The RISKS Digest

Volume 1 Issue 2

Friday, 30th August 1985

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Please try the URL privacy information features enabled by clicking the flashlight icon above. They are described in the news page. Please let the website maintainer know if you find this useful or not. As a RISKS reader, you will probably not be surprised by what is revealed...

Contents

Introduction; three more risk items
   Peter Neumann

Mariner 1 Irony
   Nicholas Spies

RISKS Forum ... [Reaction]
   Bob Carter

RISKS Forum ... [An Air Traffic Control Problem]
   Scott Rose

Risks in AI Diagnostic Aids
   Art Smith

Warning! ... [A Trojan Horse Bites Man]
   Don Malpass

SDI
   Martin Moore
   Jim Horning
   John McCarthy
   John McCarthy
   Peter Karp
   John McCarthy
   Dave Parnas
   Gary Martins
   Tom Parmenter

The Madison Paper on Computer Unreliability and Nuclear War
   Jeff Myers

Can a Computer Declare War?
   Cliff Johnson

Introduction, and more recent risk items

Peter G. Neumann <Neumann@SRI-CSL>
27 Aug 1985 23:32:01-PST

I was away during the previous three weeks, which made it difficult to put out another issue. However, the newspapers were full of excitement relevant to this forum:

- A Federal district judge awarded $1.25 million to the families of three lobstermen who were lost at sea in a storm that the National Weather Service failed to predict because its parent organization (the National Oceanic and Atmospheric Administration) had not repaired a weather buoy for three months. [NY Times 13 Aug 85]

- Another Union Carbide leak (causing 135 injuries) resulted from a computer program that was not yet
programmed to recognize aldicarb oxime, compounded by human error when the operator misinterpreted the results of the program to imply the presence of methyl isocyanate (as in Bhopal). A 20-minute delay in notifying county emergency made things worse. [NY Times 14 and 24 Aug 85 front pages] (There were two other serious Union Carbide incidents reported in August as well, although only this one had a computer link.)

- An untimely — and possibly experiment-aborting — delay of the intended 25 August launch of the space shuttle Discovery was caused when a malfunction in the backup computer was discovered just 25 minutes before the scheduled launch. The delay threatened to seriously compromise the mission. [NY Times 26 August 1985] The Times reporter John Noble Wilford wrote, "What was puzzling to engineers was that the computer had worked perfectly in tests before today. And in tests after the failure, it worked, though showing signs of trouble." Arnold Aldrich, manager of the shuttle program at Johnson, was quoted as saying "We're about 99.5% sure it's a hardware failure." (The computers are state of the art as of 1972 and are due for upgrading in 1987.) A similar failure of just the backup computer caused a one-day delay in Discovery's maiden launch last summer.

- More details are emerging on possible computer hanky-panky in elections, including the recent Philippine elections. There has been a series of articles in the past weeks by Peter Carey in the San Jose Mercury News — which I haven't seen yet but will certainly hope to report on.

I expect that future issues of this RISKS forum will appear at a higher frequency — especially if there is more interaction from our readership. I will certainly try to redistribute appropriate provocative material on a shorter fuse. I hope that we can do more than just recapture and abstract things that appear elsewhere, but that depends on some of you contributing. I will be disappointed (but not surprised) to hear complaints that we present only one side of any particular issue, particularly when no countering positions are available or when none are provoked in response; if you are bothered by only one side being represented, you must help to restore the balance. However, remember that it is often easier to criticize others than to come up with constructive alternatives, and constructive alternatives are at the heart of reducing risks. So, as I said in vol 1 no 1, let us be constructive.

Mariner 1 irony

<Nicholas.Spies@CMU-CS-H.ARPA>
16 Aug 1985 21:23-EST

My late father (Otto R. Spies) was a research scientist at Burroughs when the Mariner 1 launch failed. He brought home an internal memo that was circulated to admonish all employees to be careful in their work to prevent similar disasters in the future. (I don't recall whether Burroughs was directly involved with Mariner 1 or not.) After explaining that a critical program bombed because a period was substituted for a comma, the memo ended with the phrase 

"... no detail is to [sic] small to overlook."

My father would be deeply pleased that people who can fully appreciate this small irony are now working on ways to prevent the misapplication of computers as foible-amplifiers.

Forum on Risks to the Public in Computer Systems [Reaction]

<i>Bob <Carter@RUTGERS.ARPA>
8 Aug 85 19:10 EDT (Thu)

Thanks for the copy of Vol. I, No. 1. Herewith a brief reaction. This is sent to you directly because I'm not sure whether discussion of the digest is appropriate for inclusion in the digest.

1. Please mung RISKS so that it does not break standard undigestifying software (in my case, BABYL).

[BABYL is an EMACS-TECO hack. It seems to be a real bore to use, with lots of pitfalls still. But I'll see what I can do. Alternatively, shorter issues might help. PGN]

2. I think RISKS is clearly an idea whose time has come, but I'm not entirely sure it has been sufficiently thought through.

[I should hope not! It is a cooperative venture. I just happen to be trying to moderate it. PGN]

a. You cast your net altogether too widely, and include some topics that have been discussed extensively on widely-read mailing lists. Star Wars, the Lin paper, the Parnas resignation, and related topics have been constructively discussed on ARMS-D. I have considerable doubt about the utility of replicating this discussion. (The moderators of HUMAN-NETS and POLI-SCI have both adopted the policy of directing SDI debate to that forum. Would it be a good idea to follow that example?}
To some extent, yes. However, one cannot read ALL of the interesting BBOARDS — there are currently hundreds on the ARPANET alone, many of which have some bearing on RISKS. Also, browsers from other networks are at a huge disadvantage unless they have connections, hours of spare time, money, etc. This is a FORUM ON RISKS, and should properly address that topic. We certainly should not simply reproduce other BBOARDS, but some duplication seems tolerable. (I'll try to keep it at the end of each issue, so you won't have to wade through it.) By the way, I had originally intended to mention ARMS-D in RISKS vol 1 no 1, but did not have time to check it out in detail. For those of you who want to pursue it, next following is the essence of the blurb taken from the Network Information Center, SRI-NIC.ARPA:<NETINFO>INTEREST-GROUPS.TXT. PGN]

[ ARMS-D@MIT-MC:

The Arms-Discussion Digest is intended to be a forum for discussion of arms control and weapon system issues. Messages are collected, edited into digests and distributed as the volume of mail dictates (usually twice a week).

Old digests may be FTP'ed from MIT-MC(no login required). They are archived at BALL; ARMSD ARCN, where n is the issue no.

All requests to be added to or deleted from this list, problems, questions, etc., should be sent to Arms-D-REQUEST@MIT-MC.

Moderator: Harold G. Ancell <HGA@MIT-MC> ]

b. You do not cover the topics which, in my opinion, are going to generate more law-making than anything you do touch on. In particular, the health hazards (if any) of CRT use, and the working conditions (including automated performance testing) of "pink-collar" CRT users are going to be among the most important labor-relations issues of the next few years. Many people think these more imminent risks than those mentioned in the RISKS prospectus.

[Fine topic! PGN]

3. I think a digest is an animal that differs considerably from print media, but is no less important. I get the feeling that you consider yourself a country cousin of the ACM publications and of SEN. Wrong! You're not inferior, you are just editing in a different medium and as you put your mind to the task, I hope you come to take them with a larger grain of salt. In particular,

! Chinese computer builder electrocuted by his smart computer after he built a newer one. "Jealous Computer Zaps its Creator"! (SEN 10 1)

was a National Inquirer-style joke. The editor of SEN should not have reprinted it, and you probably should not have included it in a serious list of computer-related failures.

[The editor of SEN has sometimes been known to indulge in levity. In this case it appears that a Chinese engineer was indeed electrocuted — and that is an interesting case of computer-related disaster. On the other hand, if someone can believe that an AI automatic programming routine can write many million lines of correct code, then he might as well believe that a smart computer system could express jealousy and cause the electrocution! Actually, Bob used "PEN" throughout rather than "SEN", but "Software Engineering Notes" was the only sensible interpretation I could come up with, so I changed it. Do I have a "PEN" pal? PGN]

4. It seems to me that it is precisely in the area of serious hardware and software failures that RISKS should make its mark. Directing itself to that topic, it fills a spot no existing list touches on directly, and treats a matter that concerns every computer professional who is earning a decent living. Litigation about defective software design and programming malpractice will be the inevitable consequence of risks, and RISKS is the only place to discuss avoiding them. Please consider focussing the list more closely on that subject.

[Bob, Thanks for your comments. I heartily agree on the importance of the last item. But, I do not intend to generate all of the material for this forum, and can only smile when someone suggests that this forum is not what it should be. I look forward to your help! PGN]

[End of Bob Carter's message and my interspersions.]

RISKS forum [including An Air-Traffic Control Problem]

Scott M. Rose <rose@uw-bluechip.arpa>
16 Aug 85 21:06:39 PDT (Fri)
I had kind of hoped that somebody would submit something on the recent problem in Aurora Illinois, whereby a computer cable was cut that brought information from RADAR sensors to the regional air traffic control center there. Supposedly, the system was designed to be sufficiently redundant to handle such a failure gracefully, but this turned out not to be the case: there were several close calls as the system went up and down repeatedly. There was information about the problem in the New York Times and the Chicago Tribune, at least... but not in very good detail.

I wonder if the forum is the right format for such a group. The problem is that one may find oneself reluctant to report on such an incident that was widely reported in the popular press, and was current, for fear that a dozen others have done the same. Yet in this case, the apparent result is that NOBODY reported on it, and I think such an event ought not pass without note on this group. I might propose something more like the info-nets group, where postings are automatically forwarded to group members. If problems arose, then the postings could be filtered by the moderator... say, on a daily basis? Just an idea...

-S Rose

[Please don't feel reluctant to ask whether someone has reported an interesting event before you go to any potentially duplicate effort. We'd rather not miss out entirely.]

Risks in AI Diagnostic Aids

<Smith@UDel-Dewey.ARPA>
Sun, 18 Aug 85 12:23:25 EDT

I would enjoy a discussion on the legal and ethical problems that have come up with the creation of AI diagnostic aids for doctors. Who takes the blame if the advice of a program causes a wrong diagnosis? The doctor (if so, then who would use such a program?!?), the program's author(s) (if so, then who would write such a program?!?), the publishers/distributors of the program (if so, then who would market such a program?!?), .... These nagging questions will have to be answered before anyone is going to make general use of these programs I would be very interested in hearing what other people think about this question. It seems to me that it would be a suitable one for this bboard.

Art Smith

(smith@UDel-Dewey.ARPA)

Following are several items on the Strategic Defense Initiative and related subjects.

WARNING!! [A Trojan Horse Bites Man]

Don Malpass <malpass@ll-sst>
Thu, 15 Aug 85 11:05:48 EDT

Today's Wall St. Journal contained the following article. I think it is of enough potential significance that I'll enter the whole thing. In addition to the conclusions it states, it implies something about good backup procedure discipline.

In the hope this may save someone,

Don Malpass

************************************************************************* (8/15/85 Wall St. Journal) ARF! ARF!

Richard Streeter's bytes got bitten by an "Arf Arf," which isn't a dog but a horse.

Mr. Streeter, director of development in the engineering department of CBS Inc. and home-computer buff, was browsing recently through the offerings of Family Ledger, a computer bulletin board that can be used by anybody with a computer and a telephone to swap advice, games or programs - or to make mischief. Mr. Streeter loaded into his computer a program that was billed as enhancing his IBM program's graphics; instead it instantly wiped out the 900 accounting, word processing and game programs he had stored in his computer over the years. All that was left was a taunt glowing back at him from the screen: "Arf! Arf! Got You!"

"HACKERS" STRIKE AGAIN

This latest form of computer vandalism - dubbed for obvious reasons a Trojan Horse - is the work of the same kind of anonymous "hackers" who get their kicks stealing sensitive data from government computers or invading school computers to change grades. But instead of stealing, Trojan Horses just destroy all the data files in the computer.

Trojan Horse creators are nearly impossible to catch - they usually provide phony names and addresses with their programs - and the malevolent programs often slip by bulletin-board operators. But they are becoming a real nuisance. Several variations of the "Arf! Arf!" program have made the rounds, including one that poses as a "super-directory" that conveniently places computer files in alphabetical order.
Operators have begun to take names and addresses of electronic bulletin-board users so they can check their authenticity. When a computer vandal is uncovered, the word is passed to other operators. Special testing programs also allow them to study the wording of submitted programs and detect suspicious commands.

INTERFAKER BEWARE

But while Al Stone, the computer consultant who runs Long Island based Family Ledger, has such a testing program, he says he didn't have time to screen the "Arf! Arf!" that bit Mr. Streeter. "Don't attempt to run something unless you know its pedigree," he says.

That's good advice, because the computer pranksters are getting more clever - and nastier. They are now creating even-more-insidious programs that gradually eat away existing files as they are used. Appropriately enough, these new programs are known as "worms".

(8/15/85 Wall St. Journal) ******************************************

Software engineering and SDI

<mooremj@EGLIN-VAX>
Mon, 19 Aug 85 13:56:21 CDT

[FROM Soft-Eng Digest Fri, 23 Aug 85 Volume 1 : Issue 31]

Dr. David Parnas has quite accurately pointed out some of the dangers inherent in the software to be written for the Strategic Defense Initiative. I must take exception, however, to the following statement from the Boston Globe story quoted in Volume 1, Issue 29, of this digest:

"To imagine that Star Wars systems will work perfectly without testing is ridiculous. A realistic test of the Strategic Defense Initiative would require a practice nuclear war. Perfecting it would require a string of such wars."

There are currently many systems which cannot be fully tested. One example is the software used in our present defense early warning system. Another example, one with which I am personally familiar, is the Range Safety Command Destruct system at Cape Canaveral Air Force Station. This system provides the commands necessary to destroy errant missiles which may threaten populated areas; I wrote most of the software for the central computer in this system. The system can never be fully tested in the sense implied above, for to do so would involve the intentional destruction of a missile for testing purposes only. On the other hand, it must be reliable: a false negative (failure to destroy a missile which endangers a populated area) could cause the loss of thousands of lives; a false positive (unintentional destruction of, say, a Space Shuttle mission) is equally unthinkable. There are many techniques available to produce fault-tolerant, reliable software, just as there are for hardware; the Range Safety system was designed by some of the best people at NASA, the U. S. Air Force, and several contractors. I do not claim that a failure of this system is "impossible", but the risk of a failure, in my opinion, is acceptably low.

"But ANY risk is too great in Star Wars!"

I knew someone would say that, and I can agree with this sentiment. The only alternative, then, is not to build it, because any system at all will involve some risk (however small) of failure; and failure will, as Dr. Parnas has pointed out, lead to the Ultimate Disaster. I believe that this is what Dr. Parnas is hoping to accomplish: persuading the authorities that the risk is unacceptable.

It won't work. Oh, perhaps it will in the short run; "Star Wars" may not be built now, or ever. But sooner or later, some system will be given life-and-death authority over the entire planet, whether it is a space defense system, a launch-on-warning strategic defense system, or something else. The readers of this digest are the present and future leaders in the field of software engineering. It is our responsibility to refine the techniques now used and to develop new ones so that these systems WILL be reliable. I fear that some first-rate people may avoid working on such systems because they are "impossible"; this will result in second-rate people working on them, which is something we cannot afford. This is NOT a slur at Dr. Parnas. He has performed an invaluable service by bringing the public's attention to the problem. Now it is up to us to solve that problem.

I apologize for the length of this message. The above views are strictly my own, and do not represent my employer or any government agency.

Martin J. Moore
Senior Software Analyst
Software engineering and SDI

horning@decwrl.ARPA (Jim Horning)
21 Aug 1985 1243-PDT (Wednesday)

Subject: Trip Report: Computing in Support of Battle Management

[This is a relatively long report, because I haven't been able to come up with a simple characterization of an interesting and informative day.]

Background:

On August 13 I travelled to Marina del Rey to spend a day with the U.S. Department of Defense Strategic Defense Initiative Organization Panel on Computing in Support of Battle Management (DoD SDIO PCSBM).

SDI is the "Star Wars" antiballistic missile system; PCSBM is the panel Dave Parnas resigned from.

I wasn't really sure what to expect. As I told Richard Lau when he invited me to spend a day with them, I'd read what Parnas wrote, but hadn't seen the other side. He replied that the other side hadn't been written yet. "Come on down and talk to us. The one thing that's certain is that what we do will have an impact, whether for good or for ill."

Summary:

The good news is that the panel members are not crazies; they aren't charlatans; they aren't fools. If a solution to SDI's Battle Management Software problem can be purchased for five billion dollars (or even ten), they'll probably find it; if not, they'll eventually recognize that it can't.

The bad news is that they realize they don't have the expertise to solve the problem themselves, or even to direct its solution. They accept Dave Parnas's assessment that the software contemplated in the "Fletcher Report" cannot be produced by present techniques, and that AI, Automatic Programming, and Program Verification put together won't generate a solution. Thus their invitations to people such as myself, Bob Balzer, and Vic Vyssotsky to come discuss our views of the state and prospects of software technology.

I think a fair summary of the panel's current position is that they are not yet convinced that the problem cannot be modified to make it soluble. ("Suppose we let software concerns drive the system architecture? After all, it is one of the two key technologies.") They are trying to decide what must be done to provide the information that would be needed in the early 1990s to make a decision about deploying a system in the late 1990s.

Assumptions:

Throughout the day's discussions, there were repeated disconnects between their going-in assumptions and mine. In fairness, they tried to understand the sources of the differences, to identify their assumptions, and to get me to identify and justify mine.

- Big budgets: I've never come so close to a trillion-dollar ($10^{12}$) project before, even in the planning stage. ("The satellite launches alone will cost upwards of $500 billion, so there's not much point in scrimping elsewhere.")
  - I was unprepared for the intensity of their belief that any technical problem could be steamrollered with a budget that size.
  - They seemed surprised that I believed that progress in software research is now largely limited by the supply of first-rate people, and that the short-term effect of injecting vastly more dollars would be to slow things down by diverting researchers to administer them.
- Big software: They were surprised by my observation that for every order of magnitude in software size (measured by almost any interesting metric) a new set of problems seems to dominate.
  - This implies that no collection of experiments with million-line "prototypes" can ensure success in building a ten-million-line system. I argued that the only prototype from which they would learn much would be a full-scale, fully-functional one. Such a prototype would also reveal surprising consequences of the specification.
The FIFTEENTH LAW OF SYSTEMANTICS: A complex system that works is invariably found to have evolved from a simple system that works.

- Only Chuck Seitz and Bijoy Chatterjee seemed to fully appreciate why software doesn't just "scale up" (doubtless because of their hardware design experience). It is not a "product" that can be produced at some rate, but the design of a family of computations; it is the computations that can be easily scaled.

- Reliability: I had assumed that one of the reasons Battle Management software would be more difficult than commercial software was its more stringent reliability requirement. They assume that this is one of the parameters that can be varied to make the problem easier.

Discussion:

The Panel is still in the process of drafting its report on Battle Management Systems. Although they take the need to produce such a system as a given, almost anything else is negotiable. (In particular, they do not accept the "Fletcher Report" as anything more than a springboard for discussion, and criticize current work for following it too slavishly. The work at Rome Air Development Center—which produced estimates like 24.61 megalines of code, 18.28 gigaflops per weapons platform—was mentioned contemptuously, while the Army work at Huntsville was considered beneath contempt.)

The following comments are included merely to indicate the range and diversity of opinions expressed. They are certainly not official positions of the panel, and—after being filtered though my understanding and memory—may not even be what the speaker intended. Many of the inconsistencies are real; the panel is working to identify and resolve them.

- The problem may be easier than a banking system, because: each autonomous unit can be almost stateless; a simple kernel can monitor the system and reboot whenever a problem is detected; there are fewer people in the loop; more hardware overcapacity can be included.

- If you lose a state it will take only a few moments to build a new state. (Tracks that are more than 30 minutes old are not interesting.)

- Certain kinds of reliability aren't needed, because: a real battle would last only a few minutes; the system would be used at most once; with enough redundancy it's OK for individual weapons to fail; the system doesn't have to actually work, just be a credible deterrent; the system wouldn't control nuclear weapons—unless the Teller "pop up" scheme is adopted; the lasers won't penetrate the atmosphere, so even if the system runs amok, the worst it could do would be to intercept some innocent launch or satellite.

- We could debug the software by putting it in orbit five or ten years before the weapons are deployed, and observing it. We wouldn't even have to deploy them until the system was sufficiently reliable. Yes, but this would not test the important modes of the system.

- Dependence on communication can be minimized by distributing authority: each platform can act on its own, and treat all communication as hints.

- With a multi-level fault-tolerance scheme, each platform can monitor the state of its neighbors, and reboot or download any that seem to be malfunctioning.

- In fifteen years we can put 200 gigaflops in orbit in a teacup. Well, make that a breadbox.

- Space qualification is difficult and slow. Don't count on microprocessors of more than a few mips in orbit. Well, maybe we could use fifty of them.

- How much can we speed up computations by adding processors? With general-purpose processors, probably not much. How much should we rely on special-purpose space-qualified processors?

- Processor cost is negligible. No, it isn't. Compared to software costs or total system costs it is. No, it isn't, you are underestimating the costs of space qualification.

- 14 MeV neutron flux cannot effectively be shielded against and represents a fundamental limitation on the switching-speed, power product. Maybe we should put all the computationally intensive components under a mountain. But that increases the dependence on communication.

- Maybe we could reduce failure rates by putting the software in read-only memory. No, that makes software maintenance incredibly difficult.
Flaccidware. It's software now, but it can become hardware when necessary.

Is hardware less prone to failure if switched off? Maybe we could have large parts of the system on standby until the system goes on alert. Unfortunately, the dominant hardware failure modes continue even with power off.

The software structure must accommodate changes in virtually all component technologies (weapons, sensors, targets, communication, computer hardware) during and following deployment. But we don't have much technology for managing rapid massive changes in large systems.

Relation to Critics:

Dave Parnas's criticisms have obviously been a matter of considerable concern for the panel. Chuck Seitz and Dick Lau both said explicitly that they wouldn't be satisfied making a recommendation that failed to address the issues Dave and other critics have raised. Chuck also distributed copies of "The Star Wars Computer System" by Greg Nelson and David Redell, commending it to the attention of the panel as "Finally, some well-written and intelligent criticism."

Richard Lipton had a somewhat different attitude: How can they say that what we are going to propose is impossible, when even we don't know yet what we're going to propose? And why don't software researchers show more imagination? When a few billion dollars are dangled in front of them, the physicists will promise to improve laser output by nine decimal orders of magnitude; computer scientists won't even promise one or two for software production.

The minutes of the August 12 meeting contain the following points:

- Critics represent an unpaid "red team" and serve a useful function in identifying weak points in the program.
- Critiques should be acknowledged, and areas identified as to how we can work to overcome these problem areas.
- Throughout our discussions, and in our report we should reflect the fact that we have accepted a degree of uncertainty as an inherent part of the strategic defense system.
- How to get the system that is desired? This basic problem goes back to defining requirements—a difficult task when one is not quite sure what one wants and what has to be done.

Prospects:

After all of this, what do I think of the prospects for SDI Battle Management Software? I certainly would not be willing to take on responsibility for producing it. On the other hand, I cannot say flatly that no piece of software can be deployed in the 1990s to control a ballistic missile defense system. It all depends on how much functionality, coordination, and reliability are demanded of it.

Unfortunately, as with most other computer systems, the dimension in which the major sacrifice will probably be made is reliability. The reality of the situation is that reliability is less visible before deployment than other system parameters and can be lost by default. It is also probably the hardest to remedy post facto. Of course, with a system intended to be used "at most once," there may be no one around to care whether or not it functioned reliably.

Despite these misgivings, I am glad that this panel is taking seriously its charter to develop the information on which a deployment decision could responsibly be based.

Jim H.

Software engineering and SDI

[An earlier SU-bboard message that prompted the following sequence of replies seemed like total gibberish, so I have omitted it. PGN]

John McCarthy <JMC@SU-AI.ARPA>
13 Aug 85 1521 PDT

Subject: Forum on Risks to the Public in Computer Systems
To: su-bboards@SU-AI.ARPA
[but not To: RISKS...]

I was taking [as?] my model Petr Beckmann's book "The Health Hazards of not Going Nuclear" in which he contrasts the slight risks of nuclear energy with the very large number of deaths resulting from conventional energy sources
from, e.g. mining and air pollution. It seemed to me that your announcement was similarly one sided in its consideration in risks of on-line systems and ignoring the possibility of risks from their non-use. I won't be specific at present, but if you or anyone else wants to make the claim that there are no such risks, I'm willing to place a substantial bet.

[Clearly both inaction and non-use can be risky. The first two items at the beginning of this issue (Vol 1 no 2) — the lobstermen and the Union Carbide case — involved inaction. PGN]

Software engineering and SDI

John McCarthy <JMC@SU-AI.ARPA>
14 Aug 85 1635 PDT

Subject: IJCAI as a forum

Like Chris Stuart, I have also contemplated using IJCAI as a forum. My issue concerns the computer scientists who have claimed, in one case "for fundamental computer science reasons" that the computer programs required for the Strategic Defense Initiative (Star Wars) are impossible to write and verify without having a series of nuclear wars for practice. Much of the press (both Science magazine and the New York Times) have assumed (in my opinion correctly) that these people are speaking, not merely as individuals, but in the name of computer science itself. The phrase "for fundamental computer science reasons" was used by one of the computer scientist opponents.

In my opinion these people are claiming an authority they do not possess. There is no accepted body of computer science principles that permits concluding that some particular program that is mathematically possible cannot be written and debugged. To put it more strongly, I don't believe that there is even one published paper purporting to establish such principles. However, I am not familiar with the literature on software engineering.

I think they have allowed themselves to be tempted into exaggerating their authority in order to support the anti-SDI cause, which they support for other reasons.

I have two opportunities to counter them. First, I'm giving a speech in connection with an award I'm receiving. Since I didn't have to submit a paper, I was given carte blanche. Second, I have been asked by the local arrangements people to hold a press conference. I ask for advice on whether I should use either of these opportunities. I can probably even arrange for some journalist to ask my opinion on the Star Wars debugging issue, so I wouldn't have to raise the issue myself. Indeed since my position is increasingly public, I might be asked anyway.

To make things clear, I have no position on the feasibility of SDI, although I hope it can be made to work. Since even the physical principles that will be proposed for the SDI system haven't been determined, it isn't possible to determine what kind of programs will be required and to assess how hard they will be to write and verify. Moreover, it may be possible to develop new techniques involving both simulation and theorem proving relevant to verifying such a program. My sole present point is that no-one can claim the authority of computer science for asserting that the task is impossible or impractical.

There is even potential relevance to AI, since some of the opponents of SDI, and very likely some of the proponents, have suggested that AI techniques might be used.

I look forward to the advice of BBOARD contributors.

Software engineering and SDI

Peter Karp <KARP@SUMEX-AIM.ARPA>
Thu 15 Aug 85 00:17:09-PDT

Subject: Verifying SDI software

John McCarthy: I argue CPSR's approach is reasonable as follows:

1. I assume you admit that bugs in the SDI software would be very bad since this could quite conceivably leave our cities open Soviet attack.

2. You concede software verification theory does not permit proof of correctness of such complex programs. I concede this same theory does not show such proofs are impossible.

3. The question to responsible computer professionals then becomes: From your experience in developing and debugging complex computer systems, how likely do you believe it is that currently possible efforts could produce error-free software, or even software whose reliability is acceptable given the risks in (1)?

Clearly answering (3) requires subjective judgements, but computer professionals are among the best people to ask to
make such judgements given their expertise.

I think it would be rather amusing if you told the press what you told bboard: that you "hope they can get it to work".

**Software engineering and SDI**

*John McCarthy <JMC@SU-AI.ARPA>*

16 Aug 85 2200 PDT

I thank those who advised me on whether to say something about the SDI controversy in my lecture or at the press conference. I don't presently intend to say anything about it in my lecture. Mainly this is because thinking about what to say about a public issue would interfere with thinking about AI. I may say something or distribute a statement at the press conference.

I am not sure I understand the views of those who claim the computer part of SDI is infeasible. Namely, do they hope it won't work? If so, why? My reactionary mind thinks up hypotheses like the following. It's really just partisanship. They have been against U.S. policy in many areas including defense, that they automatically oppose any initiative and then look for arguments.

**Software engineering and SDI**

*vax-populi!dparnas@nrl-css (Dave Parnas)*

Thu, 15 Aug 85 13:01:46 PDT

Subject: Re: [John McCarthy <JMC@SU-AI.ARPA>: IJCAI as a forum ]

McCarthy is making a classic error of criticizing something that he has not read. I have not argued that any program cannot be written and debugged. I argue a much weaker and safer position, that we cannot know that the program has been debugged. There are "fundamental computer science reasons" for that, they have to do with the size of the smallest representation of the mathematical functions that describe the behaviour of computer software and our inability to know that the specifications are correct.

Dave

Date: Thu, 15 Aug 85 13:14:22 pdt From: vax-populi!dparnas@nrl-css (Dave Parnas) To: neumann@SRI-CSL.ARPA

Subject: Copy of cover letter to Prof. John McCarthy

Dear Dr. M

A friend of mine, whose principal weakness is reading the junk mail posting on bulletin boards sent me a copy of your posting with regard to SDI.

It is in general a foolish error to criticize a paper that you have not read on the basis of press reports of it.

Nobody has, in fact, claimed that any given program cannot be written and "debugged" (whatever that means). The claim is much weaker, that we cannot know with confidence that the program does meet its specification and that the specification is the right one. There is both theoretical (in the form of arguments about the minimal representation of non-continuous functions) and empirical evidence to support that claim. The fact that you do not read the literature on software engineering does not give you the authority to say that there are no papers supporting such a claim.

As I would hate to see anyone, whether he be computer scientist or AI specialist, argue on the basis of ignorance, I am enclosing ...

**Software engineering and SDI**

*Gary Martins <GARY@SRI-CSL.ARPA>*

Thu 15 Aug 85 18:50:46-PDT

Subject: Speaking Out On SDI

Dear Dr. McC -

In response to your BB announcement:

1. Given that IJCAI is by and large a forum for hucksters and crackpots of various types, it is probably a poor choice of venue for the delivery of thoughts which you'd like taken seriously by serious folks.

2. Ditto, for tying your pro-SDI arguments in with "AI"; it can only lower the general credibility of what you have to say.
3. You are certainly right that no-one can now prove that the creation of effective SDI software is mathematically impossible, and that part of your argument is beyond reproach, even if rather trivial. However, you then slip into the use of the word “impractical”, which is a very different thing, with entirely different epistemological status. On this point, you may well be entirely wrong — it is an empirical matter, of course.

I take no personal stand on the desirability or otherwise of SDI, but as a citizen I have a vested interest in seeing some discussions of the subject that are not too heavily tainted by personal bias and special pleading.

Gary R. Martins Intelligent Software Inc.

-----------------------------------------------

International Conference on Software Engineering 28-30 August 1985, London UK Feasibility of Software for Strategic Defense Panel Discussion 30 August 1985, 1:30 - 3:00 PM

Panelists: Frederick P. Brooks III, University of North Carolina David Parnas, University of Victoria Moderator: Manny Lehman, Imperial College

This panel will discuss the feasibility of building the software for the Strategic Defense System ('Star Wars') so that that software could be adequately trusted to satisfy all of the critical performance goals. The panel will focus strictly on the software engineering problems in building strategic defense systems, considering such issues as the reliability of the software and the manageability of the development.

[This should be a very exciting discussion. Fred has extensive hardware, software, and management experience from his IBM OS years. David's 8 position papers have been widely discussed — and will appear in the September American Scientist. We hope to be able to report on this panel later (or read about it in ARMS-D??). Perhaps some of you will be there and contribute your impressions. PGN]

Software engineering and SDI

Tom Parmenter <parmenter@SCRC-STONY-BROOK.ARPA>

Mon, 15 Jul 85 11:05 EDT

From an article in Technology Review by Herbert Lin on the difficulty (impossibility) of developing software for the Star Wars (Strategic Defense Initiative) system:

Are there alternatives to conventional software development? Some defense planners think so. Major Simon Worden of the SDI office has said that

"A human programmer can't do this. We're going to be developing new artificial intelligence systems to write the software. Of course, you have to debug any program. That would have to be AI too."

The Madison Paper on Computer Unreliability and Nuclear War

uwmacclmyers@wisc-rsch.arpa (Latitudinarian Lobster)

Wed, 14 Aug 85 18:08:57 CDT

The following may be reproduced in any form, as long as the text and credits remain unmodified. It is a paper especially suited to those who don't already know a lot about computing. Please mail comments or corrections to:

Jeff Myers [Something was lost here...] University of Wisconsin-Madison reflect the views of any other Madison Academic Computing Center person or group at UW-Madison. 1210 West Dayton Street Madison, WI 53706 ARPA: uwmacclmyers@wisc-rsch.ARPA UUCP: ..!{harvard,ucbvax,allegra,heurikon,ihnpc4,seismo}!uwvax!uwmacclmyers BitNet: MYERS at MACCWISC

COMPUTER UNRELIABILITY AND NUCLEAR WAR

Larry Travis, Ph.D., Professor of Computer Sciences, UW-Madison Daniel Stock, M.S., Computer Sciences, UW-Madison Michael Scott, Ph.D., Computer Sciences, UW-Madison Jeffrey D. Myers, M.S., Computer Sciences, UW-Madison James Greuel, M.S., Computer Sciences, UW-Madison James Goodman, Ph.D., Assistant Professor of Computer Sciences, UW-Madison Robin Cooper, Ph.D., Associate Professor of Linguistics, UW-Madison Greg Brewster, M.S., Computer Sciences, UW-Madison

Madison Chapter Computer Professionals for Social Responsibility June 1984
Can a computer declare war?

Clifford Johnson <GA.CJJ@Forsythe>

Wed, 21 Aug 85 17:46:55 PDT

************* CAN A COMPUTER DECLARE WAR?

Below is the transcript of a court hearing in which it is was argued by the Plaintiff that nuclear launch on warning capability (LOWC, pronounced lou-see) unconstitutionally delegates Congress’s mandated power to declare war.

The Plaintiff is a Londoner and computer professional motivated to act by the deployment of Cruise missiles in his hometown. With the advice and endorsement of Computer Professionals for Social Responsibility, on February 29, 1984, he filed a complaint in propria persona against Secretary of Defense Caspar Weinberger seeking a declaration that peacetime LOWC is unconstitutional. The first count is presented in full below; a second count alleges a violation of Article 2, Part 3 of the United Nations Charter which binds the United States to settle peacetime disputes “in such a manner that international peace and security, and justice, are not endangered”:

1. JURISDICTION: The first count arises under the Constitution of the United States at Article I, Section 8, Clause 11, which provides that “The Congress shall have Power … To declare War”; and at Article II, Section 2, Clause 1, which provides that “The President shall be Commander in Chief” of the Armed Forces.

2. Herein, "launch-on-warning-capability" is defined to be any set of procedures whereby the retaliatory launching of non-recoverable nuclear missiles may occur both in response to an electronically generated warning of attacking missiles and prior to the conclusively confirmed commencement of actual hostilities with any State presumed responsible for said attack.

3. The peacetime implementation of launch-on-warning-capability is now presumed constitutional, and its execution by Defendant and Defendant's appointed successors is openly threatened and certainly possible.

4. Launch-on-warning-capability is now subject to a response time so short as to preclude the intercession of competent judgment by the President or by his agents.

5. The essentially autonomous character of launch-on-warning-capability gives rise to a substantial probability of accidental nuclear war due to computer-related error.

6. Said probability substantially surrenders both the power of Congress to declare war and the ability of the President to command the Armed Forces, and launch-on-warning-capability is therefore doubly repugnant to the Constitution.

7. The life and property of Plaintiff are gravely jeopardized by the threat of implementation of launch-on-warning-capability.

WHEREFORE, Plaintiff prays this court declare peacetime launch-on-warning-capability unconstitutional.

************* THE HEARING IN THE COURT OF APPEALS FOLLOWS

[in the original message, and is too lengthy to include here. I presume you will find it in ARMS-D — see my interpolation into the note from Bob Carter above. Otherwise, you can FTP it from SRI-CSL:<RISKS>JOHNSON.HEARING. PGN]
Please report problems with the web pages to the maintainer