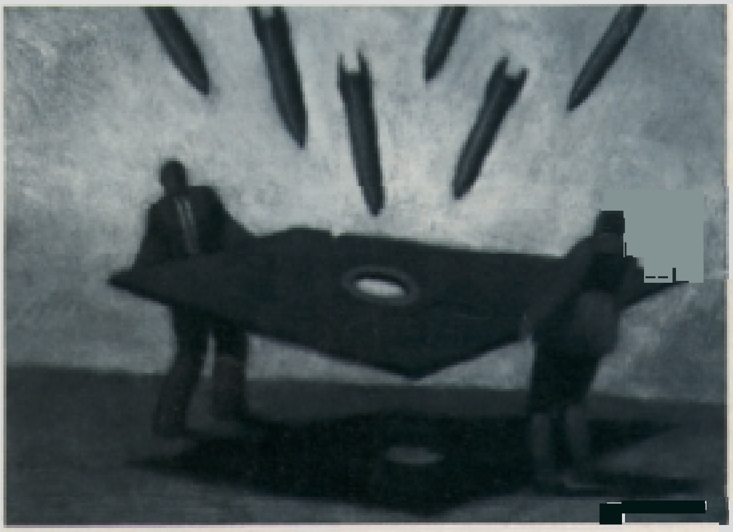
Followup/digest

A Flaw in the Shield?

Controversy builds on the software element of a Star Wars scheme.

The United States is under attack from several thousand enemy missiles. The massive assault triggers computers in U.S. defense satellites to direct space weapons and ground-based missiles to wipe out the enemy attack. But nothing happens—the computer software system of the Strategic Defense Initiative has simply foiled to respond.

This scenario illustrates the concerns many computer scientists have about the so-called Star Wars plan to develop a computer-controlled system to shield the United States from missile attack. The debate over the controlling software for an SDI system has intensified since a panel appointed by the Defense Department concluded hat previous studies overlooked the fundamental problem of how to design computer programming that could manage a space-based antimissile system.



Coupled with the resignation of prominent scientist David Pamas from the Eastport Group-a government advisory panel on SDI computing issues-the trustworthiness of software is now a paramount issue. Parnas a computer scientist at the U.S. Naval Research Laboratory in Washington, D.C., disagrees with the panel's assertion that adequate programming will soon be feasible. He argues that any SDI system would profile only false security "With this software, you'd never know whether or not it would work when needed," he says.

An SDI system would have to react quickly to many problems. Defense satellites must detect the unique infrared radiation of a missile launch, track the missiles, distinguish real ones from decoys and decide if they are going to attack. Then the satellites must pinpoint which missiles to target. verify their destruction and notify other defense satellites of any missiles that slip through the shield. At the same time, the satellites must defend themselves-and all this derision making must occur within about a half hour.

Danny Cohen, head of the Eastport Group and a member of the Information Sciences Institute at the University of Southern California, disagrees with Parnas; he asserts that SDI software can be created and can work. His group recently concluded that fast, mistake-tolerant, self-correcting, adaptable software system is indeed feasible. However, the particulars of this design must still be developed.

One chief concern is the system's complexity. To minimize it, SDI researchers have recommended a structure in which local units control their internal functions independently of other units and of the main system.

At the lowest levels in this hierarchy, software would integrate sensor data into images of missiles for evaluation. The middle levels would separate authentic missiles from decoys and coordinate attack to ensure that no missile is overlooked and no attack is duplicated. The highest levels would decide which targets to attack to ensure that a number of warheads would not reenter the atmosphere over any one target.

SDI software proposals call for a number of programs to run simultaneously on different computers distributed across the nation and in space. These computers would communicate with one another and would also work independently if others were damaged.

Parnas believes that such tight coupling is no guarantee that the systems will work. He likens the problem to rush-hour traffic: Each driver reacts to signals and to other drivers independently but also coordinates with other drivers by signaling and responding to their signals. This loosely coupled driving "system" still has traffic jams. Pamas cautions that the same things can occur in SDI.

Defense Department scientists argue that the design should not be dependent on the perfection of a system but should be formulated to cope with imperfections.

A second concern is the size of the proposed SDI computer programs. The common estimate is 10 million lines of program code. Cohen argues that very large and successful programs exist now without deadly errors: "The space shuttle flight software has about one million lines of code, but the ascent and descent code, which is considered critical, is two hundred fifty thousand lines. In spite of the recent tragedy, it still holds as an example of a big code where no errors have been found in the critical parts."

A third question is how to test such a complex system. SDI director James Abrahamson has said that one effective test of an SDI prototype would be to launch one or two real missiles with dummy warheads and at the same time simulate the launch of many missiles to see how the system reacts. However, U.S. treaty obligations now prohibit such tests, and there are no plans for them.

The SDI's key effort in this regard, then, will be the creation of the National Testbed a group of simulation facilities designed to test SDI software. However, simulation software is often as complex as the software it tests.

Pamas argues that simulation can't accurately predict what might happen, and its results, therefore, cannot be trusted. Counters Cohen, "We will do everything we can to infer performance, including testing and simulation, to assure success in preventing a nuclear holocaust. Having no system ensures destruction."

—Galen Gruman