APRIL 1989 $3.00

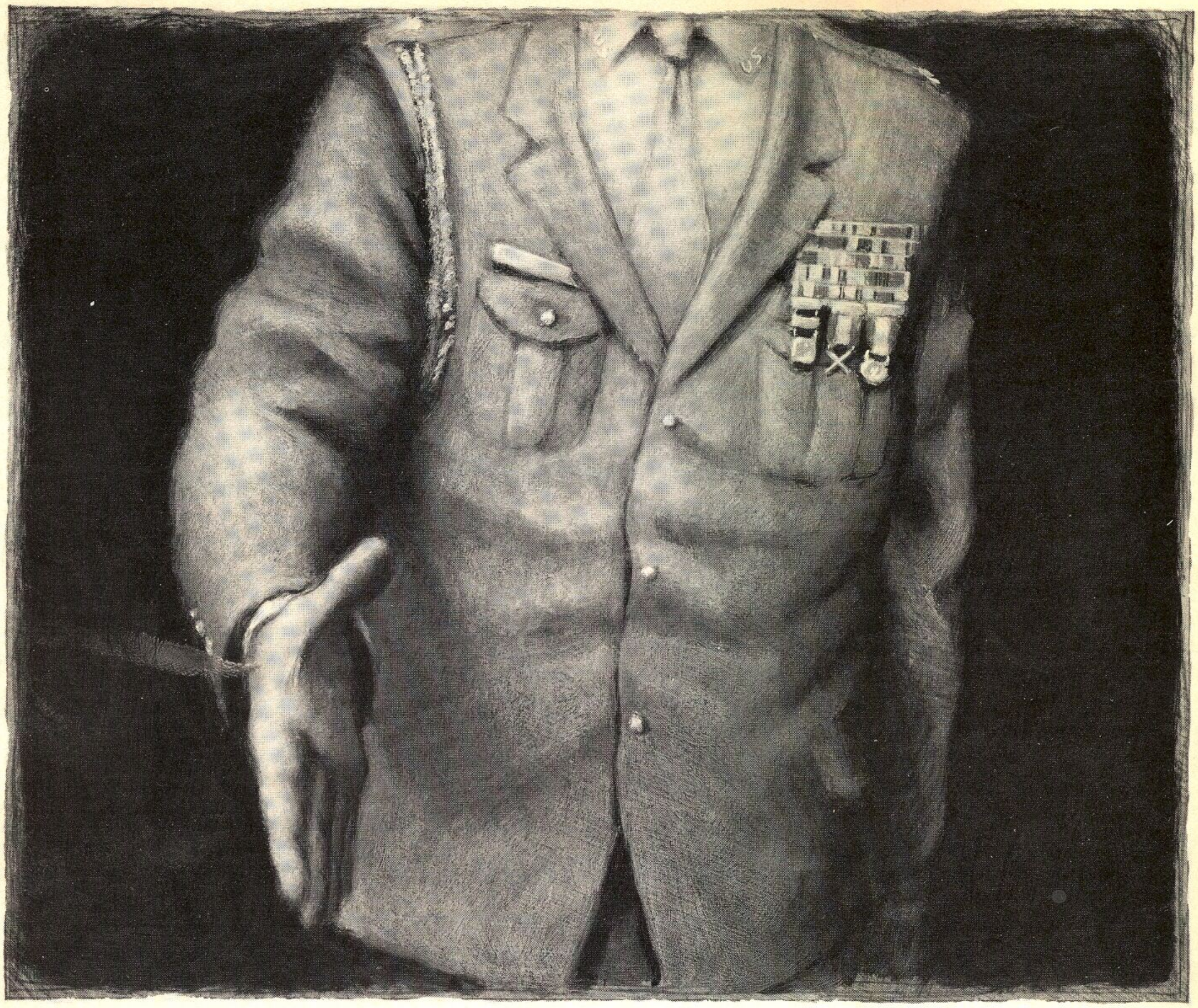
Bulletin of the Atomic Scientists

A spy in the house of war

West Germany's dangerous exports

Is academic freedom bad for business?

0 74470 73932 6 04



Illustrations by David Shannon, United States

My life as a NATO collaborator

I told the pleasant-sounding military officer that I thought some of his ideas were extremely dangerous. "In that case, I think you should come," he said. "If there are dangers, we want to make sure we know about them."

by Nathaniel S. Borenstein

I AM A PACIFIST. I abhor violence in all forms. I have been a vegetarian for 16 years, just over half my life. At age 15, I traveled 500 miles to Washington to march against the Vietnam War. I was almost disappointed when the draft ended before I was old enough to be a conscientious objector. It was, therefore, a source of considerable amazement to my old friends when in the fall of 1987 I flew to Germany to advise a NATO working group on the computer systems at the heart of modern warfare.

I couldn't quite claim that, until that week in a Bavarian mountain resort, I was entirely unsullied by any contact with the military. I had made my first accommodation some years before when I realized that the graduate program in which I was enrolled was entirely bankrolled by the Defense Department. Still, this was different. This time, I would be talking directly to military people, trying to tell them how to *improve* their computers.

Nathaniel S. Borenstein is manager of applications development and lecturer in computer science at the Information Technology Center at Carnegie Mellon University in Pittsburgh. He is the author of People Are Perverse: The Human Factor in Software Engineering *(forthcoming).*

Several months earlier, I had been surprised to receive a letter from the NATO research study group known, in typical military notation, as AC/243 (Panel 8/RSG.12). I was invited to participate in a workshop on CHICC, NATO's acronym for "computer-human interaction in command and control." The workshop was to address the serious human problems involved in the complex information management systems used by NATO as well as by the military establishments of individual NATO nations. A number of technologies were proposed in the letter as possible solutions to the problems.

I quickly concluded that NATO had no idea what they were getting into by inviting me, and my inclination was to decline. NATO, I assumed, would have no interest in hearing from someone who thought a large number of their ideas were stupid and dangerous. The thought of going just to stir up the workshop crossed my mind, but I felt I had better things to do. Still, I delayed sending a negative reply, not certain about the ethics of my position.

I was still deliberating when I received a phone call from a pleasant-sounding man who identified himself as Lt. Cmdr. David Blower, the moderator of the panel in which I had been invited to participate. I told him that I was not inclined to go, largely because I suspected that my ideas would not fit in well at the workshop. Pressed for more details, I explained that I questioned some of the panel's basic assumptions and considered some of the proposed solutions to be extremely dangerous.

"In that case, I think you should come," he told me. "If there are dangers, we want to make sure we know about them." At this point, I began to feel that it would be unethical *not* to go. After all, if I saw risks where nobody else seemed to, shouldn't I warn the people who manage the life-and-death systems? I told Blower I would come.

In a strange land

I arrived in Berchtesgaden in September, just two days before my thirtieth birthday. Berchtesgaden is a lovely little resort in the Bavarian Alps. It is standard practice, I have since learned, for NATO scientific meetings to be held in some of the loveliest spots in Europe, as this has proven effective in securing the participation of the more eminent civilian researchers. The village of Berchtesgaden lies in the shadow of spectacular mountains and is only a pleasant hiking distance from some outrageously beautiful lakes and forests. Many of the townspeople still dress in the traditional Bavarian style, not entirely for the benefit of the tourist trade.

In fact, the place is so lovely that Adolf Hitler himself maintained a summer home in Berchtesgaden throughout the war years. The week passed against an uncomfortable backdrop of reminders of the Nazi era, including several encounters with overtly antisemitic locals. Most of my relatives (all but the few who came to America) perished at the hands of the Nazis, and this was never far from my mind as we sat in an elegant conference center and genially discussed the computers that could destroy the planet.

The working group itself was very small, compared to most of the academic conferences that are the standard fare of a computer scientist's life. The official list of attendees numbered 51, and since we spent most of our time divided into four panels, there were only about a dozen who discussed the issues that concerned me.

The workshop was the culmination of a long series of meetings and deliberations by an internal NATO group addressing the general issue of CHICC. This group had determined that further work was needed in four areas, which became the panels at the workshop:

- *Decision support systems.* These systems, commonly but redundantly referred to as "DSS systems," are essentially hardware and software that make all the relevant information available to the humans at the higher levels of the chain of command. In a wartime situation, one might imagine a general asking for data from a DSS system before deciding where to send his troops, or whether to launch his missiles. Typically, such systems contain more data than a human can digest in a lifetime, and are mind-bogglingly hard to use. This is especially troubling since, if they are ever used, it may well be to help make a decision in a matter of minutes or seconds.

  Unfortunately, improving this situation is immensely difficult, and is the subject of a wide range of research efforts. Therefore it was extremely unlikely that anything new would emerge from this panel, and as far as I could tell, it didn't. I must confess, however, that I didn't always understand what the members of this group were talking about. They had an inordinate fondness for lists, as the group's spokesman made clear in his introductory speech: "Three different approaches are introduced for describing decision support systems (DSS): ten aspects that should be considered; a three-dimensional analysis, and a dynamic description based on the complexities of input, of process, and of dialogue. These approaches complement one another. The designer may use one or all of them during the design process."

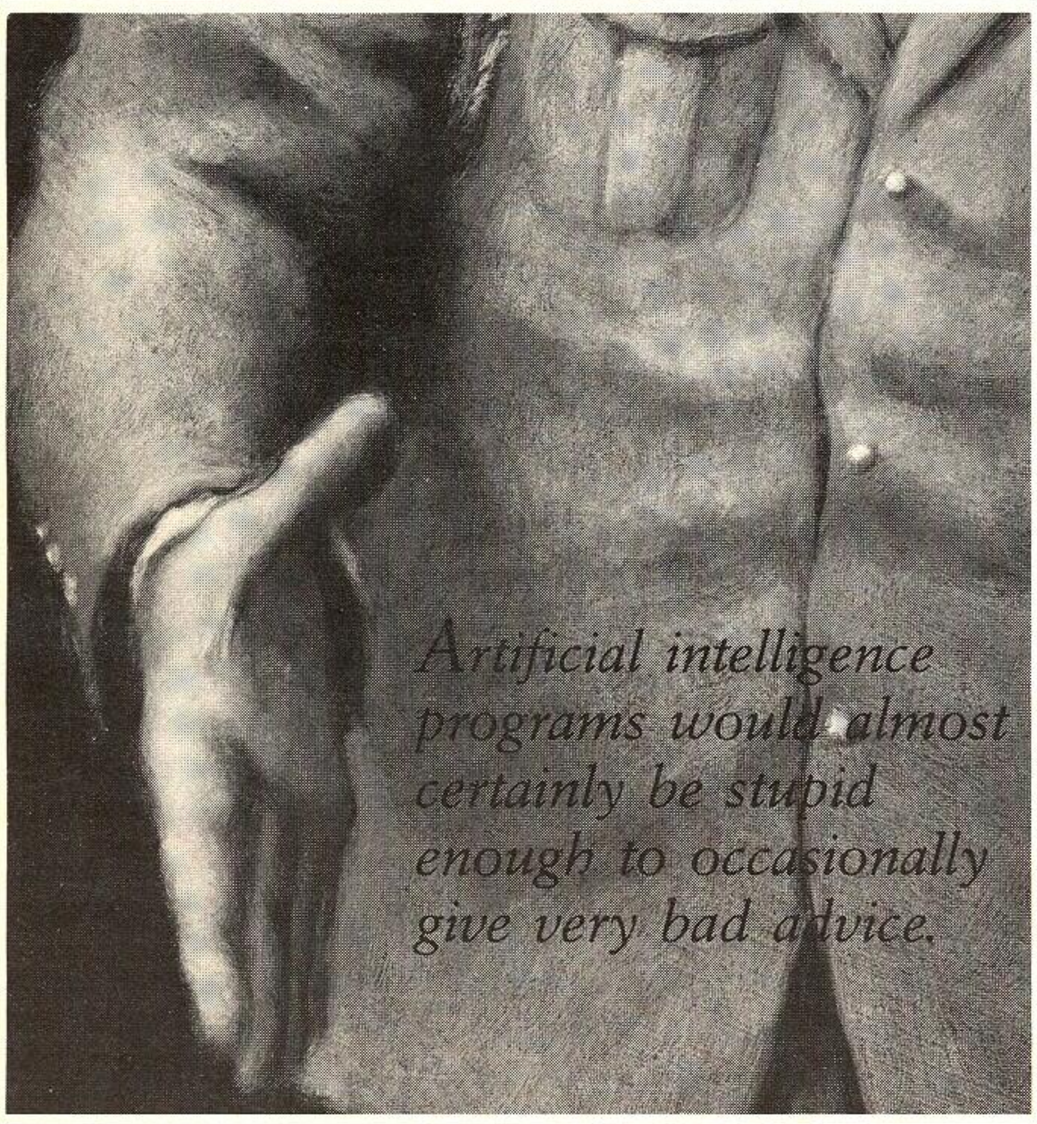
  In the end, they told NATO that further research was necessary—a remarkable understatement, but relatively harmless in the short term. If DSS systems ever become more generally useful, they may pose a whole new set of dangers, but this isn't likely to happen soon.

- *The computer as a communications medium.* In many of the communities most advanced in their use of computers,

"Communication is concerned with the passage of information only in West Germany. In the UK, knowledge is considered separately from information. Only messages occur in Holland, while Canada transmits concepts."

the machines have taken on a key role in interpersonal communication. Such tools as electronic mail, bulletin boards, conferencing systems, and calendar systems have come to seem essential. Despite the heavily computerized nature of much of the military, however, such systems have had little effect on the way NATO does business, and the working group wanted to know what benefits might be available in this area.

This group made even less sense than the previous one. Each panelist had his own research agenda which was at best vaguely related to anything NATO wanted to do. The poor mix of people led, ironically, to severe communication problems within the group, so that they ended up spending most of the week trying to define such basic terms as "communication." The problems were apparent from the initial position papers, and were ably summarized by the group's spokesman: "Communication is concerned with the passage of information only in West Germany. In the UK, knowledge is considered separately from information. Only messages occur in Holland, while Canada transmits concepts."



• *Systems that change and evolve.* It is common knowledge in the computer world that software is never finished until it is abandoned, that it requires constant tinkering, enhancements, upgrades, and bug fixes. But this notion is anathema to the military. Imagine, for example, a tank that was constantly changing on the inside. Because the military has placed such a high premium on stability and reliability, military software has been among the most monolithic and unchanging ever built. In recent years, however, many have come to realize that software might actually be more useful and reliable when it isn't set in stone.

This group had the advantage of a much clearer mandate than any of the other groups. With the absurdity of the current military procedure obvious to any computer specialist, the members of this group grappled with the pragmatic questions of how military systems could be made more flexible without endangering their reliability or, more to the point, without scaring the bureaucracy. The group managed to endorse a definitive statement that systems that change and evolve are not a bad thing, and are even good in some cases, but that the flexibility must be strictly controlled and limited to the most essential areas if reliability is to remain under control.

• *Embedded training and help.* Nowadays, nearly every complex program that deals with human beings includes a component commonly known as an "online help system," which can give the user rudimentary advice on how to use the program. The working group wanted to explore the potential of this and a related idea, "embedded training," which allows new users to practice on the actual command and control system, in a training mode in which all battle activity would be simulated. This was the panel in which I had been asked to participate.

"I could have sworn it was in simulation mode!"

My first task, coming from a background of academic research in online help systems, was to find out what kinds of help and training systems were already being used by the military. This was surprisingly difficult; I felt as though I had traveled back to the 1960s, when computers were big, unfriendly, mysterious beasts not to be approached by mortals.

Air Commodore Laurie Wing, a likeable and intelligent man recently retired from the British Royal Air Force, described what it was like to use the current generation of military command and control systems. Everything, he told me, was entered in cryptic codes of meaningless letters and symbols, and if he wanted to know, for example, how many planes were on the ground at a given base, he would have to type something like "DD c=b 27 a16." Most of the men at the upper end of the chain of command, he told me, were deeply skeptical of the systems, and reluctant to rely on them in crucial situations. Given the incredibly obtuse user interfaces he described, I figured that showed a great deal of common sense on their part.

I described to him a computer operating system known as TOPS-20, on which all commands include an integrated help mechanism that allows you to type a question mark at any point for help in figuring out what to type next. Commodore Wing was fascinated and impressed. "That's just what we need," he said. Yet he was not surprised to hear that TOPS-20 was over a decade old. "We have a terrible time getting good ideas from the research community incorporated into our systems," he admitted.

The military men at the conference were quite aware of the anachronistic nature of their current software and knew that enormous strides had been made in the research and commercial worlds. They were eager to incorporate these

improvements into the military's crucial command and control software and in fact seemed eager to embrace almost any new technology that held out the promise of improvements. This was one of the things that scared me most.

Solving one problem often creates a host of new ones. I devote my efforts rather narrow-mindedly to learning how to make computers easier for people to use. But more usable programs are not always better in every sense. Making a program seem simple to the user usually means making its internals far more complex, and this often makes it less reliable. This is not the kind of tradeoff one makes lightly when dealing with computers that control nuclear weapons.

On the other hand, a system that is totally inscrutable is a danger in itself. But this was a problem that NATO already recognized, and indeed was one of the reasons for this workshop. My fear was that the pendulum would swing too far in the other direction, without enough attention to the new risks introduced by "friendlier" software.

Embedded training, in particular, struck me as a very poor idea. Training by computer simulation has been around for a long time. Embedded training takes this one step further: it does the simulation and training on the actual command and control computer. To exaggerate slightly, whether or not anyone actually dies when you press the "launch missiles" button depends on whether or not there is a little line at the top of the screen that says "SIMULATION."

Such a system seems almost designed to promote an accidental nuclear war, and this thought was what persuaded me to attend the workshop in the first place. One can all too easily imagine human error—"I could have *sworn* it was in 'simulation' mode!"—as well as frightening technical possibilities. Perhaps, due to some minor programming bug, the word "SIMULATION" might fail to disappear when it was supposed to. Someone approaching the computer would get the wrong idea of what it was safe to type.

Beyond these problems, which I incorrectly assumed were so obvious as to nip the idea in the bud, there is the more subtle question of software complexity. Simulation software is typically less carefully engineered than the software it simulates, because it doesn't matter so much if it fails. If there's an error in the simulation program, the worst that can happen is that someone's training is delayed for a while. An error in the real "production" software can have more costly consequences, and in the case of command and control systems the possible costs of software errors are the highest imaginable.

Putting the simulation software onto the machine that actually talks to the missile launchers creates immense new areas of concern. First of all, the "wall" that the programmer builds to divide simulation from reality may not be complete. In some circumstances, the logic of the program itself might get confused about whether it is in simulation mode. Worse, when a computer program goes badly awry, it can often affect other parts of the system. Putting a simulation program on a deployed computer means, therefore, that for real security it must be as reliable as all the other deployed software. But it is unlikely that such levels of reliability will ever be demanded of simulation programs.

Dangerous simulations

U.S. military forces were sent into nuclear war alert on the morning of November 9, 1979, after computers at the North American Air Defense Command headquarters in Colorado signaled that a nuclear attack had been launched against the United States. Although the early warning computers indicated an attack by submarine-launched missiles, jet fighters were "scrambled" against a potential simultaneous bomber attack. The alert lasted six minutes; if it had gone on one minute longer the president and top military officials would have been notified.

This nuclear war false alarm was triggered by a "war game" tape. "False indications of a mass raid [were] caused by inadvertent introduction of simulated data into the NORAD Computer System," according to an October 9, 1980, Senate Armed Services Committee report. □

The dangers of help systems were more subtle but just as disturbing. Particularly troubling to me was the possibility that human operators would become overly reliant on such systems. If an operator is used to simply asking the computer what to do and then doing it, what will happen when real judgment is required? Imagine a battle situation: With only moments to respond, the operator presses the "HELP" button. The computer says, "I recommend that you press the red button." With seconds to decide, will the operator ignore the computer's advice?

Over the years there has been much well-justified resistance to any notion of "closing the loop" and making computers completely control missiles, able to fire them without any human decision. This resistance is eminently sensible, as the computers, however cleverly programmed, really don't understand what is going on. But if humans will push the button whenever the computers advise them to do so, the loop is closed almost as effectively as if humans weren't involved.

Enter artificial intelligence

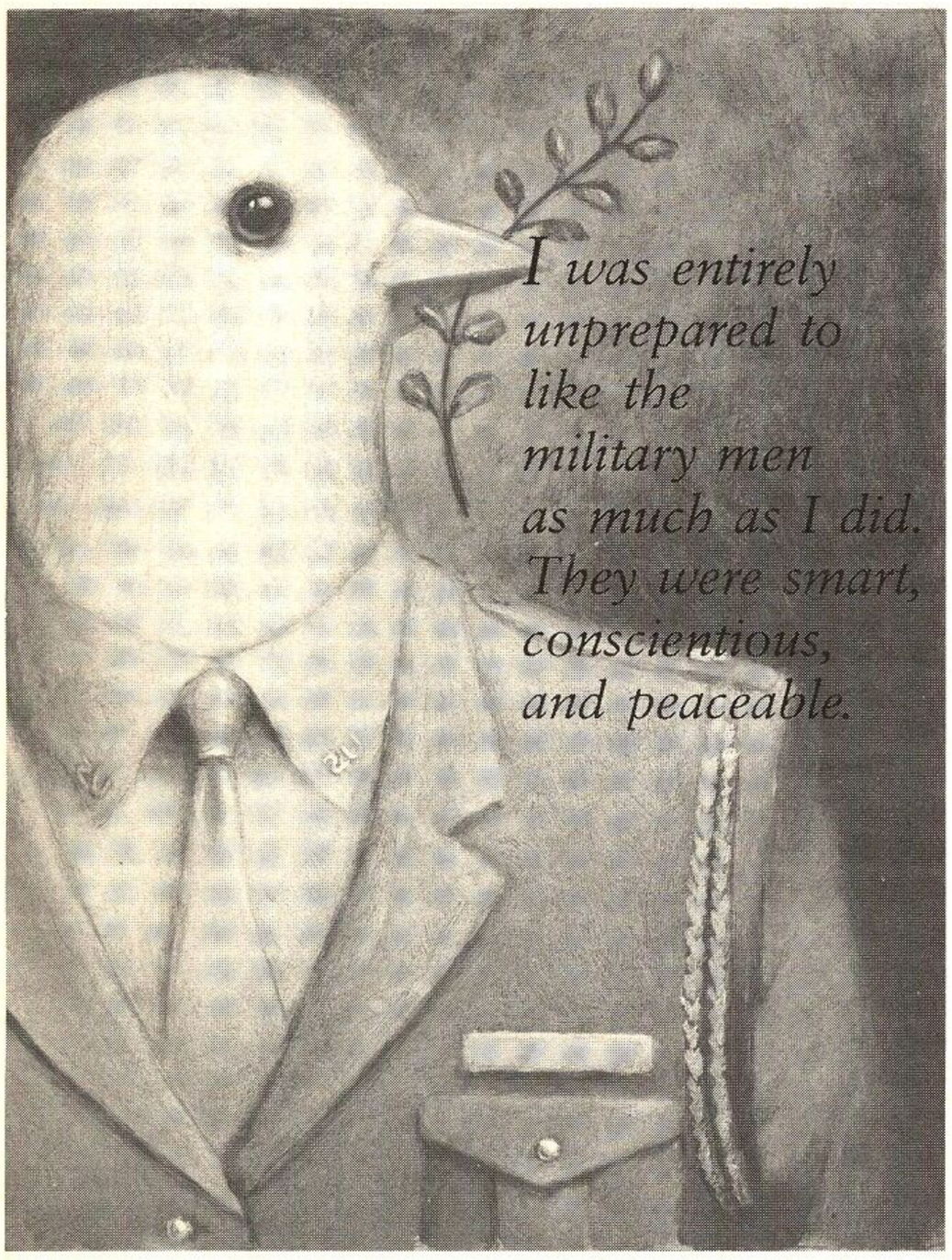
One branch of computer science that has experienced a tremendous surge in influence in recent years is "artificial intelligence" (AI)—the attempt to imbue machines with human-like intelligence. Whether such a thing is possible is the subject of unending debate; suffice it to state that the issue is not settled, and that the practitioners of AI have not even remotely approached success in that endeavor.

What they have managed to build is an extremely useful type of program called an "expert system." Expert systems can make detailed inferences about facts in a very narrow domain of expertise. They are used, for example, to do a rather good job of analyzing astronomical data from spec-

trometers, and to diagnose a patient's illness from a detailed set of symptoms and test results. They do this without ever really understanding what a "spectrometer" or a "patient" actually is, by making inferences based on rules that can be applied to these entities.

In their rush to bring expert systems to the marketplace, AI practitioners have suggested using them for an incredible range of applications. One of these, perhaps not surprisingly, is for online help systems. This was a hot topic at the NATO conference.

The idea disturbed me greatly. If people became overly



dependent on conventional help systems, the problem would be worse with so-called intelligent help systems. Such a dependency wouldn't be so bad if the system really was intelligent, but expert systems are not. They manipulate propositions about objects that they do not comprehend, and are hence incapable of spotting the most grotesque and nonsensical errors. They are only as good as the rules that define them. But rule-based programming, which is the heart of expert systems, is an entirely new way of programming, and as Tom Athanasiou pointed out in a chapter of the recent book, *Computers in Battle: Will They Work?*, nobody yet has the foggiest notion how to write such programs to be reliable or verifiable.

Using AI in help systems struck me as a particularly insidious way to make the whole system—the command and control system, the human operator, and the help program—more likely to quickly reach an undesired conclusion. The AI programs would almost certainly be stupid enough to occasionally give very bad advice, and the human user could be under enough time pressure to take that advice without sufficient consideration. The resulting disaster could be the worst imaginable: an accidental nuclear war based on erroneous interpretation of the data received by the command and control system.

That sort of question

Of course, people at this conference didn't use words like "nuclear war," "die," "kill," or "bomb." Early in the workshop, I once violated this tacit restriction by mentioning the possibility that certain programming techniques could increase the chances of accidental nuclear war. This made everyone else look uncomfortable, and although my concerns were discussed extensively, we referred, more discreetly, to the chances of an "accident."

The workshop participants were divisible into three categories. There were the academics, including myself, who were mostly scientists with strong technical reasons for attending the conference. Many of them seemed somewhat uncomfortable advising the military, although this was rarely discussed.

Second, there were professional military men, representing the armed forces of several NATO countries. They were mostly scientists as well, and I was entirely unprepared to like them as much as I did. They were smart, conscientious, peaceable, and acutely aware of the gravity of their responsibilities.

Finally, there were civilians who worked for defense-related industries, either as consultants or as employees of defense contractors. Each seemed bent on tilting NATO toward funding more of the kind of work that he did. I was surprised by this group as well. They were not merely unaware of the ethical implications of what they did; they were, for the most part, uninterested in these implications even when they were pointed out. This is not to say that they were hostile to the idea of making weapons systems safer, for example. Rather, because they were not paid to think about the issue, they preferred not to do so. They didn't resist my concerns, merely ignored them.

One incident stands out. A civilian had just finished explaining how the kind of AI system he was building could be useful in help systems, and I pressed him on what was to me the key question: wouldn't such a system be more likely than a more conventional system to lead to catastrophically wrong results? He answered, yes, it probably was more dangerous, "if you were interested in that sort of question."

How could anyone *not* be interested in "that sort of question?" I suppose this was my first direct experience of what Hannah Arendt called "the banality of evil."

Fortunately, the good sense of the military people carried the day. When one of the civilians dismissed one of my concerns as "extremely low probability" and hence not worth

discussing, the officers rallied to my cause. "We have to be concerned about anything that could go wrong, however unlikely," said Blower, "when the consequences of a mistake are so serious."

The final report

What the military also brought with them was an impressive sense of bureaucracy. The workshop, it seemed, was something like a temporary factory which had been brought into existence to produce a specific product—the final report. Blower, our group's leader, was evidently an old hand at producing successful documents. Within the first hour of our first meeting he was asking for agreement and advice on various aspects of the wording of this report, and indeed we had most of our final report drafted by the third day. The most controversial parts were heavily revised through heated debate, of course, but that didn't alter the disturbing feeling that we were arguing over the wording of our conclusions before we had reached any. Still, our final report included some strong and fresh recommendations, a tribute to Blower's ability to structure the discussion without destroying it.

I suspect that the form of the final report was far more important to the careers of the military organizers of the conference than were the contents. We could have recommended that future computers be built out of tinker toys, and it wouldn't have hurt their careers so long as they managed to produce a sufficiently solemn and credible explanation of our views. The quality that did come through in the contents, however, reflected their personal concerns and integrity. As far as the private-sector civilians were concerned, the final report was much like the U.S. defense budget: you don't complain about what others put into it so long as the things you care about get put in as well.

To my surprise, there was no serious resistance to many of my proposals. We recommended, for example, that embedded training should be avoided entirely in certain contexts, and, in any event, should be considered as potentially very dangerous. But many of the civilians, even the academics, were involved in AI-related projects and were reluctant to see AI denounced in the final report. We reached a typical compromise: we stated that AI was not "yet" ready for deployment in these systems (sidestepping the issue of whether it ever would be), but recommended continued funding of research into the possibilities.

We could have recommended tinker toys, if the explanation was solemn enough.

I think most of the AI researchers realized, at some level, the intellectual dishonesty of that recommendation, but I doubt that any of them lost sleep over it. They wanted to

do AI research, and the only way the modern world lets them do this is to get money from the military. Therefore they can't come out and say, "This won't ever be safe enough to use the way you want to use it." Relatively few are sufficiently unethical to claim that it will be safe enough any time soon, but even fewer will voluntarily make the facts clear to the people who pay for the research. The result is that those people are often tragically misinformed.

I am deeply disturbed by the corrupting effect that military funding has had on the research community.

A pacifist military?

A few years ago I was astonished to learn that one of my students, who holds pacifist views, is enrolled in Army ROTC, and that after completing his degree he will become a career officer in the military. "Yes," he confessed, "it is a really strange place for someone who basically considers himself a pacifist to spend his life. But think about it: don't you want someone like me in there to see how the weapons are being used? As long as we've got an enormous Department of Defense, what kind of people do you want to have working in it?"

I had no answer for this argument, but I assumed him to be a rather special case. Since my visit to Germany, however, I am no longer so sure. In my more optimistic moments, I now imagine a Defense Department crawling with closet pacifists, all of them doing their best to see that their jobs become obsolete. I know this isn't true and that I've been meeting an unrepresentative sample, the intellectual cream of the crop. But I can't help being encouraged by what I have seen in the military people on whom our futures depend.

On the other hand, I am deeply disturbed by the corrupting effect that military funding seems to have had on the research community. I fear that, if anything, the military may be trusting too much in the basic human sense of its contractors. What would you do if you realized that your project increases the risk of accidental nuclear war? Military people, I now believe, would make their concerns clear. Researchers, and particularly contractors, I fear, would shrug their shoulders and go back to the more "interesting" questions.

To many thoughtful members of my generation, who came of age during and after the Vietnam War, the military seemed the very incarnation of evil. By the standards of my youth, and indeed by the most rigorous pacifist standards, I got my hands dirty in Germany. But I can no longer believe that true pacifism requires one to abandon the administration of violence—for this is what the military is ultimately about—to the violent of heart. Perhaps the road to a peaceful world is not filled, as I once imagined, with millions who refuse to follow the military machine, but rather with thousands who quietly pass through the house of darkness to light a few candles. Perhaps, indeed, that is the intended nature of military forces in a democracy. □