Computer Science Curricula Design for peruvian universities: San Pablo Catholic University case study

Elizabeth Vidal-Duarte[1] and Ernesto Cuadros-Vargas[1,2]

[1] School of Computer Science, San Pablo Catholic University, Av. Salaverry 301, Cercado. Arequipa-Peru e.vidal@ucsp.edu.pe
[2] Peruvian Computer Society, ecuadros@spc.org.pe
WWW home page http://socios.spc.org.pe/ecuadros

Summary. This paper gives the basis, strategies and principles for Computer Science curricula development for peruvian universities. This work is based on CC2001, the latest curricular recommendation of ACM and IEEE-CS. As a case of study this paper proposes the guidelines for Computer Science curricula development at San Pablo Catholic University (Arequipa - Peru).

1 Introduction

Computing has become an extremely broad designation that extends well beyond the boundaries of Computer Science (CS) to encompass more interdependent disciplines such as Computer Engineering (CE) [6], Software Engineering (SE) [3], Information Systems (IS) [5, 4] and Information Technology (IT) [1].

Most of the universities in Peru offer undergraduate programmes in this area using the name "Systems Engineering" (*Ingeniería de Sistemas*) and, in a shorter number, we also have programmes called "Informatic Engineering" (*Ingeniería Informática*). This programmes are basically the same. They try to cover topics from all the five disciplines presented by the Computing Curricula 2001 [2]. Right now, it represents a serious problem for peruvian universities to distinguish the differences between their current curricula and the five disciplines presented by the Computing Curricula 2001. Even more, after 35 years using "Systems Engineering" as the reference, it is extremely difficult to accept that it does not cover the five disciplines.

"Systems Engineering" schools were created using that name because of the strong influence of IBM in the 70's. The peruvian "Informatic Engineering" was created approximately in 1990. It has the word Informatic due to the european influence. The word "Engineering" was inhereted from its predecessor "Systems Engineering" and because it was a term already related with computers. Based on their curriculas, it is clear that all of them are oriented to computing and not to engineering.

Most of the peruvian "Systems Engineering" and "Informatic Engineering" programmes are clearly oriented to both IS [5] and IT [1]. We can explain that approach due to the high presence of professionals coming from other areas specially Industrial Engineering. This kind of professionals helped to create this programme in the 70's but they are still present in a very high percentage.

On the other hand, all professionals which title contains the word "engineer" are grouped at the Peruvian Association for Engineers (*Colegio de Ingenieros del Perú*)[3]. Even considering the wrong use of the word "engineer" for IS and IT professionals, it seems to be a common mistake to call engineer to all the professionals working on computing.

This problem became worse when the *Colegio de Ingenieros del Perú* presented, at the Peruvian Congress, a proposal to create a new law which would say: "All the professionals with a the title containing the word engineer must have a membership at the *Colegio de Ingenieros del Perú* and pay monthly a fee in order to practice". This law is appliable for other areas like Civil Engineers but not for Computing specially because this field is still confuse.

In this context, the San Pablo Catholic University – *Universidad Católica San Pablo* (UCSP) was established in January 1997 in the city of Arequipa, Peru. Along with it was created the Bachelor study of "Informatics Engineering". This new bachelor program, conscious that computing has become the defining technology of our age and its influence has increased dramatically throughout the world over the last decade, has as a goal to establish a new Computer Science curricula. The task of establishing a new CS curricula in a university in a developing country is a challenge. Designing an appropriate curricula is a crucial part of this challenge. The goal of this paper is to present the guidelines to develop a CS curricula for UCSP based on CC2001 [2], the latest curricular recommendation of Association for Computing Machinery (ACM) and IEEE-Computer Society (IEEE-CS).

The rest of this paper is organized as follows. In section 2 we present a short review about CS Curricular Recommendations. In section 3 we analyze the peruvian context and present a proposal based on peruvian universities needs. Finally, conclusions and further work is presented in section 4.

---

[3] http://www.cip.org.pe

2 Review of Computer Science Curricular Recommendations CC2001

2.1 Overview CC2001

The main body of the CC2001 report consists of 13 chapters [2]. For our study only certain chapters have been considered as relevant. Chapter 3 presents technological and cultural changes in CS. These changes will be analyzed according to the reality of Peru. Chapter 5 shows the essential material to anyone obtaining an undergraduate degree in this CS. Chapters 6, 7,8 and 9 give strategies and hints for the organization of the topics in the introductory, intermediate and advanced levels of the curricula, as well as the knowledge and abilities necessary to complete the curricula in CS. Finally, Chapter 13 gives us the strategy, tactics and principles for curricula develop.

2.2 Changes in the Computer Science Discipline

These changes fell into two categories: technological and cultural, each of which have a significant effect on CS education.

Technological changes: Much of the change that affects CS comes from advances in technology. Technological advancement over the past decade has increased the importance of many curricular topics, such as the following: the World Wide Web and its applications, Networking Technologies, particularly those based on TCP/IP, Graphics and Multimedia, Embedded Systems, Relational Databases, Interoperability, Object-Oriented Programming, the use of sophisticated application programmer interfaces (APIs), Human-Computer Interaction, Software Safety, Security and Cryptography and Application Domains.

Cultural changes: Computing education is also affected by changes in the cultural and sociological context in which it occurs. The following changes, have all had an influence on the nature of the educational process: Changes in pedagogy enabled by new technologies. The enormous demand for computing expertise and the vision of large fortunes to be made has attracted many more students to the field. At the same time, the demand from industry has made it harder for most institutions to attract and retain faculty, imposing significant limits on the capacity of those institutions to meet the demand.

Cultural changes in CS discipline according to the reality of Peru are analyzed in Section 3.

2.3 CC2001-CS Body of Knowledge

The CS Body of Knowledge is organized hierarchically into three levels. The highest level of the hierarchy is represented by 14 **areas**. Each one represents a particular disciplinary subfield. The areas are broken down into smaller divisions called **units**, which represent individual thematic modules within an area. Each unit is further subdivided into a set of **topics**, which are the lowest level of the hierarchy. The details of the topics appear in the Appendix A of the Computing Curricula for CS [2].

Also, the CC2001 has defined a minimal core consisting of those units that the corresponding material is essential to anyone obtaining an undergraduate degree in this field. Units that are taught as part of an undergraduate program but which fall outside the core are considered to be elective.

The core is not a complete curricula. Because the core is defined as minimal, it does not, by itself, constitute a complete undergraduate curricula. The core must be supplemented by additional course-work. Every undergraduate program must include additional elective topics from the body of knowledge.

2.4 Overview of the Curricular Models

CC2001 report also defines detailed course implementations and strategies for assembling the individual courses into a complete undergraduate curricula. The courses described in CC2001 are divided into three categories: a) Introductory b) Intermediate and c) Advanced; according to the level at which they occur in the curricula.

Courses designated as introductory are typically entry-level courses offered in the first or second year of a university curricula. Courses listed as intermediate are usually second- or third-year courses and build a foundation for further study in the field. Courses designated as advanced are taken in later years and focus on those topics that require significant preparation in terms of earlier course-work. Although introductory and intermediate courses will certainly concentrate on core material, it is perfectly reasonable to include some elective material even in the earliest courses.

A complete undergraduate curricula consists of an introductory phase to establish basic foundations for further study, an intermediate phase to cover most of the core units in the body of knowledge, and additional advanced courses to round out the curricula. The details of this coverage are outlined below.

Introductory Courses

Most introductory CS courses have focused primarily on the development of programming skills. The CC2001 Task Force offers three implementations of a programming-first model. The programming-first implementations are [2]: Imperative-first, Objects-first and Functional-first.

Peruvian universities should analyze each approach and make decisions about which of the alternatives best address the needs of their students, the academic department, and cities.

Intermediate Courses

The intermediate courses in the curricula are designed to provide a solid foundation that serves as a base for more advanced study of particular topics. CC2001 proposes four implementations for the intermediate level of the curricula: a) Topic-based b) Compressed c) Systems-based and d) Web-based.

As with the introductory courses, peruvian universities should choose an approach to the intermediate level courses. An example of the implementation of each approach is detailed in [2].

Completing the Curriculum

The primary purpose of Sections 2.4 and 2.4 were to outline a variety of approaches for covering the core units in the body of knowledge. As CC2001 emphasizes on several occasions, the computer science core does not in itself constitute a complete curricula. To complete the curricula, peruvian universities must ensure that students have the background knowledge and skills they need to succeed as well as the chance to do advanced work that goes beyond the boundaries of the core. CS students must have a certain level of mathematics techniques and formal mathematical reasoning are integral to most areas of CS. Mathematics provides a language for working with ideas relevant to computer science, specific tools for analysis and verification, and a theoretical framework for understanding important computing ideas. CS programs must take responsibility for ensuring that students get that level on mathematics topics.

2.5 Local Adaptation and Principles for Curriculum Design

The task of designing a CS curricula is a difficult one in part because so much depends on the characteristics of the individual institution and additional factors that would influence curricula design. These factors defined by CC2001 are: a) The type of institution and the expectations for its degree programs b) The range of postgraduate options that students pursue c) The preparation and background of entering students d) The interests and expertise of the faculty

Creating a workable curricula requires finding an appropriate balance among these factors.

2.6 Professional Practice

According to CC2001, the current avenues for teaching additional material on professional practices are:

Senior Capstone courses: These courses may be from a one to a two-semester sequence during the student's last year. Usually, students must work in teams to design and implement projects. The projects may be in-class or they may be for faculty, on-campus, or industrial clients.

Practicum/Intership/Co-op programs: These programs are sponsored by the institution or department to allow students to have the opportunity to work in industry full or part-time before graduation.

Software Engineering course: These courses teach the fundamentals of software engineering and may include a team project.

These current practices in CS education described above will be analyzed in Section 3 of this report according to the reality of Peru.

2.7 CC2001 - Conclusion of Review

Currently, CC2001 can be considered to be a set of recommendations from an international perspective reflecting the current state of the art in CS and requirements for CS education in the rest of the current decade. CC2001 will be a reference model for the curricula development process at UCSP.

In section 2.2 we pointed out how the technical and cultural changes in CS have a profound effect on computer science education, affecting both content and pedagogy. The cultural changes in the computer science discipline must be analyzed according to the reality of Peru.

In section 2.3, we described the body of knowledge of CS. CC2001 has defined a minimal core consisting of those units for which there is a broad consensus that the corresponding material is essential to anyone obtaining an undergraduate degree in Computer Science. These core-units has been included in the UCSP CS curricula.

In section 2.4 it was mentioned three implementations of the introductory curricula. Section 2.4 pointed out four implementations for the intermediate level of the curricula. These implementations were presented as representative models rather than as prescriptive standards. UCSP analyzed each approach for the introductory and intermediate level and make a decision about the implementations to fit its particular characteristics and the needs. Introductory level is **Object Oriented** and the intermediate level will be **Web Based**.

In addition, to complete the curricula UCSP has already ensure that students go beyond the boundaries of the core through the general requirements and the advanced courses pointed out in Section 2.4.

In conclusion, taking CC2001 as reference, we will be able to propose guidelines for CS curricula development at the UCSP

3 CS Curricula Development in Peru

The main purpose of Section 2 was to review CC2001 curricula recommendations for CS undergraduate programmes. In this section we show a brief analysis about Peru professional practices and relevant characteristics of the universities in our country. It was considered as conditional criteria for CS curricula development process at UCSP.

3.1 Analysis of changes in the CS Discipline

To get a better understanding of the current situation of peruvian universities it is necessary to look into some relevant features in our country: a) Peru is largely considered to be a developing country. Probably the most difficult subject is that Peru lacks hight-tech industry. That is why there is not a large computing expertise demand b) Currently, the employment possibilities are rather in the area of Information Systems c) It does not exist a notable development in the industry of software development either. The type of software development is oriented to Small & Medium Enterprises (SMEs) d) The software development oriented to SMEs has not achieved higher levels of the Capability Maturity Model for Software (CMM) e) It does not exist a culture to produces high quality software in Peru.

With the background outlined above, it should be obvious this makes it difficult for Peru to take advantage of the changes in CS discipline we described in Section 2.2.

3.2 Local Adaptation for Curriculum Development in CS

As we described in Section 2.5 the task of designing a CS curricula is a difficult one in part because so much depends on the characteristics of the individual institution and additional factors that would influence curricula design. This section describes these specific factors:

- Most of the undergraduate programs follow the five-year system according to the current peruvian university system
- The academic staff does not have solid research groups in CS areas.
- Most of the students that apply to the undergraduate programs come from public High Schools. The preparation and background of entering students is of a very low level, specially in the area of mathematics. Compared to the european system, there are many topics that are not taught in our High Schools (such as: calculations, binary numbers, graphical solutions of equations, reduction of non-linear laws to linear form, irregular areas, presentation of statistical data, measures of central tendency and dispersion, probability, among others). Since in section 2.4 we pointed out that mathematics has a pervasive role in CS the lack of mathematics described above has to be considered in the designing of the introductory phase of the peruvian CS curricula.

3.3 Professional Practices

For our proposal it is important to deepen in some explanations about the professional practices in Peru in order to understand some possible limitations in the CS curricula development at UCSP:

- As we pointed out in Section 3.1, the type of software development in Peru is oriented to SMEs. Furthermore, Peru lacks of culture to produce the highest quality of software. This not only occurs on the side of the professionals dedicated to software development, but also on the side of the SMEs clients. There is no demanding proofs around quality of software.
- Since we pointed out in section 3.1, Peru lacks of industries, it is difficult for students to achieve the practicum/intership/co-op programs. Currently Peru lacks of a culture of cooperation between University-Public sector or University-Private sector. It does not allow CS students to implement these practices as they are in the developed countries. This point is also useful to observe that the implementation of the Senior Capstone course would be more oriented to in-class projects or they may be for faculty.

3.4 Strategies of Curriculum Development in Peru

General Considerations

In order to understand better our proposal, we have to consider that UCSP is already orienting our School of "Informatic Engineering" to CS. These changes also involves to adapt the name soon. From the study of CC2001, UCSP CS curricula development has considered the following recommendations:

- Introductory and intermediate courses have been mainly concentrated on core topics.
- Some elective material has been included in the earliest courses.
- UCSP CS curricula has included additional elective topics from the body of knowledge.
- UCSP curricula development has considered UCSP characteristics, the academic department expectations, the academic staff expertise, and the background of entering students described in Section 3.2.
- UCSP CS curricula development has considered professional practices in Peru pointed out in Section 3.3.

Implementing Strategies

The introductory courses are oriented to the Object Oriented approach. However, we saw excellent results using the Functional-First approach. Based on these good results we have a first course using Funtional approach and then we continue with Object-first approach.

The intermediate courses are fully oriented to the Web-oriented. It means we choose the Web-based approach for this level of courses.

Another important difference in Peru is that, when an undergraduate student finishes his ten semesters at his university, he becomes a bachelor. These bachelors do not become professionals at all until they present and deffend a thesis at his university. A bachelor can only practice when he approves his thesis. That thesis is not part of his ten semesters. Then a high percentage of our bachelors need more than 3 years to prepare their thesis. Sometimes, they do not prepare one because they do not have any pressure. In order to reduce that time to prepare their thesis we pay more attention to the Capstone Projects. This sequence takes four semesters at the UCSP and the student has to present his thesis in order to approve the last one.

As we need to incentivate the Software Industry we also introduced three courses called "Creation of Technology-Based Enterprises" in the last three semesters. It is because we aim to generate not only good employees but also possible new enterprising professionals.

We are also paying a special attention to the SE by adding five courses. It was made in this way because we think it is a specially weak area in Peru.

The tables below use the following nomenclature: T=Lecture hours (Theory), P=Practice hours, L=Laboratory hours, C=Credits.

First semester						
Code	Discipline	T	P	L	C	Prerequisites
CS101F	Introduction to Programming	2	2	2	4	
CS105	Discretes Structures 1	4	2		5	
CB101	Algebra & Geometry	4	2		5	
Second Semester						
CS100	Introduction to CS	2	2		3	
CS101O	Introduction to OO Programming	2	2	4	5	CS101F
CS106	Discretes Structures 2	4		2	5	CS105
CB102	Mathematical Analysis 1	4	2		5	CB101
Third Semester						
CS102O	Objects & Data Abstraction	2	2	4	5	CS101O
CS107	Discretes Structures 3	4		2	5	CS106
CB103	Mathematical Analysis 2	4	2		5	CB102
CB111	Physics	4		2	5	CB102
Fourth Semester						
CS103O	Algorithms & Data Structures	2	2	2	4	CS102O
CS211	Computing Theory	2	2	2	4	CS107
CS220T	Computer Architecture	2	2	2	4	CS106
CS302	Probability & Statistics	3	2		4	CB103
CB201	Mathematical Analysis 3	3	2		4	CB103
Fifth Semester						
CS290T	Software Engineering 1	2	2	4	5	CS103O
CS210T	Algorithm Design and Analysis	4	2		5	CS103O, CS211
CS225T	Operating Systems	2	2	4	5	CS220T
CS270T	Databases 1	2	2	2	4	CS103O
CS280T	Social and Professional Issues	2			2	CS100
CS250W	Human Computer-Interaction	1		2	2	CS106, CS100

Sixth Semester						
CS390	Software Engineering 2	2	2	2	4	CS290T
CS315	Advanced Data Structures	2	2	2	4	CS210T
CS343	Programming Lenguages	2	2	2	4	CS210T, CS220T
CS271T	Databases 2	2	2	2	4	CS270T
CB202	Mathematics applied on computing	2	2		3	CB103
CS281T	Ethics and computational crime	2			2	CS100, CS280T
Seventh Semester						
CS391	Software Quality	2	2		3	CS290T
CS255	Computer Graphics	2	2	2	4	CS315, CB202
CS260	Computational Logic	2	2	2	4	CS211
CS401	Capstone Project 1	4	2		5	CS290T
CS306	Numeric Analysis	2	2	2	4	CB202
Eigth Semester						
CS392	Software Engineering Topics	2	2	2	4	CS390, CS391
CS261T	Artificial Intelligence	2	2	2	4	CS260
CS240S	Compilers	2	2	2	4	CS210T; CS220T
CS402	Capstone Project 2	4	2		5	CS401, CS390
ET101	Creation of Technology-Based Enterprises 1	2	2		3	CS401
Ninth Semester						
CS360	Artificial Intelligence Topics 1	2	2	2	4	CS261T
CS230W	Net-centric computing 1	2	2	2	4	CS225T
CS370	Databases Topics	2	2	2	4	CS271T
CS403	Capstone Project 3	2	2		3	CS402
ET102	Creation of Technology-Based Enterprises 2	2	2		3	ET101
CS356	Computer Graphics Topics 2	2	2	2	4	CS255
CS393	Formal Methods	2	2	2	4	CS392
Tenth Semester						
CS361	Topics on Artificial Intelligence 2	2	2	2	4	CS360
CS231W	Net-centric computing 2	2	2	2	4	CS230W
CS404	Capstone Project	8	2		9	CS403
ET103	Creation of Technology-Based Enterprises 2	1	2		2	ET102
CS357	Computer Graphics Topics 2	2		2	3	CS356

4 Conclusions and future work

The task of design an appropriate curricula in CS in a developong country like Peru is a challenging task. This paper proposed the guidelines to the CS curricula development for San Pablo Catholic University – *Universidad Católica San Pablo* (UCSP) using CC2001 as a reerence model. Even considering that Computing Curricula was created mainly oriented to the United States, it was appliable to the peruvian context with minimal adaptations. We had to increase the courses oriented to mathematics foundations due to the low level at high schools on this topic.

In Peru, the main problem is cultural because of the existence of extrange name combinations such as "System Engineering" and "Informatic Engineering". This problem is progressively overcome using international references such as the CC2001. It is necessary to increase the number of PhDs in CS in order to improve the quality of education in undergraduate programs.

Right now, the next challenge is to tune this proposal and get all these schools aligned to the international standards.

References

1. ACM and IEEE-CS. Computing Curricula: Information Technology. Technical report, ACM, IEEE-CS, 2005.
2. Carl Chang, Peter J. Denning, James H. Cross II, Gerald Engel, Robert Sloan, Doris Carver, Richard Eckhouse, Willis King, Francis Lau, Susan Mengel, Pradip Srimani, Eric Roberts, Russel Shackelford, Richard Austing, C. Fay Cover, Gordon Davies, Andrew McGettrick, G. Michael Schneider, and Ursula Wolz. Computing curricula 2001 computer science. Technical report, ACM/IEEE, http://www.computer.org/education/cc2001/steelman/cc2001/index.htm, December 2001. Last visited March 2004.
3. Jorge L. Díaz-Herrera and Thomas B. Hilburn. Software engineering: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Technical report, ACM, IEEE, 2004. Last visit June 2004.
4. John T. Gorgone, Gordon B. Davis, Joseph S. Valacich Heikki Topi, David L. Feinstein, and Herbet E. Longenecker Jr. *Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (IS2002)*. ACM, AIS, AITP, 2002.
5. John T. Gorgone, Paul Gray, David Feinstein, George M. Kasper, Jerry N Luftman, Edward A. Stohr, Joseph S. Valacich, and Rolf Wigand. Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems. *Communications of the Association for Information Systems (CAIS))*, 3(1), Enero 2002.
6. David Soldan, James Aylor, Alan Clements, Gerald Engel, Ron Hoelzeman, Esther A. Hughes, Joseph L.A. Hughes, John Impagliazzo, Richard C. Jaeger, Robert Klenke, Douglas A. Lyon, Andrew McGettrick, Victor P. Nelson, Daniel J. Neebel, Ivor Page, Gregory D. Peterson, N. Ranganathan, Robert Sloan, Pradip K. Srimani, Mitchell D. Theys, Wayne Wolf, and Murali Varanasi. Computer Engineering: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. Technical report, ACM, IEEE-CS, 2004. Last visited June 2004.