'Almost' not enough for SDI software, Parnas maintains

Paul A. Myers

Software for the Strategic Defense Initiative has a fundamental problem — its capabilities cannot be completely tested against its operational requirements — David L. Parnas maintained in a debate at Stanford University.

"We don't know what it is supposed to do, and we don't have any way of verifying if it does what we think it should. If we don't trust our system, then we have to continue with our other preparations," Parnas, a professor of software engineering at the University of Victoria in British Columbia, Canada, told the 500 people attending the December 19 panel discussion.

The debate on the antiballistic missile defense research program was sponsored by the Stanford University Department of Computer Science and the Palo Alto, California, chapter of Computer Professionals for Social Responsibility.

Speaking in support of the software aspects of the SDI research project was Richard Lipton, a computer science professor at Princeton University and a member of the Eastport Study Group, the Defense Department advisory board on SDI computing issues.

Lipton countered that there are ways to organize the programming during the design phase to reduce the number of critical problems to a level that starts to provide great confidence in the system's capabilities.

Parnas, a noted authority on defense software, expanded on the arguments he originally made last June in his resignation letter from the Eastport Group. At that time, he argued that SDI software could never be perfect and would be impossible to completely test in any case. (See the Soft News reports in the November 1985 and January 1986 issues of *IEEE Software* for details.)

Verification problem. At the Stanford session, however, Parnas conceded that the software does not have to be perfect and that there is no fundamental law keeping the SDI system from working. Rather, he stressed the improbability of verifying that the system will do what it is supposed to do.

Developing a large body of error-free code of the size required by the SDI project is virtually impossible, Parnas argued. It might be possible to have a 10-million-line system with 100,000 errors that would still "work" in some essential sense — "however, you would have to pick your errors very carefully, and that's hard to do," he asserted.

"There is something fundamentally different about software," Parnas observed. Software always seems to be the unreliable part of the system where errors continue to be found after implementation. Even after thorough testing, it is hard to eliminate the chance of catastrophic failure from software.

He said that his experience with military real-time software showed that reliability comes from corrections made during repeated operational use. Although a high degree of fault tolerance may be achieved, that does not necessarily decrease the risk of catastrophic failure. In the case of SDI software, "almost right is not good enough," Parnas emphasized.

"Now, these are not limits on what computers can do, but rather limits on what human beings can do. We have lots of experience with building software, and we know what human beings can do — we know that they cannot build software that is correct before it is put in use," Parnas concluded.

Flexible architecture. In contrast, Lipton said that it may be possible to organize a strategic defense so it isn't a single integrated system, but rather a collection of systems of sufficient independence and diversity to achieve goals of resiliency and robustness that characterize other distributed computing systems successfully operating today.

Lipton stressed the advantages of an open architecture featuring relatively independent communications and computational structures that preserve locality, allowing autonomous action by local subunits organized into battle groups. Diversity in design and implementation would characterize the subunits.

Among the advantages of this type of architecture is a degree of independent testability allowing use of statistical inference techniques similar to those used to assess the reliability of offensive weapons systems, Lipton added.

Lipton's comments parallel arguments made by the Eastport Group. Its recent report on SDI computing issues stated, "The unique characteristic of a strategic defense system, the one that could make it difficult to test to the point of full confidence, is the extent of presumed coordination among the parts of the system."

The report adds that several approaches to this testing problem are promising. These include more use of advice-only information transfers and extensive use of simulation so the operational code and algorithms could be tested under very large numbers of battle variations.

Another approach favored by the Eastport Group is simulation for in-line testing of a deployed system with simulated data going to and from the sensor and weapons platforms.

Because the aim of the current program is to research solutions posed by a defense system's complex requirements, Lipton warned that people must be very careful at this stage about claiming that something cannot be done. Proofs of impossibility are hard to come by in mathematics and computer science, he observed. On the other hand, it is very difficult to predict the nature of future technological advances.

Data-sharing risks. In rebuttal, Parnas said the complex informational demands of responding to a missile attack will put tremendous pressures on system designers to move away from independent and autonomous computing systems back to a more centrally coordinated defensive system that relies on widely shared data — data that must be communicated. Globally based data systems accordingly risk partial or global failure, Parnas argued.

Compared to other large software systems, such as telephone switching systems and the Apollo missions to the Moon, that are often offered as examples of the feasibility of successful implementations by SDI proponents, Parnas observed that the SDI system must contend with an intelligent adversary intent on defeating the system.

Other systems are contending against a statistically predictable natural environment, but any SDI system will be involved in a battle of strategy and tactics between intelligent opponents.

This contest is not analogous to the reliability measurements of the physical world that describe essentially continuous functions as being within a certain percentage over a period of time or number of trials. "We don't rate chess players as being 97-percent effective," he said, emphasizing the win/lose nature of a nuclear defense system.

Correction

A line of text was inadvertently dropped in the January issue. The last lines on p. 62 should have read "Systems programs that are designed independently of one another often interact according to some standard pattern of usage. For example, after an editor modifies a file, the file is often passed to a text formatter." We regret the error.