

system in order to allow it to keep functioning in the event of a single component breakdown. In computers, however, "fault-tolerance" is more easily implemented in hardware than in software. If a faulty program is replaced by an identical copy, the same problem will eventually arise again.

To meet this difficulty the human programmer is duplicated, and separate teams write separate programs from the same specifications, on the theory that,"

. . . Various teams are unlikely to introduce exactly the same errors and that by comparing the answers and doing software voting (the computers compare their own answers) you can eliminate errors."

This technique of software "fault-tolerance" involves four major difficulties. First, computers cannot always compare "answers" since much information is carried along implicitly rather than explicitly. Second, the teams are not truly independent, as they are using the same, inevitably incomplete, specifications and therefore may very well build in the same wrong assumptions. Third, it is difficult to synchronize machines in order to compare "answers". Fourth, specifications are often plainly incorrect and this technique does not address that.

Ornstein considered specifications to be "the least tractable part" of the software problem. Very often people don't know what they want computers to do, and so cannot define the problem for them to solve. We can't say to computers "Protect me!" We must tell them precisely how to do it.

In problem domains there is an excess of detail, some of which it is easy to overlook. Attempts to deal with this, by having independent teams write several programs, founder on the difficulty of making any objective comparison of the results. According to Ornstein, computers are extensions of human intellect and are therefore as limited as their originator, if not more so:

*The point is that we must not expect that computers will safely solve the problem of the nuclear hair-trigger for us. They won't. They give the appearance of relieving human burden, but at the cost of increased chances of catastrophic blunder.*

Henry Thompson outlined his concerns regarding automatic decision-making systems and accidental war. He noted that as yet it is human beings who would decide to use nuclear weapons, but there have recently been some suggestions that this should be changed. The proponents of automatic launch-on-warning systems maintain that such systems would increase security through improved reliability and greater credibility. Thompson argued against the implementation of these systems on technological grounds. Automatic decision-making is subject to the present