COMPUTERS IN THE COMING SOCIETY

R.W. Hamming

Rapporteur: Mr. N.F. Hall

Abstract

It is generally agreed that computers will play a dominant role in our coming society. However, the few detailed predictions made in print seem to concentrate on the "horror stories". This talk examines various kinds of applications, and the more sociological aspects of the wide spread use of computers.

The price of computer components is falling very rapidly and is expected to continue to fall. Even though some applications of computers seem a luxury today this will help make them commonplace in a few years time. This talk begins with a personal view of software; what it was, what it is, and what I think that it is going to be.

There have been a number of trends in software through the years. First there is the trend to remove from the user the necessity, or even the possibility, of referring to specific parts of the computer. In the beginning programming was done by writing the binary names of the specific storage registers as well as the binary patterns for each specific instruction. The use of symbolic names for the operations and the addresses means that the user no longer refers to the specific register, and debugging software is needed to help find troubles in programs.

Soon there came to be monitor systems to sequence the problems, plus some special debugging features, then tape assignment programs with symbolic tape positions. Next the software systems controlled the actual storage on the tapes, and other peripheral units, so that the information was put into blocks suitable for the machine efficiency rather than in the form specified by the user.

We can expect to continue this trend away from referring to specific units until there is almost nothing definite the user can say about the particular machine that happens to solve his problem. When this trend is completed, it should be obvious that we will then have <u>machine independent coding</u>, since then nothing can depend on the particular machine. But we are a long way from that day for many types of problems.

Another trend is that, when first written, a software package is usually about five times as large as it will be some years later. This effect is partly due to greater understanding of what is to be done, partly due to progress in computer science, and partly due to the fact that when some software uses a lot of machine time then this function is likely to end up as part of the hardware of the next generation of machines.

Still another trend is that with each generation the total amount of software offered has been at least ten times larger than for the previous generation. At the beginning of the fourth generation the manufacturer would typically, for a large machine, deliver somewhere around a few million words of software, for the third generation perhaps one hundred thousand, for the second a few thousand at most, while the first generation had at most a few hundred words of software. Of course the local installation in the early days added a large percentage increase in software, and later generations relatively less, since only a few places have the manpower to write as much as a big manufacturer can do if he tries.

For many years it was a standard rule that one thousand or so lines of debugged code required a man year of work, regardless of the language used. Lately there have been significant improvements in 'debugged lines per man year', but it is still surprisingly low when the software is near the frontier of knowledge. Of course, what an expert did in the second or third generation can now be readily done by a bright college student, but software at the frontier is still (1975) laboriously hand produced by cottage craftsmen techniques and with all too many bugs left in it.

Based on these remarks, we can hope in the future for very large amounts of software, the equivalent of perhaps the Encyclopedia Britannica, with only some of it used by everyone, and much of it specialised for smaller groups of users. We can hope that it will be better written, both in size and in freedom from bugs. The work on 'proving programs will work' no doubt will help, but like proving theorems on computers it is likely to take a very long time before significant results emerge.

The presence of bugs in the software is such a sore point with the users that some types of certification, and possibly penalty clauses in contracts may come into practice, but in 1975 it is hard to find a software producing installation that will offer a reasonably good contract for error refunds.

There is currently (1975) a trend for software houses to produce packages that are sold or leased, rather than in the past depending on the manufacturer or friends for what one does not write oneself. Supposedly this trend will continue, and the 'unbundling' that was court decreed may come into effect someday, but many manufacturers still feel that the success or failure of their product line depends too vitally on the software to leave this completely to others.

It is interesting to speculate on how far we may go without keeping hordes of programmers around to write out all the details of the software system.

For analogy, we are close to the point where we decide the general structures of a proposed circuit; computer programs break the plan down into the details of gates, "and's", "or's", "nand's", etc., other programs lay these out on a planar surface; still further programs draw these circuits which are then photographed, reduced in size and used in the manufacture of integrated circuit chips. In a sense no human mind has had to think about these minute details which go to make up a computer.

It seems that in time we will be able to do even better in the area of software than we have done in the area of hardware, and do so in the near future. 'Compiler-compilers' and other software generating tools should solve the problem of creating the volume of software required.

Software is made, and sometimes used, by people who are loosely labelled 'programmers'. As noted their work has not been satisfactory on the average as judged by the complaints one hears at all levels. Recently there has been a rising tide of effort to change this situation. It is difficult to describe 'programming' with any precision. At the bottom end is the pushing of buttons on a telephone set that in turn calls in elaborate programs in a central office. Again the airlines reservation clerk who uses a terminal much of the time is not a programmer, nor is the person who uses a computer for typesetting. On the other hand, the person who submits what is usually called a program to the compiler and operating system (which sees this program as just a collection of parameters to be operated upon) may be a programmer, but there seems to be no logically clean distinction.

Even if we can neither define programming nor a programmer, it is possible to suggest a lower bound. If the special training to use the computer does not exceed a one-term college course then that person can hardly be a 'professional' programmer. Using this approach we see an increasing trend towards experts in other fields using computers, and a definite decrease in the number of programmers who are expert in computing and nothing else.

Let me now turn to possible applications. Both the hardware and the software exist to be used, and it is the applications in the long run that justify all the expense and effort. In the earliest days there was a heavy dominance on "number crunching problems". It was not that the early computer experts did not know better. Indeed, surprisingly early there were programs for analytically differentiating a function, proposals and attempts to translate from one natural language to another, as well as such items in artificial intelligence as playing chess (more or less well). The reason that so many of the problems were number crunchers was that the people with the money to pay wanted them done. As the price of computing per operation came down, the range of what was economically sound widened to include more of the non-number crunching problems. Accounting problems were, of course, done from the first, and fall close enough to the number crunching to be included in the title. Of course old applications in engineering design, weather prediction, airline reservations, computer controlled factory tools will continue to improve and proliferate; we are mainly concerned with new applications.

The big failure of machine translation from one natural language to another points up the fact that so many of the things that seem easy to do turn out to fail, not because of machine limitations, but rather from our own human limitations; we often do not understand the nature of the problem we propose to solve!

This brings up an important point about what to expect in the field of artificial intelligence, which tries to solve problems by using methods which do not ensure a perfect solution. One of the great errors repeatedly made in arithmetic problems, in accounting, and in other applications, is the slavish copying of how the problem was originally done by humans. Such direct copying is natural to try, but usually ends in disaster. Just as in mass production, the hand produced product must be modified (screws need to be replaced by rivets or welds, for example) if it is to be efficiently done.

This same effect is present in almost all aspects of engineering. Airplanes do not flap their wings as birds do, trains do not run on legs, prime mover power supplies do not depend on muscle expansion and contraction. We are heavily dependent on rotating devices, wheels, gears, etc., but nature seems never to have invented the wheel. Time and again, when engineers do something equivalent to what nature does, the means are quite different while the ends may be reasonably close.

Thus, while the study of how humans process information is, and will continue to be, an important field of computing, probably its main value will be the extent to which it illuminates our understanding of ourselves, and relatively less for the efficient use of computers.

There is a tendency to view computers as part of classical science, an extension of its possibilities to control our external worlds. There is much less understanding that in the long run computers will be more important as they reveal and supplement our internal worlds. Thus tracing past trends in applications for purposes of extrapolation is valid only if the past is properly categorised. For the extrapolation to be valid, it must be recognised how much, in the past, computers have invaded the humanities and the psychology of humans, as well as how they have changed science itself.

Computers have already (1975) greatly transformed science. Not only are they in constant use to extend the range of experiments, they are also the tool for entirely new approaches to old questions. For example, instead of measuring directly what is wanted, (say a frequency response of some device) some more easily measured item is chosen (say the impulse response) and the computer is used to transform the data (say via a fast Fourier transform) to the desired form.

In the study of vision computers have become the main laboratory tool for how humans perceive 'depth'. Many other examples may be given of how the computer has become the centre of the laboratory rather than a piece of peripheral equipment.

It is little appreciated how far computers have already transformed science and society, let alone how far they will ultimately go; already simulation has begun to replace invention. For purposes of analogy consider agriculture. At the time of the American Revolution (1776) approximately 95% of the American people were involved in agriculture. Had you then tried to explain that in 200 years less than 4% of the population would be classed as farmers and yet the whole populace would be eating better, you would probably not have been believed, nor could you have given a convincing, detailed description of how this could happen. America has already passed the point where less than half of the workers are engaged in any form of direct manufacture. With the aid of computers manufacturing will absorb probably less than 10% of the workers in the year 2000. Thus over 90% of the workers will be in service and information processing. And if this is true of society as a whole, how much more true will it be in a large research laboratory; almost all of the people employed in scientific establishments will <u>not</u> be directly concerned with the actual physical world. Impossible? Yet it is highly probable that computing machines will bring this about. Western civilization will be an information oriented, rather than a material oriented, society.

In the humanities computers almost immediately put a great deal of pressure on languages, since from the earliest days the more thoughtful computer language creators asked the classical scholars what they could say about the engineering efficiencies of languages. After a few completely blank replies the students of languages began to take a rather different turn, and to ask questions such as "How are sentences created by humans, let alone recognised as sentences?" (Classical parsing was soon shown to be a jumble of misconceptions.) "What are the efficiencies, as judged by the evolutionary standards of 'survival of the fittest', of various features of languages?" "How should the redundancy of spoken languages, and its difference from the redundancy of written language, be modified when half the dialogue is a human and the other half a machine?" "What is the value of <u>irregularity</u> that we find in almost all living languages?"

Machines were also used to attack vexing problems of authorship of various disputed writings, and in a statistical sense settled many of them. This, of course, begins to enter into the field of style, and will no doubt in the era of fifth and sixth generation computers result in formulae that indicate the main elements of a 'romantic style', a 'Gothic style', 'realism', etc.

In the field of creativity machines have been used, for example, to attack the problems of composing music, and to a great extent we have found out that we do not yet understand clearly enough how to get machines to compose good music. As yet we understand only the simplest aspects of "creativity". However, by simulating various musical instruments, by creating new sounds, and by making precisely controlled sounds, the machine offers the musical composer the gamut of possible sounds, whether or not any particular musical instrument can make them, and has given the conductor almost absolute control over the final production. Thus we can expect that machines will slowly make more and more inroads into music.

Indeed, with mini computers widely available, it is quite possible that many people will have such machines creating background music in a style they like, perhaps imitating seashore sounds during the night, springtime sounds at dawn, and during the day background music, all used to cover the unwanted noises that arise in a crowded city life. It is not hard to imagine programs that are adjusted, both in the musical composition and instruments imitated, to the individual's general tastes, though it will probably be decades until foreground music that is widely satisfactory can be so produced. Commercially sold programs will be like attachments to the older machines, the consumer buys the polished design and simply 'plugs it in'.

Similarly in the field of vision, large wall screens in rooms driven by small computers may be widely used as interior decoration features. The above mentioned seashore and spring sounds could be paralleled by appropriate pictures, which the computer can vary enough to make continually interesting as a background to living.

Game playing by machines is already a proven thing, and the range of games that humans can play with a machine is far greater than those which humans will play with each other. Thus a game with a machine where part of the problem is to find out what the rules are might well prove popular with many puzzle solvers. It is widely observable that for many people computers are as 'addictive' as are drugs.

These, and many more applications of computers for the pleasure of the individual will come in fairly soon after they are economically sound. With the coming of the fifth and sixth generation machines the costs will be so low that it is only a matter of time and social acceptance until these and many more applications of computers are routine.

In speculating about applications of computers, one should be acutely aware of the problems of legality and legal responsibility. Thus both computer medical diagnosing, and computer controlled highways are likely to be severely delayed by the legal problem of who is responsible when apparent failures occur. The legal and social framework of society will undoubtedly lag far behind the theoretical ability of computers to supply services in many areas, while in others it will lead to rapid social changes in unsuspected ways, much as television seems to change the family structure. The ultimate role of computers in the area of sex is inconceivable.

Already we have seen people who prefer, apparently, to interact with machines rather than with other humans. Sometimes this is a net gain. Thus computer controlled traffic lights are often easier for the individual automobile driver to cooperate with than is a live human directing traffic. Most of us have long ago accepted machine control via a stop light, and even when it is clearly idiotic most of us will wait for the traffic light to change. It is not a question of 'Will man accept control by machines?' it is a question of degree of control in various areas of activity. In these areas the science of human behaviour is so little developed that one can rarely avoid personal prejudices. But in any case it is fairly safe to say computers will dominate both society and our individual lives before the year 2000 (supposing, of course, that society survives and evolves much as it has in the past).

Computers are the tool for the mass production of a variable product, and thus should not be identified with a stereotyped picture of mass repetition, but rather with providing the means for individual choice over a wide range of applications.

The details of where computers will be used have largely been ignored. Inside a car there will be a computer to control the spark ratio of the petrol and to equalise the brakes. In medicine many more applications are expected. With the pacemaker we have a computer keeping people alive. (This is an example of the computer actually controlling the person.)

Small computer chips will be found everywhere. They will differ mainly in the microprogram in their read-only memories. The reason for this is cost. It is not economic to make a wide variety of chips, but it is economic to make one pattern by the million.

There are a number of applications which one should be wary of believing in. An example is the personally kept data base. People are very careless about keeping them up to date. The trouble when you are filling in your tax return is finding out what actually happened during the previous year, and finding out whether this or that rule applies; it is not how to add up the numbers. The limitations of the personal data base are determined by the effort an individual is willing to put into keeping it up to date.

There are dangers, too, in using somebody else's data base. Let us consider a data base of scientific constants, and suppose it contains the velocity of light among others. Before any scientist is going to use this constant he is going to want to know whose velocity of light. If in the future an experiment is performed that gives a slightly different value for this constant, who decides whether the value in the data base is changed? The user of the data base is not going to be pleased if the value of this constant changes from day to day and this in turn effects his results. He is faced with the problem of deciding whether the change in his results is due to a change that he has made in the experiment, a change made in the data base, or both.

When a computer is introduced into any organisation it changes the power structure a great deal. It is well known that the informal channels of communication within a large organisation are much more important than the formal ones. It is the nature of bureaucracy to be static despite the fact that we live in a dynamic world. Programs and computers reinforce that static nature. I have faith in the ability of a bureaucracy to survive!

No organisation operates exclusively by its written rules. Take the example of the union which instead of calling a strike decides to 'work to rule'. As a second example, one is not supposed to take items from the company stock room for one's personal use, yet a man who rose to become executive vice-president of Bell Laboratories once advised me: 'Go take that thing. Don't waste your time. It's too expensive for you to fiddle around getting that cheap thing!' So there is a range, an ill-defined, ill-understood level to which you take, for your personal use, items from the company stock room. We do not have fixed rules.

In our society's legal system we have remarkable flexibility. We have a forgiving society. We do not give a fixed punishment for breaking the rules. While raising children one gives them rules but enforces them selectively. One uses 'common sense'. It is my personal view that a society which does not have this flexibility and forgiving is not a pleasant society in which to live.

We do not work 'Yes-No', but machines, as they are used at present, do. If we continue to use machines in this way we will lose many of the features that our society uses, the forgiving, the 'not working by the rule-book' and the ability to change the rules with time. If we were static we could not evolve. We have a constantly changing understanding of what we may and may not do to bend the rules.

Therefore in future we will have to understand how to program in this variability, this softness of response, rather than the harsh yes-no as at present. Otherwise the society we create will not be a satisfactory one in which to live.

Returning to the subject of data bases, one may go back 2000 years and quote from the bible 'what is truth?' Should everybody be able to examine their personal data base and change anything with which they disagree? Who decides whether or not the facts are correct? The idea that we may have data bases in which all the facts are correct conflicts with our experience. After 2000 years we still do not know how to determine truth.

When in the distant future historians look back it may well be that they will regard the computer as man's most significant discovery. This has been said before, but that does not make it the less true.

Humans seem to have a deep seated, almost natural, prejudice against machines. The evidence for this remark can be seen everywhere; for example when a computer is not the world's best chess player, but plays only serious tournament play - better than most human chess players of course, but not perfect chess - then people claim that the computer is not so great. Each failure of the computer is lovingly reported in the press; its successes go almost unmentioned.

Constantly we compare the computer against perfection as we imagine it to be; and not against even average performance such as we typically have in our society. As a result we get a distorted report and deceive ourselves repeatedly. We should not make the comparison against perfection but against current practice to see if the new systems are an improvement over human behaviour.

Take again the example of data bases; I almost never hear a discussion of error rates in current data banks (even using computers) nor a discussion of the errors in judgements we are now making in the absence of adequate data banks. Instead I hear rantings about the fact that humans, being humans, will make errors in the entries and not always be prompt to correct errors. It should be obvious that the presence or absence of a data bank will, to a first order, merely result in a changing of who has the jobs, and not in how many jobs there are to be filled. And if a person now has a job that he would not have with more complete record keeping, then it follows, as the night does the day, that there is also now someone who would have the job were there adequate records. The invisible man is rarely discussed, the man who better deserves the job than the present incumbent. How seldom do I hear about the decrease in the error rate due to computers as against the hand methods still widely in use! Instead I hear mainly of the evils of having a data bank and not the evils of not having it. Of course nothing in the world is perfect and error free, we can be sure of nothing, justice is not infallible, and even the meaning of truth has its well known difficulties - how seldom even after it has happened do we know what occurred!

Therefore, if we are to discuss Computers and Society we must recognise this distaste for computers on the part of humans. Instinctively we do not want to be controlled by computers, yet we welcome the new stop and go light on the street corner, and the patient with the pacemaker attached to his heart puts his life in its hands willingly. Time and again we welcome the computer's output in the form of the details of the withholdings on our pay cheque, knowing full-well that, while the cheques are sometimes wrong, if they were done by hand there would be many more errors and a lot less details supplied to us. We complain and publicise the errors that do occur, but compare them with perfection and not hand practice.

Anyone who has watched the space shots on television has seen, if he had the wit, the central role of the computer and the peripheral role of the humans. If we think of what goes on from before testing for take-off up until the final return to earth, and how seldom humans get into the act, we see that in the symbiosis of man and machine the machine plays the larger role in space flights. How do I dare say that? Better, how do you dare to deny it? You have only to look at it through eyes that are not prejudiced in favour of the human and against the machines. Indeed, in the near disaster on a flight to the moon I saw the humans on board reduced to stooges for the computer supplying the input as best they could.

When we give up our all too human eyes and look at the situation dispassionately we see that the machine can do many things and venture into many places that the naked human is permanently barred from. And often they do the job more gracefully, not needing all the life support that man does, low accelerations, atmosphere, food, waste disposal, etc. How much less the machine requires in these matters! Yet at the same time let us not go overboard on the other side and make wild claims for the computer. There are limits on both sides of the argument of man versus computer and it is a real problem to maintain a reasonable balance between our fears and unbounded optimism when we are so intimately involved with our all too human pride.

Human culture is the difference between cave man and the present, each has different cultures and the other changes are slight indeed (so far as there is any reason to believe). Now we have to guess what man will be like in a computer culture based on a symbiosis of man and computer. First of all there will be less prejudice against the computer when more experience with it is available. Second, computers will do most of the monotonous jobs, but probably not education. It will still be slow and difficult to learn to write and express oneself well, and the good life will not necessarily be the easy life. Third, distributed brains will be the rule. In current examples of corporations there is seldom a single man in control in spite of all the titles and appearances to the contrary. Perhaps the typical university is an even better example; no one even knows who has the power to do many things, instead there comes gradually to be a consensus of opinion. Government is yet another example where the powers are not explicitly spelled out; in both the American and British governments, this is especially true.

Bureaucracy is still another example of an organisation where there is deliberately no one brain in control so that one person's whims cannot control the destiny of the many. Better a little inefficiency than the instability that can result from a poor choice of the head man.

Summary

A personal view of the future role of software has been given. The manufacture of hardware circuiting is becoming almost totally automated and this I believe will happen to an even greater extent in the field of software. That is not to say that one can automate 100% of the process but one can probably hold the error rate down in field issued software to one bug per million lines of code. Compiler-compilers exist which work in the sense that they produce reasonable code, and since the price of hardware is falling all the time, more reasonable efficiency will be regarded as satisfactory.

It is unlikely that programming is going to come under complete control. The art of programming, like the art of writing, is clear thought and clear expression. For how long have we tried to teach human beings this?

Thus programming will never be a trivial act. I have speculated on various areas of application and tried to point out both practical limitations and problem areas for society to watch.

Finally I believe I have shown the enormous problems of human acceptance of the man-machine symbiosis. Since sociology is so poorly developed, I am unable to make serious predictions in this area.

Discussion

Dr. Lavington asked 'In much of your talk especially concerning computers and leisure I was a little concerned that you kept using the word "will". I wonder if you could give your feelings on the two alternative phrases "are able to" and "ought to".'

Professor Hamming replied, 'You are asking me to make a judgement as to how people should live. I was telling you how I believe they will live. I have a distinct puritan ethic behind me, which seems to have a distaste for pleasure. But my feelings don't count when estimating what will probably happen.

Dr. Holt raised the subject of caring for land and how this should not be judged in purely monetary terms. He thought that the fact that 4% of the work force of the U.S.A. is now directly concerned with agriculture was not an achievement. Professor Hamming disagreed on the subject of the agricultural work force.

Professor Naur raised three points. First he wished to correct the impression given that the redundancy of natural language had not been investigated by Otto Jesperson at about the turn of the century. Professor Hamming said that knowledgeable linguists did not understand his questions about the engineering efficiency of natural language when he had asked them in the days of the IBM 701. The work may have been done, but it was not general knowledge amongst linguists.

Professor Naur's second question concerned the use of computers in physics. 'The problem concerns the reduction of the multi-observational data from the big machines of physics. Use is being made here of some very complicated programs and the same programs are being used by everybody. There is a risk that all productions from these vast data contain exactly the same mistakes.' Professor Hamming agreed completely.

Professor Naur's third point concerned the basic assumption of stable development and resource use that would be necessary for Professor Hamming's predictions. Professor Hamming replied that he had noted in his first talk that all predictions that he had made were based upon the assumption that we would not have tremendous revolutions and wars. He did not think that we would lack the material resources to manufacture machines, but could lack the organisation or the 'will'.

Professor Wells asked whether Professor Hamming expected that computers might be used to help remove the poverty gap between ourselves and the third world, or did he still think that the gap was going to continue to grow wider. Professor Hamming replied that he was incompetent to make that judgement.

Professor Dijkstra said that during the talk the efficiency of programmers had been measured in terms bugs per number of lines of code. He regarded the lines of code to be on the cost side and wondered what we should really use to measure programmer productivity. Professor Hamming replied that it was customary in the past to judge the cost of manufacturing software as 1000 lines of debugged code per man year. That we no longer do this is due in part to Professor Dijkstra. There did not seem to be any hard scale that could be used to measure productivity easily.

Professor Horning said 'I was a little worried about a section in your talk when you spoke of not comparing computers with perfection. You seemed to be primarily concerned with the number of errors. One of the unfortunate properties of most computer systems is that even though the errors tend to be less frequent they tend to be more catastrophic than with most manual systems. Surely it is not the frequency but the cost of the errors that one should use to judge the computer system. Professor Hamming agreed, but asked how he was to measure the cost. He believed, folk stories to the contrary, the machine errors in total were less costly than the earlier human errors that were comparable.