Architecture, simulation key to SDI success

Galen Gruman, Assistant Editor

A year after a group of computer scientists made their recommendations, the Strategic Defense Initiative Organization approved a "Star Wars" software architecture that incorporates those recommendations.

"We have an architecture now for battle management that incorporates all of the Eastport Group's concerns," said Air Force Lt. Col. David Audley in an October 1 interview in Los Angeles. Audley is an assistant director of the SDI Organization and heads the battle management and command, control, and communications (BM/C^3) program there.

The Eastport Group, as the scientists called themselves, told the SDI Organization in a December 1985 report that usual military software development and procurement tactics should *not* be followed in the SDI program. It recommended an open, decentralized architecture that favored independent nodes and the use of novel approaches to system design. (See January 1986 *IEEE Software* Soft News.)

Eastport recommendations. The advisory group approved the architecture in late September. Its concepts — some called "radical for government" by Audley — were formally accepted in February by the SDI's director, Air Force Lt. Gen. James A. Abrahamson.

The Eastport Group and its report attracted scrutiny and criticism soon after the group was formed in spring 1985. David Parnas, a long-time defense computing researcher, quit the panel, saying SDI's goal was "impossible."

However, most seem to agree with the general thrust of the Eastport Group's recommendations, whether or not they believe the recommendations will ultimately result in a successful system. "It's been regarded highly by friend and foe alike. We're really serious about doing it this way," Audley said.

"It's often said you can't build an Astrodome over America. Well, what can you do? We envision a program that will look at defensive technologies," Audley told a Los Angeles ACM meeting October 1. "We can't build all this stuff and put it in space and scratch our heads and ask 'What are we going to about it?' We have to think ahead."

In the initial SDI concept, "nobody ever said much about the battle management, command, and control software — what is really the heart of the system," he said. In fact, the only unclassified part of the 1983 report that led to the SDI program dealt with software. It was as if, Audley joked, "the software is so impossible that we can tell them what we're going to do and it still won't help."

Architecture payoffs. The biggest payoff since the recommendations' acceptance, Audley said, was that a database distributed among stations in space and on the ground is no longer considered necessary. Many researchers feared such a system would be cumbersome, difficult to develop and implement, and easy to disable.

Instead, each space-based platform in the envisaged defense system will create all data required by it. This avoids multiplatform processing and synchronization between the stations, which would have only intermittent contact because of different orbits.

The platforms — both sensors and weapons — will act instead as individual soldiers: each will take orders from a higher level authority and will communicate and coordinate with neighboring platforms, but will rely on its own resources and engagement models if necessary.

Another payoff has been the realization that "we can use existing multiprocessors for distributed operating system needs," Audley said.

One key aspect of the architecture approach is to avoid complexity. "If it's too complex then we don't want it."

The architecture approach also requires that the development of SDI software be flexible, Audley said. "We're trying different ways." The object is not to get locked into an unworkable or needlessly complex approach, he explained.

Language. One area of flexibility is in the choice of language. Ada is the preferred language, and will be required for communications applications in 1987 research projects. But computer languages, like human ones, express different things differently. Some concepts are better expressed in one language than another, so requiring that all work be done in one language would needlessly shape the programming approach, Audley said.

Audley is also uncertain about Ada's runtime performance in SDI applications. Still, he said, "I think Ada is very expressive, a rich language. If I had a dictionary of only 1000 words then I wouldn't get Tolstoy."

Simulation. Until — and if — such a system is developed and implemented, SDI remains a research project to conform with US treaty obligations. The most important part of that research effort, Audley said, was the simulation and prototyping efforts that will be undertaken in the National Testbed, a planned simulation facility distributed across hundreds of sites.

"The simulation [effort] now is uncoordinated. We need to tie it all together," Audley said. The distributed nature of the testbed will help develop the actual SDI control software, he added, because it will teach program researchers how to implement such a distributed system before developing any SDI code. "A VAX here, a VAX there, a Cray over there — we'll learn about distributed systems," Audley explained. "It's true synergy."

"The software's going to think it's really fighting a war," Audley said. The simulation may even include space-based computers. "We want to make believe we're doing it for real. We would like to, in a sense, deploy this prototype. We are not ready for the bricklayers to come in yet."

Testbed. Work on the testbed began this year. It will begin as the coordination of current simulation facilities and expand as the technology and budget allows. The development of programs under the framework of the adopted architecture also continues. Prototyping and testing will come later.

Key to the National Testbed's success will be the creation of a networking system. Now called Internet, this system will be more capable than the Defense Department's ARPAnet system. ARPAnet has a 56-kilobit packet-switching capability. Audley said Internet must handle packet-switching in the megabit range.

Audley envisions the testbed to contain many multiprocessor-based, event-driven, message-passing systems, perhaps large vector machines such as Crays and ETAs.

The testbed will allow substantial prototyping, another key area in the SDI research plans. "You learn a lot with prototypes," Audley said, invoking University of North Carolina professor Fred Brooks's famous advice, "Plan to throw one away, you will anyhow."

Once the prototyping stage is reached, the relatively open, unclassified research approach followed by the SDI Organization will change, Audley said. "We will need to ensure that the implementation and building are not bugged. The concern is physical access. We will use red teams [SDI personnel acting as enemy agents] to break into the system."

Outside pressures. While Audley insisted SDI research is not shaped by political decisions, he acknowledged that many development and implementation problems will be addressed politically. "It's like a thermostat, controlled by humans," he said.

One pressure is the decision by many scientists to not work on SDI research. Audley told of some code used in an astronomy project that "had just what we needed, but the guy who owned the code restricted it so it couldn't be applied for SDI. . . . It hurts. We need all the talent that we have."

On the other hand, the large SDI budget — $2.8 billion in 1986, compared to the National Aeronautics and Space Administration's $5 billion and the National Science Foundation's $1.3 billion — has prompted some pressure from congressmen for immediate results. Recent footage of hardware tests released to the press is an example of a limited response to this pressure.

Audley stressed that the software effort must not be railroaded by such pressure. His major concern is introducing bugs. "We prevent bugs by design. We don't like to preplan the removal of bugs. The issue is not to get things going [now] because we [will] have lots of debugging to do [later]," he stressed.