Paradigms of Computing Science

The necessity for methodological diversity

Heidi Schelhowe, Computer Science, University of Bremen, Germany
schelhow@tzi.de

Published: GTD, Gender, Technology and Development. No.3,
Sept-Dec. 2004. pp.321-334.

In Western countries the percentage of female students in computer science frequently ranges at conspicuously low levels. The debate on gender studies in this field therefore often concentrates on the problem of increasing the number of female computer professionals. Notwithstanding, I would like in this contribution to pursue the question of the occurrence of "gendering" (and the reasons for it) over the course of the discipline's historical development. It must remain uncertain whether an examination like this can explain the respective numbers of careers or quotas of women working in computer science — yet in spite of this, the following line of argument will hopefully give credibility to my claim that present-day computer science, especially in Germany, has attained a public image that does not make it easy for women to embark on a career in this field.

Germany is famously known for its extensive engineering tradition in the time before and up to the second World War. Repercussions of this reputation can still be felt today. "German" is understood to stand for high-quality technical construction, solidity and reliability through close ties with mathematics, for predictability and security. Simultaneously, education and profession of the engineer are associated with the concept of virility, in a manner less common in almost any other societal field.

The existence of a divide which separates femininity from technology has its reason not in the use of technology but instead in its constructive, engineering aspect. In almost all of the curricula of the classical engineering sciences in Germany, women represent only a vanishing minority, in most cases of less than 5%. In its time of foundation, computer science was no exception to this rule. At the beginning of the 1980s though, the share of computer science freshwomen at German universities had increased to 20%. However, this state was soon "normalized" by the discipline's growing establishment: Computer science, now equipped with an air of high technological prestige, became the unchallenged domain of men. Only in recent years has the share of women become slightly larger again.

In what follows, I want to investigate the character of computers and Computing in its scientific aspect and how they differ from conventional products of engineering and engineering sciences. It seems important to me to begin by working out aspects of distinction and novelty so that subsequently it may be shown how the study of computer science could “nonetheless” be established as a male-dominated territory.

I. The first computer

According to the "Studien- und Forschungsführer Informatik" (German guide to study and research of computer science), computer science aims for the creation of "immaterial" products. It further supplies the theories, models and concepts needed in its production processes.

Information, its symbolic representation through signs and their transformation into software are computer science's primary subjects. Yet its efforts must ultimately be directed towards a device, a physical artefact: the computer, reduced in size to the point of invisibility, and hidden behind a so-called "interface", behind input, output and peripheral devices such as screen, keyboard, mouse, printer, disk drive, camera, data helmet or washing machine.

Both the classical engineering disciplines as well as computer science involve the process of manufacturing. However, computer science does not focus on the production of machines in the conventional sense but on the achievement of intellectual results: computer programs. Here lies a crucial difference to classical engineering, which has its roots in the mathematical origins of computer science, more precisely in the "invention" of an *abstract* machine by Alan Turing.

In the 1920s David Hilbert had set down a foundational programme for mathematics. Alan Turing committed himself to one of Hilbert's proposed tasks, the so-called "decision problem". He demonstrated the insolvability of the decision problem, which asked for a *mechanical* procedure able to determine whether a given string of characters forms a valid mathematical theorem or not. Turing's paper "On Computable Numbers, with an Application to the Entscheidungsproblem" (1937) was of lasting consequence, giving (almost concurrently with Emil Post and Alonzo Church) a convincing formal definition of the concept of computability in mathematics. Alan Turing defined computability with the aid of a machine he had conceived of to exist only in his mind, an abstract machine. According to Turing, any result this machine was capable of producing could be named computable. The machine's design was based on his considerations on how a human being may "mechanically" execute calculations. In this way, a mental connection was formed between the intellectual domain of human thought and that of material production by machines — a notion that pervades the account of Andrew Hodges's Turing biography (Hodges 1989).

The origination of the first "real" computer is firmly planted in the tradition of the inventions of classical engineering. In 1941 in Germany, Konrad Zuse completed his construction of the first fully operational program-controlled calculating machine ("Z3"). (Almost at the same time the first electronic computers were built in Britain and the USA.) According to Zuse, the Z3 was designed to relieve engineers of tedious tasks of repetitive calculation. Zuse was a construction engineer, naming his company, founded while the second World War was still in progress, "Zuse Ingenieurbüro und Apparatebau, Berlin" (lit. "bureau of engineering and construction of apparatuses"). His autobiography admits a look into the engineering culture which formed the cradle for his invention, and which considers the idea of women as constructors unthinkable — the only exceptions were born out of the necessities of wartime, when male personnel was unavailable. After the war, "Veronika", the only woman who had been part of Zuse's early team, turned to a more "feminine" sphere, becoming an artist.

According to a speech given by the president of the German Patent Office on the occasion of Zuse's 70th birthday, an inventor is graced by assets which may just as well serve to distinguish a "manly hero" of the Wild West:

"The transfer into technical reality of an innovative idea spawned from technological creativity and practical inspiration requires a full-scale effort on the part of the inventor. A thought, once identified as right, must be pursued with utmost consequence. Persistence, readiness for personal sacrifice and a healthy dose of idealism are qualities reported about both famous and less famous inventors... The independent and autonomous mind is additionally marked by an entrepreneurial spirit, courage and a willingness to take risks..." (Zuse 1993, 204)

There is one particularity discerning Zuse's machine from earlier, classical machines. The Z3 is, Zuse argues in his patent application, "mathematics become flesh" (Zuse 1993, 100).

The theoretical construct of the Turing machine — not the actual machine "computer", the hardware — is regarded as *the* basis for computer science to this day.

II. Abstraction as a plan of action — artificial intelligence

After the war, the idea of mathematics given corporeal expression remains alive for Konrad Zuse. Finding himself without access to his former material resources, he designs the concept of a programming language and envisions, as did Alan Turing, an "artificial intelligence". For Zuse this is the continuation and consummation of a notion of abstraction, a programme of nature's domination and subjection by human culture, a programme of the separation of the mind from the restrictions of physicality. In the same way that formalist mathematics assumes that all mathematical statements can be logically derived from a limited number of axioms without reference to an outer reality, it seems conceivable for Turing and Zuse that a machine may be equipped with human intelligence, perhaps even with that of the whole cosmos, which would likewise have no need to refer to the world outside. According to Alan Turing, one should not and would not need to "allow this machine to roam the countryside" (Turing 1987, 97).

Historically, the concept of abstraction has been a gender-related one: Nature, mystic, intuitive, everyday knowledge, knowledge believed to be untouched by abstract principles, was associated with femininity, while male qualities were attibuted to reason, which was considered to be devoid of emotion (Scheich 1993).

Abstraction is thought of as a notion standing in opposition to nature-rooted and physical femininity — but this kind of thinking already prerequires the skill of abstraction and is thus independent from corporal conditions, including that of gender. In the history of mathematics, famous repute has therefore been repeatedly achieved by women who recognized this particular field as a means to surmount the boundaries gender historically set against them.

III. The mechanization of mental labour

Nevertheless, the success of computer science as the consummation of the principle of abstraction is owed to processes running on *real* machines, which in turn are ultimately based on physical processes. Only in this way may calculations be executed which exceed the proficiency of any human being in terms of speed and complexity. In order not to be defeated by this complexity, the new type of engineer must move away from the machines' concrete, material properties, freeing him/herself of the need to consider their physical substructures. The success story of computer science is that of its uncoupling from hardware. It is the story of its programming languages, which have in their designs become less and less aligned with the logical structures of electric circuits, now trying instead to more and more mimick the pathways of human thought.

The success story of the computer ultimately lies in the fact that mental labour, formerly carried out by humans, can now be accomplished by a machine. More and more processes have become susceptible to this treatment, including many which do not immediately come cast in purely mathematical form. Non-numerical applications have increased in importance since a time as early as the 1960s.

Thus computer science emerged as a new kind of combination of engineering science and mathematics(logic). While bearing reference to a physical machine on one hand, on the other it is concerned with the transformation of information, signs carrying meaning for humans, into a form ready to be processed by a machine which knows no meaning. At the same time it must be ensured that the processed information (data) can later be sensibly reintegrated into the flow of human actions.

The ongoing evolution of higher-level programming languages causes the activities of computer scientists to tilt increasingly to the informational side. The situation today has little in common with computer science's time of founding, when it was important to make an already formalized problem "understandable" to a machine, to translate it into a machine language. Now it has become a more and more crucial skill to be able to act within the realities of social systems, and there to identify formalizable segments in "diffuse" fields of problems. Competence in computer science, much more than in classical engineering, requires abilities of orientation and problem recognition within social environments and the creation of (formal) outlines for new forms of organization of work and the spaces we live in.

Today's very successful concept of interactivity conjoins human and machine, the machine's actions eliciting new human responses at very short intervals. This means that data, which is not invested with meaning while processed inside the machine, is continuously being interpreted by humans, becoming information as a consequence. Humans, "users", must verify the processing operations for correctness and relevance in the world of human activity, and, based on the result, initiate further processing steps, in this way "communicating" with the software.

The questioning of the use of classical models derived from the engineering sciences also affects the process of software development and its methods of "software engineering" and "software technology". Less than ever will logical consequence and physical functioning suffice as measures for "good" technology. When the development of applications is involved — which is true for the majority of jobs in computer science —, then a substantial part of the activities of the computer scientist demands an understanding of working and living processes, to the end that users may recognize data processing as meaningful and worthwhile, and become enabled to embed it into their actions. Only when humans can include software into their activities, the process of construction can be called successful.

IV. Mathematical orientation as a means of access to computer science for women

When computer science as a study programme was new at German universities, its low freshwomen quota of approximately five percentmatched the share of women in classical engineering. Was this due to a hardware-dominated view of computers and the tradition of mechanical engineering which governed computer science's public image in its days of inception?

At the end of the 1960s, the influential pioneers of computer science at German universities tried hard to avert an orientation towards the computer as a *machine*: For students of computer science, mathematical qualification was to be a significant part of their education. Logical and abstract thinking became the trademark of the accomplished computer scientist. Curricula were frequently situated inside the mathematical faculties, from where also many professors were enlisted. A still widely accepted rule for the introduction of beginners to computer science postulates that emphasis must be placed on *concepts* of algorithms, programming and data structures, not on their implementation in the syntax of an actual programming language on a specific type of computer.

This philosophy is brought to the point in Edsgar Dijkstra's opening speech to an American debate on computer science: He proposes to free neophyte students in computer science completely from contact with actual computers. His claim is that computer science is an entirely new kind of discipline, which, similar to mathematics, can be exercised in purely formal ways without any experimental approach (Dijkstra 1989).

Perhaps it was precisely this kind of perspective which enabled mathematically interested women to gain access to computer science. Women studying computer science usually had higher than average grades in mathematics at school, contrary to their fellow male students. Opinion polls I conducted in my courses suggest that most female students of computer science would rather have chosen mathematics as an alternative, but hardly any other science or engineering. This is different for male students who tended to alternate between computer science and an engineering or physical science.

In Germany, in a poll of female computer scientists holding a Ph.D., Ulrike Erb found a result which confirms this conjecture: For female computer scientists working in research, the key factor which had opened the way for them had almost exclusively been mathematics or mathematically oriented theoretical computer science.

"Many of my interview partners refer to the mathematical paradigm of computer science in the description of their interests within the discipline. The clarity of formal structures, the exactness of mathematical and logical proceeding ... are positively accentuated." (Erb 1996, 177f)

At the same time however, the interviews of Ulrike Erb show that this does not necessarily reflect a "natural" interest towards computer science in women, and that their needs are also often bypassed: According to the result of Erb's research, many female scientists turn to application-oriented fields after receiving their Ph.D.s. They criticize theoretical computer science for its weak hold on reality, and express a desire to work on practical problems.

Mathematical orientation offered a certain amount of leeway to women feeling comfortable with the employ of logically-structured methodology: The standards used in measuring successful accomplishment in this field are reasonably accessible and lucid, which makes them less vulnerable to arbitrary definition or change due to unconscious gender-specific predjudice.

V. Hands-on experience as a barrier to entry into computer science

The situation changed with the increasing complexity of the machines, the objects at which computer scientific activity ultimately aims and on which it is mounted. At some point, all those abstract concepts had to be implemented and used on real computers — in university curricula already within the first term. However, programming languages gradually became detached from hardware concepts, and moved closer to human ways of thinking. This produced a corresponding increase in the number of "layers" of software which already must exist on a computer before any program implementation can take place, rendering the grounds to be built on more and more opaque. Today, no one can still claim to be able to understand and recount in every detail the configuration of any machine in use for implementation. Nevertheless, any and every "layer" can stand in the way of success. Engagement with the computer, with its hardware, operating system, user interface or "tools", happens by experimental approach between trial and error, between asking for knowledge to gain experience and systematic learning. In this conglomerate of ways of approaching the system called computer, the official curricula of computer science are not very helpful and of little use. They (justifiably) focus on long-term or at least mid-term knowledge which can be systematically taught, the understanding being that the complementing half, experimental knowledge, can not be communicated in lectures or seminars, and will already be out of date before students leave their educational facility.

For female students, the experimental approach and, most of all, the process of gaining experience "by asking" becomes a roadblock. Experimental, hands-on knowledge is exchanged by their male counterparts between lectures, in computer pools, on trains and in pubs. This knowledge also functions as a province for the display of manliness, where the employ of coded language increases with the decrease of solid knowledge, where lies are more common than the admittance of ignorance. It is not necessarily reliable knowledge which is traded in these contexts. A lot depends on how much at home one — man or woman — feels in these circles, how one can articulate oneself, can critically question and assess the information received. Before a woman's questions are taken seriously and answered with reasonable reliability, she must first be acknowledged as a competent partner. She must also be capable of estimating whom and what she can, or cannot, believe.

Learning-by-doing and -by-asking constitute significant parts in the study of computer science. The conditions for the latter are unfavourable for women. Today, at German universities we are creating few opportunities for it and the atmosphere can hardly be called positively disposed towards "asking". Technical personnel feeling responsible for students' questions, not bothered by them, are quite scarce. We still possess little knowledge as to how we might foster a culture of questions and answers about technical matters among women.

VI. The computer as a medium

Apart from an increasing loss of transparency and clarity, there is another, for many perhaps more joyous side to the development of the computer, that of its transformation into a medium. Used in this way, it no longer seems necessary for anyone but a few system specialists to delve very deeply into the bowels of the machine. This may be one of the reasons for the popularity of digital media sciences among women. As before, let us first take a look at history.

Computers have always contained two different potentialities: On one side, they are artefacts in the way of traditional tools and machines used in manufacturing and processing. In the case of the computer though, the target of these activities is not the creation of physical products, but the generation of data and information. On the other side, computers may be counted among the category of technical artefacts used to store, transport and display signals and data. Until recently, the goal in the construction of such devices was to give sender and receiver the impression that data would pass unaltered through their "channel" of communication (avoidance of "noise", "high fidelity"). Thus computers are artefacts in the sense of classical machines and tools as well as classical (technical) media.

Early in the history of computer development, the first, "instrumental" aspect was emphasized. Today, the second aspect, pertaining to the computer as a medium, takes precedence. New interfaces, no longer forcing users to concentrate on the handling of the machine, make it possible to see the computer as a medium, a mediator for the exchange, storage and recall of information. This corresponds with the main characteristics of classical media, which are unobtrusive and unnoticeable in their use.

On the other hand, whenever we use the *computer* for communication or the acquisition of information, we employ a machine not only capable of storing and transmitting signs but also of processing them. Transmission procedures do not necessarily strive for the least possible alteration of the original data while sending it from A to B, but the computer's processing power can *consciously* be used to deconstruct messages at one side and to reconstruct them in new ways at the other.

This is of special interest in so far as programs are being used to act on *content*, on subject matter. For example, to tame the inundation of our workplaces with data from electronic networks, there now exists software to support content-based sorting and evaluation of messages: So-called "agents" will, for instance, sort incoming mail according to specific criteria or display some particular information on the screen at the appropriate time.

VII. Recent concepts of computer science and discussions on its theoretical foundations: Limits of formalism and new ways of access for women?

Our contemporary way of using the computer is described by the term "interactivity". Computer programs are designed to give continuous feedback to their surroundings, their users and their interpretation by users. Human and machine communicate using "contents", displayed and, in part, also created by the computer. This means that the usage of computers has changed: Instead of an understanding of mechanical processes (which can even be counterproductive), now an ability is required to grasp and comprehend contents intuitively, to interpret them, fit them within the current context and communicate about them (possibly with the machine).

These new ways do not stop at *usage* of the computer. The required tasks for the *construction* of software are changing as well. Today, the field of application development is clearly moving towards areas where neither the virtues of engineering nor formal and abstract methods alone can lead to satisfying results.

In their textbook on computer science, Abelson and Sussman call the occupation of programming a means to express ideas (in a formal manner). Programming languages, to their minds, are not simply ways to make a machine do something, but progams "must be written for people to read" (Abelson / Sussman 1989, XV). It is true, however, for the process of software development as a whole (from first analysis to maintenance), that it is identical neither to mathematical formalization nor to classical engineering. Software development does not aim for or end with the construction of a physical machine, whose success is demonstrated by working independently from outside control. In most cases of its application, software development does not hinge exclusively on the question whether a program can be implemented and run. Much more than this, successful software development requires that humans will be able to interpret and accept the signals which appear on the screen: Signals must become signs, making sense to the user in their context and enabling him/her to act appropriately. Peter Bogh Andersen, considering the development of applications, radically enunciates that software developing should be regarded as similar to the writing of a play, the success of which is contingent on the comprehension and approval of an "audience" (Andersen 1990). In practice, computer scientists find themselves increasingly confronted with demands for "social competence". They must be willing and able to respect and learn about the skills and the needs of users.

As a consequence, ideas about a new theoretical foundation of computer science have emerged. Peter Wegner speaks of "Interaction Theory" (Wegner 1997). In this theory, the consideration of the inclusion of computers into social organizations and human activities is regarded as a necessity, or is at least referred to when the old theoretical concept of computer science is shown to be lacking. On the other hand, however, Wegner looks for a (renewed) closing of the new theory in a purely formal way.

These new views of computer science — neither formal nor within the range of traditional engineering — together with the notions of the computer as a medium, are not yet gender-specifically marked and could therefore open new ways of access for women. Social intercourse and competence, the creation, processing and representation of informational and communicational content are not regarded as exclusively male areas.

For that reason, it is important that the contradictory state in which computer science currently finds itself is kept alive. A closure of computer science by a centering on formal or engineering methods (perhaps enhanced by some kind of "social-skills training" outside of computer science itself) can not be an appropriate answer to the contradictions which have come to light.

VIII. Consequences for the issue of gender

The example of computer science clearly shows the untenability of the traditional partitioning of the sciences into exact or discursive, "hard" or "soft", analytic or constructive segments, or into natural, engineering and social sciences. Movements within the edifice of science and between classical dichotomies are of special interest to gender research. Emerging contradictions can be used to dispel traditional associations between gender and certain ways of thinking, methodologies or disciplines. It has thus been noted that even the so-called exact sciences are based on discourse. In the case of computer science this has become especially clear from its inner contradictions as well as its applications.

In recent years, the percentage of female beginner students in Germany has shown a slight, still reluctant, increase. If a proper tendency is to come forth from this, an atmosphere of awakening, as it seemed to become perceptible in the society of the 1970s, when women tried to invade areas of technological influence, then a multitude of factors will have to be considered, not just those concerning the scientific development of computer science.

Nevertheless, I believe that computer science itself must provide a significant contribution: It must be resolved how experimental knowledge about successful implementation on an actual computer can be acquired and combined with a systematic approach. It is unacceptable for computer science to continue to leave this kind of knowledge — which constitutes a decisive professional skill — to the mercy of a more or less opaque, random and gender-specifically influenced learning process.

Additionally, the opening of computer science to non-formal methods, which in practice has long since happened, must become a conscious effort, leading to an inclusion into the discipline's theoretical concept and its public image.

Technology is a social construct, and in reverse, technological knowledge can help to illuminate our understanding of society and the world. The clarification and comprehension of this could make computer science education at schools and universities more interesting and exciting for both sexes.

References

Abelson, Harold; Sussman, Gerald Jay; Sussman, Julie (1989): Structure and Interpretation of Computer Programs. The Massachusetts Institute of Technology, 10.Auflage.

Andersen, Peter Bøgh (1990): A Theory of Computer Semiotics. Semiotic Approaches to Construction and Assessment of Computer Systems. Cambridge University Press.

Dijkstra, Edsger, W. (1989): On the Cruelty of Really Teaching Computing Science. In: Communications of the ACM, Vol.32, 1989, S.1398-1404.

Erb, Ulrike (1996): Frauenperspektiven auf die Informatik. Informatikerinnen im Spannungsfeld zwischen Distanz und Nähe zur Technik. Münster: Westfälisches Dampfboot.

Hodges, Andrew (1989): Alan Turing, Enigma. Berlin, Kammerer & Unverzagt.

Scheich, Elvira (1993): Naturbeherrschung und Weiblichkeit. Pfaffenweiler, Centaurus.

Schelhowe, Heidi (1997): Das Medium aus der Maschine. Zur Metamorphose des Computers. Frankfurt, Campus.

Studien- und Forschungsführer Informatik (1996): Wissenschaftliche Hochschulen und Forschungseinrichtungen, Brauer, Wilfried; Münch, Siegfried (Hg.), Berlin, Heidelberg, New York, Springer (3., völlig überarb. Aufl.).

Turing, Alan M. (1987): On Computable Numbers, With an Application to the Entscheidungsproblem. (Original in: Proceedings of the London Mathematical Society Heft 2, No. 42, 1937). Neu aufgelegt in der deutschen Ausgabe: Über berechenbare Zahlen mit einer Anwendung auf das Entscheidungsproblem. In: Alan Turing, Intelligence Service. in: Dotzler, Bernhard; Kittler, Friedrich (Hg.), Berlin, Brinkmann und Bose 1987, S.19–60.

Wegner, Peter (1997): Why Interaction Is More Powerful Than Algorithms. In: CACM, May 1997, Vol. 40, No.5, S.81-91.

Zuse, Konrad (1993): Der Computer – Mein Lebenswerk. Berlin, 3. Aufl., Springer (1. Aufl. 1984).