# HUMAN ASPECTS OF COMPUTING IN LARGE PHYSICS COLLABORATIONS

*Vladimir Chaloupka*
University of Washington, Seattle, USA

**Abstract**
In these lectures, we discuss the link between human interactions in large communities on the one hand, and the modern, state-of-the-art computing and communication tools on the other hand. The scope of the lectures is broader than the original title indicates. After a review of the relevant aspects of the psychology of programming, some remarks on problems specific to High Energy Physics are made. The second lecture deals with general aspects of Computers and Society, and addresses the concerns of "post-modern" critics of science and technology.

## 1 INTRODUCTION

Traditionally, computing is discussed in terms of hardware and software. In these lectures, we will introduce a division across different lines: technical aspects versus human aspects, and we will concentrate on the latter category.

Examples of technical aspects, taken from both the hardware and software categories, are:

- Pentium vs. PowerPC chip
- high bandwidth network technology
- C++ vs. JAVA

Examples of the human aspects might be:

- inter-national [sic] ego problems in collaborations
- management of the software process
- use of English by non-native speakers (and listeners)

There are many connections between the technical and human aspects, but the distinction is useful. While preparing the Lectures (originally limited to computing in High Energy Physics), I found a set of issues which seemed very interesting, very important and very appropriate for this School. The questions are: what is the real, as contrasted with the hyped-up, promise and future of computing and networking in the 21st century, and what is the desirability, as opposed to just inevitability, of particular developments? These issues are part of a broader concern about the role of science and technology in society. Of course, there is another question: do Physics Computing Experts have something special to say, and/or do they have special reasons to worry about these issues? Needless to say, my answer is yes, we indeed do have a good reason, even duty, to worry about this, and we do have something to contribute.

It so happens that at this particular School, there is an additional motivation for a Lecture or two on these esoteric subjects. We are in the country of playwright/philosopher/king Vaclav Havel, who is very concerned with the role of science and technology in society. His views are "post-modern": quite critical and controversial, and it appears appropriate for you to learn about them.

So, the first Lecture will address the human aspects specific to the psychology of computer programming, and to computing in large physics collaborations in particular. The second Lecture will deal with the human aspects of computing in general. Both sets of issues are quite personal (as human issues tend to be) and many of the statements people make are controversial. In addition, not much is really known, and the issues are rarely discussed at professional meetings.

## 2 PSYCHOLOGY OF PROGRAMMING, AND COMPUTING IN LARGE COLLABORATIONS.

The first part of this section consists of a review of four works on the subject; additional references can be found by searching for concepts such as "groupware", "collaboratory", "virtual laboratory" and/or "Computer Supported Collaborative Work" (CSCW). These searches tend to produce references to rather technical discussions of tools, methods and applications. Recent examples, with many references to other works, are "Groupware: Technology and Applications" (COLEMAN 95) and "Towards a CSCW Framework for Scientific Cooperation in Europe' (LUBICH 95). A nice collection of selected papers (including the Bush and Engelbart classics) is in GREIF 88. A very large (and up-to-date) collection of references is included in the "Rise and Resurrection of the American Programmer" (YOURDON 97).

And now on to the works selected for more detailed review and comments.

### 2.1 Psychology of Computer Programming

This book (WEINBERG 71) became something of a watershed in the field. It is based on experimental studies of the behavior of programmers, based on anecdotal stories, and/or on controlled experiments.

The stories are good: interesting and useful. They range from the description of "interference of the observer with the subject being observed - a kind of uncertainty principle", to a nice anecdote leading to the concept of "rotating devil's advocate". And the experiments are good, too: see e.g. the remarkable experiments on "cognitive dissonance". This leads Weinberg to the suggestion of "ego-less programming" (which is probably not so easy to accomplish and maintain). In any case, the book is also an example of a delightfully self-deprecating, "ego-less psychology" (e.g. when Weinberg points out that "Essentially all psychological tests assume that the psychologists who made the tests are smarter than the people who take them.")

After a good discussion of "essential personality factors" of programmers (which are identified as humility, assertiveness and sense of humor), the book ends with an excellent (if somewhat emotional) epilogue on the social responsibility of programmers/software scientists. This theme is in fact quite relevant to the second Lecture.

Overall conclusion: Weinberg hoped that his book would "trigger the beginning of a new field of study", and indeed, there has been much research, conferences and books on this subject (for examples, in ten year intervals from Weinberg, see SHNEIDERMAN 80 and HOC 90.) However, the Weinberg book, dated as it is, is still a classic, and a pleasure, to read.

### 2.2 The Mythical Man-Month; Essays on Software Engineering

The original 1975 edition is perhaps the most quoted book of its kind. The 1995 edition (BROOKS 95) contains 4 additional chapters with updates.

The "title argument" is about the question "how many man-months will it take?". Criticizing this standard measure of software complexity, Brooks says: "The bearing of a child takes nine months, no matter how many women are assigned. Many software tasks have this characteristic.... /in a team work/ the added burden is made up of training and communication." Since the "added burden" increases with more personnel, this results in an optimum manpower for each project, and sometimes more people will produce less ...

An important argument is about "aristocracy vs. democracy" in software engineering:

"Conceptual integrity dictates that design must proceed from one mind, or from a very small number of agreeing resonant minds." This means no "design by committee", no "consensus building", but rather an architect (or a few architects) is needed, with a hierarchy of supporting roles (implementers, testers, ...and managers)

It is interesting to see early mentions of encapsulation and structured programming. Some arguments are dated (memory limits etc.) and some opinions are corrected as mistaken in the updates

(e.g. the 1975 opinion about information hiding). When considering the issues of reuse, Brooks quotes David Parnas (the author of the information hiding concept, and the author of "Software Aspects of SDI"): "Reuse is something which is far easier to say than to do. Doing it requires both good design and very good documentation." (and also DeMarco: "...there is a big expense in making things reusable.") In any case, it is good to keep in mind the 1996 $500 million Ariane crash, caused by a re-use specification error!

Of special relevance for High Energy Physics may be Brooks' claim that "Object-oriented techniques will not make the first project development any faster, or the next one. The fifth one in that family will go blazingly fast." But of course, in HEP there will hardly ever be anything of the "fifth one in that family"!

Another important problem is that with increased abstraction come increasingly large vocabularies (and semantics, syntax, grammar, ...), and programming becomes increasingly more difficult to master. Brooks argues that the natural languages which we all manage to learn (English, French, Czech, ...) are even more complex, but I should like to point out that we only learn at most few of them, it takes a long time, the difficulty increases with age, and the native accent remains!

The most famous argument is the 1986 claim of "No Silver Bullet": that "a decade would not see any programming technique that would by itself bring an order-of-magnitude improvement in software productivity". The reasoning is that software construction consists of "essential tasks" (creation of complex conceptual structures), plus "accidental tasks" (coding etc.). Most of the big past gains in software productivity came from dealing with the "accidental tasks"; the progress in "essential tasks" will be much harder and slower. This means that "building software will always be hard. There is inherently no silver bullet." Chapter 17 reviews (in 1995) the rebuttals of "No Silver Bullet" (Brooks says that most of the rebuttals "agree with most of the arguments in NSB, but then go on to assert that there is indeed a silver bullet, which the author [of the rebuttal] has invented.")

The book ends with an almost unbearably excellent "The Mythical Man-Month after 20 Years" (chapter 19) and Epilogue: pondering why the book is still relevant, Brooks writes:

> " ...The Mythical Man-Month is only incidentally about software but primarily about how people in teams make things. .... Human history is a drama in which stories stay the same, the scripts of those stories change slowly with evolving cultures, and the stage settings change all the time. So it is that we see our twentieth-century selves mirrored in Shakespeare, Homer, and the Bible. So to the extent The MM-M is about people and teams, obsolescence should be slow."

Conclusion: the 1995 edition of The MM-M should be in everyone's library.

### 2.3 Peopleware: Productive Projects and Teams

This is another classic (DEMARCO 87). Everybody quotes it. That is all I have to say about it (except this: they don't even quote Weinberg. Nor do they quote "The Mythical Man-Month"! )

### 2.4 C++??, Critique of C++

This is an unusually excellent web publication (JOYNER 96), with a severe condemnation of C++, but not overly argumentative, and with many detailed comparisons and insights about a broad range of issues in contemporary software engineering. You may disagree with some points, but you will learn a lot! It is substantially more "technical" than the first three books, but it shows that many of the "technical" issues are in fact human issues. A minor but good example is a short discussion of the issue of case-sensitivity in UNIX: "Case distinction adds cognitive difficulty. Good language design takes into account such psychological considerations in these small but important details, being designed towards the ways humans work, not computers."

The main arguments discussed in the book are:

- C++ is too large and complex:

C++ "claims to be standardized, but this work is still very much in progress. The number of issues to be addressed by the [standardization] Committee keeps increasing rather than decreasing. ....I have already heard stories of C++ tool vendors complaining that the standard is too horrendous to understand."

- compatibility with C is BAD:

"Adoption of C++ does not suddenly transform C programmers into object-oriented programmers. A complete change of thinking is required, and C++ actually makes this difficult."

- High level is more important then Object Orientation:

" The ability to remove the bookkeeping burden from the programmer in order to enhance the speed of development, maintainability and flexibility" is more important than object-orientation itself. C++ more than cancels the benefits of OO by requiring programmers to perform much of the bookkeeping."

Joyner's general C++ conclusions are severe (but written in rather beautiful language):

"The advantages of the OO paradigm are so effectively undermined in C++ as to be worse than useless. .... Today's C++ programs will be tomorrow's unmaintainable legacy code. The seeds of software disasters for decades to come have already been planted and well fertilized. "

## 2.5 Issues specific to High Energy Physics

This section addresses HEP-specific issues, but to the extent to which HEP represents a collaborative, international effort of large teams at the frontier of technology, many arguments should have broader validity.

Here is my summary of a typical HEP experiment (ATLAS/CMS/)

1600 physicists from the 5 continents will have spent 15 years designing and building a $500 10**6 detector, then running it for 10 years, getting 10**15 bytes of data each year out of 10**9 interactions per second. Online and offline processing at about 10**7 MIPS each.

As for the HEP computing environment, we are in a transition period between:

- recent past: disorganized and dispersed group of amateurs coding large programs in FORTRAN
- future (for LHC experiments): disorganized and dispersed group of amateurs coding large programs, dealing with obscenely large amounts of data, as well as dealing with new, often esoteric languages, concepts and methods, and at the same time with the inertia of the FORTRAN generation(s)

Please note that the word "amateur" is not pejorative (it comes from "amare"). In this case, the amateur is a physicist. This is both a plus (most are bright, dedicated) and a minus (most are undisciplined, and are not full-time programmers). This is one of the two main differences between the HEP computing environment, and that of other large software efforts (at Microsoft, AT&T, Boeing Computer Services, ...), where, at least in principle, the programmers, and managers, are professionals. (The other difference is the management structure discussed below).

### *2.5.1 New problems inherent in the new paradigm*

The principal change which is well underway is the extreme increase in abstraction involved in modern software engineering: the concepts of Object Orientation (encapsulation, polymorphism, genericity, templates, multiple inheritance, virtual classes, ...) are considerably more difficult to learn (and to teach) then the concepts of do-loops and go-to statements of the past. One has to recognize and evaluate the hype and propaganda. From a recent LHC-experiment software document I quote: "Object Orientation allows huge software systems to be built easily and safely and allows them to be changed readily." This is, of course, strictly speaking, complete nonsense: the words "easily", "safely" and "readily" will never be applicable to huge software systems. For another example of hype,

consider that while a JavaSoft document describes JAVA as being "extremely simple to program", the JAVA Tutorial has 800 pages!

A significant part of the difficulty is connected with the extreme increase in verbosity, and in natural language abstraction, used in defining, describing and discussing the new concepts. When this is combined with the fact that the English language is often mishandled (by non-native and native English speakers alike), the result may be a sentence which requires some time to decipher. Some random examples of the Newspeak:

> "A class in and of itself is not an object"
>
> "There is a class named Class, instances of which contain run-time definitions."
>
> "In this approach, a base class called NamingMsgType is defined and a subclass of NamingMsgType is created for each type of NamingMsgType."

Sometimes the desire for precise formulation leads in fact to an error. Consider the following example from a HEP computing paper:

> "We adopt the Liskov Substitution Principle for proper arrangement of classes in inheritance hierarchies:
> If for each object o1 of type S there is another object o2 of type T such that for all programs P defined in terms of T, the behavior of P is unchanged when o1 is substituted for o2, then S is a subtype of T.
> In other terms, derived classes must be usable through the base class interface without the need for the user to know the difference."

There is a logical error in the above (for explanation see the footnote[i] - but please do spend a couple of minutes trying to find the error first), and it illustrates the qualitatively new degree of difficulty associated with the new computing paradigm.

This increased abstraction and difficulty in fact represents a "paradigm shift" as introduced by Kuhn in his book "Structure of Scientific Revolutions". Programming used to be easy. Any bright kid, without any formal education in Computer Science (or in anything else), was able to make a major contribution (and we all know famous examples of kids who did just that, in the garages of their parents.) Now programming is becoming very difficult, with large demands on programmer's capability of abstraction. Above all, programmers must be flexible: willing and able to learn new things continuously, even long after getting their degrees ... It is now much more difficult to be a non-professional, part-time (i.e. amateur) programmer. I should stress that all the above is not meant as a criticism of the software engineering developments - with the new tools and within the new paradigm, we can hope to accomplish projects that would have been inconceivable in the past.

### *2.5.2 Management, funding and other political problems:*

This is the second area in which the HEP environment differs from that of other large software projects. In HEP, the hierarchical structure is very weak. The absence of central funding results into an absence of central authority, and it leads to design by committee and decisions by consensus, in direct violation of the recommendations of Brooks and others about the significance of design integrity. This is also true for collaborations in hardware design and production, but it is a well known fact that while hardware collaborations in HEP are mostly successful, software collaborations are often very difficult. I suggest that hardware specifications are much easier to produce and enforce (and understand, even for non-experts, and for bosses!).

---

[i] The formal definition (the second paragraph of the quote) has the structure "if A then B". The third paragraph, supposedly explaining the formal definition in English, says "if B then A."

As to the support from Funding Agencies and Other High Places, I will limit myself to an anecdote. During a recent Site Visit by a Cognizant Officer of my Funding Agency, I extolled the potential of the WEB to improve communications and documentation, to save on travel expenses, etc. etc., only to be told (and I quote): "Real Men don't do Web". So I switched my presentation to describing my projects to develop Virtual Reality tools to simulate complex equipment, only to be told: "If you keep doing Virtual Reality, you will get virtual money." This is not a complaint: as you can imagine, funding agencies must see to it that their subjects are doing useful work, and it is not always easy ....

As for "pure politics", I must mention the frequent, and regrettable, association of knowledge and information with control and power. This makes local/subsystem managers reluctant to make promptly available all information they have, under all kinds of pretenses and excuses. This is especially evident in the documentation and in the database issues.

Regarding the "Software Process" : appointing people to roles of Chief Architect, Domain Architect, Abstractionist, Chief Tester, all the way down to just plain programmer ....is recommended by The MM-M and others, and it may or may not work for Boeing or Microsoft. It will be very difficult but crucial to do this properly in HEP experiments! I predict that, basically, experiments where this will be done right will have well-designed and well-performing software; the other experiments will have terrible headaches.

#### *2.5.3 Communication and Documentation:*

In spite of all its considerable accomplishments (see these Proceedings), I claim that the WWW has not yet delivered on its promise: it does not (yet) replace many meetings. The problem is not the lack of a particular technology, but human factors: people still don't believe and/or practice the advance distribution of reasoned presentations. One of the reasons for this is the well known "it is a waste of time to document" phenomenon. Another, somewhat less excusable reason is the "surprise attack" phenomenon: if "winning" at a meeting is the goal, then it is seen as counterproductive to distribute the presentation in advance of the meeting .... In any case, I claim that, in many cases, teleconferencing is really a step backwards. Even real meetings, not to mention teleconferencing, are often a waste of time and money, compared to an exchange of carefully written arguments and reasoning. I believe that more emphasis should be directed towards replacing many meetings, real or virtual, with information exchange.

## 3 HUMAN ASPECTS OF COMPUTING IN 21ST CENTURY

As in the first Lecture, I will present much of my material in the form of a review of three books and one article, which represent a wide-ranging and serious critique of the role of technology in modern society. You will see that I disagree with many of the arguments I review, yet I think it is very important for you to know about them. First, it is necessary to be aware of unjustified anti-science and anti-technology arguments, to better be able to defend what we do. Even more importantly, it is crucial that we absorb and react to the parts of the criticism that are not unjustified.

### 3.1 Silicon Snake Oil

(STOLL 95, by Clifford Stoll who earlier tracked down hackers selling information to KGB, and wrote a book about it. He is therefore computer-literate, which makes what follows even more strange ...)

This is 240 pages of an incessant, rambling attack on the WWW, the Internet, computers in schools, even on EMAIL ... on the whole gamut of present day computing and networking. Every now and then, Stoll comes up with a valid point (e.g. hidden costs of the Internet; lack of structure, reviews, editing, recommendations on the WWW, etc.), but most of his complaints are those of an addict, or are addressed to an addict (“Birds don’t sing on the Internet.” or "During that week you spend online, you could have planted a tomato garden, volunteered at a hospital, spoken with your child's teacher, ..."). Nostalgic denials of progress are often supported by quotes from well known thinkers of the past:

" 'Our inventions are wont to be pretty toys, which distract our attention from serious things. They are but unimproved means to an unimproved end.' Thoreau writes in Walden. 'We are in great haste to construct a magnetic telegraph from Maine to Texas; but Maine and Texas, it may be, have nothing important to communicate.' "

To this, Stoll adds: "That magnetic telegraph has evolved into Internet."

Most curiously, Stoll prefers the paper card catalogues in libraries, and he hates Internet searches. Presumably, this includes the Science Citation Index, which enables you to look up all subsequently published papers which cited the one you are interested in (and what they had to say about it ...). I find it hard to overestimate the importance of this subject. The contribution of distributed hypertext may end up ranking with the previous improvements in our ability to learn from the cumulative accomplishments (and mistakes) of the previous generations (the invention of writing, or printing .... ). The form of the Citation Search as pioneered *e.g.* by the Los Alamos preprint server (where not only the citations, but the Abstracts, and the papers themselves, are easily available) provides a preview of the future: removal of the unnecessary difficulties in finding out what is known on any question and issue, large or small.

But this is perhaps not surprising. Stoll recalls being humbled, with his computer, by Feng Li doing a Fourier transform on twelve abacuses. Stoll's results were incorrect because of "computer's [sic] assumptions" about the data which were not justified!

Conclusion: the Silicon Snake Oil is a famous book, for all the wrong reasons, and an important book not to be misled by.

### 3.2 The Future Does Not Compute

(TALBOTT 95; published by O'Reilly who says "When a leading publisher of computer and Internet books also publishes a book questioning the role of computers in our society, it just might make people think.")

This is another famous book, more voluminous, and a class above Snake Oil in reasonableness, thoughtfulness and language (“ ... we choose to inflict the computer upon millions of schoolchildren who have not asked for this reductive assault upon their higher capacities ...” - isn't this a well-turned sentence?). Talbott's essential points and worries lead to those of Technopoly and V. Havel:

> “It is not that society and culture are managing to assimilate technology. Rather, technology is swallowing culture.”

And incidentally: he is not impressed by modern physics:

> “... today the physicist plays in the realm of number, equation, and probability, disavowing all attempts to assign meaning to his constructs.”

If you have the time, you can learn quite a bit from this book. Otherwise, go for the much leaner Technopoly, or leaner yet Havel.

### 3.3 Technopoly

(POSTMAN 92. Neil Postman is the chair of the Department of Communication Arts and Sciences at New York University, and a winner of the 1987 George Orwell Award for Clarity in Language.)

The book opens with a story from Plato's Phaedrus, and goes beyond computing and the Internet. Postman defines technopoly as “submission of all forms of cultural life to sovereignty of technique and technology. ”USA is the first Technopoly"; some other countries are getting there. Then

"scientism"[ii] is described as a "pillar of technopoly"; a surprising but uncontestable example of an advocate of scientism is Ronald Reagan!

But before we are done, Mr. Postman provides us with this bit of food for thought (p. 62):

> "In 1480, before the information explosion, there were thirty-four schools in all of England. By 1660, there were 444, one school for every twelve square miles."

Can you tell what is wrong with this, and why I find it interesting?[iii]

Conclusion: for its discussion of scientism alone, this book is well worth reading.

### 3.4 The End of Modern Era

There are now a number of additional books objecting to an uncritical attitude towards science and technology. It so happens that the least exaggerated, most coherent, and at the same time the most compact, statement of the "post-modern anti-scientism" was given by Vaclav Havel, in his speech at the 1992 World Economic Forum (HAVEL 92). And he even managed to mix in the Fall of Communism:

> "The modern era has been dominated by the culminating belief that the world is a wholly knowable system governed by a finite number of universal laws that man can grasp and rationally direct for his own benefits. … This, in turn, gave rise to the proud belief that man, as the pinnacle of everything that exists, was capable of objectively describing, explaining and controlling everything that exists, and of possessing the one and only truth about the world.. Communism was the perverse extreme of this trend ... /and its/ fall can be regarded as a sign that modern thought ... has come to a final crisis. The end of Communism is a serious warning to all mankind. It is a signal that the era of arrogant, absolutist reason is drawing to a close ..."

And since this was the most coherent and least exaggerated[iv] account, it caused a fairly large upheaval on the pages of the American Journal of Physics (Sept. 1992, p. 779) and Physics Today (Aug. 1993, p. 11). A number of replies and comments on this can be found in the issue of Physics Today of June 1994, including a Letter from one V. Chaloupka, who manages to mix the Theories of Everything into all this (see CHALOUPKA 93 for copies of some of these arguments.)

What is the driving force behind the post-modern negative attitude to science and technology? The critics are certainly not simply neo-Luddites (in 1811 - 1816, Luddites revolted in England against the machines, mainly because of the loss of jobs), although you can find, in the books reviewed above, a fairly negative evaluation of just about every technological progress (from the invention of writing and printing, to the building of the US Interstate highway system.)

Rather, the main factor seems to me that the post-modernists regret the loss of "meaning" and "security" of the past, not realizing that such meaning and security were false and illusory. Quoting Postman for the last time:

---

[ii] Scientism is an uncritical attitude towards science, believing that science can and should be used for purposes which in fact are beyond the competence or the range of scientific methods. The example given by Postman is excellent: Ronald Reagan said that he personally opposes abortion, but that it is up to Science to tell us when does life begin.

[iii] [iii] Mr. Postman confuses "12 square miles" with "12 miles" squared. Everyone can make a mistake like that, but there may be a symbolic meaning in the fact that the result did not strike him as obviously incorrect.

[iv] I should point out that the three books reviewed above do not represent, by far, the most exaggerated anti-science and anti-technology positions. The virulence and the intensity of some papers and publications is remarkable - see e.g. the collection "Resisting the Virtual Life" (J.Brook and I.A.Boal, eds., City Lights 1995)

> "What Havel is saying is that it is not enough for his nation to liberate itself from one flawed theory; it is necessary to find another, and he worries that Technopoly provides no answer."

I am not sure Havel would agree with this interpretation of his views[v], but I think it is exactly to the point, and it gives us the best cue as to what we should do to counter the post-modernist charges. We should make it clear that science does not, not at all, aspire to solve human spiritual and moral problems, and even for problems where it does apply, it represents only the tools, not the solution. In particular, all talk about the "Theory of Everything", "Knowing the Mind of God" or "Reading His Thoughts" should be avoided, as it does disservice both to Science and to Religion, and it is a clear example of the "scientism" as discussed above. This is not the time nor the place for a detailed discussion of this issue, so I will give just one example. In his "Brief History of Time", Stephen Hawking writes:

> "If we do discover a complete theory of the universe, it should in time be understood in broad principle by everyone, not just a few scientists. Then we, philosophers, scientists, and just ordinary people will be able to take part in the discovery of why it is that we and the universe exist. If we find the answer to that, it would be the ultimate triumph of human reason: for then we would know the mind of God."

Assume, just for the sake of the argument, that the "M-theory" will be actually worked out, and that it will turn out to represent "the superunification of the forces of nature." Will you then point out to your barber or to your priest that "in uncompactified ten-dimensional Minkowski space, the strong-coupling limit of the type I superstring is the weakly coupled heterotic SO(32) superstring ..." ? (the quotes are from the remarkable article by WITTEN 97.) In general, even if we should become able to predict the probabilities for any result of any experiment, how will this help Stephen Hawkings or anybody else to understand "where we came from and why are we here"?

There is much more that could be said about this, but we have to leave it for another day. Perhaps I should just mention that some people seem to think that a little dramatization of Physics is helpful to our funding prospects. I believe that the implied arrogance and maybe even blasphemy of the "Mind of God" claims is actually hurting us.

Coming back to computing and information processing, I do have some concerns:

- The WWW does need structure. It is not clear to what extent the present anarchy is a result of an original design flaw, and it is not clear how the necessary structure should be achieved. My expectation is that the WWW will actually be replaced by something structured; not just modified.
- I am very concerned about the increasing vulnerability of society due to the dependence on increasingly complicated software. Even now, some quite simple tasks turn out to be major problems (take e.g. the "year 2000 phenomenon". US Congressman S. Horn, who oversees the information-technology issues, says: "The problem is that we do not know which programs will fail, what problems their failure will create, and disastrous will be the consequencies.") Just imagine the vulnerability of society to large scale natural or societal upheavals in a few years, when computerization will reach everyone's home on a scale similar to the one described by GATES 96.
- The optimistic advocates of progress-at-full-speed claim that computers represent an equalizer in society. I am not so sure. The extreme complexity of modern computing could result in exactly the opposite effect. The reason is very fundamental. The complexity and dynamics of modern computing and information processing require, more than the acquisition of some body of knowledge, the ability to learn continuously. Only a small fraction of people may be capable of

---

[v] Is it necessary to find another FLAWED theory? Perhaps searching for a meaning is preferable to an illusion of having found one.

this, and they will become the new elite - privileged by their brains ... (of course, smart people are always likely to be better off than the not-so-smart people - I am worried about the extremes of such inequities...)

- The most significant concern is the undeniable fact that computers (as well as other modern technology - just recall nuclear energy and nuclear weapons) have the potential and the tendency to magnify human follies. And it is here that the combination of our expertise in Physics and in Computing gives us special duties and responsibilities. We all should be spending at least a small fraction of our time explaining to society what it is that we do, and what are not only the potential benefits, but especially what are the possible dangers.

However, for most of our time, we should just keep on doing what we do. Quoting from that remarkable MM-M book once more:

"To only a fraction of the human race does God give the privilege of earning one's bread doing what one would have gladly pursued free, for passion."

I think a fair number of us belong to that fortunate fraction, and we should rejoice in that.

And, quoting from MM-M for one last time:

> "The tar pit of software engineering will continue to be sticky for a long time to come. One can expect the human race to continue attempting systems just within or just beyond our reach; and software systems are perhaps the most intricate of man's handiwork."

To which I would add (see CHALOUPKA 97):

Perhaps only the Art of Fugue is more intricate.

**REFERENCES**


BROOKS 95: F. Brooks, Jr., "The Mythical Man-Month", Addison-Wesley 1995 (20th anniversary edition)

CHALOUPKA 89: Variations on GEBH, Proceedings of the BNL Workshop on Hybrid and Exotic Mesons, also available from my WEB page (see below).

CHALOUPKA 93: see my WWW page (http://www.phys.washington.edu/~vladi)

CHALOUPKA 94; V. Chaloupka, Letter to Physics Today, June 1994, p. 95. Also available on WWW (see above).

CHALOUPKA 97: V. Chaloupka, "Art of the Fugue" (this Proceedings).

COLEMAN 95: D. Coleman and R. Khanna (eds.), "Groupware: Technology and Applications", Prentice Hall 95

DEMARCO 87: T.DeMarco and T. Lister, "Peopleware: Productive Projects and Teams", Dorset House 1987

GATES 96: Bill Gates, "The Road Ahead", Penguin Books 1996

GREIF 88: I. Greif, "Computer-Supported Collaborative Work: A Book of Readings", Morgan Kaufmann Publishers 1988. For recent results, see also Proceedings of the CSCW Meetings (e.g. CSCW '96, M.S.Ackerman, ed.)

HAVEL 92: V. Havel, "The End of the Modern Era", NY Times, 1 March 1992. See also Havel's speech on ""The Need for Transcendence in the Postmodern World", delivered in the Independence Hall, Philadelphia, on July 4, 1994. Both documents are available from my WWW page - see CHALOUPKA 93.

HOC 90: J.-M. Hoc et al, "Psychology of Programming", Academic Press 1990

JOYNER 96: Ian Joyner: C++?? - A Critique of C++ and Programming and Language Trends of the 1990s; a WEB publication. Search Alta Vista for +joyner +critique, to find the best site.

KLEPPNER 93 D.Kleppner, "Thoughts on Being Bad", Physics Today, August 1993, p. 9

LUBICH 95: H.P.Lubich, "Towards a CSCW Framework for Scientific Cooperation in Europe", Springer-Verlag 95

POSTMAN 92: N. Postman, "Technopoly: The Surrender of Culture to Technology", Vintage Books 1993

SHNEIDERMAN 80: B.Shneiderman, "Software Psychology: Human Factors in Computer and Information Systems", Winthrop Pub. 1980

STOLL 95: C. Stoll, "Silicon Snake Oil: Second Thoughts of the Information Superhighway", Doubleday 1995

TALBOTT 95: S.L.Talbott, "The Future Does Not Compute: Transcending the Machines in Our Midst", O'Reilly 1995

WEINBERG 71: G. Weinberg, "The Psychology of Computer Programming", Van Nostrand Reinhold 1971

WITTEN 97: E.Witten, "Duality, Spacetime and Quantum Mechanics", Physics Today, May 1997, p. 28

YOURDON 97: E. Yourdon, "The Rise & Resurrection of the American Programmer", Yourdon Press 1997. (Yourdon's previous book was the "Decline and Fall of the American Programmer" - I wonder what will come next.)