Deriving Different Computer Science Curricula from a Common Core of Disciplines

F. G. Moraes, A. F. Zorzo, N. L. V. Calazans
{moraes, zorzo, calazans}@inf.pucrs.br
Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil

Abstract: This paper addresses the construction of three different computer science curricula starting from a common core of disciplines. The curricula are named Information Systems, Computer Science and Computer Engineering. The first curriculum targets the formation of professionals intended to use computer as tools defined by the applications they run. The second prepares professionals to develop computer applications ranging from specific systems to basic software. The third is destined to produce professionals able to explore the hardware software interface in detail to implement basic software, such as drivers and operating system kernels.

Keywords: Computer Science Curricula, Computer Engineering Curricula, Information Systems Curricula.

1. INTRODUCTION

The advent of the electronic computer and its popularisation in the last decades as a widespread commodity brought about several changes in well-established academic sectors like Electrical Engineering, Physics, Mathematics and Business Administration departments. The creation of

Computer Science courses emerging from single or joint efforts of such departments was a first change. Presently, besides the stable status acquired by independent Computer Science departments, there is a significant trend to offer degrees in subjects that are intermediate between Computer Science and many other disciplines. These range from Computer Engineering, Systems Analysis and Computational Mathematics to multidisciplinary efforts unlikely to make sense just a few years ago, such as degrees mixing computers and Arts or computers and Law. Even in the least technical degrees, it is necessary to provide the student with a basic set of concepts about hardware, computer programming and computer science theoretical concepts.

This paper addresses the construction of three different computer science curricula starting from a common core of disciplines. The first is an 8-semester curriculum called Information Systems (IS). It targets the formation of professionals intended to use computer as tools defined by the applications they run, e.g. network administrators and Information Technology systems experts. The second is a 9-semester curriculum named Computer Science (CS). It prepares professionals to develop computer applications ranging from specific systems to basic software, with a strong emphasis in computer architecture and organisation. The third is a 10-semester curriculum called Computer Engineering (CE). It is destined to produce professionals that are able to explore the hardware software interface in detail to implement basic software, such as drivers and operating system kernels, specially in the domain of embedded systems, as applied in industrial automation processes.

The paper is organised as follows. Section 2 presents the common core of disciplines. Section 3 explores the individual curricula organisation. Section 4 provides a set of conclusions.

2. COMMON CORE OF DISCIPLINES

During the past years, several Computer Science educators have studied how to apply the basic aspects of computing in undergraduate curricula. Different from other countries, e.g. USA where students have some background in programming before starting their computing undergraduate course [1], in Brazil most students do not have any previous experience in programming. Based on this, it is very important to any proposed common core to consider the social context in which the curricula will be applied to. A second important aspect that has to be considered when applying a core of disciplines in several curricula in the same institution is the flexibility

students have to migrate from one curriculum to another. This aspect is very important, because many students do not have clear idea of which profile is better suited to them at the starting of their undergraduate course.

Based on those two considerations, the common core of disciplines proposed here to support all three curricula is divided into three domains: (*i*) theoretical aspects of computer science; (*ii*) basic programming techniques; (*iii*) computing technology.

Subjects covered in the first domain include algebraic structures, logic and computability theory. Algebraic structures explore the mathematical basis around Boolean algebra, such as set theory, partial orders, lattices, rings and fields. Logic covers mostly propositional calculus and first-order logics. Computability theory studies the most important models underlying computers, i.e. Turing machines and the concepts of decidability, solvability, reducibility and computability. The implementation of these subjects in courses is very similar in the CS and CE curricula, comprising one course for each domain. The IS curriculum compacts the three domains in two courses, due to the curriculum profile requirements.

The second domain includes two theoretical courses and two practical courses on algorithms. The theoretical courses give an overview of programming techniques and data structuring principles, while the practical courses exercise the techniques and principles using one or more imperative languages. Each curriculum has its own choice of languages. These languages are used to cover the two basic programming paradigms: structured and object-oriented programming, in this order. For example, IS and CS curricula employ C and Java languages, while CE uses C and C++ languages.

The last domain covers basic aspects of hardware like digital circuits, computer organisation, computer architecture, software engineering, databases, computer networks and operating systems. All these aspects are the seed for a set of courses in each of the computing curricula. The IS curriculum emphasises software engineering as its mainstream, hence several related courses are derived including databases, integrated systems, applied software engineering and software auditing. On the other hand, the CE curriculum emphasises computer organisation and architecture as its mainstream. Examples of derived courses are VLSI design, microprocessors, digital signal processors, and digital control systems. Finally, the CS curriculum values all of the above aspects, giving the opportunity to students to deepen their knowledge in a subset of aspects by means of elective courses.

The common core discussed here is covered in each curriculum by different sets of courses, as illustrated in Table 1. A slight difference among the sets in the computing technology domain is due to the specific

curriculum profile. For example, the principles of digital hardware are covered in one course in IS, while these are covered in three different courses in CS and CE. The same is applied to software engineering, which is covered in one course in CE and in three courses in CS and IS.

Table 1 - Set of courses of the common core.

	IS	CS	CE
Theoretical Aspects	Introd. to Computing Introduction to logic	Algebraic structures Logic for computing Computability theory	Algebraic structures Logic for computing Computability theory
	120 hours	180 hours	180 hours
Basic Programming	Alg. and Data Struc. A Alg. and Data Struc. B Programming Lab. A Programming Lab. B	Alg. and Data Struc. I Alg. and Data Struc. II Programming Lab. I Programming Lab. II	Alg. and Data Struc. I Alg. and Data Struc. II Programming Lab. I Programming Lab. II
	240 hours	240 hours	240 hours
Computing Technology	Comp. org. and arch. Systems theory Software engineering A Databases A Operating systems Computer networks	Digital circuits Computer organisation Comp. org. lab. System modelling Software engineering I Databases I Operating systems Data commun. and TP	Digital circuits Computer organisation Comp. org. lab. Software engineering Operating systems Data commun. and TP
	360 hours	450 hours	330 hours

Alg. and Data Struc - Algorithms and Data Structures

The above common core is restricted to computing subjects only. Besides this core, each of these curricula also includes several courses on mathematics, statistics, management, sociology and philosophy. This common set of disciplines could be further discussed but it will be overlooked here because of paper space limitations. Table 2 shows the non-computing science courses of the common core of disciplines in the three Computing Science curricula.

3. CURRICULA DESCRIPTION

The Computer Science curriculum is central in the Faculty. It contains very strong aspects in all subjects, i.e. the three domains mentioned before are considered quite relevant in this curriculum. The theoretical aspects domain is extended with formal languages, formal systems, systems modelling, systems evaluation, and systems simulation. The second domain, i.e. basic programming, is extended with three more courses on theoretical and practical aspects of algorithms, and a strong basis on programming

languages. Besides the courses shown in Table 1, the computing technology domain includes major studies in several fields in the computer science, for example, artificial intelligence, computer graphics, distributed systems, computer networks, and network management. One aspect in the Computer Science curriculum is the possibility the students have to elect at least four from a set of twelve disciplines. These disciplines are named special topics on theoretical computer science (2 courses), applied computer science (2 courses), software engineering (2 courses), computer graphics, distributed systems, digital systems (2 courses), databases, and artificial intelligence.

Table 2 - Set of non-computing science courses of the common core.

	IS	**CS**	**CE**
Mathematics	Calculus I Calculus II Probability and Statistics Linear Algebra	Calculus A Calculus B Probability and Statistics Linear Algebra	Calculus I Calculus II Probability and Statistics Linear Systems Analysis
	210 hours	240 hours	240 hours
Management	Entrepreneurship Org. Behaviour	Entrepreneurship Org. Behaviour	Entrepreneurship Org. Behaviour
	120 hours	120 hours	120 hours
Law	Legislation on Informatics	Legislation on Informatics	Legislation on Informatics
	30 hours	30 hours	30 hours
Social	Philosophy I Philosophy II Sociology I Sociology II	Philosophy I Philosophy II Sociology I Sociology II	Philosophy Sociology
	120 hours	120 hours	120 hours

The Information Systems curriculum has a strong emphasis on software engineering aspects of computer science and management. Like all other curricula, the Information Systems curriculum contemplates the common core of disciplines from the theoretical aspects and basic programming domains. The basic programming domain is extended with one more course on programming and another on algorithms. The main derivation in this curriculum is on the computing technology domain. In this domain, several courses are added to the core courses shown in Table 1, including management of software projects, software engineering (2 courses), database systems, computer networks management, integrated systems, distributed systems, special topics on information systems, and software auditing and security. Furthermore, five courses on management are included: management (2 courses), systems integration, organisational behaviour and entrepreneurship.

The Computer Engineering curriculum is a joint effort between Computer Science and Electrical Engineering departments. As such, this is a regular engineering curriculum in many aspects. For example, it includes stronger basis on Mathematics and Physics. The three domains mentioned above are contemplated in distinct ways here. The first two, theoretical aspects and basic programming are limited to the core courses only. However, the third domain is extended to include four large subjects. The first subject is basic software development, comprising operating systems, real time systems, reliability, and parallel and distributed programming. The second subject includes computer networks and telecommunications. The third subject includes the same digital hardware core courses of the Computer Science curriculum, adding courses to address circuit theory, basics on analogue electronics and VLSI systems. The digital hardware courses are explored in more detail in [2]. Finally, the fourth subject is automation, which is the main emphasis of the curriculum. Similar to the CS curriculum, at the end of the course, a CE student can choose three from a set of eight elective courses. This set of three disciplines will specialise the student in one of the following areas: automation, VLSI design, computer networks, embedded systems or control systems.

4. CONCLUSIONS

Common Computing Science curricula is one of the major issues being discussed and tried in the past decades. See, for example, [3] [4] [5] [6], and several other joint efforts around the world, for example the Computing Science reference curriculum of the Brazilian Computer Society [7].

This paper has presented the efforts of educators from the Computing Science Faculty at the Pontifical Catholic University of Rio Grande do Sul in Brazil to implement three Computing Science curricula. The main goal of these curricula was to have a common core of disciplines that could be applied to all of them. This common core has shown to be complete and to contain the major prerequisite subjects to subsequent courses in each curriculum. Another very important aspect is the easiness the students have in moving from one course to another depending on the profile that they believe is the right for them, due to the characteristics of the proposed common core.

Although some disciplines in the three curricula have different names, they currently present the same set of contents. This is due to the different times the curricula were implemented (CC - 1997, SI - 1999, EC – 2002). By the time the SI curriculum was proposed there was no consensus that the

content of the common core disciplines should indeed be the same. This was clear when the EC curriculum was elaborated.

5. ACKNOWLEDGEMENTS

The authors thank educators from the Faculty of Computing Science at PUCRS that have discussed several issues that resulted in the curricula presented in this paper. They also thank professors Luís Lamb and Celso Maciel da Costa (Faculty of Computing Science), and Luís Fernando Pereira, Rubem Dutra Fagundes and Sérgio Haffner (Faculty of Engineering) for their work on the CE curriculum, which raised several questions that are now answered in this paper. A. Zorzo is a researcher financed by the Brazilian agency CNPq (350277/2000-1). F. Moraes and N. Calazans have their research partially financed by CNPq and FAPERGS.

6. REFERENCES

1. Carley, L.R.; Khosla, P.; Unetich, R. Teaching 'Introduction to Electrical and Computer Engineering' in Context. *Proceedings of the IEEE*. 88(1), 8-22, January 2000.
2. Calazans, N.L.V.; Moraes, F.G. "Integrating the Teaching of Computer Organization and Architecture with Digital VLSI Hardware Design Early in Undergraduate Courses". *IEEE Transactions on Education*, v. 44, n. 2, p. 109-119, 2001.
3. Educational Committee of the IEEE Computer Society. A curriculum in computer science and engineering. Publication EHO 119-8, Computer Science of IEEE, January 1977.
4. ACM Curriculum Committee on Computer Science. Curriculum '78: Recommendations for the undergraduate program in computer science. *Communications of the ACM*, 22(3):147-166, March 1979.
5. Allen B. Tucker, et. al. Computing Curricula '91. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991.
6. ACM and IEEE Computing Science Joint Task Force on Computing Curricula. Computing Curricula 2001 – Computer Science: Final Report. December 2001.
7. Nunes, D.; Bigonha, R.; Costa, T.; Setzer, V. Reference Computing Science Curriculum of the Brazilian Computing Society, June 1991. (in Portuguese)

F.G. Moraes is co-ordinator of the Computer Engineering course at PUCRS. He is a member of the committee that implemented the CE curriculum at PUCRS. He has published some papers about how to teach Digital Systems and Computer Architecture in Computing Science courses.

A.F. Zorzo is co-ordinator of the Applied Computing Department at PUCRS, and is involved in the co-ordination of disciplines linked to the three Computing Science curricula. He is a member of the committee that implemented the CE curriculum at PUCRS.

N.L.V. Calazans was member of the committee that implemented the CS and CE curricula at PUCRS. He has published some papers about how to teach Digital Systems and Computer Architecture in Computing Science courses. He is a former co-ordinator of the Post-graduate Programme in Computing Science at PUCRS.