information that comes along every day and has yet to make it into any manual or document. Moreover, new TSR’s are coming on board all the time, as a result of turnover or expansion, and they are starting from scratch. Compaq uses a related technology called case-based reasoning to help all of their TSR’s, both new and experienced, share with each other the problems they have solved for their hotline callers. The money here, the high impact possibility for automation, is in reducing the amount of time and training it takes to get a new hire up to speed. This involves knowledge that most everybody already knows, not just the expertise of the most experienced.

Figure 1. Because of the long learning curve and frequent staff turnover at the help desk, it often makes better economic sense to focus on distributing the knowledge that lies below the “three-ring binder level” of common knowledge than on the high-level “expertise” encoded in classical problem-solving systems (shaded area).

---

In summary, the economic problem at most help desks is in communications, not diagnosis. And at the help desk, expert systems technology has evolved from focusing on all the knowledge needed for problem diagnosis to building information sharing systems informed by explicitly-encoded knowledge about when the information might be relevant. With several dozen new software products introduced in the last two years, and more coming every month, help desk applications have become a particularly active area for knowledge systems development. These systems have become popular because, instead of being passive resources for problem solvers, they actively help users find the information they need.\footnote{Most of these systems use a related AI technology called cased-based retrieval. See Avron Barr, Active Information Systems, which appeared as the “Out of the Lab” column in IEEE Expert, July 1993.}

**Knowledge publishing.** In knowledge publishing, knowledge-embedding technology is used to effect the widespread distribution of knowledge. Just as at the help desk, the knowledge is not necessarily expert-level knowledge. These systems allow end users to have access to the knowledge they need. For example, the most efficient way of supporting would-be callers to a product support hotline would be to avoid these calls by giving customers the means to solve their own problems. Both established companies and new software entrepreneurs are getting the idea of using expert systems technology to “publish” knowledge in the form of software products that can actively help the “reader” find the information they need. Some examples:

- For help desk applications, two firms, ServiceWare (Verona, PA) and Knowledge Broker, Inc. (Reno, NV) are now publishing dozens of titles about common problems in many familiar PC products and operating systems. Some vendors, like IBM Personal Systems Products (Austin, TX) and Compaq (Houston, TX), have begun publishing in this form as well. Compaq is even incorporating similar knowledge-based advisors into its products, such as its Presario line of PC’s, when they ship. Remember, this is not just on-line documentation: these knowledge systems can help naive users, who might not even know what questions to ask or what terminology to use, find relevant information.

- TaxCut is a document generation program — it generates your federal and state tax forms. Unlike other tax programs, however, TaxCut has a knowledge base that is used to give advice about how to fill out the forms and what is typical/acceptable in critical entries, much like the advice you’d get from a human tax expert. The program, originally developed by Legal Knowledge Systems (Newton, MA) entrepreneur Dan Caine and is now published by H&R Block.

- Companies like KnowledgePoint (Petaluma, CA) and Lawgic (San Rafael, CA) have developed similar document-generation products for small businesses and for attorneys, respectively. These products are conceived of and marketed as titles from a publisher, but they function as software tools incorporating explicitly-encoded knowledge.
Like help desk applications, knowledge publishing pushes the technology away from problem solving and towards communications and information retrieval.

**Embedded expert systems.** Another direction for expert systems technology involves embedding explicit knowledge within specific modules of conventional programs so that they can be updated and maintained more conveniently. Ten years ago, AI companies selling expert systems technology began to talk about “integration” as a key issue. The thinking was that expert systems were but one component, albeit a central component, of a system that also involved some user interface, some database, perhaps some transaction processing, or simulation code. As time went on, the centrality of the expert systems component was less and less apparent. We recall a payroll system at a major oil company where the “AI” was a little bit of code that processed the pay rules for each time card and the remainder of the code was a traditional database application. However, because the pay rules changed with each union negotiation and government regulation, it was easier to maintain that part of the code as a rule-based knowledge system.

In fact, some of the most successful, and certainly the most widespread, applications of expert systems technology have embedded knowledge so deep within a mountain of other functionality that the expert systems technology is anything but central. Nevertheless, this growing trend of enhancing applications with some behavior that is based on explicitly-encoded knowledge is an important part of the evolution of expert systems. Some examples:

- The intelligent assistants in desktop productivity products, e.g. Microsoft’s Tip Wizard and Autocorrect in Word and Excel, and the intelligent defaults and “experts” in Borland’s (Scotts Valley, CA) Paradox and C++.\(^{12}\) Network General’s (Menlo Park, CA) Expert Sniffer, which, among its many network management capabilities, can use a knowledge base to detect common problems.
- Trilogy’s (Austin, TX) SalesBuilder uses explicitly-encoded knowledge about a company’s product line to configure and check complex orders.
- Teknosys’s (Tampa, FL) Help! uses a knowledge base of thousands of rules about the symptoms of known problems and conflicts in its detection of problems in Macintosh PC’s.

As we have discussed in this section, classical problem-solving applications can be very useful in the right situations, but new types of applications for expert system technology have emerged that do not emulate expert human problem solvers. Help desk applications refocus the technology on distributing knowledge around an organization and sharing knowledge effectively. In both embedded expert systems and knowledge publishing, expert systems technology is no longer even the central component — it is dominated either by other aspects of the software or by the knowledge itself. Developers of these newer applications take the technology for

---

On the Horizon

There are three forces driving the continued development of applications using explicitly-encoded knowledge. Each takes expert systems technology in yet another direction. Each is extremely important to the evolution of computing.

The information web. The deployment of a computer network that links every office and every home to everyone else and gives us all access to unlimited information is already having a big impact on the future of computing. This information web is the new platform — one universal computer. Everyone sees the potential for this new platform and for its “intelligence” in a different way. Whether you look at the possibilities in terms of gathering information efficiently (using “intelligent agents” or “information filtering”), communicating with colleagues (“collaborative engineering”), or just navigating through the vastness of the network, programs that actively help people find information will be based on expert systems technology and explicitly-encoded knowledge, and are critical to the success of the new medium.

Customer-centered corporate computing. The revolution in corporate IS departments which we mentioned earlier was in part due to the rigidity of their highly-evolved systems development methodology. These techniques, which had been developed to ensure correctness, stability and reliability of complex information systems, were breaking down for new applications, causing significant delays and often causing a failure to deploy the new systems required for their companies to compete.

As computers, especially PC’s, moved out from the glass house into every department and onto everyone’s desk, new classes of applications were envisioned. Information systems working in the heart of the corporation, like payroll, accounting and even manufacturing, have a slower rate of change than do systems for customer service and sales support, and other systems focused on the customer and the market environment. These newer, customer-oriented systems could not be specified as easily or with the same detail as conventional information systems. Furthermore, the business managers kept “changing their minds” as external contingencies changed their business.

All of this change and uncertainty just didn’t compute in the waterfall model of software development. While IS held onto the stability-oriented methodology they had developed, the friction between them and the perceived needs of their business unit partners caused many years of tension, and eventually, the downfall of IS from the position of power it had had in many companies.

What was needed was a new methodology: techniques for eliciting systems requirements from business managers based on the way people know and learn, and tools for encoding those specifications into programs that could be modified conveniently as the business situation changed. Yes, to build the applications that will be at the center of competitive automation for the next 20 years, all systems analysts
must learn to be knowledge engineers. Knowledge engineers, not in the sense of the LISP hackers of old, but rather in terms of their attitude toward their business partners’ expertise, and in their tools and methods for organizing and codifying business knowledge in vitro.\textsuperscript{13}

**Knowledge asset management.** There are many kinds of corporate knowledge assets, including patents, trade secrets, enlightened policies, well-worked-out procedures and, of course, the uncaptured experience and know-how of the entire staff and even some of the customers. Although most often these assets don't appear on the ledger books, there is a growing awareness that they are critical to effective management decisions and long-term success.\textsuperscript{14} However, no matter how valuable it is on paper, knowledge can only realize its value if it is applied on the job.

Knowledge-based systems, which offer an effective way to codify and distribute their most important knowledge, also help organizations recognize the true value of their knowledge assets. As corporations realize the implications of their new ability to share knowledge on the network, to communicate among staff and with customers, and to do effectively with knowledge systems what they have long tried and failed to do with training, endless publications, and “passive” information systems, the competitive value of corporate knowledge assets will be seen in a new light. Moreover, when organizations begin to recognize and actively manage their knowledge assets, they are very likely to discover new uses for expert systems technology that help them attend to these knowledge assets more effectively.

**What This Planet Really Needs**

Our bet is that this realization of the value of knowledge, and the power of on-line knowledge systems to realize that value, will first manifest itself in the area of regulatory compliance.

We all complain about over-regulation, and yet regulations are the sole product of our biggest organizations, our governments. We ask those governments to regulate our lives so that we can get what we want while not being disturbed or inconvenienced by others. As our world becomes more complex and our knowledge about the consequences of our actions more complete, we need more regulations. We want more regulations! We just don’t want them to interfere with our lives, to burden our workday, or to put us at a disadvantage to those who ignore the rules or manipulate the rule makers and judges.

\textsuperscript{13} Avron Barr, IS 2000: A New Role for Corporate IS Departments and People, Software Development Conference, San Jose, 1993.


What we need is a vehicle for the rational formulation of policy and the fair administration of the resulting regulations. Fair to everyone. And more participatory than our modern-day interpretation of Jeffersonian democracy, which seems to have become somewhat bogged down and less effective as the US population grew two or three orders of magnitude in two hundred years. In fact, what we need is a system that helps the planet’s several billion people feel that we are justly governing ourselves. We need a way to formulate our goals, make plans, set the rules, elect administrators and, generally, realize our common vision.

What does this grand scheme have to do with our little expert systems technology? Egon Loebner, inventor of the LED and a student of invention, once said that technologies evolve like species evolve. In nature, the evolution of a species is determined by two factors: its genetic potential and the space of environmental niches. Loebner believed that technologies have similar evolutionary pressures: that the evolution of technology is shaped by the possibilities of the inventor’s imagination and the space of human needs. In other words, technologies evolve toward human needs. We believe that networked knowledge systems will be a factor in the future of democratic government because, in the space of human needs, fair and effective government is what we need the most.

Expert systems emerged from the laboratory ten years before its time. Today, however, as a result of networks and more powerful personal computers, and the invention of new applications of expert systems, like embedded knowledge systems, “active” information systems and knowledge publishing, the technology remains alive and well. As the world becomes more networked, expert systems’ time will come.

About the authors:

Avron Barr and Shirley Tessler, principals of Aldo Ventures, Inc., Palo Alto, CA, are knowledge systems consultants, and currently, co-directors of the Software Industry Study at Stanford University’s Computer Industry Project. Mr. Barr was co-editor, with Edward A. Feigenbaum and Paul R. Cohen, of The Handbook of Artificial Intelligence, published by Addison-Wesley.

---

Technologies used in Expert Systems in Artificial Intelligence includes: a. Expert Systems Development Environment. Basically, hardware and tools are included in it. They are...

As a result, we have studied Expert Systems in artificial intelligence. Also, learned characteristics, components, types and benefits along with expert systems applications of AI. As we have images that will help you in better understanding.

**Artificial Intelligence - Expert Systems**

- **Expert systems (ES)** are one of the prominent research domains of AI. It is introduced by the researchers at Stanford University, Computer Science Department.

- The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

**Characteristics of Expert Systems.**

- **High performance.** It considers all the facts and rules, and sorts them before concluding to a solution. This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.

- **Backward Chaining.**