collision of Europe and North America.

The second episode of compression would have squeezed much the same region in a northwest-southeast direction. That formed two other mountain fold belts, Akna Montes and Freyja Montes, but it also sliced up Maxwell Montes along nine San Andreas–like faults, squashing the long, linear belt into its present roughly rectangular shape. Part of the compressed crust seems to have been squirted to the side, as has happened to parts of China caught between the colliding Indian and Asian plates.

"The exciting thing is that we are seeing features very similar to those on Earth that are not just randomly scattered," says Head, "but appear to be integrated in regional patterns. We're starting to see tectonics very different from that on the small terrestrial planets," whose surfaces cooled to form a single thick plate without horizontal motions developing.

Indeed, the Brown group sees indirect evidence of many plates or segments of crust moving about the surface of Venus. Crust seems to converge at Ishtar Terra and possibly at two other areas in mid- to high latitudes while extension and volcanism tend to dominate equatorial highland regions, in particular Aphrodite Terra. As at terrestrial mid-ocean ridges, most of the heat lost from the interior would escape in regions of extension by conduction through the thinned plate and increased volcanism. Head and his colleagues stress that they cannot say whether the crust is moving relatively rapidly and diving back into the mantle, as on Earth, moving just enough to form the extensional zones without plate recycling, or moving at rates between those two extremes in ways that are uniquely venusian.

Although other planetary geologists also now see abundant evidence on Venus of horizontal motions previously known only on Earth, they are still reluctant to construct such detailed geologic scenarios. "There seems to be horizontal tectonism," says Harold Masursky of the U.S. Geological Survey in Flagstaff, "but it is so complex. Perhaps the best approach now is to map geologic features, then make grand inferences." Geologists have a few more years in which to make inferences before the arrival of the next radar mapper at Venus, the U.S. spacecraft Magellan scheduled to arrive by the end of the decade. Everyone agrees that its resolution, the highest ever, will allow sequencing of events and assignment of tectonic mechanisms. Toward that end, Soviet scientists presented their American counterparts at the meeting with Venera 15 and 16 imaging radar data tapes and maps derived from them. ■ RICHARD A. KERR

Resolving the Star Wars Software Dilemma

A panel of computer scientists has concluded that computers will be able to manage a strategic defense system—but only if battle management is designed in from the beginning

SEVERAL months ago, a panel of computer scientists convened by the Pentagon's "Star Wars" Strategic Defense Initiative Organization (SDIO) quietly released a report concluding that the creation of battle management software for the Star Wars system will indeed be feasible.

Since most people would hardly expect an advisory panel handpicked by the SDIO to conclude anything else, the report seems to have aroused little public interest. In this case, however, the report is worth a closer look. Written by the nine-member Eastport Study Group, which is chaired by the Israeli-born computer scientist Danny Cohen of the University of Southern California, the report is in fact a scathing critique of the way the Pentagon handles high-technology weapons design in general and software development in particular. It challenges a number of tacit assumptions being made on all sides of the Star Wars debate. It deals with important questions about the limits of computing, the nature of reliability, the organization of large, complex systems, and the nature of strategic defense itself.

And in a striking paradox, it validates what the program's many critics have been saying about the *in*feasibility of Star Wars software.

This last conclusion is particularly ironic, because the software issue first gained widespread attention a year ago when David L. Parnas, a computer scientist from the University of Victoria, British Columbia, resigned from the panel on the grounds that Star Wars battle management software could never be made reliable. Since that time, Parnas and many other critics have continued to insist that software is the Achilles' heel of the entire strategic defense project.

Even if one assumes that space-based antimissile weaponry can be made to work, they argue—admittedly a big assumption—it will be computers that manage the battle. Human reaction times are simply too slow. Computers will process the raw data from the sensors. Computers will detect the missile firings, determine the source of the attack, and compute the attacking trajectories. Computers will discriminate between warheads and decoys. Computers will coordinate the activities of the battle stations. Computers will aim and fire the weapons. And computers will assess whether the warheads have been destroyed.

As a result, the Star Wars battle management system will be by far the most complex body of software ever devised. By one estimate it will require up to 100 million lines of computer code written by hundreds or thousands of individual programmers. Obviously, no one is going to be able to write that much software without making mistakes. But much more serious, say critics, no one will be able to trust the battle management system because no one will be able to test it under realistic conditions—realistic conditions being a full-scale nuclear war.

"It's not enough to put a bunch of killer satellites into space and call it a missile defense system," says Parnas. "We have to have confidence in that system. We have to *know* what it will do, because a weapon you can't trust is of no use to you. *We* will make decisions as if it was not there, and *they* will make decisions as if it might work. If we continue in that way, it's my fear that we will end up in a much weaker strategic position than ever before."

Judging from their report, the Eastport panelists are in complete agreement with that premise. Where they differ, however, is in their conclusions—or more specifically, in their assumptions about what strategic defense is supposed to be and how battle management is supposed to work. Parnas, like most members of the news media and the general public, explicitly takes the President at his word that Star Wars is supposed to make nuclear weapons impotent and obsolete. "I've taken my requirements from the very highest and most reliable source, Ronald Reagan," he says. And if that is the requirement, he adds, then the software clearly has to be perfect, or at least guaranteed free of catastrophic defects.

However, Cohen and his colleagues on the Eastport panel take the point of view held almost everywhere in the defense community outside the Oval Office: that Star

Wars will primarily serve to strengthen deterrence. "The panel does not expect small-scale and/or early technology deployments that might occur during the 1990's to provide a 'near-perfect' defense," they write. "Rather, initial deployments influence our strategic position largely in their ability to intercept enough incoming warheads to enhance Soviet attack uncertainties. . . . The United States would then no longer need to depend solely on an offensive deterrence."

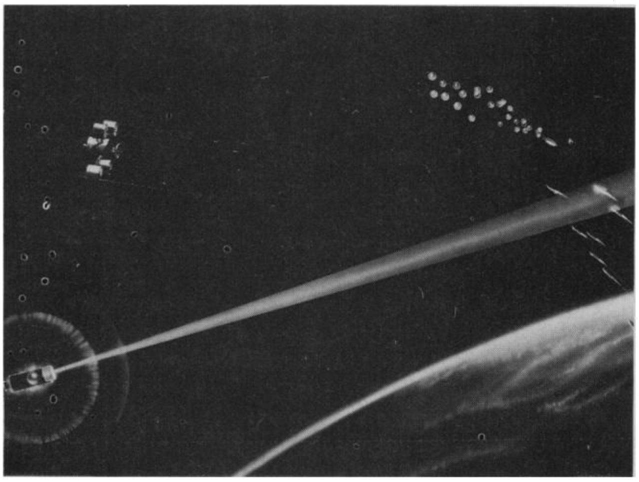
Depending on whom one asks, this latter strategy will either lead to a catastrophic new arms race, or to a new era in which both the United States and the Soviet Union feel safe in reducing their offensive nuclear arsenals. However, without trying to take sides in that controversy, the Eastport group does point out that the goal of making an attack uncertain is much more tractable and realistic than the goal of building a totally leak-proof defense against everything. And given that definition of Star Wars, say the panelists, it *is* possible to have effective battle management—with two critical caveats.

First, they say, battle management is tractable only if SDIO and its defense-industry contractors give up their tacit assumption that software is an "appliqué," something that can be sprinkled on preexisting weapons and sensors like pixie dust to turn them into a working defense system. This assumption was quite evident in SDIO's so-called "Phase I" architecture studies, which were completed in 1985 and which seemed to concentrate almost exclusively on hardware.

Instead, says the Eastport report, battle management must be designed into the strategic defense system from the beginning. "SDIO must not assume that any architecture with sufficient weapons and sensors in the right places is also a *feasible* architecture, that is, one that can be implemented successfully," they write. In particular, they add, despite the optimism heard in some quarters of the defense community, no miracles are going to occur in the process of software creation. It will remain a slow, painstaking, labor-intensive task, which means that the system will have to be designed so as to minimize the complexity of the programming. Indeed, "we find it a bit troubling to be discussing whether radical advances in software technology would enhance the quality of a new defense system, when we are aware that many of the DOD's biggest software development contractors are presently literally decades behind the state of the art—an art that is only a few decades old. Suppose new technologies come into being. Are we sure they will be used?"

The second caveat, says the report, is that developers must give up their tacit assumption that strategic defense will be a tightly coupled "monolithic" system—a single giant organism capable of meeting an offensive strike with millions of individual actions coordinated to the millisecond.

In fairness, say the panelists, this is an easy trap to fall into, especially when the costs of weapons and sensors so thoroughly dominate the cost of the system as a whole. Even the most grandiose estimates put the software cost at no more than a few percent of the total. Thus, the temptation is overwhelming to optimize the hardware, to keep everything carefully coordinated so that every shot will count. However, as panel member Charles L. Seitz of the California Institute of Technology points out, the whole idea of a monolithic, tightly coordinated system rests on a fallacy: that by optimizing every piece of the system, one optimizes the system as a whole. "That's just not true," he says. "The hidden costs of that kind of optimization—in the tremendous load of communications and coordination required, in the rigidity and brittleness of the system—far outweigh the benefits." A monolithic architecture is unworkable in a strategic defense system for much the same reason that the Normandy invasion would have been unworkable if Eisenhower had personally tried to coordinate the actions of every soldier who went ashore on D-Day.



LLNL painting by George P. Dooley

Battle management by computer

Any strategic defense system will have to be commanded by computers: human reaction times are too slow. Computers will process the sensor data, track the missiles, discriminate between warheads and decoys, aim and fire the weapons, and assess whether any warheads have survived.

On the other hand, that same battlefield metaphor suggests a much more feasible approach to battle management, say the panelists. Instead of organizing the strategic defense system as "a big octopus with one brain and thousands of arms," as Cohen describes it, organize the system like a military chain of command, with low-level tasks delegated to the parts of the system actually doing the work.

As an example, the panelists imagine a small battle group consisting of several sensors and weapon platforms, all within a few hundred kilometers of each other and all just a few hundred kilometers above a hostile launch site at the beginning of a full-scale attack. The immediate problem for such a battle group is to identify and track the missiles in its own area. This means coordinating and processing large volumes of data from many different sensors, with updates every 10 milliseconds or so. Clearly, this involves a massive amount of computation. On the other hand, it does *not* involve millisecond by millisecond instructions from a higher authority. Nor does it require detailed information from other battle groups. The sensors only have to communicate and coordinate among themselves over a relatively short distance.

Meanwhile, the battle group also reports a condensed summary of the situation to higher authorities in the battle management hierarchy. These computers combine the information from many such battle groups and from high-altitude sensors into a summary of the overall threat assessment, which they then pass up the line to the top command authority. This part of the battle management process obviously involves communication and coordination over a large area. However, by this point the data are already highly condensed, and only need to be updated every second or so.

Indeed, the panelists point out that this battle group example illustrates a general principle of management by hierarchy: the functions that have to be performed rapidly only have to be coordinated over a small area. Conversely, the functions that require coordination over a large area can be allowed to take more time. And the reason, they say, is elementary physics: the targets always move very slowly compared to the communications signals.

In the computer science community this kind of organization is generally known as a "decentralized" or "distributed" architecture. It is a relatively new approach to designing large computer systems and, as discussed below, it is still on the cutting edge of research. Nonetheless, the Eastport panelists point to some critical advantages it offers for battle management:

Simplicity. By eliminating needless coordination, they say, a decentralized architecture can significantly reduce the complexity of the battle management software. As an example they imagine the most decentralized strategy possible, in which weapon platforms choose their targets independently. A preliminary analysis suggests that this strategy would require only about 20% more shots to destroy the same number of enemy missiles as would a perfectly coordinated system. But the payoff would be a greatly decreased burden of communication between the platforms and a relaxation of the need for split-second timing.

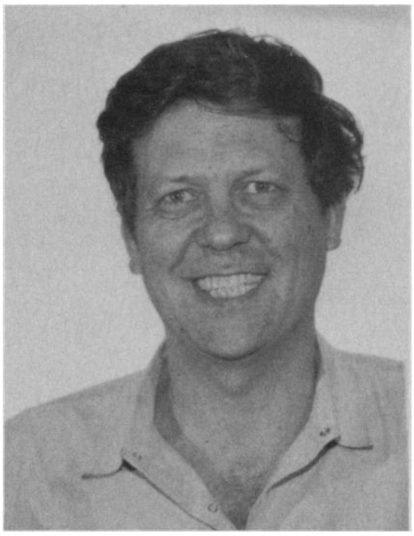
Robustness and durability. In a highly centralized, monolithic architecture, damage to one platform or battle group could bring the whole system to a halt. For that matter, so could a single bug in the software. In a decentralized architecture, problems in one area would have only a local effect.

Evolvability and diversity. "No matter what you do now," says Cohen, "the system will change. So you have to devise the system so that it *can* change." New technologies will have to be incorporated. New enemy threats will arise and will have to be countered. And this is clearly much easier to achieve in a decentralized system where changes can be made in one place without requiring changes everywhere else. For much the same reasons, decentralized architectures also allow an easy evolution from small-scale to large-scale deployment. Moreover, they allow for healthy diversity, in that the various elements of the system can be developed by different vendors; thus, errors or vulnerabilities in one element are unlikely to be duplicated elsewhere.

Testability. If the strategic defense system is designed to operate in a monolithic fashion, say the panelists, then the accusation made by Parnas and other software critics is absolutely correct: it cannot be adequately tested in any situation short of a real nuclear war. However, when the elements of the system—whether platforms or battle groups—are capable of independent action, then the system *can* be tested. "The idea is simple," they write: "Test each independent platform or group separately. If it works, then its independence allows one to infer that the whole will work also."

The Eastport panelists do not claim that implementing a decentralized battle management architecture will be easy. Quite the opposite. What they have done, in fact, is to identify an intriguing dilemma. On the one hand, their report confirms the objections of Parnas and the other software critics: so long as the designers of Star Wars are talking about a centralized, monolithic architecture, and so long as they are thinking of hardware first and seeing software as an appliqué, then strategic defense is guaranteed to be a failure.

On the other hand, the panel has proposed a decentralized approach to battle management that answers the objections—and yet is so new and untried that it poses a technical challenge as daunting as the Star Wars weapons and sensors themselves. So the question is whether it can be done at all. In particular, the report identifies a number of research and development needs:



Eastport chairman Danny Cohen

Battle management has to be designed in from the beginning.

■ Coordination and communication are reduced in a decentralized architecture, but they are not eliminated. Some forms of coordination are in fact quite important. A good example is the projection of ballistic trajectories in the mid-course phase of an attack, and the assignment of priorities to targets in order to prevent any single area from receiving a high concentration of warheads. This would provide useful information for the terminal defense portion of the system and would reduce the total number of warheads that actually leak through the defense. However, the panel points out that this kind of long-range coordination is difficult to test directly. It will have to be done through simulations. Such simulations are at or beyond the state of the art, says the panel, and SDIO will need to devote substantial resources to basic research in the area—particularly to the critical problem of validation, making sure that the simulations actually model the real world.

■ While it is unlikely that radical improvements can be made in the way software is developed, the process can be made easier. For example, new high-speed computer work stations allow programmers to do things that previously seemed prohibitively time-consumming, such as tearing a program apart and putting it back together again in a new structure. New high-speed computer networks likewise allow software development to be distributed among small, informally organized groups of programmers, who could conceivably make software changes easily and rapidly while avoiding mistakes. SDIO should be willing to experiment with such approaches.

Meanwhile, the advent of a new generation of ultrahigh-speed parallel-processing computers may make it possible to substitute abundant processing power for software complexity. "In general, the most error-prone part of software stems from trying to optimize its performance," says the report. "If simple, 'brute-force' algorithms rather than complex, special case-laden methods are used, then the software is likely to be more reliable."

■ The whole area of distributed processing is a wide open research problem, and not just for Star Wars. The same difficulties crop up in attempts to devise an automated air traffic control system, or even an automated factory. For example, when computers are operating autonomously their databases very quickly become inconsistent with each other, for much the same reason that every soldier on a battlefield has a different, and probably rather confused, impression of what is going on. So how does a computer decide what to do when it receives contradictory messages? How can machines be programmed to cope with unreliable communications? How can they be programmed to recognize casualties elsewhere in the system and to work around those casualties?

Meanwhile, it is one thing to say that the strategic defense system should evolve gracefully and be open to new technologies. But it is quite something else again to design a system that can do it. As a beginning, says the panel, SDIO could try to devise a core of communication and information exchange protocols, so that the interfaces between various parts of the system would be relatively clean, and the addition of new kinds of sensors and weapons would be relatively straightforward. On the other hand, these protocols have to be chosen very carefully, so that they do not become obsolete before the system is even deployed. As the panel points out, the design of such "open" systems is still a matter of forefront research.

Finally, the battle management dilemma posed by the Eastport group is not just a matter of software, but a matter of management: Will SDIO and its contractors take the recommendations seriously?

Officials at SDIO headquarters are certainly saying the right words. The report has been sent to all contractors with a cover letter from SDIO director Lieutenant General James A. Abrahamson in which he calls battle management "the long pole in the tent," and adds that "The SDIO endorses the spirit and content of the report of the Eastport Study Group. It is found to be in harmony with the needs of the SDI program and its rapid implementation shall be pursued throughout the R&D effort."

The panel members themselves say they are quite pleased with the response. SDIO did not ask the panel for a yes-man report, says Seitz, and it certainly did not get one. "There was a very wide spectrum of political opinion on the panel," he says. "If anything, it leaned toward the liberal side. Furthermore, we felt very free to look at the whole problem, not just an isolated piece of it.

"Everything we've heard from SDIO suggests that they are listening," he adds. "In fact, they've threatened to take money away from contractors who don't listen."

Nonetheless, there is still plenty of room for skepticism. "The Eastport report calls for a profound cultural change in the way weapons contractors operate," says John Pike, a defense analyst for the Federation of American Scientists. But is that really happening? "To fully implement the Eastport recommendations," he says, "you would have to put all of SDIO's hardware projects and field demonstrations on hold for several years. Then you would concentrate your efforts on some very basic research into the fundamental concepts of ballistic missile defense, until you had the software problem completely worked.

"And yet," he says, "SDIO is still spending horrendous amounts of money on hardware. From a bureaucratic point of view you can see why they're doing it that way. Software isn't tangible, you can't show it to anybody. Hardware *is* tangible. But it means that inevitably you're going to get a situation a few years down the road when SDIO says, 'Hey, we've got all this neat hardware. Can you guys make it fit?"

For a long time, says Pike, there has been a big debate in the software community: Will battle management be workable in the 21st century? "Politically," he says, "one of the effects of the Eastport report may be to unite the software community behind the idea that SDIO is doing the wrong thing." ■ M. MITCHELL WALDROP

ADDITIONAL READING

Eastport Study Group, *Summer Study 1985: A Report to the Director of the Strategic Defense Initiative Organization*, December 1985.

R. J. Smith, "New doubts about Star Wars feasibility," *Science* 229, 367 (1985).

Briefing:

Charge Density Waves Seen in Potassium

Because of their comparatively simple electronic structure, the alkali metals are sometimes considered a test-bed for understanding the behavior of electrons in solids. Concepts proven to be sound in such an uncluttered environment can then be extended to more complicated materials. A new neutron diffraction study of potassium, however, seems to support an old and controversial assertion that theorists have incorrectly treated some aspects of even the alkali metals. Physicists contacted by *Science* regard the finding as significant but would like to see it confirmed before calling the controversy settled.

The new study, reported by Tomasz Giebultowicz of the National Bureau of Standards (NBS), Albert Overhauser of Purdue University, and Samuel Werner of the University of Missouri at Columbia, provides direct evidence for the existence of charge density waves in potassium, thereby confirming a 1964 prediction by Overhauser, who also coined the term "charge density wave."

In brief, Overhauser had calculated that, when postassium is in its lowest energy or ground state, the free electrons responsible for potassium's metallic character do not remain uniformly distributed throughout the material, as the then current thinking held. Instead, the electron density varies sinusoidally with a characteristic wavelength (hence the name charge density wave) that is generally not an integral multiple of the crystal lattice constant.

The reason for the sinusoidal clumping of electrons is that it lowers their energy. In the jargon, the exchange and correlation energies are reduced. A consequence of the clumping is that the lattice undergoes a distortion in an attempt to reduce the huge electric fields generated by the separation between the positive charge of the potassium ions and the negative charge of the electrons.

Overhauser's ideas have never been well received, but in 1964 there were no experimental examples of charge density waves, so the question was somewhat academic. Some years later, researchers began finding a phenomenon like charge density waves in so-called layered materials, those in which the electrons effectively move in only two dimensions, and in linear conductors in which the motion is nominally one dimensional.

Instead of adopting Overhauser's explanation, however, solid-state physicists attributed the observations to another effect called the Peierls instability (after Rudolf Peierls of the University of Oxford), which automatically occurs in linear conductors, but they kept the name charge density waves. The Peierls instability also involves a lowering of electron energy and a lattice distortion but the mechanism, which depends on an interaction between the electrons and lattice vibrations, is different from that proposed by Overhauser.

In particular, both models allow for large effects in lower dimensional materials but the Peierls instability is thought not to occur in simple three-dimensional metals, such as potassium. The new neutron diffraction study is by far the most direct evidence for charge density waves in this material, although anomalies (see the Additional Reading for the most recent example) in several of its properties have raised the possibility of their existence. If confirmed, the finding means that theorists will need to modify their thinking about the complicated ways in which electrons behave in solids by incorporating Overhauser's ideas.

Neutron diffraction is so helpful because it is sensitive to the small displacements in the positions of the ions in the distorted crystal lattice. Near each Bragg diffraction spot there are much less intense satellite spots whose shape (intensity as a function of diffraction angle or, equivalently, momentum transfer) provides information about the distortions. In the case of potassium, not only are the satellite spots dim ($1/10^5$ as intense as the Bragg spot they are associated with) but they are so close to the Bragg spot that a very high resolution neutron spectrometer is needed to see them. In addition, a large, defect-free single crystal is needed, the growing of which is a major project in itself.

The difficulty of the experiment is the reason that it has taken so many years to come up with the evidence for charge density waves in potassium. The measurements, described as an experimental tour de force by one physicist, were done at the NBS neutron scattering center in Gaithersburg, Maryland. While there is little question about the data and although the most obvious alternatives to charge density waves seem inconsistent with the findings, physicists would like to see confirming evidence elsewhere. Other explantions may yet be found as well. "It's hard to rule out what hasn't been thought of yet," says Overhauser. ■ ARTHUR L ROBINSON

ADDITIONAL READING

T. M. Giebultowicz, A. W. Overhauser, S. A. Werner, *Phys. Rev. Lett.* 56, 1485 (1986).

E. Jensen and E. W. Plummer, *ibid.* 55, 1912 (1985).