On Methodic and Didactics of the History of Informatics

Laszlo Böszörmenyi
Klagenfurt University, Department for Information Technology

"And God saith, `Let Us make man in Our image, according to Our likeness …'"
Genesis, 1, 26

"In the moment that I allowed the computed data to influence the program – for that only a small wire connecting the arithmetic unit and the stored program is required – I could no longer monitor the calculations. I had a lot of respect for that little wire, because I felt as soon as this wire is there, Mephisto stands behind me. ... With it a programmer can do the most amazing things."
Konrad Zuse in a talk, given at the ETH Zurich in 1992

"As the years went by, I became convinced that the influence of automatic computers in their capacity of tools would only be a ripple on the surface of our society, compared with the deep influence they were bound to have on our culture in their capacity of intellectual challenge to Mankind that was totally without precedent."
Dijkstra, personal notes, 1975

I. Introduction – The unfinished man and his creations

What does it mean that God makes man to "his likeness"? Is this the external image, the body? I doubt it! We know only one single characteristics of God at this time: the he "makes" that he creates. This is presumably the idea: that man is made to God's image in that man should become a *creator* himself (or herself – there are good reasons to assume and many traditions say that "man" was created first as an *andogyn* being, man **and** female). People are the only beings on earth that are born 'unfinished', in the sense that people are able to be educated as children, and able to educate, to develop themselves as adults. Whether we believe in a creation by God or not, this specialty of the human being is obvious – even if some animals have a certain, limited level of capability to be trained. The creations of people are usually 'finished'. If we make a chair, we know the purpose of it and we do not wish any surprise (a chair with missing or shaky legs is simply a bad chair). Nevertheless, the idea, to create something, which is 'unfinished' yet, created to "our likeness", which has its own "life", analogous to the idea of a "homunculus", is very old. Is Mephisto, who "stood behind" Konrad Zuse, the same who assists Wagner, the scholar of Faust in Goethe's *Faust*, in creating the homunculus? I guess, yes (Zuse knew Faust surely well). The computer comes closer to the idea of an 'unfinished' machine than any other equipment created by humans. The idea was surely present also in the universality of the Turing machine and in the generality of the programmable John von Neumann machine.

I definitely do not want to say that by inventing the computer "man" succeeded in imitating the creation of God. What I want to say is that the old desire of creating an 'unfinished', educable being might have inspired the invention of the computer considerably. And I think that this is one of the main reasons that computers are still "interesting" for the public, not only from the point of view of practical use, but also philosophically, sociologically, psychologically. This is probably also the reason, why the "death of computing science" – as predicted by some authors, such as Neil McBride –, might be not so near as assumed. Computers have become a part of our culture; they – actually not the machines themselves, much rather our way of dealing with them – have influenced both our every day life and our way of thinking to an extent, which is "totally without precedent".

This phenomenon is interesting and the history of computing science (or informatics) could reveal at least a bit of its mystery.

II. Methodic questions

The history of informatics has both *internal* and *external* questions. An internal question can be answered by looking only at informatics itself, the external (or interface) questions relate to the interaction between informatics and the rest of the world (other sciences e.g.). The external questions are obviously more general, more interesting for the public, but cannot be answered without a thorough investigation of internal questions. The probably most relevant external questions are: (1) Where do the ideals of computing science come from? (2) How does the development of computing science influence our ideals? We will see that a methodic study of both questions is fairly similar. In some cases we are more interested in the first, in others in the second one. For example, it is interesting to ask: which external inspirations played a role in the emergence of software engineering? The influence of the latter on the rest of the world might be less interesting, at least from a theoretical point of view. On the other hand, in the case of digital media, the really thrilling question is related to their influence on our habits of reading and learning.

1. Where do the ideals of informatics come from?

Quite a lot of effort has been invested into technology assessment, into predicting the consequences of technological development. A not less interesting, but much less investigated question is where do the ideals of technical development come from? In the every day practice of computer scientists we get our ideas – to be honest – mainly from each other. Researchers read the papers of each others; sometimes they are also involved in industrial projects, which put somewhat different questions. Based on these interactions, technology produces an impressive development in rather small steps. However, at certain times, some ideas emerge that could be regarded as revolutionary, that do not represent just small modifications (mostly, but by far not in all cases, improvements) of the actual state.

For example, the idea of *structured programming* – later rephrased as software engineering – was inspired by the ideal of elevating computer programming to a science, to a discipline that is as sound and exact as theoretical physics and as effective as traditional engineering. I think it is fair to say that this ideal has never been reached, not even approximately. Why? An answer that people dealing with computers are not as smart as physicists or mechanical engineers is hardly satisfactory. The reason must lie deeper in the nature of creating software and in the usage of computers in most different fields of everyday life. A scientific analysis of such a question could bring a lot of light into the nature of computing science. Actually, similar questions have been put by many exponents of software engineering; To my best knowledge, however, such questions were frequently aimed at finding arguments for the questioner's own work. A methodically correct, scientific research should address such a question without giving favor to any actual technique. I try to give an example of possible steps of such a research:

1) Analyze the usual techniques of software development before the idea of structured programming has emerged.
2) Summarize the essence of the proposals made by Dijkstra, Hoare, Dahl and many others in order to agree on a more disciplined way of programming than before.
3) Look for alternative, already forgotten proposals of the same time.

4) Try to reconstruct the discussion on this topic – including the implicit, not obvious threads of the opinion exchange.
5) Figure out the reasons for the acceptance and rejection of ideas – not only the obvious, verbal ones, but also hidden, political, economic and psychological motivations, interests, suppressions etc.
6) Draw an analysis of the current situation in the light of the insights gained in the previous investigation.
7) Deduce predictions and proposals for the future development.
8) Give examples of the consequences of ignoring resp. following sound proposals – in order to make the issue understandable also for non-technical people, especially for politicians and managers (including science mangers).
9) Find analogies with other sciences and other fields of the cultural life.

This list is neither exhaustive nor can it be followed sequentially. It is an example for a possible guide to research addressing the initial question.

2. How does the development of computing science influence our ideals?

This question is more related to the issues of technology assessment. However, I think that a historical view, paired with knowledge in computing science, could put and answer additional, interesting questions. Let's take the example of *digital media,* in order to present some possible steps of a methodic investigation:

1) Investigate the historical development of communication media.
2) Investigate the role of different communication media in the individual development of children and adults (how we learn to speak, to draw, to write etc.).
3) Describe the situation, when the first ideas of digital media emerge. E.g. Donald Knuth and his Tex system, which enables computer-supported composing of printed text, and the introduction of pixel graphics in the Alto computer at Xerox PARC, which goes one step further and enables low-cost graphics, including "what you see is what you get" images of printed text.
4) Follow the main steps in the development of digital media (vector and pixel graphics, color, 2D and 3D graphics and animation, audio and video, development of input and output devices in the service of digital media).
5) Try to find – already forgotten – ideas for digital media.
6) Figure out the reasons for the acceptance and rejection of ideas – again, not only the obvious, verbal ones.
7) Figure out the influence of digital media on the usage of traditional media (such as reading) and also the changes in using digital media. Ivan Illich describes in his amazing book "In the Vineyard of Text" how the mode of reading changes in Middle Age, and how this process correlates to the changes in consciousness of reading people – and society in general. A similar investigation of the transformation of modes of reading and its influence on our consciousness would be one of the most interesting studies that could be imagined.
8) Deduce predictions and proposals for the future development.

This list has many similarities to the previous one, what I regard as an advantage. We could presumably develop a merged list for both main questions. Nevertheless, I refrain from this, because I try to resist the temptation of over-abstraction – a typical occupational disease of computing scientists.

III. Didactic issues

Didactic is interesting in double sense: (1) How to teach the history of informatics? (2) How can we improve teaching of informatics itself, by relying on knowledge in the history of informatics? A third question could be, how to integrate the history of informatics into other subjects, such as history, sociology etc. – this question is, however, out of scope of this paper.

1. How to teach the history of informatics?

This question is much harder than it might seem. As long as history of informatics is not a real science, as long as we miss a more or less accepted methodology, it is almost impossible to answer this question. Moreover, it is not sure at all that the history of informatics should be taught in separated, own courses. It could be much more important to integrate a historical view into technological courses in order to broaden the prospect of students – and teachers.

2. How can we improve teaching of informatics, by taking a historical view?

This question was discussed by the author elsewhere[1]. The essential points are the following:

1. A historical view can give informatics a human face. Instead of speaking about technologies, products etc. we can speak about people and their ideas.
2. By showing different – often complicated – ways of historical development we can encourage critical thinking in an excellent way. We can show our pupils that
    a. Ideas do not grow from the ground, they need people.
    b. Even good ideas may disappear; often worse ideas conquer better ones.
    c. Everybody can have excellent ideas and thus, change the history a bit.
3. We can relate informatics to other sciences and the everyday life.
4. We can address a number of external issues, especially ethical ones, such as the responsibility of people dealing with informatics.

IV. Conclusion – The unfinished man recreating himself

Probably the most important general lesson what we can learn from the history of informatics is that even though computing science is a very successful technology, the computer is still a very poor imitation of God's creation. This could us lead to the consequence that the most creative act of humans remains learning, educating – and thus recreating – himself/herself. Knowledge cannot be quantified and cannot be measured. The real goal of learning is not to agglomerate knowledge, but to keep fresh our capacity to learn new things, to receive new ideas. If we look at the example of great computing scientists, such as Dijkstra, Dahl or Nygaard then we find exactly this attitude[2]. This is, of course, not a privilege of informatics, but it is present here as elsewhere. Informatics did not make us more similar to God than any other science – but we may hope that it did not make us much less similar either.

[1] Böszörmenyi L.: Teaching: People to People About People (A plea for the historic and human view) In: Mittermeir, Roland T (Ed.): From Computer Literacy to Informatics Fundamentals. Heidelberg, Germany: Springer Verlag, February 2005 (LNCS, 3422), pp. 93-103.
[2] Böszörmenyi L., Podlipnig St.: People behind Informatics, Universität, Klagenfurt August 2003, 121 pp.