6^{th} International Conference on Applied Informatics
Eger, Hungary, January 27–31, 2004.

Trends in Computer Science Curricula at the beginning of the 21^{th} Century

Malgorzata S. Zuberek

Department of Computer Science, Memorial University, Canada
e-mail: malgosia@cs.mun.ca

Abstract

The constantly changing nature of Computer Science discipline affects computer education significantly. It makes the process of shaping and reviewing curricula frequent and demanding. In the past few years we have seen dramatic changes in both computing curricula and accreditation requirements. Pedagogy, as well as teaching infrastructure, has also changed. What is a nature of these changes? What kind of graduates are we trying to produce for the next decade? This paper addresses some of these issues in the context of the new ACM/IEEE recommendations for computing curricula.

Categories and Subject Descriptors: K [Computing milieux]; K.3 [Computers and Education]; K.3.2 Computer and Information Science Education: Computer Science Education, Curriculum

Key Words and Phrases: Curriculum Development, Undergraduate Curricula, Computing Curricula 2001

1. Introduction

Global reality, no doubt, has significantly affected the computing landscape. A decade in a computer related field sounds like a century. Rapid changes in computer technology are shaping job market today. In order to keep pace with the fast growth in computer industry, academic institutions are adjusting curricula frequently. But how often should universities revise their curricula? To what extend educational institutions should concentrate on training graduates for the job market, which is changing so fast? The graduates in the post millennium era are expected to demonstrate, in addition to solid foundations, a broad view of their discipline [6], including versatile skills and professional ethics standards. In addition to depth, more breadth is required. How to ensure additional professional qualities without compromising the academic standards? These issues are currently subject to major concerns and lengthy deliberations in our departmental programme committee of which I have been a member for a prolonged time.

2. Factors that largely influenced Computer Science education

The predominate, but not necessarily independent factors that affected computer science education to a great extent may be summarized as follows:

- Rapid advances in communication technology
- Efforts to keep pace with fast growth of industry
- Shaping education towards industrial needs
- Globalisation and pressure to train for the Internet
- International scope of computing
- Growing role of Web and net-centric software
- New set of curricular guidelines released
- New accreditation criteria implemented
- Competitive job market and the students' expectations
- Changing background of university candidates
- More flexible university admission criteria
- Competition between universities for student enrolments
- Attempts to separate Software Engineering as a discipline

The two factors, which made the strongest impact on the modern curricula, are globalization and shift in the main language paradigm. Since technology has enabled wide world communication, business began to operate globally. A consequence of the global reality is a pressure to educate for the Internet, Web interactive technologies and net-centric computing. The context in which computer professionals function these days has widened significantly [5].

3. Shaping a curriculum

Curriculum design, according to the best educational practise, is a challenging and a time consuming process. A good curriculum needs a vision and a focus. Curriculum should be designed by the most experienced professors, whenever possible. A well-designed curriculum has clearly stated aims, learning outcomes and objectives [3]. A curriculum also needs a time for fine-tuning and assessment. However, by the time a curriculum fully matures, a new set of standards is likely to be published. How important is it to keep the curriculum up-to-date? Curricular issues have always been subject to heated controversy. Advocates of evolutionary approach implement curricular changes by introducing new topics to already full programmes, to the point when further expansion is not possible. On the other extreme, partisans of revolutionary approach present their vanguard proposals, and find rather a few followers.

The latest standard for computing curriculum (CC2001), adopted in December 2001, provides guidelines for shaping modern academic programmes in computer related disciplines for the next decade. The new guidelines act already as a catalyst for revamping the inflated curricula. Not all educational institutions, however, follow the recommendations of educational experts in the area. Some instead are trying to please their industrial partners. Their programmes, to large extent, are composed of 'cocktail' courses, loaded with fancy terms but lacking a substance. Frequently, in a single course unit students are taught everything and nothing, there is no focus, nor depth. How to avoid erosive trends and how to ensure quality? How will the modern education style, driven by globalisation and influenced by powerful computer manufacturers, large software houses and giant publishing companies, affect the academic standards? [7]

4. Emerging trends in Computing Curricula

A comprehensive computing curriculum, as suggested in CC2001, must be "international in scope". One of the guiding principles in CC2001 was to ensure that the computing curriculum provides a sufficient exposure to social, political, legal, and economic aspects of computing. Legal differences between nations arise in the areas of privacy law, civil liberty laws and intellectual property laws. These regulations differ from country to country and discussion of these issues should take place in the global context. Computing professionals are also expected to be aware of many issues in social sciences and business areas in the context of their profession. The modern trends observed in computing curricula, may be summarized as follows:

- Growing tendency to offer flexible programmes
- Reduced core component, larger slate of electives
- Streams/specializations introduced
- Stronger emphasis on mathematical foundations
- Initiating computer studies early in a programme
- Adopting OO paradigm with the objects first approach
- Changing the main language to Java or C#

- Emphasis on human communication skills
- Exposure to social and legal aspects of computing
- Introducing bridging course options
- Unloading course work at low-levels to minimize drop out

Object-oriented languages have been moving into the mainstream of the undergraduate curriculum at an accelerating rate. Java, with its consistent opinion of an excellent language for teaching the object-oriented paradigm, is being widely adopted as the main programming language [2].

Recently many educators have been promoting the notion of teaching about classes and objects first, before students write code. This helps students to adopt the object-oriented paradigm at an early stage and encourages them to focus on the application structure before implementation. It appears that "an effective perception of object technology is best attained by exposing students to the object paradigm from day one" [8]. Students dealing with modern software perceive this paradigm as natural. While this method has clear advantages, it is not easy to realize in practise. Although the OO paradigm is more complex, hence possibly more difficult to learn, pedagogic theory suggest that unlearning is more difficult than learning. Thus, objects first approach is gaining popularity.

5. The curricula new standards and the graduates' profile

The Computing Curricula CC2001 provides a set of guidelines for educating modern and competent computer science professionals. The guidelines focus on knowledge and skills that would match the latest development in computing technology. The document concentrates on computing education as a whole with four specializations identified: Computer Science, Computer Engineering, Software Engineering and Information Systems. Capabilities and skills, which students are expected to develop through a well-integrated, up-to-date and robust curriculum, are grouped into three categories (in the context of computer based systems): cognitive capabilities, practical skills and general skills [1].

Cognitive capabilities include the following:

: Knowledge and understanding of essential facts, concepts, principles and theorems
: Modelling and design of computer based systems
: Analysis and specification of problems and planning strategies for their solution
: Critical evaluation and testing
: Familiarity with methods and tools
: Professional responsibility

Practical skills include:

: Design, implementation and evaluation
: Information management
: Human computer interaction
: Risk assessment
: Operational proficiency

General skills include:

: Communication
: Teamwork
: Numeracy
: Self-management
: Professional development

What are the main characteristics a successful graduate should posses?

According to CC2001 these characteristics may be summarized as follows [1]:

- a high level understanding of the complex nature of computer systems and processes involved in their construction and analysis
- an appreciation of the interplay between theory and practice with full understanding of how theory influences practice
- a recognition of a broad application of abstraction, complexity and evolutionary change to the field of Computer Science
- a significant practical experience through involvement in at least one substantial software project
- a solid foundation, which will allow maintaining the skills as the field evolves.

These characteristics of graduates will not come out as a by-product. These skills must be explicitly included in the learning objectives and then carefully shaped by charismatic teachers.

6. Shift of emphasis in CC2001 when compared with CC1991

Tab. 1: CC2001 knowledge areas and number of core hours for each area [1].

Knowledge Area	**Area Code**	**CC2001 hours**	**Percent**
Discrete Structures	DS	43	18
Programming Fundamentals	PF	38	14
Algorithms and Complexity	AL	31	11
Programming Languages	PL	21	7
Architecture	AR	36	13
Operating Systems	OS	18	6
Human Computer Interaction	HC	8	3
Graphics, Visualization and Multimedia	GV	3	1
Intelligent Systems	IS	10	3
Information Management	IM	10	3
Net-Centric Computing	NC	15	5
Software Engineering	SE	31	11
Computational Science	CN	0	0
Social, Ethical and Professional Issues	SP	16	5
Total		280	100

Tab. 2: The subject areas of Computer Science in CC'91 guidelines [10].

Subject Area	**CCŠ91 hours**	**Percent**
Algorithms and Data Structures	47	17
Architecture	59	21
Artificial Intelligence and Robotics	9	3
Database and Information Retrieval	9	3
Human-Computer Communication	8	3
Numerical and Symbolic Computation	7	2
Operating Systems	31	11
Programming Languages	46	16
Programming in the Small	12	4
Software Methodology	44	16
Social, Ethical and Professional Issues	11	4
Total	283	100

The above summary illustrates that within approximately the size of core component, the emphasis have been shifted towards Discrete Structures and Computer Programming, including the Web.

7. Accreditation requirements

The academic units are encouraged to seek accreditation of their programmes in order to demonstrate that certain academic requirements for professional practice have been met. In the large picture the new accreditation criteria emphasize, beyond traditional qualities, the following educational standards [4].

- Balance between the breadth and the depth of the program
- Restrictive core component for each specialization
- Sufficient coverage of mathematics, discrete mathematics, probability and statistics
- Significant coverage of relevant science, depending on specialization
- Sufficient course work beyond core component
- High level of critical thinking and problem solving skills
- Proficiency in programming, computer networking and hardware
- Proficiency in Web technologies
- Exposure to importance of HCI
- The ability to communicate clearly and consistently both orally and in writing

- Familiarity with social and ethical implications of computing
- General collaborative skills
- Project managerial skills
- Professional responsibility
- Organizational leadership
- Life-long learning perspective

8. Conclusions

Computing has change in many ways that have profound effect on curriculum design and pedagogy. The importance of stating educational objectives for a curriculum and developing a well-balanced curriculum structure is critical. The new curriculum style is recommended to be more flexible and adaptable in order to accommodate future advances. The accreditation requirements are well synchronized with the new curricula recommendations. In order to satisfy the accreditation requirements, many academic institutions will have to follow curricular standards more closely. In the new educational style, the graduates are expected to be better prepared for a quick adaptation to a versatile work environment. Evidently, the nature of universities is changing to more business-like institution style in order to produce well-trained professionals and to cope with permanent financial difficulties. The outcome of this new educational style is yet to be seen.

References

[1] ACM/IEEE Computing Curricula 2001,<http://www.acm.org/sigcse/cc2001>

[2] Culvin F., Justify Java, SIGPLAN, Vol. 33, No 4, 31-4

[3] Diamond R. Designing and Accessing Courses and Curricula, Jossey-Bass Publishers, 1998

[4] Criteria for accrediting Computing Programs, <http://www.abet.org/>

[5] James J., Globalization, Information Technology and Development, St. Martin's Press, NY, 1999

[6] Roberge J. & Carlson C.R., Broadening the Computer Science Curriculum, SIGCSE, Vol.29/1, 320

[7] Niklaus Wirth: Computing Science Education: The Road not Taken, SIGCSE, June 24-28, 2002, 1

[8] Culvin F., Object Imperatives, SIGCSE, Vol. 31/1, 31-36

[9] Lewis J, Myths about Object-Orientation and Its Pedagogy, SIGCSE, Vol.32/1, 245-249

[10] ACM/IEEE Computing Curricula 1991, ACM Press, New York

Postal address

Malgorzata Zuberek
Department of Computer Science
Memorial University
St. John's, NF, Canada A1B 3X5