Bridging ICT and CS – Educational Standards for Computer Science in Lower Secondary Education

Torsten Brinda
Univ. of Erlangen-Nuremberg
Didactics of Informatics
Martensstr. 3
91058 Erlangen, Germany
brinda@cs.fau.de

Hermann Puhlmann
Leibniz-Gymnasium Altdorf
Fischbacher Str. 23
90518 Altdorf, Germany
puhlmann@ic4life.net

Carsten Schulte
Freie Universität Berlin
Didactics of Informatics
Königin-Luise Str. 24
14195 Berlin, Germany
schulte@inf.fu-berlin.de

ABSTRACT

Recently, the importance of computer science education in secondary schools has been coming more and more into focus. Students' interests and motivation to pursue a career in CS related fields are highly influenced by school. Also beliefs in the nature of CS are influenced, and necessary foundations in knowledge and skills are built. However, a major problem is the gap between information and communication technologies (ICT) and computer science (CS). Often pupils have only experiences in ICT, and therefore develop inadequate beliefs about CS. We propose educational standards for CS in lower high school in order to bridge ICT and CS.

Categories and Subject Descriptors

K3.2 [**Computers & Education**]: Computer and Information Science Education – *computer science education, information systems education*.

General Terms

Measurement, Performance, Experimentation, Human Factors, Standardization.

Keywords

Secondary CS Education, K-12, Didactics of Informatics, CS Ed Research, Educational Standards, Pedagogy, Gender.

1. INTRODUCTION

Looking at students in general we have to admit that computer science has never been popular among all of them. Of all students beginning post high school studies in Germany in 2005, only 7.6% decided to do a major in CS. With a dropout rate of 38%, CS has headed the statistics in Germany for years [15]. The decrease in enrolments of all CS students between 2000 and 2005 adds up to 23.5%. In the United States, CS departments also faced a dramatic decrease in enrolments of even 60% between 2000 and 2004 [28]. This situation is even more dramatic for female students [5], [24], [27]. A great deal of research – mostly in the field of gender – has been done to understand this situation and many different reasons have been given [8]. Good, Estrella and Margolis ([12], pp. 91ff) summarize the following reasons for the under-representation of females in CS at US high schools: few opportunities, limited notion of relevance due to presenting a narrow view of CS, negative teaching climate, accumulated negative experiences, and teaching concepts which fail to provide pathways into CS at postsecondary level. They demand to develop teacher training in CS, and collaboration of university and high school in developing didactical approaches ([12], pp. 110ff). The situation in Germany is comparable [9]. In Germany, coordinated by the authors of the paper at hand, a working committee of the German Informatics Society (Gesellschaft für Informatik – GI) [10] developed educational standards for lower secondary CS education [11] which target the major issues, as they define a change in content, role of teachers and provide guidelines for didactic approaches in CS education. A main issue was to bridge the gap between ICT and CS. The working group consisted of persons from universities and high schools. In the following section we will discuss the crucial role of this gap between ICT and CS in further detail. After that an overview of the standards for CS in lower secondary education will be given and examples provided which show how ICT and CS can be bridged.

2. THE GAP BETWEEN ICT AND CS

Besides the above discussed reasons for low levels of interest and high dropout rates, students' preconceptions about the subject matter of CS are further issues to be considered.

Greening [13] surveyed students of a secondary-level computing course and observed that 92.9% possessed and used a computer at home, but "the majority of students (over 58 percent) were unable to approximate a definition of computer science." (p. 149). Carter mentions in her study that students who never think of majoring in CS lack subject information, and the most part of the students is unable to explain what CS topics are [6]. Beaubouef and Mason state also that "students often have misconceptions about the field of computer science. Many of them take a computer literacy course, do well in it, and believe that computer science is all about word processors, spreadsheets, or web browsers" ([3], p. 103). CS freshmen often believe CS is primarily concerned with using and administrating computers, hardware and programming [19]. Students who are less interested in computers never think of studying this subject, while others who do well in using it believe to be able to cope with CS as well [1].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ITICSE'09, July 6-9, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-381-5/09/07...$5.00.

For postsecondary education Cohoon and Asprey ([7], pp. 149f) conclude that there is a consensus that computing experiences (in a broad definition) have an impact on how students conceptualize CS. The same hypothesis can be found in [4] pp. 50f, [7], p. 171, and in [13], [14], [18], [22]. However, others believe that computing experiences in programming have only minor or no impact (e.g. [20], p. 80). In addition there are the effects of the hidden curriculum. In other words, students' conceptualizations of the field may only in a minor part be shaped by prior experiences and more influenced by the teaching approach and implicit messages about the characteristics of CS. Overall, there are no direct relations between prior computing experiences and participation and success in CS [6]. That is, computing experiences work as a starting point for some students, whereas they are a barrier for others [26].

We think the role of computer science education in lower secondary education is to build on these experiences in computer usage and to develop a more appropriate insight into the discipline and the technology behind its applications. Teachers should take into account prior experiences of their students and be aware of students' preconceptions. As a consequence didactic approaches that bridge the gap between computer usage and computer science are needed. The proposed educational standards for CS in lower secondary education are a means to bridge this gap.

3. EDUCATIONAL STANDARDS

3.1 Overview

The "Principles and Standards for School Informatics – Educational Standards of Informatics in Lower Secondary Education" (translation of the title [11]) of the GI address CS teachers, CS teacher-trainers, and decision makers in the educational administration. They were stimulated by conclusions of the results of the OECD PISA studies [17].

It is the aim to foster a contemporary and professionally substantial CS education in high schools. Nowadays CS is important in more and more areas of everyday life. The IT industry needs qualified employees, and learners need professional orientation to relate CS to their personal environments and also to acquire knowledge to be able to consider subsequent CS-related educational phases on a more realistic perspective. The best way towards this aim is to acquire competences in a CS subject of its own in lower secondary education as early as possible. That is, to teach CS instead of teaching ICT, computer literacy, or "computer driving licences" with focus on computer usage only.

During the development of educational standards for computer science in lower secondary education the working committee of the GI acted on the assumption that CS is a subject of its own, which is taught from year 5 to 10 (ages 11 to 16 years) at an average of one lesson per week. The standards define competences that are to be developed after year 10. Some of these are needed earlier so that various subjects in school can rely on the students' abilities. Therefore, the standards are subdivided into the grade groups "grades 5 to 7" as a foundation for the further academic learning and "grades 8 to 10" as a general CS education to which vocational and higher secondary education can connect to. This arrangement into groups of grades makes different implementations in the different federal countries possible, but it is also the demand that every learner (of every school type) must get the chance to acquire these CS competences at school. In the introductory principles of a good CS education this is stressed in the topics *equal opportunities* and *digital inclusion* (instead of digital divide). In this spirit these standards are opportunity-to-learn standards. By also specifying the competences, which should be acquired after 7th and 10th grade, they are also outcome-oriented. Moreover, they are minimum standards. The examples of those German federal countries, in which CS is a regular subject in lower secondary education, make reaching the competence demands at least on a simple level possible. In 2008, in Germany 3 of 16 federal countries have compulsory CS classes in lower secondary education, the others offer facultative classes.

The standards and their structure have been developed gradually in various workshops with numerous CS teachers and scientists (also from Austria and Switzerland) in a perennial discussion process since 2003. The standards document [11] is structured into two main parts: the *principles* and the *standards*.

The *principles* draw a picture of a desirable CS subject. They specify the conditions under which the demands of the standards should be reached. They deal with equal opportunities, the curriculum, teaching and learning, quality assurance, ICT usage, and the multidisciplinarity of a CS subject. So they are about institutional conditions as well as teaching methods. Both are strongly interrelated in the implicitness that teachers need a profound pre- and in-service training to give good CS lessons now and in future.

The *standards* specify the competences the learners should acquire under the conditions of the CS subject described before. The basic subdivision into content and process standards was taken from the NCTM standards [21], but other structures might also be possible and reasonable (see Figure 1).

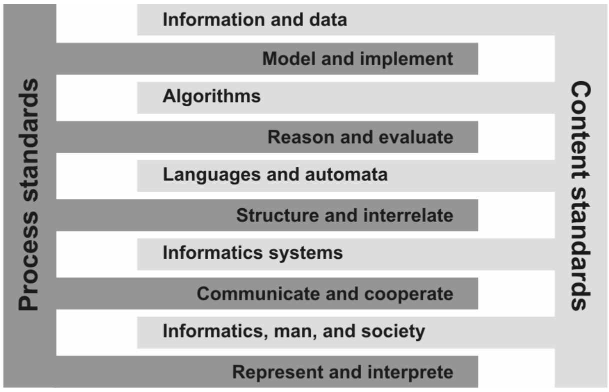

Figure 1. Content/process standards ([11], p. 11)

The content standards characterize the minimum of wanted professional competences. The process standards specify how the learners should be able to deal with the professional content. The content standards though should not be misinterpreted as content check lists to "be just completed" (see Table 1).

Good lessons evolve from stimulating examples, in which content standards are interrelated and teaching methods are chosen in a way the learners can practice actively the handling of the contents, which is specified in the process standards.

Learners of all years should …
Information and Data - … understand the connection between data and information and different forms of representations of data - … understand operations on data, and interpret these with regard to the represented information - … be able to trigger suitable operations on data
Algorithms - … be aware of algorithms to solve problems in different areas - … read and interpret given algorithms - … design and implement algorithms, using fundamental programming concepts - … represent algorithms
Languages and Automata - … make use of formal languages for interaction with ICT and for problem solving - … analyze and model automata
Informatics systems - … understand functionalities and structures of informatics systems - … be able to use informatics systems properly - … be able to explore new informatics systems
Informatics, Man and Society - … be able to state interrelations between informatics systems and their social context - … be aware of being able to make decisions in the use of informatics systems and to adhere to social norms in usage - … be able to react to risks involved in using informatics systems

Table 1. Content competences

To what extend do these content and process standards cover essential competences of secondary CS education? The main goal of a general secondary CS education is to prepare learners as good as possible for life in an "information society", which is significantly affected by the widespread usage of ICT in the private as well as in the professional area.

Every learner should understand the design and the functionality of *informatics systems* on a level that allows using them when solving problems, but he should also be able to understand other systems of the same class more easily. Learning about the design and the functionality of informatics systems must go beyond the usage of the user interface, which may change in the next product version or when using another product. The starting point for a product-independent approach is given by the *representation* of *information* to problems relevant to the learners by *data* in informatics systems of different application classes [16] (e. g. 4.2). So the learners learn to adequately *interpret* the data produced by the systems. Furthermore, they understand that information needs to be represented by the use of certain *languages* to ensure that an informatics system can process them with *automata* and *algorithms*. This opens up an intuitive approach to the *modelling* of the structure and the functionality of informatics systems (e. g. 4.1) and their exemplary *implementation*. This also helps the learners to uncover the possibilities and dangers in principle and to react adequately. They discover and evaluate relevant interrelations between *informatics, man and society*.

All this takes place in good CS lessons which focus on the principles (see above) and stimulate the learners to an adequate *communication* using correct CS terminology, to *structuring*, *reasoning*, *evaluating* and *cooperating* and which finally *interrelate* CS results with those from outside.

3.2 Short Comparison to other Approaches

The standards described here are compared to subsequently selected other approaches.

The idea of *IT fluency* [22] aims at the development of the capability to not only be able to use today's technology in one's own field (which is defined as IT literacy), but also to be able to independently learn and use new technology as it evolves, including algorithmic thinking and problem solving. The idea of IT fluency was proposed as a minimum standard that all college students should achieve by the time they graduate. "Fluent" students would master IT on three orthogonal axes namely concepts, capabilities, and skills. Coming to the specified content and the processes there are big similarities with the minimum standards presented here. But the learners addressed here are significantly younger than those of the IT fluency framework. This must be taken into account by the teaching concepts.

The *ACM model curriculum for K-12 CS* [1] is structured into four levels. Level I provides the "Foundations of Computer Science" to K-8 learners. "Level I […] should provide elementary school students foundational concepts in computer science by integrating basis skills in technology with simple ideas of algorithmic thinking. This can be best accomplished by adding short modules to existing science, mathematics and social science units." ([1], p. 7). The standards proposed here for the grades 5 to 7 go beyond that, because they combine IT literacy (e.g. usage of a word processor) with basic CS concepts (e.g. analyzing the similarities of all word processors by describing the data structures of word processor documents using OOM, see 4.2) in a subject of its own with well trained CS teachers. Previous experiments with the integration of basic ICT and CS issues into other classes did not work, mainly because of the lack of the other subject's teachers basic ICT/CS knowledge. In Level II "Computer Science in the Modern World" a general CS education for all learners of grades 9 or 10 in separate CS classes is described. The German standards describe competences for the grades 8 to 10 and also address general education, but the focus is less on hardware and programming issues and more on modelling aspects (see 4.1).

4. BRIDGES

In this section, we will sketch several possible bridges between ICT and CS. Both examples address learners of the grades 5 to 7 (11 to 13 years old), the first one can be found in the content standard "languages and automata", the second one in "information and data".

4.1 Example 1: Automata

Often, learners find aspects of theoretical computer science to be hard to learn and unrelated to their computer usage. In fact, the subjects' aim is to explore fundamental possibilities and limitations of e.g. certain kinds of automata. In reasoning about them mathematically, a restriction to abstract alphabets is handy, such that automata theory appears as the discipline about words like *aab*, *abba* and *babbba*. If we restrict ourselves to finite state automata and state diagrams, however, the theory is not complex any longer, and, moreover, students can experience its usefulness.

As an example, meet Luise, a 9-year-old girl. She got a new MP3-player in order to be entertained on a long train journey. When she did not know how to operate the device and the badly translated, purely textual manual did not help her, her father gave her the state diagram shown below (Fig. 2, translated into English), showing part of the device's functionality.

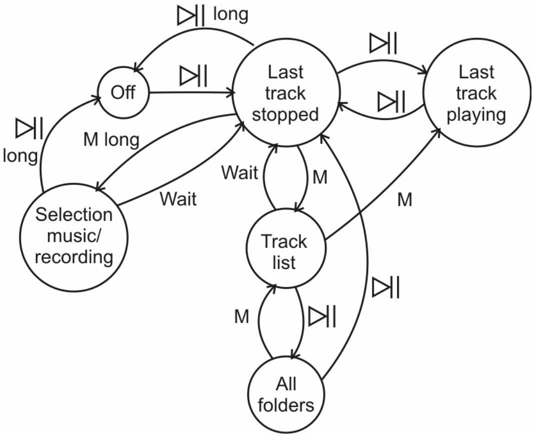

Figure 2. State diagram of a MP3-player

Intuitively, Luise managed to operate the MP3-player. Subsequently, she even added further functionalities of the device to the diagram. In order to do this, she explored the MP3-Player in a trial-and-error fashion. In summary, she "analyzed and modeled automata" as it is required within the educational standards of formal languages and automata. In a classroom context further examples can be added, such as operating a digital alarm clock. The experience with many examples will then lead to discussing the common principles and a theory will evolve.

4.2 Example 2: Word Processing and OOM

Word processing is a core part of ICT and it can be taught with focus on software available in the classroom. The aim is then to get a "word processor driving license" for a specific product, involving the danger of too much specialization. An alternative way of teaching the issue is to highlight common principles of word processors. Still, there will be a product that is used most of the time, but the teacher might make several other pieces of software available, so that the learners find out that they have in fact learned to use a whole class of products, i.e. word processors in general. The advantage of OOM in this context (and thereby of using CS in an ICT context) is that it supplies a language and means of visualization to speak about general and transferable concepts. The effort seems worthwhile e.g. in the next step, when hypertexts are addressed. The students will recognize the old structure of texts augmented with certain features which again can be expressed in the same framework. One common principles is e.g. that font, font size and font color are associated with each character – using object oriented concepts, we can represent this principle as the concept of a class: Class 'Character' has attributes like 'font', 'font size' etc. Other attributes, e.g. alignment, belong to entire paragraphs of a text. This allows to understand the structure of digital text (see figure 3). Learners will analyze which parts of a text can easily and uniformly be manipulated, thus finding characters and paragraphs as classes (with concrete characters and paragraphs, the objects, as their instantiations). They will explore their attributes, and by which means (the methods) they can change them. A class diagram (see Fig. 3, didactically reduced for the beginning (learners of age 12); the black circle at the end of the connecting line represents a one-to-many relationship) can show the structure, including the fact that paragraphs are built out of characters.

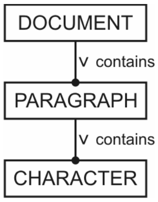

Figure 3. Data structures of a word processor document

5. CONCLUSION AND OUTLOOK

The "Educational Standards for Computer Science in Lower Secondary Education" are a milestone in the development of secondary CS education in the German speaking countries, because for the first time and with various contributions of numerous relevant actors in the field of secondary CS education a consensus about the competences to be acquired by every learner of every school type was achieved. It can be assumed that these standards will significantly influence the next revisions of the federal countries' CS school curricula and thus contribute on the one hand to better comparable learning outcomes in the CS education field and on the other to a more realistic picture of all learners about what CS is all about (see Section 1).

The work on educational standards is not completed though, as the standards mark a stage of development, which especially specifies the desired professional breadth in the view of all contributors. Statements about the professional depth especially with regard to different school types are planned. The further discussion and the usage of the standards in the everyday life at school will also show to what extend the formulation was understandable and comprehensible and to what extend the specified competence demands are reachable in practice. Therefore practicable measuring methods are needed (e. g. [25]). Moreover, there is a strong need for well trained CS teachers, who conduct the whole process actively by describing experiences and discussing critically. Finally, learning and teaching material needs to be developed to stimulate the learners and the teachers for their work.

In this spirit, the work on the "Educational Standards of Informatics" has just begun.

6. REFERENCES

[1] ACM K-12 Task Force Curriculum Committee 2006. A Model Curriculum for K-12 Computer Science. 2nd edition. URL: http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf.

[2] Beaubouef, T. 2003. Why computer science students need language. ACM SIGCSE Bull. 35, 4, 51–54.

[3] Beaubouef, T. and Mason, J. 2005. Why the high attrition rate for computer science students: some thoughts and observations. ACM SIGCSE Bull. 37, 2, 103–106.

[4] Byrne, P. and Lyons, G. 2001. The effect of student attributes on success in programming. In Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2001, ACM Press, New York, 49–52.

[5] Camp, T. 2002. The incredible shrinking pipeline. ACM SIGCSE Bull. 34, 2, 129–134.

[6] Carter, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2006, ACM Press New York, 27-31.

[7] Cohoon, J. M. and Aspray, W. 2006. A Critical Review of the Research on Women's Participation in Postsecondary Computing Education. In Cohoon, J. McGrath and Aspray, W. Ed. Women and Information Technology: Research on Underrepresentation. MIT Press, 137-180.

[8] Cohoon, J. M. and Aspray, W. 2006. Women and Information Technology: Research on Underrepresentation. MIT Press.

[9] Faulstich-Wieland, H. 2004. Mädchen und Naturwissenschaften in der Schule, Expertise für das Landesinstitut für Lehrerbildung und Schulentwicklung (in German). Hamburg.

[10] German Informatics society (GI) 2008. Homepage – Welcome to GI. URL: http://www.gi-ev.de/english/at-a-glance/.

[11] German Informatics society (GI) 2008. Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I (in German). URL: http://www.informatikstandards.de/.

[12] Good, J., Estrella, R., and Margolis, J. 2006. Lost in translation: Gender and High School Computer Science. In Cohoon, J. McGrath and Aspray, W. Ed. Women and Information Technology: Research on Underrepresentation. MIT Press, 89-114.

[13] Greening, T. 1998. Computer science: through the eyes of potential students. In Proceedings of the 3rd Australasian Conference on Computer Science Education, ACSE, ACM Press New York, 145-154.

[14] Guzdial, M. and Soloway, E. 2002. Teaching the Nintendo generation to program. In Commun. ACM 45, 4, 17-21.

[15] Heublein, U., Schmelzer, R., Sommer, D. et. al. 2005. Studienabbruchstudie 2005 (in German). HIS: Hochschul-Informations-System, Hannover, URL: http://www.bmbf.de /pub/studienabbruchstudie_2005.pdf.

[16] Hubwieser, P., Broy, M., and Brauer, W. 1997. A new approach to teaching information technologies: shifting emphasis from technology to information. In Passey, D. and Samways, B. Ed. Information Technology. Supporting change through teacher education. Chapman & Hall, London, 115-121.

[17] Klieme, E., et al. 2004. The Development of National Educational Standards. An Expertise. Federal Ministry of Education and Research. Berlin. http://www.bmbf.de/pub/the_ development_of_national_educationel_standards.pdf.

[18] Ladd, B. C. 2006. The curse of Monkey Island: holding the attention of students weaned on computer games. Journal of Computing Sciences in Colleges 21, 6, 162-174.

[19] Maaß, S. and Wiesner, H. 2006. Programmieren, Mathe und ein bisschen Hardware...Wen lockt dies Bild der Informatik? (in German), Informatik Spektrum 29, 1, 125-132.

[20] Murphy, L., Richards, B., McCauley, R., Morrison, B. B., Westbrook, S., and Fossum, T. 2006. Women catch up: gender differences in learning programming concepts. In Proceedings of the 37th SIGCSE technical symposium on Computer science education, SIGCSE 2006, ACM Press, New York, 17-21.

[21] National Council of Teachers of Mathematics. 2005. Principles and Standards for School Mathematics. 4. print. Reston.

[22] National Research Council Commitee on Information Technology Literacy 1999. Being Fluent with Information Technology, National Academy Press, Washington, DC, URL: http://www.nap.edu/catalog/6482.html.

[23] Romeike, R. and Schwill, A. 2005. Das Studium könnte zu schwierig für mich sein – Langzeitbefragung zur Studienwahl Informatik am Institut für Informatik der Universität Potsdam (in German). Institut für Informatik der Universität Potsdam.

[24] Schinzel, B. 1993. Zur Gleichstellung von Frauen und Männern in der Informatik: Curriculare Vorschläge. Infotech 5, 4, 7-15.

[25] Schlüter, K. and Brinda, T. 2008. From exercise characteristics to competence dimensions exemplified by theoretical computer science in secondary education. In Proceedings of the Joint Open and Working IFIP Conference on ICT and Learning for the Net Generation (LYICT 2008), Kuala Lumpur, Malaysia), URL: http://cs.anu.edu.au/iojs/index.php /ifip/article/viewFile/1012/16.

[26] Schulte, C. and Knobelsdorf, M. 2007. Attitudes towards Computer Science - Computing Experiences as a Starting Point and Barrier to Computer Science. In Proceedings of the 2007 International Workshop on Computing Education Research, ICER 2007, ACM Press, New York, 27-38.

[27] Statistisches Bundesamt Deutschland (Federal Statistical Office Germany) 2005. Hochschulstandort Deutschland 2005. http://www.destatis.de/.

[28] Vegso, J. 2005. Interest in CS as a major drop among incoming freshmen. Computing Research News 17, 3, 126-140.