IEEE supports 'innovation incentive' in court case

Citing the importance of "incentive for innovation," the IEEE has asked the California Court of Appeals to reverse a lower court's ruling on ownership of intellectual property.

While the case between Cubic Corp. of San Diego and William B. Marty debates when ownership originated on an electronic warfare training device, the outcome could affect pre-invention assignment agreements for software producers as well.

A state court in San Diego had concluded intellectual property, such as the training device, is not invented until a patent application had been filed or the property was "reduced to common practice."

The IEEE contended to the appeals court that the ownership of the patent was "determined at the time of conception," in this case at the time the inventor prepared a manuscript describing the invention.

The IEEE further contended the lower court's interpretation of a California law restricting employers' pre-invention assignment results. The court had said the law did not apply in Cubic Corp. v. Marty. The law "was enacted to protect the employee," said the IEEE's November 22 friend-of-the-court brief to the appeals court. "However, if the viewpoint of the trial court is adopted, the statute will be emasculated."

The California law excludes any provisions in pre-invention assignment agreements that require an employee to assign his rights in an invention if the invention (1) was developed without using the employer's equipment, supplies, facilities, or trade secrets; (2) was developed on the employee's own time; (3) does not relate to the employer's business or actual research effort; and (4) does not result from work performed by the employee for the employer.

Meanwhile, the IEEE US Activities Board adopted a statement establishing disclosure standards for all pre-invention assignment agreements. The required terms and covenants of such agreements should be disclosed before or at the same time paid employment is offered, the statement said.

NSF helps colleges begin research centers

The National Science Foundation's Coordinated Experimental Research Program has granted nearly $12.5 million to five universities to help establish experimental computer research centers.

The NSF gave $4,682,247 to the University of California, $3,197,254 to the University of Colorado at Boulder, $2,166,099 to Princeton University, $1,495,604 to the University of Minnesota, and $920,868 to the University of Washington. Except for the University of Washington's two-year grant, the funds are earmarked for five years.

The Massachusetts project will aid research in cooperative distributed computing based on a tightly coupled multiprocessor with 64 or 128 elements. The Colorado program will help establish a loosely coupled network of scientific workstations.

The Princeton effort will investigate supercomputers with tens of billions of bytes of physical memory. The hypothesis is that such machines may change in fundamental ways how certain classes of problems are solved and may lead to major performance improvements. Cosponsors include the NSF's Office of Advanced Scientific Computing, the Defense Department's Advanced Research Projects Agency, and the Office of Naval Research.

Minnesota's project is concerned with using supercomputers to design and analyze algorithms and applications software. The Washington effort will study distributed computer systems, focusing on operating system software that allows tight transparent resource sharing among essentially identical workstations and on the problems of a significant but looser integration of dissimilar workstations.

Frank Mathur has joined the IEEE Software Editorial Board.

Magazine adds board member

Frank Mathur, a professor of mathematics at the California State Polytechnic University at Pomona, has been appointed the 16th member of the IEEE Software Editorial Board, Editor-in-Chief Bruce D. Shriver announced.

He has assumed the editorship of the New Products and Product Highlights departments. The former editor, Richard Eckhouse, Jr., of Moco, Inc., has become editor for the New Product Reviews department, which he started as a section of New Products last July.

Mathur invented the hybrid-redundancy technique and developed the Computer-Aided Reliability Estimation program for the National Aeronautics and Space Administration. Mathur is a consultant with the Jet Propulsion Laboratory for artificial intelligence, spaceborne computing, and reliability modeling.

He has published extensively in fault-tolerant computing. Mathur received his PhD in computer science from the University of California at Los Angeles in 1970.
Aerospace conference survey AI technology

Galen Gruman, Assistant Editor

The difficulties of applying artificial intelligence systems to the increasingly complex aerospace industry programs was a predominant theme at the American Institute for Aeronautics and Astronautics Computers in Aerospace Conference held in Long Beach, California, in late October.

The “Future Directions in AI” panel focused primarily on the emerging technology’s applications. The reports presented showed that much more needs to be done before AI can be used to dramatically improve software creation, flexibility, automation, and reusability.

Most expert systems follow essentially the same approach: Data is evaluated against a knowledge base and a rule base. An inference engine decides how to correct contradictions between what is expected to happen and what is happening.

A user interface connects this looping, self-evaluating system to people. With the interface, users command the system, which then decides how to carry out commands based on what is occurring in the system. A knowledge acquisition tool in the user interface can send data to the knowledge base and inference engine to teach them about new situations and solutions.

The expert-system model parallels the traditional learning combination of rule-based and experience-based education. Most expert systems use rules—basically sets of if-then procedures—to make their decisions.

Flight monitor. Victoria Regenie of the National Aeronautics and Space Administration’s Ames Research and Dryden Flight Research centers in California described work on an expert-system-based flight-systems monitor. The monitor’s goal is to downlink satellite telemetry to an on-board expert system so a computer can help pilot an aircraft.

In the foreground, the system would process information at flight control rates for real-time analysis. In the background, it would answer pilot queries using available time. So far, those in the project have designed indicator rules for knowledge acquisition, completed half the expert system monitor, and constructed about three quarters of the knowledge base.

The planned system will have a flight base and simulation base to compare reality with what the system expects to happen. The team is developing a knowledge-acquisition tool that will maintain clause consistency by making new rules from old ones.

The flight system’s decision making would be essentially carried out through multiple systems that access a common knowledge base. Each system evaluates the data according to its area of expertise. Voting between the systems’ actions determines the solutions followed (this is the system a space shuttle’s on-board computers use).

Blackboard approach. William Erickson of the Ames Research Center’s space station automation research group described an increasingly popular approach to multiple-expert-system decision making: the blackboard. The blackboard essentially controls the flow of discussion between the systems.

Each expert contributes its knowledge and analysis to a blackboard accessible by all experts. Each can change, delete, or add to the others’ contributions. However, only one expert may write on the blackboard at any time. A moderator decides who may contribute when. Usually, a common decision based on the systems’ combined expertise is reached if not, voting takes place.

As had many at the session in their own research, the Ames/Dryden group greatly underestimated the number of rules their system’s rule base needed. Writing in Common Lisp, they had expected to need about 200 rules. Instead, they ended up with more than 1000. “It’s an escalating process,” Regenie said. It can also lead to unwieldy and slow computations.

Fault evaluation. Michael Geoffr of SRI International in Menlo Park, California, discussed fault evaluation on the space shuttle, an area where “there’s a strong incentive to automate.” However, it is also an area difficult to automate—especially because of the thousands of rules involved.

The current system—a collection of thousands of pages of manuals that contain procedures that call up other procedures that in turn call up other procedures—is nonextensible and hard to modify. “We need a system that can explain what it’s trying to do,” Geoffr said. Standard programming languages fail to do that because verification across procedure calls are difficult.

If you use traditional expert-system tools, you have little control structure, essentially because nothing keeps track of past events that may have an effect on the solution of a problem. Such control conditions become so cumbersome and complex that you lose understanding and efficiency, he explained.

For fault evaluation and correction, an expert system must also make a decision in a reasonable amount of time and be able to put problems in perspective. One floor participant illustrated this point by describing an expert system that was given a chess problem to solve while the room it was in was set afire. Faced with this situation, an expert system, the storyteller warned, had better not wait until it solved the chess problem before deciding to leave the room or put out the fire.

Geoffr said he is exploring a space shuttle fault-correction automation system that has no rules. Instead, the system would use generalized procedures and specifications. The context of the problem would be part of the procedures rather than part of a long string of unwieldy if-then statements that invoke a procedure.

Essentially, he explained, the system would be a set of executable specifications with associated triggers or demons that call up the procedures. Such a design could circumvent the large, cumbersome lists of rules used in traditional systems, Geoffr said.

Fuzzy logic. Lofti Zadeh of the University of California at Berkeley dealt with the issue of logic and meaning in AI rather than with specific applications. He described the calculus of his fuzzy logic, which tries to quantify degrees of usuality.

“Many in AI feel classical logic can deal with common-sense reasoning. I disagree. Facts and rules are imprecise.” Zadeh said that most facts and rules have elastic meanings. For example, a hotel that tells its guests “checkout time is 11 a.m.” might mean anything from “checkout time is any time before 11 a.m.” to “checkout time is any time before noon.”

The rule is context-dependent, subject to local differences from hotel to hotel and from clerk to clerk. What rule-based system should do, Zadeh argued, is “try to describe the meaning in a context rather than legislate it.” He used the term dispositional evaluations to mean a collection of unrugged dispositions (a disposition is a “fact” that is preponderantly, but not always, true).

A person, he said, interprets the phrase “young men like young women” to mean “(most) young men like (mostly) young women.” What AI systems must do is figure out how often the strict interpretation of “young men like young women” is true. Knowing the degree of usuality allows expert system to handle—indeed, expect—deviations from the norm.
‘Star Wars’ software special report

SDI feasibility questions mount

Galen Gruman, Assistant Editor
On March 23, 1983, President Ronald Reagan announced the Strategic Defense Initiative, a plan to shield the United States and its European allies from ballistic nuclear missiles. The initiative called for a research program into the appropriate technologies.

Soon after Reagan’s announcement, a scenario emerged of computer-controlled, space-based weapons. As described in the Defense Technology Study, a report issued just before the SDI effort was formed, the threefold system in that scenario would react to the three phases of a ballistic missile attack (see accompanying report on SDI requirements).

The concept is simple: “What if a free people could be secure in the knowledge that their security did not rest on the threat of instant US retaliation to deter a Soviet attack? That we could intercept and destroy strategic ballistic missiles before they reached our own soil or that of our allies?” Reagan asked a national TV audience in March 1983.

The national press popularized the “Star Wars” label, taken from the 1977 space epic, over the objections of the president (who prefers the term “peace shield”) and George Lucas, the Star Wars creator.

And soon the controversy began. Could such a shield work? Couldn’t the Soviets easily render it useless? Did’t it violate several treaties? Would it destabilize Soviet-American relations? Would it be affordable? How perfect would it be? Should the US change its deterrence strategy from the long-standing deterrent philosophy of mutually assured destruction?

Ironically, the SDI area in which the Defense Department believes the US is far ahead of the Soviets is software, an area that US scientists appear to have the most doubts about.

Technology. The initial controversy over whether the technology could be created resurfaced last summer when David L. Parnas, one of the fathers of modern software, quit a Defense Department advisory board on SDI computing issues; the Eastport Group, and published a set of papers arguing the proposed program could not work. (American Scientist published the papers in their entirety in its September-October 1985 issue.)

Questions about hardening tracking and weapons satellites against Soviet attacks, about keeping the communications systems intact during the electromagnetic pulses of nuclear explosions, and about creating directed energy weapons such as X-ray lasers continue to be raised.

In November, an internal report from the Stanford University laboratory investigating X-ray laser weaponry for the SDI program revealed that the sensors used to detect X-ray lasers may not be capable of detecting them, thus making all data gathered so far unreliable. On the other hand, an antiballistic missile fired from a fighter plane was successfully tested from California’s Vandenberg Air Force Base in October.

Reliability and testability. The software question is whether such a complex system can be effectively tested and made reliable, Parnas has argued in debates last July and October at the Software Engineering Conference and the Massachusetts Institute of Technology. Another debate will be held at Compcon Spring 86 in San Francisco, California, March 3-6. (See “Can ‘Star Wars’ Software Be Built?” Soft News, November 1985 IEEE Software for a report on the Software Engineering Conference debate.)

Parnas told the MIT panel, “It might not be impossible to put something up there that will work, but there will never be a day when we will trust it, and thus we will get no benefit from it. . . . One never really knows when the last bug has been removed.”

Reagan administration officials have pointed to the Apollo missions to the moon, the B-1 bomber development, and the space shuttle as examples of complex systems that work. Danny Cohen, a professor at the University of Southern California and chairman of the Eastport Group software advisory panel, told the MIT panel, “All of them have bugs. Yet they are trustworthy, they work.”

Joseph Weizenbaum, an MIT professor debating on Parnas’s team, argued back, “In a space shuttle launch, the adversary is nature, the conditions are known in advance, and we still have failures. In SDI, there is an adversary who is trying to blow up your whole system.”

Fred Brooks, a professor at the University of North Carolina at Chapel Hill and the man who headed the development of the IBM System 360, told the Senate Armed Forces committee in late October, “I see nothing that means we could not build the kind of software system that ‘Star Wars’ requires with the software engineering technology that we have today.”

At the Senate hearing, Brooks said he disagrees with some of Parnas’s assumptions. “One of them is that he takes as a criterion of success the construction of an absolutely perfect system that has no flaws. I think the real requirement is that it be perfectly guarded against catastrophe.

“But if the system, for example, assigns two weapons against one threat when it should have assigned only one, that is not the kind of flaw that one could lose sleep over.”

A larger question is how can one test such a system. When asked that question by the House Appropriations Committee in late October, SDI director Lt. Gen. James Abrahamson answered that the US could launch a few real missiles with dummy warheads while at the same time simulating the launch of other missiles to see how the SDI system reacts.

Brooks pointed out he does not necessarily believe that SDI software will work. “Now I would not be so bold as to assert that the SDI software can surely be built. What I would say unequivocally is that it is much too soon for gloom. . . . The fact that a defense is difficult to build and the fact that the ultimate outcome of any military system is uncertain is not a reason not to attempt one,” he argued.

Program size. Among the specific issues cited are the size of the required program to coordinate tracking, situation assessment, weapons control, reconfiguration, and the other system elements.

“The problem is not size,” Brooks maintained. “You hear people talking about 10 million lines of source code, but I have no basis whatever that leads me to believe that 10 million is anything other than really a guess.”

Given a 10-million-line requirement, “instead of building one 10-million-line system, it might be simpler to build 10 one-million-line systems with standard interfaces. As simplistic as it sounds, it may make the systems development much easier to accomplish,” said Navy Cmdr. James Offutt, the SDI’s assistant director for system architecture, at the Computers in Aerospace Conference, held October 21-23 in Long Beach, California.

Such size would lead to a large number of potentially fatal errors, opponents argue. Reacting to news accounts predicting large numbers of errors in any SDI
software, the president of AT&T's Technology Systems Group, Thomas R. Thomsen, defended the SDI effort.

In a November letter to the SDI's Abrahamson, Thomsen cited "a Bell Labs survey on software faults [that] found that errors typically range from 0.5 to 3.0 occurrences per 1000 lines of pro-gram" to show how few errors would occur. Yet that could translate to 5000 to 30,000 errors in the predicted 10 million lines of SDI code.

Government response. When asked about these problems, Defense Department officials responded that the point of the SDI effort was to research solutions to such problems.

System architecture is considered the key to any defense system's success, say SDI scientists. In the past, hardware had been the limiting factor in systems design. Now, software development is the issue, the SDI's Offutt told the Computers in Aerospace Conference. Offutt said, "An architecture needs to take into account the strength of computer power and the weaknesses of the state of the art of software."

The SDI's goal is a fast, ultrareliable, electromagnetic-pulse-resistant, fault-tolerant, mistake-recoverable, reconfigurable system. "The requirements are now beyond the state of the art yet demonstrated. Software is the major issue. Even so, the SDI requirements do not seem to be out of reach," Offutt concluded.

The Eastport Group agrees. In its November report addressing some of the criticism about software feasibility, the panel said it thought the "Star Wars" program was achievable—but it also said that any such system would have to have an unspecified unconventional architecture to succeed.

The panel recommended a tree, or hierarchical, structure. "This communication and computation structure preserves locality and allows for autonomous actions from local subunits, which have the same tree-like structure as the entire system. When the communications are augmented with lateral paths, the hierarchy can also be made fault tolerant," it said.

At the lowest levels in the hierarchy, the Eastport Group proposed, software would handle sensor-fusion operations that integrate the data into an accurate "stereo" image. In the middle levels would be target discrimination and attack coordination to avoid multiple shooting and assure complete coverage. The highest levels would assign priorities to the targets in midcourse to prevent too many warheads to begin reentry over any one area.

The panel put much stock in simulation as a tool to solve SDI software problems. It recommended several simulation systems to evaluate battle management architectures and strategies and also urged the use of third-party validators whenever possible, including efforts by research and academic groups.

The Eastport Group report admits that, because of its large size, any SDI system would contain errors. "All systems of useful complexity contain software errors. Therefore, for the level of reliability required for a Strategic Defense System, the central question is how to design this system such that errors are first minimized and then tolerated."

Proposals flawed. The Eastport Group report argued that the traditional Defense Department approach to systems design would not work for a strategic defense system. In fact, the panel rejected Phase I software efforts by SDI contractors because they used the current "appliqué approach." The Phase I architecture proposals sent to the panel by companies proposing to gain SDI contracts "are deficient because they fail these technical issues," the panel said.

"They have developed their proposed architectures around the sensors and weapons and have paid only 'lip service' to the structure of the software that must control and coordinate the entire system," the Eastport Group found.

The five contractors invited back for Phase II proposals—Martin Marietta, Rockwell, Science Applications International, Sparta, and TRW—have paid attention to the criticism. Preliminary results from some of the Phase II work shows use of the decentralized, distributed system urged by the Eastport Group, said Air Force Major David

Technical areas SDI software must address

Galen Gruman, Assistant Editor

Whatever their personal views on the Strategic Defense Initiative, software scientists seem to agree on the areas that need research. And almost everyone working on a missile defense—with the notable exception of President Ronald Reagan and Stanford University physicist Edward Teller—agrees that no SDI system would be 100-percent effective. Some missiles would get through.

Even a "near-perfect" system could not be expected until about the year 2000, admitted the Eastport Group, a panel of scientists advising the Defense Department on defense system software, in a November report.

Objectives and limits. If a strategic defense system is in place when intercontinental ballistic missiles are launched against the US or its allies, it would have to deal with three phases of attack: missile launch (boost phase), warhead targeting and dispersion (midcourse phase), and warhead homing (reentry phase).

Each phase requires different weapons countermeasures and different software techniques for discrimination and interception. The times available for a defense system's reaction range from minutes in the first phase to a half hour for the second to 40 seconds for the third.

Estimates on how many "leakers"—the defense community term for warheads that get through the defense system—range from 0.01 percent to 20 percent. That translates to a range of three to 6000 hits on US targets if all 30,000 warheads now aimed at the US were launched.

Furthermore, the SDI program addresses only ballistic missiles fired from the ground or from submarines. Bomber-delivered missiles would have to be detected by the Defense Early Warning radar system now in place and intercepted by US fighters or surface-to-air missiles. The SDI also does not address cruise missiles, which don't have the characteristic infrared signatures of ballistic missiles and which can fly below radar detection range.

Software requirements. Major Walter Seward of the Air Force Institute of Technology summarized the problems any SDI software would face in its responses to a missile attack at an SDI tutorial session at the Computers in Aerospace Conference, held in Long Beach, California, in late October.

The SDI effort must investigate and solve target acquisition, target classification, target tracking, resource allocation, and communications, control, and command problems, Seward said. Any defense system will be constrained by time and battle space.
Audley, the SDI's program manager for battle management and command, control, and communications.

**Non-technical issues.** Even if the software and hardware questions are resolved, a future administration will need to decide whether to go ahead with a missile defense system. Three treaties—the 1967 Outer Space Treaty, the 1968 Limited Test Ban Treaty, and the 1972 Anti-Ballistic Missile Treaty—are commonly cited in the SDI debate, but both proponents and opponents of the program have been able to interpret them to their advantage. While the treaties' provisions are debated, both the US and the Soviet Union are pursuing missile defense research.

The Europeans are split on the SDI. Britain, West Germany, and Italy have agreed to cooperate on a government-to-government level in SDI research. France, Canada, The Netherlands, Norway, and Denmark have all officially declined government-to-government cooperation on SDI work. Eight other European nations have made no official decision. But all allow private concerns to accept SDI contracts.

The SDI marks a major shift from the policy of mutually assured destruction followed by both superpowers for four decades (and for two decades by France and Britain, which have small national nuclear forces). An SDI system, which could be in place between 2000 and 2010, "could start to provide a sufficiently effective defense that no Soviet planner could be reasonably 'assured' of success of a ballistic missile attack," the Eastport Group proposed.

Under the mutually assured destruction deterrence, Soviet planners know that an attack on the US will result in an equally destructive counterattack by the US. Thus both sides know that the results of an attack on the other are unacceptably high. If the US had a defensive system, however, Soviet planners would not know how many missiles might strike the US, or where these "leakers" would strike. Thus both sides would never know if an attack was worth the cost of retaliation—and thus not take the risk.

Any SDI system would not be the sole missile deterrent, however. It would be coordinated with strategic modernization and conventional force improvements for a "balanced national military policy," said the SDI's Offsett. The defensive system would supplement—not replace—the offensive system.

**Costs.** The cost estimates for the total SDI program vary from $800 billion (the October 1984 report to Congress) to $2 trillion (a Defense Department spokesman in November 1985). The 1984 SDI budget was $50 million, while in 1985 it was $1.4 billion. The Department of Defense asked for $3.7 billion for 1986, but received about $2.7 billion.

The percent of the Defense Department research budget taken by the SDI is about five percent of the department's total research budget, or about three percent of the national research spending, the Council on Economic Priorities reported. By 1990, the council predicted, the SDI would account for 13 percent of the defense research budget and five percent of the nation's.

So far, the software costs have been small, as most of the money has gone to the directed energy and kinetic energy weaponry and to the hardware programs. Processors and software technology will account for roughly $70 million this year, a spokesman said. The software priority this year is the National Testbed, which will be used to test any SDI software developed later, the SDI's Offsett said.

The potential costs of an SDI system have not gone unnoticed. With a national deficit surpassing $2 trillion, the House Appropriations Committee has recently asked SDI officials to defend the program and its potential costs.

As part of that defense, the president issued National Security Decision Directive 172 last spring. The directive mandates that the maintenance of any system be cheaper than the costs an enemy incurs in countering the system once it becomes operative. Officials will not describe the directive further, saying it is classified.

*IEEE Software* has requested the National Security Council to make public its unclassified portions under the provisions of the Freedom of Information Act.

---

The computer system implemented will have to be independent of its physical structure. Problems will have to be classified as local or global. Seward defined target acquisition, target classification, and resource allocation as local, or intralayer, problems. He defined surveillance, rules of engagement, delegation, mutual defense, and situation assessment as global, or interlayer, issues.

Seward further broke down the local issues into two areas: tracking and classification.

Tracking involves predicting trajectory and future state, maintaining a track file, and handling multiple sensors. These require, respectively, high-speed computing, a large-scale distributed database, and some kind of array-processor expert system.

Classification involves discriminating decoys from warheads, providing kill and damage assessment, dynamic allocation of sensors and weapons (for present and future situations to avoid flooding one area or ignoring another), and listing priorities by comparing strategies with track files and system status. These would require expert systems.

Seward broke the global issues into surveillance, delegation, mutual defense, and situation assessment.

Surveillance involves attack verification, attack assessment, and response selection from rules of engagement. These would require array processors and expert systems.

Delegation involves coordination among the layers, as well as handling of track file data and control information. These would require large-scale databases and high-speed computation.

Mutual defense involves defense of the defense system itself, the US, and US allies.

Situation assessment involves an ongoing assessment of attack status, defense status, and system status.

Any SDI system will require redundancy, Seward said, in which the logical nodes don't equal the physical nodes. To prevent single points of failure, the system would need distributed physical elements. Each layer would consist of logical nodes in a layer spanning layers (multidimensional parallel processing).

The computation would be geographically distributed and the processing distributed among physical nodes and among logical nodes. Processing would be parallel within physical nodes.

These requirements in turn will require research into parallel processing software, knowledge-based expert systems, algorithms for distributed resource allocation, distributed database management systems, and algorithms for distributed real-time control, Seward said.

Obviously, making SDI work will require a great deal of research in several areas of software design—and then in integrating the areas to make a cohesive system.
Panel delineates software initiatives’ differences

Galen Gruman, Assistant Editor

Representatives of four US software initiative programs—MCC, SPC, STARS, and SEI—defined the differences among their programs and spoke of their common problems and objectives in a panel discussion at the Computers in Aerospace Conference. The conference, held in Long Beach, California, October 21-23, was sponsored by the American Institute of Aeronautics and Astronautics.

The audience was largely concerned with the implementation of the software initiatives: where would the staffing come from, would they compete or cooperate, and what were the programs’ motives?

MCC. Johannes Grande, of the Microelectronics and Computer Technology Corp.’s Software Technology Program, told the audience that MCC is pursuing high-risk, long-term projects to bridge the research and implementation gaps between the corporate and academic communities. MCC’s targets are software technology, advanced computer technology (such as human interfaces, artificial intelligence, databases, and parallel processing), VLSI circuit computer-aided design, and VLSI circuit packaging.

The software program will focus on requirements for high-level design, which Grande called “the front end of the life cycle,” for the complex-system challenges introduced by the Strategic Defense Initiative, the US manned space station, and tactical fighters.

SPC. The Software Productivity Consortium will address near- to midterm programming issues facing its 13 member aerospace companies (see the report on the consortium below). Jack Foidl, an SPC technical member from TRW’s Defense Systems Group, said the SPC intends to develop software development tools to help speed and ease the process.

Its initial thrusts are reusable software, software prototyping, knowledge-based expert-system software development, and software systems engineering (which includes technology transfer, productivity impact evaluation, technology acquisition, and engineering support).

STARS. The Department of Defense’s Software Technology for Advanced Reliable Systems, or STARS, program will assume the role of development shepherd for at least the defense industry. “We believe labor-intensive factors must be turned over to automation. We want to promote an appropriate selection of standards to promote technology,” said Clarence Giese, director of the STARS Technical Program Office.

The intent is to create a government software environment across the Department of Defense, the National Aeronautics and Space Administration, the National Security Agency, and the defense industry that covers interface specifications, interface frameworks, generic tools, tool environments, and application tools.

The cost of underwriting a software environment (roughly $150 million) is more than any large company would want to spend, Giese said. “STARS will be a seed of investment that will be placed in industry and hopefully germinate.”

SEI. The Software Engineering Institute will work with academia and government on both improved software engineering methods and improved software engineering education, said Mario Barbacci, associate director for project engineering. The SEI is part of the same Department of Defense software initiative that created STARS and the Ada Joint Program Office.

With the existence of two government-supported and two company-supported software initiatives, several members of the audience asked whether there would be any cooperation between the programs and if there would be duplication of efforts.

Because the STARS and SEI programs are part of the same government initiative, there should be no duplication of

Parallel machine beats
computer chess champ

The new Hitech chess computer designed at Carnegie-Mellon University beat the reigning world computer chess champion, the Cray Blitz, at the North American computer chess championship, held in October in Denver, Colorado.

The supercomputer-based Blitz came in fourth, after the parallel-processor-based Hitech, the BEBE system, and, surprisingly, the Apple-based Intelligent Software.

The Hitech machine, running on a Sun minicomputer, uses an oracle to analyze chess positions and a parallel-processor searcher to find the best move. The oracle evaluates what information each of the searcher-controlled 64 chips must look for.

Hitech’s advantage over the Cray’s faster processing is its ability to take a close look at a relatively small number of positions rather than quickly scan millions of positions before making a move.

The world computer chess championships will be held in Cologne, West Germany, in June.

Software Productivity Consortium announced

Twelve of the nation’s largest aerospace companies officially announced the formation of the Software Productivity Consortium on October 23 from the center’s headquarters in Reston, Virginia.

“The demand for high-quality software for application to critical government missions and private sector programs exceeds the capacity of individual companies to develop,” said V. Edward Jones, interim president of the consortium.

The consortium will focus on three areas: reusable software, software prototyping, and knowledge-based systems for software development. Ten products in these areas are scheduled to be developed within five years. The goal is to provide a 400-percent improvement in software productivity of the member companies.

The results of the SPC’s work will be transferred to 33 sites among the 12 companies, each of which has at least 500 software engineers. The simultaneous transfer is expected to speed technical transition. The first products are scheduled for delivery to the member companies 18 months after formal work begins on them. (That work is expected to begin this spring.) The delivery plans include a followup period of one to two years for feedback.

“Since the Department of Defense will spend over $12 billion this year on embedded weapon-system and support software, our goal is to markedly improve the productivity of the member companies’ software engineers,” Jones said. That spending is expected to increase to more than $32 billion each year by 1990.

The companies now involved are Allied-Signal (Bendix Aerospace), Boeing, Ford Aerospace and Communications, General Dynamics, Grumman, Lockheed Missiles and Space, McDonnell Douglas, Northrop, Science Applications International, TRW, United Technologies, and Vitro. Harris and Martin Marietta have indicated their desire to join as limited partners.
effort, SEI's Barbacci responded. For the same reason, the two programs will cooperate on appropriate projects, he added.

For the commercial efforts, "the difference is one of time frame," MCC's Grande said. MCC is aiming to develop its prototypes to 10 years, while SPC "is addressing near- to mid-term issues," SPC's Foidl explained.

"MCC is looking for a few revolutionary ideas that may or may not become a reality. We look at SPC as something that is required for MCC to be successful," Grande continued. "No argument," Foidl responded.

**Staffing.** The panel's moderator, Robert Jones of Hughes Aircraft, asked the participants, "Where do you get the people? Are we biting off more than we can chew?" Barbacci answered, "We're really drawing people from the same pool."

Foidl pointed out that the SPC's participating companies have guaranteed to assign to the consortium a minimum number of in-house people to help staff SPC's 172 positions. However, as Grande said, these same companies are giving people to MCC and STARS as well. At MCC, he said, the companies originally planned to supply 80 percent of the staff. They were talked down to 50 percent, but the actual figure is closer to 20 percent, he added.

Furthermore, Grande continued, "It has been very difficult to get your hands on the irreplaceable, the experts, the very good people in the company." Thus, despite the participants' commitments, the initiatives still need to attract large numbers of well-qualified researchers.

The other panelists concurred. At the SEI, the staffing plan calls for the hiring of a new software engineer each week, Barbacci noted. Foidl only half-joked that the SPC was accepting resumes.

**Benefits.** A member of the audience asked what the benefit was to the companies of participation in the SPC and MCC. Why share among competitors? he asked. Foidl responded, "The same thing that causes them to compete: the profit motive."

It is cheaper to research cooperatively than to duplicate an effort across several dozen companies, he explained.

However, competition between the member companies will still exist because antitrust considerations have limited the consortiums to research work only. Proprietary information cannot be transferred between the companies and the SPC, Foidl explained.

The result, Grande added, is that each company will have to act on—by itself—the information and technology transferred to it from the SPC or MCC. "The management of it is what will make or break the company."

Foidl picked up the argument by stressing that each company will use the tools differently. "The challenge to the company is to take the promised, guaranteed tool and enhance it for their application in the marketplace," he said. The SPC will not support any enhanced versions, so the companies will have to make their enhancements wisely, he added.

The SPC will not extend its efforts to software work outside the realm of the interests of the aerospace consortium's companies. "SPC will not go after SEI or SDI work," Foidl said, because its member companies already compete among themselves for work on programs like STARS and SDI. Furthermore, work for the government initiatives, competition is not an issue because of the coordination in the Department of Defense. However, STARS's Giese said in response to an audience question, the issue of rivalry and cooperation between the military services involved in the research remains a "delicate" matter.

That careful response drew knowing chuckles from many in the audience.

---

**WANG INSTITUTE OF GRADUATE STUDIES**  
School of Information Technology  
Visiting Scholar  
in  
Software Engineering  
The School of Information Technology offers visiting appointments (1985/86) for scholars of demonstrated software engineering skill and accomplishment. The focus of the School is a graduate curriculum in software engineering. A primary goal for the School is to bridge the gap between academic and industrial approaches to software research and development.

The Institute offers excellent facilities for professional development, including modern computing systems, an excellent software engineering library, graduate assistants, and secretarial support. The student/faculty ratio is 6 to 1. The balance between teaching and research efforts is negotiable; however, it is expected the visitor will have a definite plan of activities for the year.

Interested parties should submit a letter of application to the Chair of the Faculty, including a discussion of plans for the visit, required technical support, names of three references, and a resume. Compensation and benefits are appropriate to the position.

Richard E. Fairley  
Chair of the Faculty  
Wang Institute of Graduate Studies  
Tyng Road  
Tysterboro, Massachusetts 01879  
fairley@wanginst - CaNet  
... decvax twanginst !fairley

WANG INSTITUTE OF GRADUATE STUDIES  
School of Information Technology  
Software Engineering Faculty  
Wang Institute provides a unique educational environment that combines the best features of academic and industry. Faculty members teach one graduate course per semester, interact closely with small numbers of graduate students who have industrial experience, supervise production quality software projects, and pursue their individual research interests.

Positions are available for full-time faculty members and one-year visitors who have experience in using and teaching the technical and managerial principles of software engineering. Faculty members receive industrial-level salaries and are provided with sufficient equipment and support staff to achieve their professional goals. Each faculty member is expected to maintain a reasonable balance between research in software engineering and exposition of that material in the classroom. Teaching ability and an interest in graduate-level education are essential.

The campus of the School of Information Technology is located near the Boston hi-tech and university areas which provide rich environments of academic, cultural, and professional activities. Instructional computing facilities at Wang Institute include a VAX 11/750 (VMS), a VAX 11/780 (UNIX), a Prime 750, a Wang VS100, Apollo Domain, and IBM, Apple, Wang, and Hewlett-Packard personal/professional computers. A growing collection of software tools for software engineering is maintained by the staff of the Wang Institute Software Environment project.

Applicants should submit the names of three references and a resume that indicates professional specialty areas, teaching experience, and experience in the practice of software development and/or software management to: Richard E. Fairley, Faculty Chair, Wang Institute of Graduate Studies, Tyng Road, Tyngsboro, Massachusetts 01879.

---

**Classified Ads**

RATES: $5 per line, $50.00 minimum charge (up to ten lines). Average six typewritten words per line, nine lines per column inch. Add $4 for box number. Send copy at least six weeks prior to month of publication to: Sandra J. Arteaga, Classified Advertising, IEEE SOFTWARE Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, IEEE SOFTWARE may reject any advertisement containing any of these phrases or similar ones: "...recent college graduates...", "...1-4 years maximum experience...", "...up to 5 years experience...", or "...10 years maximum experience." IEEE SOFTWARE reserves the right to append to any advertisement, without specific notice to the advertiser, "Experience ranges are suggested minimum requirements, not maximums." IEEE SOFTWARE assumes that, since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only.

---

*January 1986*