Technology Transfer Crises in the 1980's: Mishaps at the Human Interface

Danny Kopec
Computer Science Department
University of Maine
Orono, ME 04469

July, 1990
Report No. 90-3

Abstract: Technology Transfer Crises in the 1980's: Mishaps at the Human Interface

By Danny Kopec

Automation and computers have become standard modes of operation in our technological society. By and large technological advances have improved economies, productivity, transportation, and defense systems. In addition, facilities and technologies for information access and storage have become abundant.

However, as technology and computers have become integral to our lives, so have increased the number, scope, and severity of technological mishaps. This paper will explore specific case studies where computers have been involved in crises resulting in the breakdown of complex systems. Potentials for the field of artificial intelligence to help distinguish mere data from knowledge and how it can effectively be used in decision-making will also be examined.

Background

Our earlier work in 1982 addressed the dangers of society's increasing dependence on technology. (1) Specific examples of mismatches between complex technological systems and the humans who operate them were cited from four case studies:

(1) The Three Mile Island-2 (TMI-2) Nuclear Power Station

(2) Air Traffic Control

(3) The NORAD Military Computer

(4) The Hoogovens Royal Dutch Steel Plant

In that report we discuss the dangers of allowing machines to perform tasks which had previously been performed by humans, particularly on a standalone basis, and the need for mankind to "maintain charge" of the relationship between people and machines. As computer systems and their complex interrelationships become prevalent it becomes it becomes critical that we address the possibility of their malfunction:

> "The mismatch between a technological system and the humans who operate it can be either at a "syntactic" or "semantic" level. A knowledge representation can be syntactically correctable. But if its semantic structure is wrong, then no amount of human technology can correct it. The nature of the underlying causes of mismatch between man and machine, whether essentially syntactic or semantic is the issue to which our investigations will be oriented.
>
> It is no longer just laymen who are unable to comprehend how computers and the advanced information technology under which they operate, can co-ordinate within large sophisticated systems; control rooms in nuclear power stations, in air traffic, and oil platforms are instances where large systems are now generally beyond the technical and scientific sophistication of those who operate them." (ibid. p2)

The complex computer networks of the 1980's and 1990's pose a new problem which heretofore was not relevant. Such networks, to control diverse tasks from advanced weapons systems to telecommunications, and the stock market, " . . . are subject to a mathematical concept called *chaos,* a natural phenemenon that leads to turbulence in rapidly moving water or in the atmosphere." (2) Advances in software and hardware design enable the development of networks of immense complexity. These force us to change the way we think about computers. Bernardo Huberman at Xerox Palo Alto Research Center likens the behavior of such computer networks to any society, and coins the phrase "computational ecology". While the behavior of individual computers is highly structured and disciplined, a network of computers can exhibit "wild oscillations and unstable behavior." (2) Such loosely configured networks, without a central controlling computer, will often be forced to make decisions based on an incomplete knowledge about the status of the machines in the network. Much of this problem is related to the issue of concurrency control in computing systems -- how do we most efficiently keep a network of computers busy while ensuring that their computational data and status is correctly communicated, understood, employed and updated. In other words, the whole is not the sum of its parts. Furthermore the complexity of such systems cannot be satisfactorily decrypted through models used to simulate them. Any model used will be missing some critical component which contributes to the real world complexity of the system. This is the main criticism against the Strategic Defense Initiative -- that its aim is to build a system so complex that it can never be fully tested, except when in full-scale employment.

Not far removed from the problem of chaos is the notion of a *normal accident*, which is the title of Charles Perrow's comprehensive study of such systems:

> "A normal or system accident ... is meant to signal that, given the system characteristics, multiple and unexpected interactions of failures are inevitable. This is an expression of an integral characteristic of the system, not a statement of frequency. It is normal for us to die, but we only do it once. System accidents are

> uncommon, even rare; yet this is not all that reassuring, if they can produce catastrophies." (3, p5.)

Perrow goes on to define normal accidents as having two major characteristics which can be largely attributed to the underlying cause(s) of such accidents: 1. *multiple features* and 2.*tight coupling.* 1. Multiple features typically involve problems in six areas: design, equipment, procedures, operators, supplies and materials, and environment. 2. Tight coupling refers failures in different parts of the system which are quite dependent on one another. Perrow gives the example of a transportation system where a bus strike results in a shortage of taxis. (3, p8). Usually tightly coupled events also follow each other closely in time. He further asserts: "In complex industrial, space, and military systems, the normal accident generally (not always) means that the interactions are not only unexpected, but are incomprehensible for some critical period of time." (ibid. p9) Finally he adds that 60-80 percent of accidents are attributed to operator errors, although the real causes often involve mysterious interactions involving bizarre or unusual events, "Patient accident reconstruction reveals the banality and triviality behind most catastrophies." (ibid, p9)

Case Studies

Software has been called the "invisible Achilles's heel" of the computer revolution. (4) Although the trends of the past few generations have involved smaller, more powerful computers which control nearly every aspect of life, e.g. including cars, microwaves, traffic lights, telephone networks, air traffic, etc., the programs which run them have become huge and unmanageable. That is they often involve thousands of lines of code which are virtually impossible to test and difficult to comprehend.

(1) The AT&T Breakdown: On Monday, January 22, 1990, a computer program which determines the most efficient path for routing long distance calls, had a serious breakdown which caused a nine hour collapse in AT&T's well-publicized and heretofore flawless nationwide network. An unkown combination of events (calls) caused malfunctions across 114 switching centers nationwide, causing 65 million calls to be unconnected. This caused considerable embarassment to the corporate giant amid its campaign committing millions of dollars to advertisement of reliability claims and a "virtually foolproof system". (ibid. p70) Although normal service was restored in a matter of hours, more frightening is the fact that the exact cause of the malfunction (in terms of the software) was not positively identified. Had such a malfunction occurred in a socially critical task domain, i.e. air traffic control or nuclear power stations, then it is easy to understand why Perrow would consider such catastrophies just "normal accidents".

(2) The Case of Bank of New York: In 1985 a single software malfunction caused the Bank of New York to borrow $24 billion to cover its accounts temporarily. The net result was a loss of $5 million in extra interest. Complex computerized financial networks for effecting nationwide transfers of trillions of dollars daily are regularly vulnerable to errors of this kind. (ibid. p71)

(3) Washington ATM: Next there is the oft-cited case of the Washington, D.C. automatic teller machine (ATM) which, due to a single programming errors, allowed cardholders to make unlimited cash withdrawals regardless of their account balance.

(4) Canadian Cancer-Therapy Machine: In the spring of 1986 at the East Texas Cancer Center in Tyler, Texas, there were two serious mishaps involving involving the Canadian-made Therac 25 linear accelerator. In one case an unpredictable combination of software commands caused radiation output 100 times the intended dosage and the subsequent death of a 66-year-old man. Earlier 33-year-old man received an excessive dose of radiation after an apparent operator error, subsequent machine shutdown and restart. The same product, developed by Atomic Energy of Canada, caused a deep, discolored hole, larger than a quarter, to develop in the collarbone of a 62-year-old woman patient at the Kennestone Regional Oncology Center in Marietta, Georgia. It seems that the cause of the software malfunction was a certain group of commands typed into the machine at a very rapid speed. In the East Texas Center the further use of the machine was discontinued pending investigations into the accidents which have been ongoing for several years. (Ref.?)

(5) Case of the Government Accounting Office: It is not uncommon for the development of large software programs to run far over budget. Cases in point are the Government Accounting Office's Satellite Tracking Program which is $250 over budget and its Radar Jamming System Software which is $1 billion over budget. Here the lesson to bear in mind is that bigger and more complex programs are not necessarily better.

(6) The Case of the AT&T Net 1000: In 1986 AT&T dropped its Net 1000 project after an investment of $1 billion and eleven years. There had been promises of unbelievable software that would allow disparate machines and networks to communicate effortlessly. The Advanced Communication Network (ACS), was designed to depend on timesharing which quickly became outdated in the 1980's as a means of supporting multi-user systems. The epitaph here was the choice of the wrong software for a particular task -- i.e. a lack of vision or ability to predict future trends. (5)

Risky Systems

We live in a world which has put too much faith into technology. Perhaps the underlying cause of this unpallatable state of affairs is the rapidity with which technological progress has been made during the past few decades.

This has led to a "blind faith" in technology whereby risky systems have been allowed to coexist with us in our daily lives. We have come to accept such potentially catastrophich systems despite awareness of their fallability. The characteristics of risky systems may be summarized as follows: (1) They are deemed essential (2) their dangers are often covered up or denied (3) they will not be abandoned or made safe and (4) they have numerous potential causes including poor design, poor operators, faulty equipment or an uncooperative environment.

Some examples of high risk technologies include nuclear power stations, nuclear weapons systems, recombinant DNA production, and ships carrying toxic or explosive charges. The normal accident in such systems will typically entail an "interactive complexity" overlooked by its designers -- that is there might be two or more failures in components which interact in some unexpected way. (5) These will involve tightly coupled processes which cannot easily be stopped, whose parts cannot be isolated, while critical decisions must be made in short time intervals. (6) Air traffic control is an example of a risky system where such tight coupling and interactive complexity have been somewhat reduced by better organization and "technological fixes". Nonetheless, most complex systems are not immune and have large internal contradictions.

The catastrophies at Bhopal (1984), the Challenger accident (January, 1986) and at the Chernobyl Nuclear Power Station (April, 1986) are all classified as normal accidents by Perrow. (6) Each is a dramatic example of blind faith in technology, accompanied by human character flaws, critical decisions made by leaders of organizations for political and economic reasons, ignoring the potential dangers involved. For example, although the shuttle Challenger accident was primarily caused by faulty O-ring design and operation in relatively cold temperative conditions, low level engineers at Morton Thiokol (their manufacturers) were well aware of this but their warnings were not heeded by NASA.

An important point which Perrow makes is that there were many equipment malfunctions during the previous 24 shuttle missions including faulty landing gear, tire blowout, a premature engine shutoff, etc. Thus any unexpected combination of such equipment failures could have occurred with disasterous effects. Furthermore, according to Air Force sources, the booster rockets used in the shuttle have a 1 in 70 chance of failure. Therefore the two booster rockets employed by the Challenger had a 1 in 35 chance of failure. (6)

Conclusions: The Potential Role of Artificial Intelligence in Complex Technological Systems

In our 1982 study we stressed our position that standalone computer systems to control complex technological systems for socially critical tasks should not be allowed. (1) There, in an effort to quantify the bounds of scrutability of computer solutions to difficult problems solved by humans, we defined the notion of a "human window". That is, solutions whose "grain size" is neither too large (intensional) or too small (extensional) according to the limitations of the human brain in terms of computational speed and memory storage capabilities. In order for the discipline of artificial intelligence to gain further credibility as a science I feel that it has golden opportunity to serve in the domain of facilitation of accident prevention in complex technological systems used for socially critical tasks. Human window solutions are not only necessary, but essential. Success in this endeavor requires cooperation amongst specialists in many disciplines, including artificial intelligence researchers, engineers, psychologists, ergonomists, educationists, etc. This kind of communication is not easy effect, but it is certain to ensure a safer world where we can live and thrive with technological progress.

Acknowledgement: The author is indebted to Malik Faisal for assistance in gathering research material for this paper.

References

(1) Kopec, D. and Michie, D. (1982) Mismatch between machine representations ands human concepts: dangers and remedies. **Report to Commission of European Communities, Subprogram FAST**, Brussels, Belgium.

(2) Markoff, J. (1988) In computer behavior, elements of chaos, **New York Times, Sept.** 11, p.6E.

(3) Perrow, C. (1984) **Normal Accidents.** Basic Books, Inc., New York.

(4) Rogers, M. and Gonzalez, D.L., (1990) Can We Trust Our Software? **NEWSWEEK**, January 29, pp. 70-71,73.

(5) Howe, C.L. (1986) Another tangles network: AT&T[s Net 1000 is only the latest network collapse in a business where almost everyone seems to get a wrong number. **Datamation**, 15 Mar. pp. 64,66,68:

(6) Perrow, C. (1986) Risky Systems: the habit of courting disaster; Bhopal, Chernobyl, and Challenger accidents. The Nation, Oct. 11, pp. 329-39.