Journal of Universal Computer Science, vol. 7, no. 5 (2001), 366-378
submitted: 14/4/01, accepted: 5/5/01, appeared: 28/5/01 © Springer Pub. Co.

Glimpses into the Future of Computer Science Education

Gitta Domik
(University of Paderborn, Germany
domik@upb.de)

Abstract: This paper discusses necessary changes to the computer science curriculum at universities for the future. The alterations are grouped into the following five areas: content and body of knowledge; pedagogy; audience; training-on-the-job; and professional skills. The paper argues that extending the scope of knowledge beyond the narrow borders of primary computer science topics ("breadth in computer science") will lay a solid foundation for building the necessary skills for the future work force.

Keywords: education, curriculum, breadth in computer science, women in computer science
Category: K.3

1 Introduction

During the time this paper is being written (February to May 2001), enormous changes in the headlines concerning the IT market have occurred. The future for computer science graduates can be seen as ambiguous: Financial Times, February 2001, records 500.000 IT jobs open in the US, 1.2 Mio IT jobs open in Europe, with an estimated 5 Mio open IT jobs globally by 2005. The share of the IT market in the GNP in Europe is estimated to double from 5% by 2010. In April 2001, IT news turned unpleasant: The Washington Post reports a decrease of 44% in the demand of IT workers in the last year, with a continuing slow-down in IT hiring. USA Today reports that tech spending fell 6.4 percent during the first quarter of 2001 for the first time in 10 years.

One of the most extensive reports to evaluate the software and IT-service market was recently finished for Germany [Stahl et al., 2000]. By December of 2000, 28.000 jobs were claimed to be open at that time in Germany, with 55.000 software development jobs expected to be open within a range of 12 months ("bare minimum of needs"). The job market in these fields was predicted to grow by 120% until 2005.

Many enthusiastic headlines about the IT job market during the past two years have increased the number of students entering their first year of a computer science program at a university. Requests by industry have been met to provide for more IT workers with a solid background within a shorter time frame. Germany saw the start of many "Bachelor Degree" programs for computer science students at universities. This decreased their typical nine semester curriculum to (in most cases) six semesters, with an option to continue the regular program of nine semesters, then finishing as "Diplom-Informatiker/in". The consequences of the new, now pessimistic headlines will still have to be seen. It is the opinion of the author, supported by this year's computer science graduates in the US (May 2001), that potential changes in the hiring policy will affect graduates from education systems at levels other than universities. US computer science graduates found a pleasant job market in spite of the above quoted headlines.

The content of this paper, however, is not driven by the declining or inclining Nasdaq stock market, though economy is always influencing implementation of necessary transformations at universities. After all, universities need funding for any changes and additions and funding is driven by the need as seen by society at large and in particular by government and industry. This paper is about changes pertaining to the curricula of computer science *at universities* as a preparation to the *variety* of jobs open to our computer science graduates. The *amount* of students to teach plays a lesser role in this paper.

2 What Changes Will We Observe

Changes that are necessary to the education in computer science are best grouped into the following five categories: content and body of knowledge; pedagogy; audience; training-on-the-job; and professional skills.

2.1 Change in Content Within an Ever Increasing Body of Knowledge

Curriculum '68, Curriculum '78, and Computing Curricula 1991 have documented some of the past efforts to continuously adapt computer science education to the ever expanding body of knowledge that encompasses computer science. Recently the draft of Computing Curricula 2001 (called "Ironman Draft") has become available on the web [Curricula 2001] as a documentation of the ongoing effort of the "The Joint Task Force on Computing Curricula" including education leaders chosen by the IEEE Computer Society and Association for Computing Machinery (ACM) to make recommendation for computer science education in the new millennium. With reference to the content, the report points out the need for more education in areas where the importance has significantly grown over the past ten years, such as networking, security, object oriented programming, or embedded systems. Shifts in the focus of teaching include the evaluation of existing software solutions versus new software developments, or the increase in the use of multimedia and graphics. Computer science has also become broader – many application areas have grown together with the field of computer science and the curriculum needs to lay a foundation for an understanding of the issues involved. [Curricula 2001] discusses in detail recommendations for various computer science areas and their extent in a computer science curriculum. Table I summarizes their suggestions for area and size of fourteen main themes.

1. Discrete Structures (43 core hours)
2. Programming Fundamentals (54 core hours)
3. Algorithms and Complexity (31 core hours)
4. Programming Languages (6 core hours)
5. Architecture and Organization (36 core hours)
6. Operating Systems (18 core hours)
7. Net-Centric Computing (15 core hours)
8. Human-Computer Interaction (6 core hours)
9. Graphics and Visual Computing (5 core hours)
10. Intelligent Systems (10 core hours)
11. Information Management (10 core hours)
12. Software Engineering (30 core hours)
13. Social and Professional Issues (16 core hours)
14. Computational Science (no core hours)

Table I: Fourteen computer science areas and their extent as recommended by [Curricula 2001]

More important than individual changes to the courses is the recommended teaching structure. The constant increase in the body of knowledge of computer science demands a number of *core units* that are relevant to any computer science undergraduate and therefore required in the curriculum. Additional knowledge is then added on in *elective units*, where units are individual divisions of predefined computer science areas. This will allow to set a “standard” for computer science graduates while at the same time making room for new or applied topics to come into the curriculum.

The main – and complex - effort here is for universities and industry to agree on standards and on accreditations [Curricula 2001], [Clark, 00].

2.2 Change in Pedagogy Through Technology

New technology is changing our teaching and learning styles. Ready access to information and lectures on the Web, sharing curricula across the Internet, joint lecturing across distant locations, distance learning and other teaching and learning modes reach new kinds of students, create new responsibilities for professors and change the teaching and learning environments everlastingly. In April of 2001, the Massachusetts Institute of Technology announced that it will make the materials for nearly all its courses freely available on the Internet over the next ten years. The new program, called MIT OpenCourseWare, is the largest in a series of similar efforts across the world. While these efforts seem to offer “knowledge for everyone”, the much discussed *Digital Divide* is progressing (because of it) at similar speed.

Many of the pedagogical changes have already occurred at the Universities without much effort: seminars are taking advantage of on-line information, lecture notes are being modified and published immediately during the lecture, homework may be submitted from home and discussed with the professor or with peers in an interactive distant mode. While distant learning is currently still reaching the “odd” and not the regular student, universities will need to carry out long-term changes to find their global role in education [Tsichritzis, 99]. If they do not worry about these issues now, their funding might become unstable.

The Digital Divide, meaning the separation of people and organizations having access to the Internet from those who do not, has vast economical influence. Programs to prevent economical disaster to development countries and to certain social structures within well-to-do economies might be hurt by the current down-turn of IT economy. An effort must be made within the education community to build bridges across that rift.

2.3 Change in Audience

Since it seems like a good economical decision to prepare for a job in the areas of information technologies, we see a strong increase in our number of students studying computer science but also a stronger *diversity* in the types of students we host in our computer science programs. Some students bring extended knowledge about areas of computing with them when they enter their first year of study, some bring solely their interest to learn about it. Teaching styles at the University must accommodate all kinds of students, taking maximum advantage of their background and release them with a comparable set of skills. Diversity in skills is particularly noticeable at lectures at entry level. Some universities have started to set up different levels of lab exercises in the beginning year, working towards a more homogeneous group in the second year.

While the level of skills at entry level is diverse, so are the interests of our "new" students. Presently we attract students of a broader variety of interests, having in common only the need for the necessary understanding of underlying computer science fundamentals and technology that they want to use in a diversity of jobs. Many of these students are not only interested in the "core curriculum" of computer science, but in principles of economy, art, sciences or any other area, in which they will seek a job later on, though wanting to rely on a solid knowledge of the technology and its fundamentals on which to base their work and interests.

Last but not least computer science lecturers see an increase of students not majoring in computer science. Demand for *service* to other departments has increased to a point where, together with the increase of their own computer science students, it is often not manageable for the computer science department.

An approach that encompasses students at different levels of interest and entry knowledge is called *breadth first*. This approach gives a holistic view of the field of computer science in a first (series of) introductory lecture(s). The "breadth first" method therefore prepares computer science students for the coming depth of each area covered while leaving students, who are only going to look into specific topics, without total blanks in their knowledge of the whole field. Even students who will not take any more courses in computer science will have received an appreciation of what computer science encompasses. While many educators believe in such an approach, it has not been a popular form of a first year course of computer science, but we will most likely see this approach more often in the future.

These last years have seen the start of many new interdisciplinary courses and projects at the lower, mid and upper level of a curriculum. These are usually in a smaller setting than beginner's lectures, often project or seminar oriented and have shown good results in developing both academic and professional skills. The *breadth approach* therefore needs not end after the first semester, but can be followed throughout a whole curriculum.

2.4 Less Training-for-the-Job

The last turn of the century has seen an immense demand for IT workers. In a time where students are recruited while still at high-school and computer science graduates can almost pick their salaries, people with appropriate skills are put to work as soon as they show up for the job. There is little time to do extensive training-for-the-job [Edwards, 00]. Moreover, many of these IT experts work at start-ups where no one is available to show them the ropes. While the tech market takes a decline at the first and second quarter of 2001, companies are interested in short-term employment, meaning again that training is often eliminated. At the same time work as "free agents" and/or teleworking leaves many of the work force out of the otherwise offered training at the company base.

What students can not gather on the job they need to learn at the university. The demand is therefore high to match students and industry needs early on. This may be done during practice semesters, summer internships, but also by involving practitioners from industry with university courses. Such cooperation *must become part of the curriculum* to guarantee a smooth transition from graduation to the workforce [El-Rewini and Mulder, 97]. It is also important to encourage and train a life-long learning concept while students are still working on their first degree. The most important ingredient in a life-long learning concept is the solid basis to build new understanding on, which the university must provide early on.

2.5 More Need for Professional Skills

Two major changes in the job world of computer scientists drive new needs at the education level. The first change is from programmer level to management level, where we find many of our graduates just a few years after their graduation. The other change is from being the acknowledged expert ("god") in a technological field to being the service (wo)man on a piece of equipment everyone needs and uses. *Management and communication skills, acceptable English in written and oral form, customer orientation, social competence (including ethics)* are the skills that were not foremost demanded of computer science graduates in the 80´s or even 90´s, but are in high demand now and in the future [Lethbridge, 00], [Mahn et al., 99].

Most universities are moving towards including several spots in their curriculum to develop and train these skills, starting during the first year. While training is sometimes done in specially targeted courses, these values and abilities are mostly being trained during courses, seminars and projects with a focus on specific computer science topics. Courses in software engineering lend themselves well to develop skills for working in teams during first semesters and develop skills for project leadership in later semesters. Communication skills are enforced in oral presentations and written reports. While this is not a new concept, giving feedback to the student and evaluating these values and abilities as part of a grade is recent enforcement. In many non-English speaking countries a section of the curriculum, if not all, is being held in English, to prepare for work in the Global Market (and to include foreign students to enhance the internationality of the University). Training the service angle can again be done as part of course work (e.g. usability, human computer interfaces, e-commerce), or as part of a project with local industry, or as part of an interdisciplinary project, where computer science students are building to the demands of other students.

3 Computer Science Expands

3.1 The Demand for "Breadth" in Computer Science

The burden computer science departments currently carry is that

- the demand (and need) for more service to other departments increases at a much faster pace than funding for the department
- both the scope of knowledge to teach our students and their interests are growing and spilling over into neighbouring disciplines
- there is more need to prepare students for self-training ("life-long learning") and development of professional skills

All of these demands can be met by keeping a broad view on the computer science areas throughout the computer science curriculum. To begin with, a "breadth first" approach will improve the capacity of computer science education by teaching a first course to *all* students, computer science students and others alike. Similar, with known prerequisites, students of other disciplines can join with the regular computer science students a number of computer science courses during their curricula. For computer science students, keeping the broad approach will improve their knowledge of applied information technology, and make them understand the various areas of computer science while they will only be able to learn in-depth about a core curriculum and their specially elected courses. For students of other disciplines, taking computer science courses will saturate their understanding of their discipline with relevant computer science knowledge and at the same time give them more marketable job skills. One of the main principles of Computing Curricula 2001 is "*Computing is a broad field that extends well beyond the boundaries of computer science.*" and can be best served by such a broad, interdisciplinary approach.

In the interdisciplinary working of computer science and other students lies the secret to learning many professional skills for the computer science students. Their team members of other disciplines can act as "customers", bringing applied topics to joint projects. Such projects mix application and technical depth with each student bringing valuable knowledge to the project. Team building and communication skills are different and rather reality-like in such mixed environments, with the advantage that there is still guidance provided.

There are more and more universities fusing their computer science curricula with interdisciplinary courses and even generating interdisciplinary curricula around computer science. In most cases, these changes come about through the interest and far-sightedness of individuals fighting an uphill battle with university regulations. While many believe that the demands for a better work force can be met this way, universities themselves are hard to change, especially in a short time frame. What we can see is the change in the social and professional network between educators of various disciplines and industry leaders, who can move faster into the new direction than the university as a whole can [King, 01]. Degree programs in interdisciplinary areas surrounding information technology have arisen, because of the described obstacles usually in form of *initiatives* and *certificates* rather than approved *degrees*, and in all cases (known to the author) have drawn high numbers of students. In the following sections I will summarize some instances to such interdisciplinary courses supporting the notion of a broad computer science curriculum:

3.2 ATLAS at the University of Colorado at Boulder

The Alliance for Technology, Learning, and Society (ATLAS) is a campus-wide academic initiative at the University of Colorado at Boulder (CU) that started in 1997 and integrates information and communication technology into its various curricula. ATLAS reaches out to the K-12 system, penetrates the teaching and learning mode by offering innovative and effective uses of technology in the class room, and spurns and supports new (usually interdisciplinary) courses and research activities. ATLAS is fundamentally interdisciplinary. Its target is to prepare students for a well-informed use of information technology [ATLAS].

Fostered by ATLAS, the certificate-granting multidisciplinary curriculum in Technology, Arts and Media (TAM) has been available to undergraduate students of any discipline (non-technical or technical, humanistic, scientific, or otherwise) since January 2000. By April 2001 it had over 150 full-time students enrolled. Enrollments had tripled in the preceding six months. Students come from almost every discipline, with Communication and Journalism majors having the highest representation. It is interesting to note that TAM courses attract a high amount of women (above 60%), though many of its courses are technical in nature.

TAM naturally has a strong home in the computer science department. Several courses are offered jointly for the TAM program and for computer scientists, two of these are described in more detail below. TAM courses taught by the computer science faculty strive to mix the technical in-depth knowledge of computer science students with background knowledge from students of other disciplines to design and develop new technologies for the future. While intended to broaden the knowledge of computer scientists and training professional skills not offered in other courses, TAM has also been a solution to students switching from computer science to a less technical field (e.g. Fine Arts) while still taking advantage of their technological skills by mastering the TAM certificate on their way to a “Bachelor Degree”. TAM is targeted for a larger role in the computer science curriculum by encouraging more participation by computer science students to broaden their knowledge particularly in the media/design directions.

3.3 Technology for the Community

One of the courses offered within the TAM program, and also open to computer science students in their regular undergraduate degree program, is a course “Technology for the Community”, developed by Professor Liz Jessup (computer science department). For the first course in Spring of 2001, a truly heterogeneous mix of students signed up, where computer science students were in the minority and women were in the majority. The course was advertised to “develop computational products designed to serve the needs of local community service organizations.” In a two-day workshop and together with experts of various fields (e.g. teachers, community leaders, community volunteers) the students formed and discussed several ideas of needs of their community that technology could solve. In the first part of the course students learned methods and tools to develop technology (e.g. data bases, web technology, user interface design) the second part saw the development of two projects to meet needs of the community. One project transformed a paper-based evaluation sheet for a local school into a computer based, and thus reusable, evaluation report. The second project developed an on-line brokerage for offering donations from individuals to appropriate organizations. The students had opportunity to meet with their “customers” throughout the project. During one of the last weeks of the semester students presented their projects at a national conference. Throughout the second part of the course, several lecture times were taken up by guest speakers, linking technology and community needs from their daily work experience. It was a pleasure to personally experience the success of the resulting projects and the intellectual and professional development of the students during this semester.

3.4 Courses as Seeds

The course “Designing the Information Society of the New Millennium” by Professor Gerhard Fischer (computer science department) falls into the same category as the course described above: offered as a TAM course as well as within the regular curriculum for computer science students. The goal is ”to explore how new media will impact learning, designing and collaboration in the information society of the next millennium.” The class, open to both graduate and undergraduate students, was announced as an interdisciplinary course in an attempt to attract participants from a wide range of backgrounds, interests, and experiences. The majority of the 30 students who enrolled during the Spring semester 2000 were undergraduate (67%), male (70%), and from computer science (86%). A new teaching model, the *courses as seeds model*, designed to understand and learn to resolve open-ended and multidisciplinary problems, was implemented for this course [see dePaula, Fischer, and Ostwald, 01]. The summary here is taken from this reference.

The kernel of the *courses as seeds* model is the *seeding, evolutionary growth, and reseeding model* (SER model), where *seeding* describes the creation of the initial state of a system that is intended to evolve, *evolutionary growth* describes the phase of unplanned evolution as the seed is used by the members of a community to do work, and *reseeding* is a deliberate effort to organize, formalize, and generalize knowledge created during the evolutionary growth phase.

Journal of Universal Computer Science, vol. 7, no. 5 (2001), 366-378
submitted: 14/4/01, accepted: 5/5/01, appeared: 28/5/01 © *Springer Pub. Co.*

The *courses as seeds* model lends itself well to interdisciplinary courses in a technological setting. It fosters self-directed learning, active collaboration, and consideration of multiple perspectives. It is a valid form of teaching, when problems do not have *right* answers, and the knowledge to understand and resolve them is changing rapidly, thus requiring an ongoing and evolutionary approach to learning. Educators are *guides* in this process who also do not have all of the relevant knowledge but are learners as well. While the *courses as seeds* model offers an excellent method to follow for "breadth" courses in computer science, it also contains a number of possible pitfalls, which are well described in the named reference.

Examples of final projects over the last two years include "Defeating the digital divide--creating user designers", "LifeStyles - A game about the global ecological impact of your lifestyle", or "Wireless Access Retrieval Protocol for Enabled Devices".

3.5 Augmented Reality as a Bridge Between Technology and Application

At the University of Paderborn the course "Pictures in a Computer" was developed and implemented by the author in 2000/2001. The one semester course was offered only to students with no or little technical background. Its primary goal was to teach methods of digital image processing to these students, allowing them to develop the fundamental understanding necessary to train themselves in the use of any commercial image processing software of the future. Its secondary goal was to knock down barriers to using technology in an everyday setting. The third goal was to support (and observe) a corporation between computer science students and non-technical students.

The first goal, instilling a fundamental knowledge of the methods of digital image processing, was accomplished in a classical class room setting extended by on-line teaching technology. Additionally, once a week students met in a PC lab and performed supervised lab exercises using an interactive programming language suitable for image processing. Two computer science students (graduate level) were hired as student assistants to supervise the lab and help with all assignments. At the end of the semester, students took a final (written) exam over questions of contrast enhancements, filtering techniques, compression techniques, and colour models.

During the first weeks of the semester, examples of virtual and augmented reality (VR and AR) worlds were introduced. Students had the opportunity to wear VR equipment and get the feeling for VR and AR applications. Soon after this introduction, a brainstorming session on "How can technology help to overcome problems in our everyday lives?" was held in the classroom with only the students, two professors (of media sciences and computer science) and the two computer science students present. By mid-semester, the students had to propose possible topics for a final project. They were told that they could rely on technical help from the two computer science students and had time to meet independently of the instructor with the computer science students. From six proposed topics, the lecturer chose "The virtual jungle map", an orientation walk through the University of Paderborn for first year students.

None of the students had worked with AR or VR equipment before this class. Students developed quickly the necessary self-confidence in using the equipment and various software packages to formulate their input into the system, still under the guidance of the two computer science students. The finished project offered help in form of orientation maps throughout the university campus and information on key personnel (e.g. staff and faculty of specific departments) through the AR equipment at appropriate times.

The course consisted of a small amount of students from the media department, including eight female and three male students. Two project teams were formed between these students and the two (graduate, male) computer sciences students. Collaboration between all of these students worked out extremely well, leading to an excellent evaluation of the course by participation students. Therefore, the second and third goal of the course were met as well.

4 Women will Shape the Information Society

4.1 Current Situation

In Germany, the number of women opting for a degree in computer science has been below 10% for most of the ´90s and climbed just recently to 13-15%. At the same time, the number of women finishing a computer science bachelor's degree in the US is quoted by Financial Times in February of 2001 as below 28%. The two universities used as samples throughout this paper, the University of Paderborn, Germany, and the University of Colorado at Boulder, US, show currently 15% and 14 %, respectively, of women entering the computer science program. These numbers stand for many Western European countries and the US with the possibility of a minor deviation. These numbers are lower than the 25% or the 34 %, respectively, Germany or the US have seen in the participation of women in computer science during the ´80s. Possibilities for the decline may be the overemphasis of programming in the introduction of computer science in high school, the misconception over computer science (or for that matter IT) jobs, lack of role models and other reasons. The slight increase during the last years has been brought on by dedicated programs to motivate girls in high school to consider computer science as their choice of university degree. Girls seem to have as much interest in computers as boys during their teenage years, but rather in the computer as a tool to use and not as an item of interest by itself. As a consequence, applied computer science programs in Germany as they have been available over the last years, such as “Computervisualistik”, “Medieninformatik”, or “Bioinformatik” have typically 30% of women or more in their first year. In several countries in Europe, including Scandinavian countries, Spain and Italy, participation of women in computer science and engineering has been more encouraging. This section will particularly address Germany and the US and countries with similarly low numbers in the participation of women in the field of computer science.

4.2 Is There Need to Increase the Amount of Women in Computer Science

In those countries with a significantly low participation of women in computer science programs, governments and industries have recently spent much money to increase that percentage. The outcry for more IT workers and the fear of an economic slow-down if not enough IT workers can be produced has led to the discovery of those *reserves in the work force*. While this might have been the driving force over the past two years, there are other reasons in our economy that speak for a stronger participation of women. For any society it should hold true that its members should be able to participate in shaping, operating and controlling it, and defining its standards and values. In our current society much of this is performed and defined over technology and women have an insignificant input into the process. How can the technological basis for safety and medical care for children and the elderly be (almost) solely controlled by men while the responsibility for these groups lay (almost) solely in the hands of women? Similarly women have a strong impact on health and education in general, and still have (almost) no part in defining the technological basis for either. Our society misses out on new ideas and useful products that would impact its members in a very positive way if we had more women involved in the process. Any process that is singly based on the ideas of one gender but produces for two genders will be too narrow in scope. Therefore it should be a desire for the society to produce more technologically interested and knowledgeable women. The question of how to perform this task is harder to answer.

4.3 How to Include More Women

The interest of women (generally speaking) in any technology related subject is characterized by a broad view of how that technology can be used and how it impacts humans. Pursuing a narrow technological topic into its very depth (such as becoming an expert in Java) is rarely in the interest of young girls, while developing a school newspaper with the use of newest and most up-to-date technology conveying critical insights into life at and outside school may be of great interest. Computer science at high school level must therefore keep its topics broad enough to draw interest from both genders.

Gender separation in high school has produced good results in the past in drawing girls into computer science and engineering. Many of the women now on an advanced career path in the areas of computer science and engineering had sprung from girls-only schools, and had received education in math and sciences among their own gender. With the popularity of girls-only schools dropping, girls were educated in math, sciences and computer science under the same value system boys appreciated in these areas. The consequence of this (among other reasons) was the decline in the number of girls pursuing a university degree in technological areas. To oppose this trend, high schools are again offering gender separated courses. Such settings allow the development of interest in math, sciences and computer science within a value system girls appreciate, *if* well guided. Such a setting also allows women mentors from universities and industries to address girls among themselves to correct any misconception of computer science related jobs or inform about entry level abilities for a university degree in computer science. Universities and businesses also offer special workshops for girls to learn methods and tools of the trade leading to more self-confidence in using and defining technology. It is through such programs that the percentage of women in computer science programs has been increasing again. Similar gender separation has worked well at the university, where the most crucial time for it seems to be the entry level courses, in particular programming courses. Women-only university programs in computer science have also shown to be attractive for the purpose.

It seems that once the notion of *narrow mindedness* of computer science is vanishing in the mind of women, their interest for technology increases. The ATLAS program at the University of Colorado had continuously above 50% women in their programs, at peak times even 68%. Both here described courses “Technology for the Community” and “Pictures in a Computer” hosted a majority of women. Access to girls in high school can therefore be used to break down prejudices and will often lead them into computer science programs as long as the computer science programs at the universities are broad in their view and allow for an inclusion of application in their curriculum.

5 Conclusions

Some aspects of computer science education at the university level will be affected by the economy of technology. The here discussed issues are – at least to a certain extend – independent of the ups and downs of technology stock. It is continually necessary to apply changes to the content of our courses taught at the University in a computer science curriculum, currently these changes are well documented by the [Curricula 2001] Committee. Curricula that support breadth in computer science have the advantage of reducing service load to the computer science department, fitting more knowledge and professional skills into the curriculum, and of increasing the amount of women in computer science.

Acknowledgements

This paper is dedicated to my teacher and mentor Professor Dr. Dr. h.c. Hermann Maurer who taught me more than twenty years ago the fundamental issues of computer science that I can still build upon today and has since inspired me to think about a wide varieties of topics on and off computer science.

I want to thank the University of Paderborn for granting me a sabbatical to take time to pursue the topic of education in computer science. I also want to thank the University of Colorado at Boulder for opening their doors to me to let me look at their education programs. Of particular help to this paper were Professor Gerhard Fischer, Dr. Kathy Garvin-Doxas, Professor and Assoc. Vice Chancellor Bobby Schnabel, and especially Professor Liz Jessup. The most essential help to write this paper was given to me by Ingrid Fischer who understood that a women researcher needed a school for her son, (free) babysitting facilities and plenty of other help to organize work and family in a provisional environment. I wholeheartedly thank all of these people.

References

[ATLAS] http://www.colorado.edu/ATLAS/ ATLAS and TAM at the University of Colorado at Boulder, version May 2001.

[Clark, 00] Clark, D.: "Are too Many Programmers Too Narrowly Trained?", IEEE Computer, June 2000.

[Curricula 2001] http://computer.org/education Ironman Draft, version February 2001.

[dePaula, Fischer, and Ostwald, 01] dePaula, R., Fischer, G., and Ostwald, J.: "Courses as Seeds: Expectations and Realities", Proceedings of The European Conference on Computer-Supported Collaborative Learning 2001 (Euro-CSCL 2001), Maastricht, The Netherlands, March 22-24, 2001, 494-501

[Edwards, 00] Edwards, J.: "Redefining IT Career Paths for the New Millennium", IEEE Computer, January 2000, 16-20.

[El-Rewini and Mulder, 97] El-Rewini, H., Mulder, M.C. (ed): "Keeping Pace with an Information Society", IEEE Computer, November 1997, 46-57.

[King, 01] John L. King: "Evolution of the Information Disciplines", talk at the University of Colorado at Boulder, April 19, 2001.

[Lethbridge 00] Lethbridge T.: "What Knowledge Is Important to a Software Professional?", IEEE Computer, May 2000, 44-50.

[Mahn et al., 99] Mahn, A., Brauer, W., Freytag, J., Hantzschmann, K., Jandach, T., Kysler, N., Schröter, G.: „Empfehlungen der Gesellschaft für Informatik e.V. zur Stärkung der Anwendungsorientierung in Diplom-Studiengängen der Informatik der Universitäten", Informatik Spektrum, Dezember 1999, 444-448.

[Stahl et al., 2000] Stahl, P., Wucher, R., Rombach, D., Hartkopf, S., Kohler, K., Friedewald, M., Kimpeler, S., Zoche, P., Broy, M., Krüger, I.: „Analyse und Evaluation der Softwareentwicklung in Deutschland, Dezember 2000, Eine Studie für das Bundesministerium für Bildung und Forschung, http://www.dlr.de/IT/IV/Studien/evasoft_abschlussbericht.pdf

[Tsichritzis, 99] Tsichritzis, D.: „Reengineering the University", Communications of the ACM, June 1999, Vol. 42, No. 6., 93-99.