Supercomputer access update

Galen Gruman, Assistant Editor

Representatives from the four university supercomputing centers sponsored by the National Science Foundation met September 10 with the government board trying to devise a national policy aimed at restricting access to supercomputers by communist nations, particularly the Soviet bloc.

The meeting was held after the academic community's outcry against the proposed restrictions, which were advanced without university or National Science Foundation input (see Soft News, September 1985 IEEE Software). Representatives from the centers at Princeton University, the University of Illinois, the University of California at San Diego, and Cornell University held discussions with the Senior Interagency Group on Technology Transfer.

That group's members include under-secretary-level people from the Departments of State, Commerce, Energy, and Defense, the Central Intelligence Agency, the Defense Intelligence Agency, the National Aeronautics and Space Administration, the National Science Foundation, and the National Security Agency.

The interagency group is charged with monitoring the flow of technology from the US to nations abroad and devising policies to limit or halt any flaw considered an advantage, military or otherwise, to "unfriendly" nations such as those in the Soviet bloc.

At the September meeting with university representatives, the government presented its case for restrictions on supercomputer access. The main concern is keeping the Soviets away from the machines, which the Soviets do not have.

Another reason for the proposed limits on access within the US was to put pressure on European allies to improve similar limits against the Soviets, said Arthur Kusinski, assistant general counsel for the National Science Foundation. The interagency group has been negotiating such limits with the Europeans for about two years.

"The universities and the government were not talking," Kusinski said, explaining the academic community's heated reaction. Indeed, the National Science Foundation was also surprised to hear of the limits on supercomputers, including those in the centers it sponsors, he said. The purpose of the centers was to provide common supercomputer access, and the proposed limits at least appear to contradict that purpose.

As it stands now, there are no limits on access. Kusinski expects no proposal to be made until summer or fall 1986. Any proposal would have to be approved by the National Security Council, and then by the president.

One plan being discussed would restrict access to supercomputers by placing limits in the visas of Soviet-bloc researchers. These visa limits would prohibit direct or indirect access to all US supercomputers, Kusinski said. Universities would not be responsible for the enforcement of the visa limits at their facilities, he added.

"We would not accept an edict that occurs without our being involved," said Alfred Brenner, president of the Consortium for Scientific Computing, the Princeton University supercomputer center.

"There appears to be a window of compromise" that continued university-government discussions might lead to regulations acceptable to all parties, Brenner said. On the other hand, he cautioned, "There are lots of possibilities of this falling apart."

The issue is essentially a quantitative one. "Everyone will agree that if one gave away a full year's supercomputer time [about 8000 hours], that would be unhealthy," he explained. "Most reasonable, responsible technical people in the government would say five microseconds [of access] is not a problem."

"The question is 'What is the breakpoint between five microseconds and 8000 hours?,'" Brenner posed.
Supercomputer aids mapping of cold virus

Using fast Fourier transformations and a supercomputer to process their data, researchers from Purdue University and the University of Wisconsin have mapped the atomic structure of a common cold virus, rhinovirus-14. The mapping could lead to the development of vaccines for colds and other viral ailments because it provides insight into how a virus works.

Michael G. Rossmann of Purdue led the research effort. Using a Cornell University synchrotron—which accelerates electrons to near-light speeds—as an X-ray machine with 100 times the resolution of conventional X-rays, the team solved the virus's structure to a resolution of three angstroms.

The X-ray diffraction data was translated into a map of the electrons’ patterns with novel fast Fourier transformation techniques developed by Rossmann, said John Connolly, head of the National Science Foundation's Office of Advanced Supercomputer. The office funded the supercomputing and computer graphics phase of the research.

The final set of computations that determined the virus structure took about a month to complete on a Cyber 205 supercomputer—compared to 10 years without a supercomputer, the NSF said.

Connolly described Rossmann's work as “a straightforward technique to examine viral structures.” The research will be used to examine other troublesome viruses in the search for vaccines against them. There are about 100 strains of cold-causing rhinoviruses to be mapped, as well as the hepatitis and acquired immunity deficiency syndrome (AIDS) viruses. Rossmann will focus his efforts on the latter two viruses.

New British tool center a response to software complexity

Ware Myers, Contributing Editor
The United Kingdom has decided to set up a software tool demonstration center in Manchester, England, next year, based at the National Computing Centre there. Geoffrey Pattie, Britain's minister of state for industry and information technology, made the announcement at the Eighth International Conference on Software Engineering in London, England, on August 28.

"It will provide a center where tool users can gain experience on tools currently in the market and receive impartial advice," said Pattie, whose responsibilities include oversight of the Alvey computer research program.

The center will provide an outlet for products of the Alvey and Esprit software engineering programs, as well as for products of private vendors.

The tool center is a response to the British government's belief that "software has become a crucial enabling technology that has a fundamental effect on the competitiveness of a growing range of industries," Pattie said.

Depending on the elaborate software systems "increasingly permeates the entire infrastructure of our societies, inevitably raising more and more questions about the safety and reliability of those systems," he declared.

Blunt talk. As a result, software quality and reliability have become important concerns of governments. In turn, members of the software engineering community bear "immense responsibility for insuring that the systems they design function correctly," Pattie said.

"To put the point very bluntly—and I suspect that everyone here will accept that a problem exists—too much delivered software is still unsatisfactory," he said. "It is still too often delivered late, costs more than expected, sometimes fails to work in the way required, and quite often consumes excessive resources in what is euphemistically called maintenance.

"Even more seriously, many of today's most advanced systems are of such size and complexity as to be approaching the limits of the human capability to design and build them with any reasonable certainty of correct operation," Pattie told the conference.

When building systems that involve critical control applications, such as nuclear power plants or air traffic control centers, much more than reasonable certainty is required, he asserted. "Total reliability is imperative."

Correction
In the September 1985 issue of IEEE Software, the biographical sketch of Nathan H. Petschenik, author of "Practical Priorities in Systems Testing," stated his company's name incorrectly. Petschenik works at Bell Communications Research, which is jointly owned by the seven regional Bell companies and has no affiliation with AT&T.
Can ‘Star Wars’ software be built?

Successful implementation called improbable at best

Ware Myers, Contributing Editor

Building software to implement President Ronald Reagan’s Strategic Defense Initiative, or “Star Wars” antimissile defense system, is “impossible rather than just unlikely,” said David Parnas of the University of Victoria in British Columbia, Canada.

Not quite so, rejoined Fred Brooks of the University of North Carolina. Brooks is a member of the Department of Defense’s Defense Science Board. “We cannot prove that it is impossible, but I agree that the successful prosecution of this endeavor is indeed improbable.”

“It is very, very unlikely that such a system can be built,” said Alan Perlis of Yale University, the man who coined the term “software engineering.”

Agreeing considerably on general issues—but not on whether ultimately such a system was impossible or only improbable—the three distinguished software engineers analyzed the technical feasibility of SDI software at the Eighth International Conference on Software Engineering August 30 in London, England. “Our collective qualification is technical,” said M. M. Lehman, who organized the session and served as conference chairman.

Members of a government scientific advisory panel invited to the London conference declined to attend. “The panel felt, since it was still deliberating and investigating, it was inappropriate to take a position or be forced to take a position at this time,” an SDI official in Washington, DC, said.

The conference advised their first report in mid-October and planned to attend a Compon Spring 86 panel discussion in San Francisco, California, March 3-6. (The Department of Defense’s positions on the SDI program are detailed in the accompanying segment.)

The need for reliability. “Every engineering product that is reliable now has been validated by one of three methods: mathematical analysis, exhaustive analysis of cases, and realistic testing,” Parnas said. Parnas is a former member of the Eastport Group, a Department of Defense SDI panel on computing support. He resigned after the group’s first meeting upon concluding it was impossible to build successful software for a very large system of this kind.

The fundamental reason for the difficulty in verifying software is that “the tools that deal with systems that can be described by continuous functions are much more powerful than the tools that deal with systems that have discontinuities,” he said.

Case analysis is successful only if the number of possible cases is small or the cases are repetitive. Testing all possible cases of a complex system might take 1000 years.

Moreover, if these methods are to be trusted, the system must operate under either controlled or completely predictable conditions, Parnas continued. SDI operation, however, is neither controlled nor predictable. Rather, its operation is subject to unknowable perturbations introduced by an intelligent opponent.

The consequence of these difficulties in validation is that system reliability normally increases—as bugs are corrected—the longer a system is used. Since SDI technology would not be used regularly, its reliability would remain low.

Future hopes. Will artificial intelligence provide a solution to these problems? In Parnas’s eyes, “expert-system building is just an ad hoc, cut-and-try method of programming. It gives you little basis for confidence because you never know if you have told it enough. I don’t think there is any magic in artificial intelligence,” he said.

Others believe that new programming languages, automatic programming, or new verification methods promise workable solutions. “They [such new methods] will help, but I don’t expect any major breakthroughs,” Parnas responded.

The issue of proving programs focuses on the right problem, but “we don’t have the technology to do it at all.” Advocates of proof methods consider 500 lines a large program. “Even if they were to achieve an order-of-magnitude improvement, they are not near the requirement,” he said.

Underlying the problem of proof is the matter of function continuity. Because software is based on discrete functions, the descriptions do not condense to a compact form that is easy to study. “Because we do not have compact representations of software, program verification is not going to become practical,” Parnas concluded. “I don’t even see any way to approach it.”

The net sum. The University of North Carolina’s Brooks agreed with Parnas’s claim that there is no magic coming over the horizon. In particular, having just spent a year surveying artificial intelligence techniques, Brooks found that the techniques “stand to provide no serious help on this problem in the foreseeable future.”

It follows then, he continued, that “if SDI cannot be done with today’s techniques and technologies, it cannot be done at all.” On the other hand, Brooks did not agree that “the difference between continuous systems and discrete systems is as pervasive as David [Parnas] sees it.”

More fundamentally, “the very argument that we do not have a mathematical or physics basis for software discipline means that we cannot prove things impossible in the sense that we can prove things mathematically impossible by appealing to an axiomatic system or physically impossible by appealing to known physical laws,” Brooks said.

As Brooks saw it, Parnas’s net position comes to the following: “If you gave me the job and all the resources I wanted, I could not do it.”

Brooks’s own position is “I don’t know whether we should do it, but if you gave me the job and all the resources, I think we could do it.”

Official admits difficulties, defends SDI research

Galen Gruman, Assistant Editor

Creating software for President Ronald Reagan’s proposed defensive antimissile system, the Strategic Defense Initiative, requires addressing thorny problems, US Air Force Major David Audley acknowledged, but research into potential solutions should continue until a “Star Wars” system is created or is proved impossible to create.

“The SDI is a research program, and we don’t have preconceived answers to questions we are still seeking to pose,” said Audley, the program manager for battle management and command control and communications in the Defense Department’s SDI Organization. There can be no answers yet in a research program begun just over a year ago, he said.

While Audley acknowledges that the reliability and validation problems discussed by a panel at the Software Engineering Conference do exist and
must be addressed, he disagreed with the panel’s conclusion that the success of such research was improbable and perhaps impossible. (The details of the panel’s discussion are in the accompanying report.)

**Monolithic system not wanted.** “People think of a highly synchronized, end-to-end system” when they think of the SDI, he said. Most popular scenarios depict a space-based, computer-controlled tracking system that coordinates the destruction of incoming missiles through response missiles and lasers.

“That kind of system is impossible. . . . We intend to develop a system that does not rely on a monolithic, 10-million-line, homogeneous software system,” Audley explained. “We need to structure it so it is not so complex.

“We have to have a system that drives down complexity. We need to promote independence of elements in this system,” he stressed. The system must be modular, open-ended, and built with redundancy, he added. The space-based scenario is only one of several, Audley said. He said most elements of a defensive system would likely be ground-based.

Audley compared the government’s envisaged system to the nation’s air traffic control system. That system relies on multiple, independent nodes. Airplanes carry flight control computers that allow independent navigation. Control towers are dispersed throughout the country to guide pilots.

“We could replace all the flight control computers with a bunch of Crays centralized on the ground somewhere,” Audley said. “But we know that it’s ludicrous to even think that way. One problem could cause all the planes to fall out of the sky.”

**Strategic coordination.** No SDI system would be software-intensive, Audley said. Instead, it would “take advantage of the strengths of computer hardware and at the same time minimize the vulnerability of the computing software.”

Audley used the phrase “strategic coordination” to describe the government’s conception of the system. As an example, he described the nation’s strategic offensive system, which consists of independent missiles, aircraft, submarines, and the like. If one of 10 missiles tested fail, military analysts expect an overall 10-percent failure rate. However, those 10 percent failures do not stop the 90 percent successes because each element is independent from the others, despite the fact that all the elements are coordinated, Audley explained.

**Year of the software initiatives**

*Ward Myers, Contributing Editor*

“Those of us concerned with software have been hoping that industry and governments would recognize the need for software engineering institutions,” said Laszlo A. “Les” Belady, vice president and program director of the Microelectronics and Computer Technology Corp., as he opened a panel discussion at the Eighth International Conference on Software Engineering in London, England, on August 29. 1985 saw those hopes recognized, Belady said, leading him to label it the year of the software initiatives. The new organizations seem to be positioning themselves between academic research and advanced development, he noted.

**The institutions.** Belady brought together representatives of nine institutions, including one contributor from the floor. The institutions were:

- Microelectronics and Computer Technology Corp. (MCC) in Austin, Texas,
- Esprit Software Technology Program, supported by 10 member nations of the European Economic Community,
- Alvey Software Engineering Program, established by the British government,
- Joint Software Engineering Program in Singapore,
- Sigma Project, sponsored by Japan’s Ministry of International Trade and Industry,
- Software Engineering Institute, sponsored by the US Department of Defense at Carnegie-Mellon University,
- Software Productivity Consortium (SPC), which will be located in Virginia and supported by major aerospace contractors,
- Software Plant Project in Brazil, and
- a $35-million program in Italy with five projects (reported by a floor participant).

One program, Alvey, began in 1983. Two, Esprit and MCC, started in 1984. The rest began this year. All are long-range and all are planned to operate indefinitely. No significant results are expected for four to eight years.

Several of the US programs are consortia, with MCC composed primarily of computer industry members and SPC of defense contractor members. The foreign programs are government-funded. Some of the institutions will do most of their work in-house, while others will contract out to universities and other institutions.

Most of the results will be generally available—within the limits of national security concerns—but the consortia are bound to transfer technology to their members first, while the national programs will apply their findings in their own countries first.

**Goals and methods.** In general, the programs’ aims are to improve the quality of software products and to reduce the cost of software development. The national programs aim to achieve competitiveness in international markets, as John Elmore of Esprit pointed out. Similarly, the goal of Singapore’s Joint Software Engineering Program is to make that country the software center of Southeast Asia, Lim Swee Say affirmed.

The institutions plan to achieve these goals through precompetitive research, development to the prototype (or, in some cases, the marketable) stage, and transfer of the resulting technology to program clients.

The panel members emphasized that the research effort is meant to cover the entire software development cycle, from capture of specifications to production of code and on through maintenance.

Esprit’s Elmore and Alvey’s R.W. Witty stressed research in formal design techniques. Witty and Say mentioned software metrics. Other activities cited included reusability of software components, rapid prototyping, integration of software environments, and standards development.

The Japanese Sigma Project plans to establish a nationwide computer network to connect major software houses, service bureaus, and large computer users. Several service centers on the network will develop and distribute tools within a standard environment based on Unix.

Several programs intend to study the applicability of knowledge-based systems to software development. Among other efforts, the Software Engineering Institute is analyzing the legal problems of software licensing and is coordinating a coherent set of courses for the masters degree in software engineering.

Unlike earlier research institutions, which often ended their work upon publication, the new software institutes are charged with transferring the outcome of their work to their members within their countries. Various transitioning means are planned, ranging from prototypes to marketable tools and environments.
Space factory contains own control software

Galen Gruman, Assistant Editor

McDonnell Douglas will place a fully automated space factory in a space shuttle cargo bay in summer 1986. The factory will contain its own control software, developed by Robert J. Wood, a lead software engineer and network architect for the production prototype.

The remote-processing factory will make purer pharmaceuticals than is possible on Earth through electrophoresis, a separation process that uses an electrical field to separate biological and chemical substances.

All such substances have inherent electrical charges and separate in relation to those respective charges when in an electrical field. The lack of gravity makes such separated substances about four times as pure as those separated in full Earth gravity.

The language used to develop the factory's software was a version of Forth, an interactive metalanguage "well-suited to process control and data acquisition issues," Wood said. "I can do a lot more with the payload because I can modify the program during flight."

The language's quick diagnostics translated into a 40 percent savings in the space factory's software development time, an important issue for commercial space applications, Wood said. "It was significantly less expensive than comparable Ada or Pascal systems," he added.

Wood will be on board the 1986 shuttle mission as payload specialist to monitor and fine-tune the system, which has been designed for commercial-scale production. McDonnell Douglas expects to orbit the factory on three to six flights a year, Wood said.

Wood will also be an alternate, ground-based payload specialist for a space shuttle mission late this month that will orbit a smaller-scale (about four percent output of the commercial-scale factory) version of the factory that has been tested and been in limited production during six flights since 1982.

The automated factory's computer will evaluate new pictures of the electrophoresis process every few seconds. As each image is evaluated, the software locates newly produced material and controls the movable collection mechanism to pick up the material. Support modules control the filtering of buffer solutions, the material flow, and the cooling of the factory.

A shuttle would put the factory in low Earth orbit (roughly 300 miles above the planet), and another would retrieve it and its refined contents a half year later.

While data will be sent to ground stations from the unattended orbiting factory, McDonnell Douglas decided to have the factory monitor itself to avoid the problems caused by the split-second delay in ground-to-space communications and by the periods in which the satellite would be out of receiver range.

"The computer can easily recognize when the process is wrong," Wood said. However, "It's easier to tell if something is wrong than to make it right again," he added. Should a problem arise, commands from the ground would be sent to reconfigure and continue processing. Severe problems could require a space shuttle to pick up the orbiting factory earlier than planned so an on-board specialist could try to fix the problem.

Longer-range plans include a version of the factory for the US manned space station, construction of which is scheduled for the early 1990s. As the program develops, "we will consider inference machines to look at the recurring, more complex problems so we don't have to deal with them again and again," Wood said.
IJCAI program reflects ‘substantial progress’

Marilyn Potes, Managing Editor

Solid contributions based on prior work, rather than major breakthroughs, were characteristic of the technical program at the International Joint Conference on Artificial Intelligence. According to the program chairman, Aravind K. Joshi of the University of Pennsylvania, this building on the theoretical foundations of the past indicates the growing maturity of the field.

Joshi’s remarks at the wrap-up press conference on the final day of IJCAI, held August 18-23 at the University of California at Los Angeles and attended by 3200 people, were reinforced by the general chairman, Alan K. Mackworth of the University of British Columbia.

Mackworth mentioned eight topics “representing areas of substantial progress” reported at the conference: theory, robotics, vision systems, personal computer systems, AI learning, qualitative reasoning, connectionism models, and expert systems.

Among the papers Mackworth cited as contributing to the theoretical framework of artificial intelligence were “Belief, Awareness, and Limited Reasoning” by Ronald Fagin and Joseph Y. Halpern of the IBM San Jose Research Laboratory; “Self-Knowledge and Self-Representation” by John Perry of Stanford University; and “An Essential Hybrid Reasoning System: Knowledge and Symbol Level Account of Krypton” by Ronald J. Brachman of AT&T Bell Laboratories, Victoria Pigman Gilbert of Schlumberger Palo Alto Research, and Hector J. Levesque of the University of Toronto.

For his work in knowledge representation, Levesque received IJCAI’s 1985 Computers and Thought Award, a biennial award that recognizes an outstanding young scientist working in AI. Fagin and Halpern received the MIT Press Publisher’s Prize for the best paper at IJCAI. They presented mathematical models that account for the imperfect reasoning inherent in human logic.

The program also featured “some nice work on robotics,” Mackworth said. One paper, “Building a Bridge Between AI and Robotics” by Hiroki Inoue of the University of Tokyo, described a parallel arm implementation that can be positioned in any orientation and is fast and easy to control.

Advances in AI small systems were especially evident at the trade show, which featured “very practical work on expert systems on PCs,” Mackworth said. “An important development for us as a community is Common Lisp,” he continued. “We now have standard programming environments accepted by all the major manufacturers in support of all major research labs, and we’re getting tools built on top of that.”

AAAI president assesses research, calls for ‘bold experiments’

Artificial intelligence research is “headed generally in the right direction” but needs “bold experiments” rather than “bold bragging,” said Woodrow W. Bledsoe, outgoing president of the American Association for Artificial Intelligence. Bledsoe’s remarks were part of his AAAI presidential address, delivered Monday evening, August 19, at the opening session of the Ninth International Joint Conference on Artificial Intelligence.

In his speech, titled “I Had a Dream,” Bledsoe described a vision he had 25 years ago—one filled with the “wild excitement of seeing a machine act like a human being, at least in many ways.” This “daydream” led to his “calling” and 25-year involvement in AI research. Currently, while on leave from the University of Texas at Austin, Bledsoe is a vice president of the Microelectronics and Computer Technology Corp. where he directs the AI/ Knowledge-Based Systems Project.

“These 25 years have not been totally kind” to his dream, Bledsoe acknowledged, but he remained in the AI fold because they “have also been fruitful and exciting. We have much to be proud of,” he continued, “with much left to be done.”

To date, most AI accomplishments have been of the partial kind, Bledsoe noted. For example, natural language processors handle only a subset of English or French. Character recognition machines read only typewritten characters. Programs exhibit only an elementary level of learning and reasoning by analogy. But these partial results, he continued, “have helped unearth the roadblocks that stand between where we are and where we want to go.”

Bledsoe favors the bold approach in AI research. He cited the mechanical translation work of the early 1960s as an example of a “bold experiment that had to be made.” Although it “seems rather obvious now that you cannot have mechanical translation without language understanding,” it was these early experiments, he noted, that helped focus research on natural language processing, currently an exciting area for research that is leading to the resurgence of mechanical translation.

In the next decade, new “super expert systems” will absorb a large percentage of the research and development effort, and rightfully so, according to Bledsoe. He sees these as expert systems that have been endowed with large structured knowledge bases, ability to reason through various causality levels, limited ability to learn automatically from experience and to accumulate knowledge by analogy, truth maintenance systems, and enhanced human interfacing to facilitate knowledge acquisition.

Thus, one of the challenges for AI researchers is the development of common-sense knowledge bases to supplement the textbook and expert knowledge now used in expert systems. Other challenges mentioned by Bledsoe include rethinking the learning and analogy process and improving qualitative reasoning to allow greater flexibility in levels of causality.

See pp. 86-88 for announcements of new products exhibited at IJCAI.
**Expert systems: How far can they go?**

*Marilyn Potes, Managing Editor*

Expert systems function well as paraprofessionals within a systematic, well-defined domain, according to Terry Winograd of Stanford University, chairman of an IJCAI panel, “Expert Systems: How Far Can They Go?” But the expert system user must, he emphasized, understand the domain and be responsible for all decisions, especially in critical situations.

The limitation Winograd established—that they be used only by experts to handle particular aspects of their jobs—was virtually unchallenged by the other panelists: Stuart Dreyfus of the University of California at Berkeley, Brian Smith of Xerox PARC, and Randy Davis of the Massachusetts Institute of Technology.

There were, using Winograd’s definition, no “TV evangelists” on the panel. Winograd had opened the Thursday, September 13, session in UCLA’s Royce Hall by assuring the audience that this would not be a “religious discussion” of expert system potential involving “not possible” atheists, “not sure” agnostics, or “send in your money and receive miracles” TV evangelists. He promised instead to focus on practical questions, the larger view, and the body of techniques.

There is still uncertainty—among professionals, press, public, and users alike—as to what expert systems can and cannot do, Winograd said. Although he acknowledged the possibility of an “AI winter or dark age” because of initial “oversell,” his particular concern was that misplaced confidence might lead to overexpectations and obscuring of responsibilities. “If the knowledge base were called an ‘opinion base,’” he said, “perhaps things would be different.”

As it is, the knowledge engineer creates an artificial domain with the expectation that it will correspond to the real world. But, because he focuses on some things and drops others, the expert system he creates have a very narrow structure. This structure works well for precise technical areas, but when applied to other areas, it “creates an inevitable blindness” in a system that has “no way to step back and apply common sense.”

The appeal of rule-based systems is that we can seemingly add more rules to cover omissions. “But that’s wishful thinking,” Winograd said, “We’ve already created the blindness.”

Dreyfus reinforced this evaluation, saying, “Rules cannot get to intelligence.” People have what rule-based systems will never have: intuition. People have “the ability to see situations, recognize similarity to previous situations, and, based on what worked/didn’t work, will know what to do/not do.”

As an example, Dreyfus cited an experiment in which world-class chess players simultaneously added a column of figures, at a rate predetermined to require full attention, and beat the best-available chess-playing program. “Expert systems cannot be viewed as a repository of human expertise. They do not respond as well,” and, “Dreyfus added, “there is no reason to be optimistic about their doing so.”

Dreyfus placed expert systems at the third of five skill levels, which he defined. The first level, that of the novice who is taught context-free facts, is attained by computers. The second-level advanced beginner thinkers in terms of plans and goals. Expert systems reach the third level, competence, of conscious and deliberate application of facts. They cannot duplicate the skills of levels four and five: the proficient performer, who intuitively recognizes similarities, and the expert, who knows the sense of a situation and what to do about it. Thus, domains requiring only level-three abilities—for example, the VAX system-planning function of DEC’s R1—are good areas for expert systems, Dreyfus concluded.

Smith, the third speaker, remarked that panel agreement was “not as close” as it might appear. All models deal with a certain level of abstraction, he pointed out. “They have to ignore things, that’s how they work.” We must live in the gray area in between the model and the real world, he added, relying on people to make the flexible assessments that expert systems cannot make for them.

Davis maintained that the limitations of expert systems “should not disqualify them.” Human experts also develop “tunnel vision,” he reminded.

“The technology is sufficiently new to have low predictability,” Davis said. It is our fate to deal with incompletely understood tasks, but expert systems provide a useful set of tools for dealing with ideas while suppressing uninteresting detail.

To avoid the benefits of expert systems yet avoid the pitfalls, Davis recommended that we:

1. recognize the inevitability of failure, giving the machine the freedom to fail;
2. choose problems wisely, building off-line training systems and assistants instead of experts; and
3. keep people in the loop.

Similar guidelines were endorsed by all the panelists, each of whom mentioned proper domain selection and human decision-making as essential criteria.

The panel’s view of the future for expert systems included the concept of assistance; panelist Smith can continue to be “glad a computer is helping the pilot land the plane.” It did not, however, extend to areas involving critical decisions; that could lead to scenarios where, as Winograd put it, an expert system “plots a perfect chess move while the room is on fire.”