

# A Personal History of CMC

**Jacob Palme**

Dpt. of Computer and Systems Science  
Stockholm University and KTH Technical University  
Electrum 230  
164 40 Kista  
jpalme@dsv.su.se  
SWEDEN

Published in *Creative Crossroads – Electronic Honorary Publication*  
*Dedicated to Yvonne Wærn on Her Retirement, april 2003.*

## ABSTRACT

Based on more than 30 years of experience in using and designing computer systems, this paper summarizes a somewhat controversial view on how software should be designed to be liked by its users. Basic to this view is that it is dangerous to use computers to ensure adherence to rules, laws and regulations. When rules are interpreted by humans, the humans are capable of interpreting the rules with discretion and understand that quite often, the rule should not be adhered to 100 % in every case. Computers are not so good at making such judgements.

## Keywords

CMC, Computer Mediated Communication, HMI, Human-Machine-Interaction, User influence in software design, Personal privacy, Power, Control, Influence.

## INTRODUCTION

Some people feel threatened by computer. Other people experience computers as enhancing their opportunities. I have been using computers since 1963 and been working in the area of CMC (Computer Mediated Communication) since 1975. This paper tries to summarize my experiences and the explanatory system which I have built to understand these issues. This system can be a tool in learning how to design user-friendly computer interfaces.

## PUBLIC VIEW OF COMPUTERS IN THE 1970s

The public view of computers in the 1970s can (somewhat simplified) be described as consisting of three main ideas:

1. Computers are used to register a lot of information about people. This registration is a threat to personal privacy.
2. Computers are used to impoverish people, taking away the interest and value from their work tasks, preventing them from using their knowledge and competence.
3. In spite of these two important drawbacks with computers, we must anyway use them in order to keep the competitiveness of our economy.

Nowadays (this is written in June 2000) people have a more varied and positive view of computers. But there is still reason to try to analyze and understand the issues which were so controversial in the 1970s, and which can still

teach us much about how to use computers in human-friendly ways.

## A threat to personal privacy

**Case 1:** “The credit card company notes that mr X registers with his wife for a double room in a hotel in London. At the same time, his wife uses her credit card to pay her hairdresser in Sweden.” This is a typical example of the arguments given by people about the threat of computers to personal privacy.

**Case 2:** An American computer company decided to add, to their internal e-mail system, a facility which let the sender check if and when an e-mail was read. This caused an uproar among employees who felt this to be an invasion of their personal privacy.

**Case 3:** In 1979, we introduced the KOM forum system. KOM gave users a lot of opportunity to check on each other. A KOM user could see which forums another user subscribed to, when this user last visited this forum, when a user read a personal message, etc. In spite of this, there were very few complaints that this was an invasion of privacy. Why? We did add a facility for KOM users to say that their personal information should not be shown to other users. Almost no users used this facility to protect their personal privacy. Why, if there is such a large risk with intrusion on personal privacy through computers?

**Case 4:** But in one case there was a conflict. The director of studies at a university department started a forum for information to teachers. After some weeks, he made a printout of the KOM page, which shows which teachers had not participated regularly in this forum, and put a copy into all their physical mail boxes. He wrote on this copy: “You are all obliged to participate in this Forum!”. This caused an uproar among the teachers.

In trying to analyze these cases, my conclusion is as follows: When people complain about “a threat to personal privacy”, their real complaint is actually against using computers to try to control them. The reason so few KOM users (Case 3 above) complained, was that KOM was designed very much to be controlled by each user. KOM users experience KOM as giving them much ability to control their usage. They can choose which forums to

subscribe to, when to go to a forum, in which order to read news, what to read and what to skip. Because of this design, they did not feel that KOM was used to control them.

People felt a threat to personal privacy, when they felt that large companies and organizations “spied upon them” and used this information to gain power over them. Because KOM was not (usually, case 4 is an exception) used in this way, people did not feel that KOM threatened their personal privacy, even though KOM allowed users to see a lot of personal information about other KOM users.

This is very important. Because if we believe that the problem is with a threat to personal privacy, we may resolve this problem with methods which makes the problem worse and not better. By understanding, that the real issue is about control and power, we can solve the real problem, by trying to design the software to empowerish the user, and to make it difficult for large organizations to use the computer to control people.

### **The computer impoverishes people**

While many people experienced, and sometimes even today experience, computers as impoverishing them, other people have the opposite view, that computers increases their capabilities, empower them instead of impoverish them.

The more people are experts on the usage of the computer, the more they tend to feel that the computer empowerishes them. Especially programmers, who can actually make the computer do what they want by writing programs, feel that the computer empowerishes them.

### **HELPLESSNESS CAUSES DEPRESSION**

Seligman 1975 studied people who were depressed, and concluded that a real or experienced feeling of helplessness, of inability to influence one's life, was a major cause of depression.

Gordon 1970 describes how human conflicts are best solved. The major steps in solving a conflict, is that the parties try to understand each other. When they have learnt to understand each other, they can try to find solutions which are acceptable to both parties, so-called win-win-solutions.

Harris 1969 describes how a positive view of people accepting and respecting each other is the basis of solving problems and causing working relations.

How can the computer influence this? Well, when a user is using a computer program, this can be seen as a communication between the designers and programmers who made the program, and the user who uses them. But this communication usually very badly fulfills the requirements of Gordon and Harris. The user has usually little option to really communicate with those who made the program. Even when the user can communicate, the communicating is very much delayed. It often takes year

from a user complaint to a change in the computer program, even in the best cases.

### **Newspeak**

One way of seeing this is to look at the famous novel *Nineteen eighty-four* by George Orwell [7]. This book describes a future earth, controlled by stalinistic dictatorships who use all possible tools to control people. One chapter in the book discusses how to prevent people from thinking thoughts which the government does not want them to think. The chapter proposes the invention of a new language *Newspeak*. This language would be designed in such a way that people cannot think unwanted thoughts. If everyone was allowed to use only this language, the government could control people's thoughts.

Computers have some similarities with Newspeak. Computers allow you to communicate only using the user interface of a particular program – i.e. a special language which only allows permitted thoughts. The user interface does not allow anything except what the program was designed for.

One article which discusses this issue, and which has meant much for me, is Hoare 1975. Hoare describes a common process of software development, where developers and advanced users go through an iterative process which makes the software more useful for the expert user, but which also makes the software very difficult to learn, and thus impoverishes the non-expert user. So user influence on software design can actually make the software worse. I am fully aware that this is a very controversial statement, and it is not really my wish to argue against user influence in software design. There is strong reason to believe that user influence in software design is important. But one should be aware, that there are risks. There is often a tendency that the most advanced and expert users are best able to speak to the developers and have their wishes realized in new versions of the software.

### **Language = Power**

One should note that language is actually very often used as a tool for power. For example, medical doctors have a specialized language, which gives them more power over the patients, by describing the medical problems in words which the patient cannot understand. Lawyers and courts are famous for using legalize, a specialized language which keeps the ordinary people out. In courts, there are even rules about who is allowed to say what at what time. If you know these rules, you can accomodate them to your needs, if you do not know them, you are impoverished compared to the legal experts who know the rules. Witnesses are not allowed to listen to the trial before they are called. The reason for this is claimed to be that they should not be influenced by other witnesses. But one could as well argue for the reverse: By allowing the witnesses to listen, they can note lies and falsehoods in what other witnesses say. But the real reason for keeping witnesses outside may be an

issue of power and control: By not letting the witnesses listen, their information is less, and their control of their situation is reduced. The power of the insiders is increased, the outsiders are kept down.

There is ample reason why discussions about language often gets so heated. People defend their language, because they defend the security which their ability to communicate in this language give them.

### Cultural fields

The french sociologist Pierre Bourdieu [1] describes how groups of people form so-called cultural fields, areas in which insiders follow certain behavioural rules, and outsiders are kept outside unless they learn to adhere to the insiders' thinking and behaviour. In the computer area, the number of cultural fields is large. Every software system or network protocol has its own insiders, those who are experts on the software, knows all its facilities and knows how to get the software to produce what they want. Outside this group, there is a large group of casual users who struggle with the software and more or less succeed in getting the desired result. With time, some of the casual users become experts and get admitted to the inside circles.

### CAN COMPUTERS DECIDE WHAT IS RIGHT AND WRONG?

It is dangerous to use computers to ensure adherence to rules, laws and regulations. When rules are interpreted by humans, the humans are capable of interpreting the rules with discretion and understand that quite often, the rule



should not be adhered to 100 % in every case. Computers are not so good at making such judgements. Because of this, it can be unsuitable to put too much of rule-checking into software.

### Introducing the Issue by an Example

**Example:** Suppose you equip a motorcar with a computer-controlled device, a so-called breathalyzer, which makes it impossible for you to drive when your blood alcohol level is above a certain level. And suppose there is an exceptional case. John has a heart attack, and the only

person available to drive to the hospital, Mary, has drunk a little too much. The computer stops Mary from driving the car, and John dies.

**My argument:** This is an example of a computer making a decision. The computer decides that John is not allowed to drive the car, and in this particular case, the decision made by the computer might be ethically wrong. This is no easy issue, the best may still be to have such a computer device in the car. But the example illustrates the problem and the danger of programming computers to make decisions about right and wrong.

**Counter-argument:** No, the computer did not make any decision. The decision was made by the humans who programmed the computer. They may have weighed pros and cons, and decided that the advantage with such a drunk-driving-protection device is worth the risk that in some exceptional cases the outcome may be wrong.

**My counter-counter-argument:** This is becoming a discussion of the meaning of words. You do not accept that the computer made a *decision*. OK, let us then say that the computer made a *ruling*, or whatever word you prefer to apply to the case where a computer prohibits you from doing something. You are avoiding the ethical issues: In what way should we program computers to control human beings.

I am not a fanatical liberal who is against all laws and rules. I am quite willing to accept that in some cases it may be ethically right to program computers to prohibit you from drunken driving or stop children from downloading bomb-making recipes from the Internet. But I am advocating that in those cases you are programming the computer into making *decisions*, or *rulings* or whatever word you prefer to use. And this can be dangerous and you should be aware of the risks.

**Solution:** In this special case, a solution might be to allow the driver of the car to communicate with SOS Alarm, and allow them to send a code to override the breathalyzer lock in some very exceptional cases. By doing this, we are moving the final decision from computers to humans.

### Explanation of Problem

One mode of human communication is the setting of rules. Some human beings make a list of rules. The rules may be a law, a local ordinance, ethical rules of a professional organizations, company rules for employees or published in other ways. The human beings may also introduce ways of enforcing the rules, such as courts of law, committees on ethical conduct, etc. This is sometimes (not always) necessary even though the rulings made are sometimes wrong, like convicting innocent people. But we accept that this risk must be taken because without law and order society would not work.

An example: Even if it is forbidden for a pedestrian to step into the street against a red light, there are special cases



where this rule does not apply. Suppose a child runs out on the street, and the only way to stop the child from getting run over by a car is to run out and catch the child. Such special cases are easily handled by humans. No human court would sentence a person for running across a red light in such a case. But it is not so easy to teach a computer to understand such exceptions.

The danger is that people do not always understand, that putting such rules or laws into a computer, and programming the computer to enforce the rules, is something very different from having humans implement the rules by human decisions. Humans can understand the special conditions of special circumstances. A human might decide, in the example above, that in this special case the importance of getting John to a hospital is higher than the risk of Mary driving while intoxicated.

**Counter-argument:** A counter-argument [6] is that in this case it was humans who made the decisions, by programming the car computer, but their decisions were wrong. They did not take all circumstances into account. They should have made a more advanced program, which could take into account the special circumstance of the heart attack situation.

**Counter-counter-argument:** What you are doing, with this kind of argument, is to make the computer program more advanced and complicated, to reduce the risk that the decisions/rulings made by the computer are wrong. The path you are treading may make things worse instead of better. More complex and advanced computer rulings may increase the risk of wrong decisions instead of reducing them. The right solution may sometimes instead be to accept that the computer is not perfect, and thus that all rules do not have to be enforced by computers alone.

#### **“Italian Strike” Example**

A well-known method for strikes is to continue to work, but to adhere 100 % to all rules while working. This rapidly causes many businesses to a complete standstill, or at least makes them work much slower and less efficiently before.

The reason for this is that even good and benevolent rules can have disastrous effects if adhered to 100 %, the way a computer would do if programmed to enforce them.

#### **Filling in a Form Example**

If you fill in a form manually, you have the option of adding an accompanying sheet of paper explaining why you filled in the form in a particular way. In most forms, where there is a “yes-no” question, it is quite possible to omit the checkboxes and write a more nuanced explanation below or above. The human who receives the form will read and understand and interpret this. When the form becomes computerized, this freedom to not follow the prescribed way of filling in the form often disappears.

#### **Society Evolves by Many People doing Things in Better Ways**

An important way in which human society is evolving is that many people make small and large decisions to try out new and better ways of doing things. If the computer program stops them from doing things in other ways that those specifically allowed, this will prevent people from finding better ways of doing things, and thus stop improvements.

Sometimes this may be necessary. For example, there is a human tendency to stop performing actions which are necessary only to avoid seldom occurring risks. Example: A pilot forgets an item on the pre-flight check list, or a night watchman forgets to go to a normally empty part of the building. In such cases, it may be necessary to use technical means to ensure that the human follows the rules, for example the night watchman must turn a key to show that he has passed that part of the building. But this does not forbid the night watchman from disobeying the rules in special cases, for example skip the empty corridor if there is a thief in another part of the building. The danger is when the computer does not allow you to do things in other ways than those foreseen when programming it.

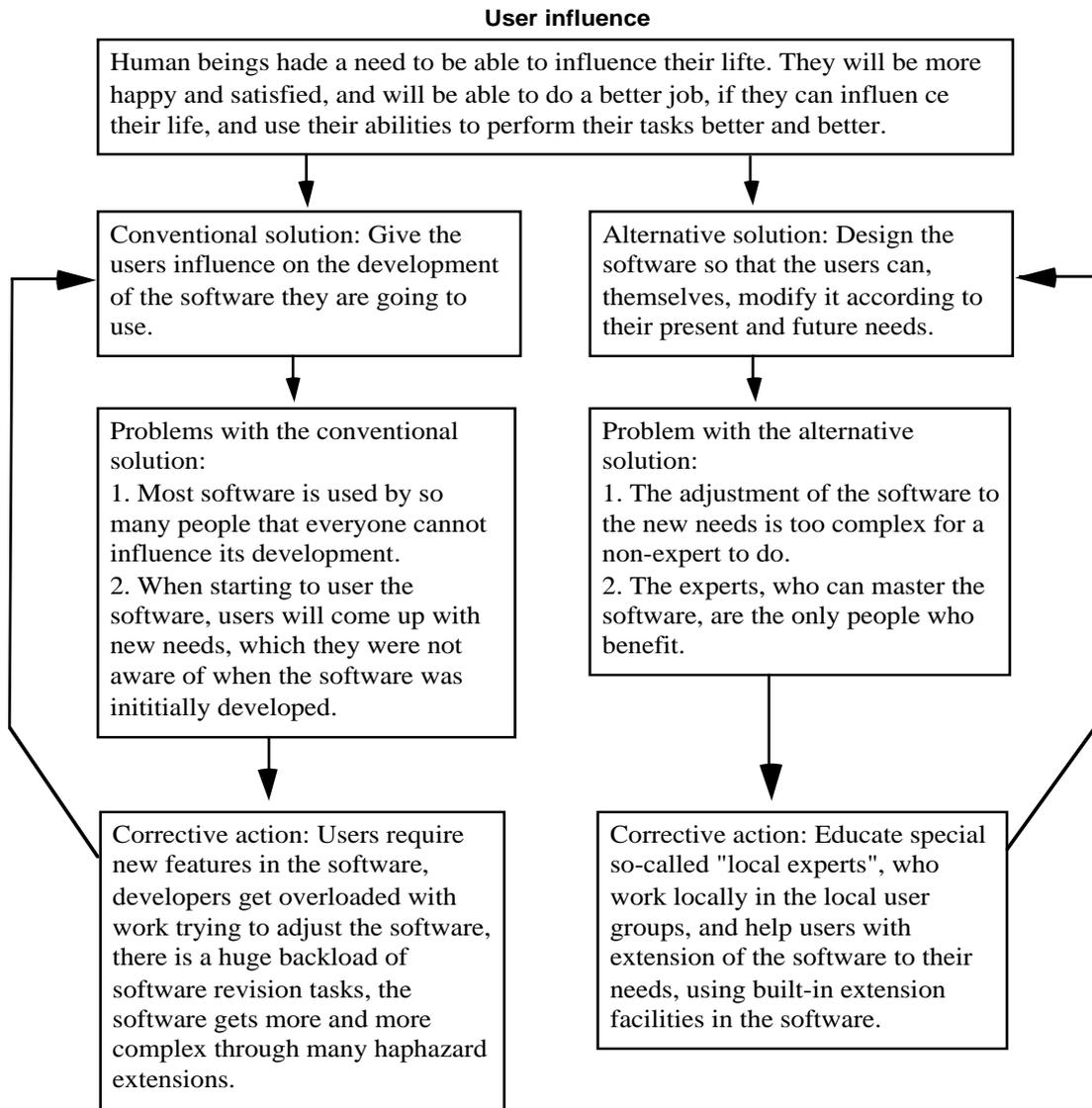
#### **Two-sided Communication is Better than Enforcing one Solution**

It is a well-known fact that power is addictive. That is why we design human societies with so many safe-guards against giving individual people too much power. We must understand that the power to control other humans by design of computer software can also be addictive [5]. Only by understanding this, can we stop people from putting too much control of humans into their software. I am not arguing that there should be no control of humans by software. The common “Are you sure?” dialog boxes are often motivated, even though they are sometimes a nuisance. But those who design computer software should be aware of the risks of putting too much control of humans into the software.

This will reduce the possibilities for people to influence their environment, and will create a feeling of helplessness,

which may cause dissatisfaction and depression [11]. When two humans communicate regarding a task, the outcome of their discussion is usually twosided or so called win-win solution, a solution where the needs of both are taken into account [5, 2]. When the task is controlled by a computer (even though a human did originally program the

computer) the interaction necessary to achieve a win-win solution is often not possible. It is well-known that such situations easily cause frustration and dissatisfaction and also often mean that a less good way of performing the task has to be used.



**Stopping the Porn**

An example of an application area, which illustrates the problems with computers deciding what is right and wrong, is the area of the porn-blockers, program modules meant to prevent use of the Internet for unsuitable purposes [6]. They are used by parents who do not want their children to download porn on the net, by schools and libraries, and also by employers to stop employees misusing their office computers, and even by countries to control the flow of

information and stop undesirable information. The People's republic of China and Singapore are examples of countries who want to stop unacceptable information, such as views by so-called dissidents.

These programs, however, have severe problems. Either they permit only access to listed and allowed sites. But since the developers cannot keep up with all pages on the Internet, only a small subset is listed. Alternatively, they try to guess whether a document is suitable or not, this is done

by scanning for certain character strings, like “sex”, “breast” and “xxx”. This has led to horrendous mistakes, such as prohibiting information from Middlesex (a local government in England) or prohibiting information about breast cancer. A computer user complained that when he downloaded code in a particular script language from the Internet, the code was distorted in funny ways. For example, the following piece of script code:

```
#define one 1 /* foo menu */
#define two 2 /* bar baz */
```

Was corrupted in the following way:

```
#define one 1 /* foo      */
# fine two 2 /* bar baz */
```

I leave it to the reader to compare the scripts and conclude what “Cybersitter” had done with his script code and why.

### Is the Internet Illegal

Actually, almost all usage of the Internet is illegal according to the privacy protection laws in many countries. These laws prohibit all transport of personal data from one country to another without permission from the government. If these laws had been programmed into the computers, then we could not have had the Internet we have today. We should be happy that the laws are enforced by humans who understand that the intention of privacy protection laws are not to prohibit free speech. You cannot be sure of this. I ran one of the first Swedish BBS-es in 1978. We were forbidden to run our BBS by the Swedish Data Inspection Directorate. Later on, we were allowed to start it again, provided that we did not allow anyone to write any political or religious opinions in forums on the BBS (Since the Swedish Data Act forbade the creation of registries of political and religious opinions, except in certain special cases, but in contradiction to the Swedish constitution, which specifically says that the right to communicate freely on political and religious issues should be protected). We continued to use our BBS including some discussions of political or religious issues. No one prosecuted us. But what would have happened if the computer had been programmed to recognize and automatically prohibit any message with political or religious content?

### The Computer need not Stop all Unwanted Behavior

The idea that human rulebooks should be programmed into computers is closely connected to another faulty idea. This other faulty idea is that anything is legal, which the computer permits you to do. “The computer did not stop me from accessing this data”, is the standard defense from the cracker who breaks into a computer.

If you believe that anything allowed by the computer is legal, then obviously you have to program the computer to prohibit all unpermitted behavior. One can understand the danger of this by trying to envision a society where all illegal acts are made impossible to perform. Hammers are

not allowed, since you can kill people with a hammer. Suppose you need a hammer in woodworking. Tough luck, this is illegal, hammers are inherently dangerous. In order to prevent crime, every movement from one place to another without permission might be prohibited and monitored. Is this the kind of society we want?

A real example which I have actually seen: A building where I worked was split into zones. Whenever you moved from one zone to another, you had to insert a keycard into a slot to open the door. A person inserted the keycard, opened the door, then dropped the keycard, bent to pick it up, while the door closed with the person still in front of the door. The door locked automatically, so the person inserted the keycard again to open the door. This did not work. The computer obviously reasoned as follows: “This person has already passed into the new zone. She cannot be in front of the door. So her keycard must be falsified or wrongly used.”



Faulty programming of the computer? Perhaps, but you can never be sure that your program is perfect. And making the program more complex by taking into account more special cases in deciding what the computer allows and prohibits may introduce more bugs, while removing old bugs in the software. The new bugs may be more insidious and difficult to find. All problems are not best solved by making computer software more complex. Some problems are better solved by letting humans, instead of computers, make decisions!

### Calendar Scheduling

A good example to discuss these issues is the use of computers to schedule meetings. This may at first seem like a good way of using computers. But the more you look at the problem, the more you find that real meeting scheduling includes so many special cases, where human judgment is needed, that it becomes very difficult to get the computer to

do this automatically. For example, some meetings are more important than other meetings, and may cause other meetings to be rescheduled, but such a decision cannot be done by a computer. And there are contextual factors, like knowing that you should never schedule a meeting with a certain person on a Monday morning, which everyone knows, but which are difficult to put into the computer.

### INTELLIGENT USER INTERFACES

When we designed the KOM forum system user interface in the 1980s, we wanted to make the software easy to use and adaptable to user needs. To achieve this, a user could rapidly scan through everything new to this user, by a very simple repeated command. But different forums have different importance to different people. We then let users order the list of subscribed forums according to their personal preferences. The default method of scanning through news made this in the user-preferred order.

But how should users tell the computer which forums are of special importance? In the first version, we did not want to burden users with special commands to reorder the list of forums. Instead, we designed the software so that whenever a user exited from a forum, without reading everything new in this forum, the system automatically concluded that the forum left should be lowered in priority, and the forum to which the user moved should get increased priority.

Users, however, did not like this. They felt that the computer tried to control them, by reordering the list of forums automatically. So in a later version, we removed the automatic reordering of forums, and gave users explicit commands to do this reordering instead.

### Conclusions

- ⇒ The successes of human society is based on the flexibility of humans and their willingness to adapt their activities to different circumstances.
- ⇒ Humans are most happy and productive if they can influence their living environment and contribute to solving problems together.
- ⇒ Laws and regulation are a form of communication between humans. They are in reality only guidelines, people have to adapt to varying circumstances and interpret and apply the rules with understanding and human compassion. If everyone had to adhere 100 % to all laws and regulations, human societies would not work any more.
- ⇒ This is usually no problem when the laws and regulations are written on paper. But if the laws and regulations are programmed into computers, so that the computers control what is allowed and not allowed, serious problems will often occur. In the best case, people will only be unhappy and unproductive, in the worst case, major catastrophs can occur.

- ⇒ Computer software must be designed to allow flexibility and human choice. Laws and regulations should be interpreted by humans, not by machines.
- ⇒ Making the software more complex, to include in it more different special handling of special circumstances, will often only make it worse. Instead of complex software, software should be flexible and open-ended.
- ⇒ There is a human tendency when designing software to want to include in it “proper procedure” and “experience how things should be done”. This tendency can easily produce unusable or unsuitable software.
- ⇒ Possible exception: Certain security rules, where enforcement is needed to overcome human weaknesses.

### REFERENCES

1. Broady, D.: 1988: *Kulturens fält - om Pierre Bourdieus sociologi*. Masskommunikation och kultur, Nordicom-Nytt/Sverige Nr 1-2, 1988.  
<http://dsv.su.se/jpalme/society/pierre.html>
2. Gordon, T. 1970: *P. E. T. - Parent Effectiveness Training - the tested new way to raise responsible children*. New York : P.H. Wyden 1970, ISBN: 0-88326-039-5
3. Grip: *ADB-system och kommunikation (Data processing and communication)*. Hermods-studentlitteratur, Lund, Sweden, 1974.
4. Harris, T. A.: *I'm OK - You're OK, by Thomas A. Harris, 1969. I'm OK, you're OK: a practical guide to transactional analysis*. New York : Harper & Row 1969.
5. Hoare, T.: Software Design: a Parable. In *Software World*, vol. 5, No. 9&10, 1975.  
<http://dsv.su.se/jpalme/s1/hoare.html>
6. Martin, C.D.: *Empowering Educators and Parents: Content Advisories for the Internet*. By C. Dianne Martin, *Proceedings of the ITiCSE ACM conference*, June 1997. URL:  
<http://www.dsv.su.se/~jpalme/reports/iticse-notes.html#martin>
7. Orwell, G.: *Nineteen eighty-four*. 1949, ISBN: 0-14-012671-6
8. Palme, J.: *Interactive Software for Humans*. Management Informatics vol. 7(1976). At URL: (HTML version):  
<http://info.dsv.su.se/~jpalme/reports/interactive-software.html> and URL (Acrobat version):  
<http://info.dsv.su.se/~jpalme/reports/interactive-software.pdf> .

9. Palme, J.: User influence on software design may give less good software, 1997. <http://dsv.su.se/jpalme/s1/control-power-home.html>
10. Palme, J.: *Can computers decide what is right and wrong?* 1997. <http://dsv.su.se/jpalme/reports/right-wrong.html>
11. Seligman, M.E.P.: *Helplessness: On depression, development and death*. W.H. Freeman, San Francisco 1975.