

Consequences of IT-use, good and bad.

What can and should we, computer scientists and others, do about it?

Talk, session 10 “What is so fascinating with Computer Science?”

Dirk Siefkes, Technische Universität Berlin, Fak. Elektrotechnik und Informatik

In my abstract for the session I explained how I understand fascination: as a mixture of excitement and fear/anger, of good and bad. IT fascinates, I said. Here are two examples: With IT we communicate easier and much more. Formerly we wrote letters, not too many, but thought through and well formulated. Today we write quick emails or just rub our thumbs to disperse our ideas. Formerly with work we depended upon competent people; today we need people who fit into slots of the hierarchy. Thus today work is narrow, prefabricated, not our own. If we want a new job, mostly we have a wide choice; but we have to fit in, we have to be flexible, which might corrode our character (Sen98). - Fascination has for a long time played an important role in my scientific career and my personal development. Thus I follow that line.

From mathematical to cultural theory

I studied mathematics, did my PhD in mathematical logic at the University of Heidelberg, and then visited Purdue University, USA, for a year. I was invited by the Math Department; they, however, didn't have money, so I had to teach computer science. Thus I got to know CS by teaching it, mostly a week ahead of the students. When I came back to Germany CS was just established as a new field and I was offered a chair in Theoretical CS, i.e. in mathematical theory. I seemed on my way in a mathematical career, although in CS.

During the year at Purdue, however, another important thing happened. My wife and me wanted to do something together (besides raise our four boys) and to improve our English. So we took courses in English Literature. It was a course on “Literature of Ecology” that caught me. We read the classics like “Walden” by Henry David Thoreau, and I got to know ecology: To understand something you have to see it evolve within its environment. I tried to apply the approach to my formal work, see the report (Sie92) “Fish in schools or fish in cans. Evolutionary thinking and formalization”.

I called the result “small systems thinking” (Sie82ff): (1) Never consider the individual alone, but always as a member of the small groups it is living, working, learning, conversing in; and conversely, see how such groups are determined in their development by their members. Thus, understand groups and individuals as mutually dependent. Here a system is “small”, if it evolves according to this interrelation; for which smallness in number is necessary, but not sufficient. Do the same with groups within their schools, schools within their institutions, and so on up to societies, cultures, nations as ultimate frames. (2) As important: Never look at states, always look for processes. Small systems are not static, they develop, changing thereby. They evolve.

Consequently, I started to discuss with some colleagues how we could develop a theory for CS appropriate for that thinking. As computers had been developed in the 1940s by engineers and mathematicians, CS had begun as engineering plus programming plus mathematics as its formal underpinning. That was historically correct. But when you looked at computers within the environments they were used in, you needed a theory drawing from sociology, psychology, etc., from all of the humanities, besides from mathematics. So Wolfgang Coy started a project “Theorie der Informatik” (Theory of CS) where we all participated, discussing, writing, learning together for some years (Tdi92). Ten years later Frieder Nake, Arno Rolf and I took up the task with similar work (Tdi01ff). At last I continued the work under the name “Cultural Theory of CS”.

But before that my group in Berlin did something that deeply influenced my development. A true small systems event. I remember vividly, how during a sabbatical at Berkeley University an assistant called me from Berlin: “We want to do an interdisciplinary project “Sozialgeschichte der Informatik” (Social History of CS). Will you join us?” I didn't know anything about the subject, but of course I wanted to participate, and of course I had to be the boss. So when the project started in 1994, we looked at the early papers on computers and programming, including the great paper by John van Neumann 1945 where he describes the new machine as having a ”memory”, not a store. What we found struck us (SGI97ff): All the authors, whether they write on planning, building, developing, using the new machine that will do some human task, while doing this look at man and machine at the same time. They don't identify the two, but they *hybridize* them, as we called it, melting them into one hybrid object. Independently and at about the same time Christiane Floyd and Frieder Nake found the same phenomenon; they called the hybrid objects “autooperational form” and “algorithmic sign” resp. (Flo97, Nak01).

And I maintain: We still do, we cannot deal with computers, with IT, otherwise. This is good, enables our work. And it is bad: All the errors in developing and using IT result from it. Think of the “software crisis” which began in the 1970s and never ended. As man and machine are in fact different, software never works, it always contains errors. And is the situation today different? We rarely feel that IT fails; but is this because we are technologically advanced, or because IT is so overwhelmingly “smart”, or because we have reduced our expectations to what IT can do? More general: Our mental activities and our technical constructions influence each other. What does that tell us about us? Great questions. Fascinating!

IT and CS

In the title of the session I combine two topics: Fascination with CS and Dealing with IT. Up to now I have talked about fascination with IT, nothing much about CS. So how relate the two? How do they influence each other? To get answers we will consider two types of questions.

First: How does dealing with IT influence the user? We mentioned good and bad consequences, see the examples above. But can we agree on good and bad, at least in some cases? Or are *digital natives* like the younger ones among us and *digital immigrants* like we traditional scientists forever separated into two cultures? (See Hyp12, Sie13a,b). I got the distinction from Hyperkult, a great series of yearly conferences, like STS, which the organizers propose to shut down in 2015. Or can we enjoy the variety, looking for others whom we could join or who would join us? This would be appropriate with small systems thinking; thus this is my way. Then we can also enjoy all the consequences of IT-use without hesitation as we are never alone on the way.

Second: How can we influence this fascination through our work in CS? Can we change CS, this hardcore science of engineering, software and mathematics, into a science of design, as Schulz-Schaeffer would have it in his talk? Or can we transform what we learn in CS about IT into a language for everybody? Teresa Macchia reports in her talk how she designed museums in such a way that the visitors, IT-guided, can and should find their own way(s!) through the exhibitions. Thus this “language of CS” does not consist in commands and directions, but in suggestions, as should be the case when people talk to each other. Therefore everybody should be able to understand what we tell him/her through CS. Or can we simply enjoy the parts of CS attractive to us, leaving the other parts to other people who like them? See the book “Vielfalt der Informatik” (Variety of CS) by Anja Zeising and others (VdI14) who report from a project InformAttrac (Attraktive Informatik, Attractive CS). What I like about the book: It maintains a feministic approach (“Informatik-Professorinnen für Innovation und Profilbildung. Eine Informatik, die für Frauen und Mädchen attraktiv ist” - “Women CS

professors for innovation and profilization. A CS attractive for women and girls”), but it does not push aside those parts of CS attractive for, or even dominated by men; rather it lets those to others, men or women, who like them. Thus there could evolve a CS where men and women do not hinder, but help each other, to find their approaches. Hopefully they can even cooperate in that search. What I don't like: The authors seem to take all of IT as “products of CS” (informatische Produkte). I think that it is mainly producers, industry and the like, and politicians who dominate the development of IT. The question is how we, through our work as computer scientists, can influence that development (see above).

Promoting small systems (thinking) in CS

Can we break down the borders between the different fields of science? Borders help us find others with whom we can work, but they involve aggression, deprehension, fear. Do we have to settle down between the traditional fields, seeking collaborators, a strange colored bird between the white chickens? If so, where? Will CS still be a field of engineering? Or of the humanities? Or begin a new type of science? Who then are we? Strange. No, fascinating!

References

- (Bau01) K. Bauknecht et al. (Hg.): Informatik 2001, Jahrestagung GI & OCG. Workshop "Erkenntnistheorie - Semiotik - Ontologie". Österreich. Computergesellschaft, Wien.
- (Flo97) Christiane Floyd: Autooperationale Form und situiertes Handeln. In C. Hubig (Hrsg.): *Cognitio humana - Dynamik des Wissens und der Werte*. XVII. Deutscher Kongress für Philosophie, Vorträge und Kolloquien, S. 237-252.
- (Hyp12) Martin Wancke et al.: Digital Nativity – die Normalität des Digitalen; HyperKult XXI, 12.-14. Juli 2012, Lüneburg.
- (Hyp13) -“- : Standards, Normen, Protokolle; HyperKult XXII; 4.-6. Juli 2013, Lüneburg.
- (Nak01) Frieder Nake: Das algorithmische Zeichen. In Bau01, S.736-742.
- (Sen98) Richard Sennett: *The Corrosion of Character*. New York: Norton. - Dt.: *Der flexible Mensch. Die Kultur des neuen Kapitalismus*. Berlin Verlag.
- (SGI97) Peter Eulenhöfer et al.: Die Konstruktion von Hybridobjekten als Orientierungsmuster in der Informatik. TU Berlin, FB Informatik, Bericht 97-23.
- (SGI98a) -“- : Sozialgeschichte der Informatik. FIF-Kommunikation 2/98, S. 3-4, 28-48.
- (SGI98b) Dirk Siefkes: Sozialgeschichte der Informatik. Kulturelle Praktiken und Orientierungen. Deutscher Universitätsverlag 1998.
- (Sie82) Dirk Siefkes: *Kleine Systeme*. TU Berlin, FB Informatik, Bericht 82-14. Engl.: *Small Systems*. Purdue University, Computer Science, CSD-TR 435, 1983.
- (Sie92) -“- : Fish in Schools or Fish in Cans. Evolutionary Thinking and Formalization. International Computer Science Institute Berkeley, TR-92/009.
- (Sie01) -“- : Informatikobjekte entstehen durch Hybridisierung. In Bau01, S. 798-803.
- (Sie13a) - „, - : Glauben, Wissen, Können, Müssen – und Wollen. Wie wir mit Digital Natives eine nicht normierte Informatik gestalten können. Abstract Hyp13, 2 S.
- (Sie13b) - „, - : Digital Natives in Symbiosis with IT-systems? Abstract, Conf. „Futures Past: Design and the Machine”, MIT, November 21-23, 2013; 1 S.
- (Sie14) -“- : What is so fascinating with Computer Science? How we, Computer Scientists and others, deal with IT. Session, STS-conference Graz Mai 2015. Book in preparation.
- (Tdi92) Wolfgang Coy, Frieder Nake, Jörg Pflüger, Arno Rolf, Jürgen Seetzen, Dirk Siefkes, Reinhard Stransfeld (Hg.): *Sichtweisen der Informatik*. Vieweg.
- (Tdi01) Frieder Nake, Arno Rolf, Dirk Siefkes (Hg.): *Informatik - Aufregung zu einer Disziplin*. Tagung zur Theorie der Informatik 2001. Uni Hamburg, FB Informatik, Bericht 235.
- (Tdi02) -“- : Wozu Informatik? Theorie zwischen Ideologie, Utopie, Phantasie. Tagung zur Theorie der Informatik 2002. TU Berlin, Fak. Elektrotechnik & Informatik, Bericht 02-25.

- (Tdi03) -'' : Informatik zwischen Konstruktion und Verwertung. Tagung zur Theorie der Informatik 2003. Uni Bremen, FB Mathematik & Informatik, Bericht 1/04.
- (Tdi09) Andreas Möller, Frieder Nake, Arno Rolf, Dirk Siefkes (Hg.): Beiträge zu einer Theorie der Informatik. Zum kritischen Selbstverständnis einer Disziplin. E-Journal Communication, Cooperation, Participation, Heft 5 (Sonderausgabe), 08/2009.
http://195.37.26.249/ijsc/en/special_edition.php
- (Vdi14) Anja Zeising, Claude Draude, Heidi Schelhowe, Susanne Maaß (Hg.): Vielfalt der Informatik - Ein Beitrag zu Selbstverständnis und Außenwirkung. Projekt InformAttraktiv. Staats- und Universitätsbibliothek Bremen 2014
<http://nbn-resolving.de/urn:nbn:de:gbv:46-00104194-14>
- (Zu meinen Publikationen siehe auch <http://tal.cs.tu-berlin.de/siefkes>)