Teaching Human Computer Interaction: First Experiences

Marta E. Calderón
Universidad de Costa Rica, Escuela de Ciencias de la Computación e Informática
San Pedro, Costa Rica
marta.calderon@ecci.ucr.ac.cr

Abstract

Human-computer interaction is a very recent discipline at the Universidad de Costa Rica. In this paper we present the experiences of the first academic year the first courses about human-computer interaction, an undergraduate course and a Masters course, were designed and taught. The HCI course introduction strategy consisted of two steps: 1) to initiate a dedicated undergraduate course during the first term, and 2) to initiate a dedicated Masters course during the second term, simultaneously taught with the undergraduate course. Both courses share the outline. However, due to differences among undergraduate and graduate students and among undergraduate and Masters courses, evaluation methodology differences were implemented, resulting in more assignments and a higher exigency level for graduate students. Work in the classroom is different for each of the courses, because graduate students can build their own knowledge based on their previous working experience and on the exchange of ideas with other students. In both undergraduate and Masters courses, emphasis is set on practice supported by theory.

Keywords: human-computer interaction, teaching human-computer interaction

1. INTRODUCTION

Human-computer interaction (HCI) is considered by the Association for Computing Machinery (ACM) one of the fourteen fundamental areas that represent the body of knowledge of computer science [6] . The Escuela de Ciencias de la Computación e Informática (ECCI) of the Universidad de Costa Rica (UCR) has not included a required course on this area on its bachelor degree program. However, the ECCI has considered necessary to teach an elective course covering this area, because of the importance that HCI has taken and the growth that the software development industry has experienced in Costa Rica. On the other had, the Masters Program on Computer Sciences, associated to the ECCI, has created an elective course on HCI. The undergraduate and graduate courses are relatively new and are taught by the author. It is then possible to compare both of them in order to reveal methodological differences that can arise when teaching courses on the same topic at different academic levels.

In this paper we describe our experience on designing and teaching courses on HCI during the two academic semesters when the first introductory HCI courses were introduced, showing methodological differences due to differences on the characteristics of the students and other factors.

The structure of this paper is as follows. Section 2 describes the context of the Bachelor and Masters programs. Section 3 presents the justification for teaching HCI within the two programs and the strategy followed. Section 4 describes the characteristics of undergraduate and graduate students, showing evidence of differences between the two groups. Section 5 presents the curricular design of both HCI courses. Section 6 describes and compares evaluation schemata of both courses. Section 7 presents the course schedule. Section 8 describes how lessons are developed. Section 9 describes the results of teaching the HCI courses over two semesters. Section 10 and Section 11 present future work and conclusions, respectively.

2. CONTEXT OF THE COMPUTER SCIENCE PROGRAMS

The faculty members of the ECCI at the UCR reviewed and updated the Bachelor in Computer Science program in 1999. This program, with some few changes, is still in use. This undergraduate program is a four-year program which includes 41 courses, reaching a total of 139 credits. Credits are a measurement unit of the student academic activity within public universities in Costa Rica [5]. This unit was defined in 1997 with the purpose of unifying the credit definition within the superior education system of Costa Rica. Specifically at the UCR, one credit represents three hours of weekly work during fifteen weeks, applied to an activity supervised, evaluated and approved by a professor [4].

Thirty seven (37) courses out of the 41 courses within the undergraduate program are required and 4 are elective. Academic semesters are 16 weeks long. Elective courses allow students to take courses oriented towards their professional interests. These are 4-credit courses which provide flexibility in order to maintain an updated program through special topics courses. The course on HCI has been taught twice since the second academic semester of 2006 as a special topics course.

The Masters Program on Computer Sciences at the UCR started in 1995. In the program all courses are elective. Two options are available: academic (thesis-option) and professional (non-thesis option). In the professional option, research is oriented towards practical application of topics covered in courses. In the academic option, research is mostly oriented to the creation of new knowledge [5].

Courses in the Masters program were structured as two paired sub-courses: a four-credit theory course and a two-credit laboratory course. Theory courses are dedicated to the theory and are developed during lecturing hours. Laboratory courses are dedicated to applied research, in which students develop a practical project. Students in the professional option have to register in both the theory and the laboratory courses, whereas students in the academic option only take the theory course. Within this context, the author designed a Masters course on HCI and started to teach it during the first academic semester of 2007.

A non-academic difference between Masters courses and undergraduate courses is that Masters course fees are much higher, not only due to the higher number of credits, but all because credits are more expensive.

3. JUSTIFICATION AND STRATEGY FOR A HCI COURSE

According to [6, p. 5], human-computer interaction is "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." The area of HCI has experimented an enormous growth during the last ten years. However, not all universities include a course on HCI in their programs [10]. Even more, some universities include HCI as part of programming courses [10]. This happens even when organizations such as ACM and the Institute of Electric and Electronic Engineers (IEEE) have defined the importance of teaching about HCI.

In 1991, ACM and IEEE defined nine areas which comprise computer science, one on them being HCI [14]. In 2001, ACM/IEEE published the final report of Computer Curricula 2001 project [2], in which fourteen fundamental areas representing the body of knowledge of computer science are identified. One of the fourteen areas is HCI.

On the other hand, the ACM Special Interest Group on Computer-Human Interaction (ACM/SIGCHI) is an organization formed by people working on the tasks of design, evaluation, implementation, and study of interactive computer systems to be used by human beings [6]. This group and other HCI specialist organizations prepared a document on guides and recommendations on HCI education and published the "*Curricula for Human-Computer Interaction.*"

The documents published by ACM and IEEE emphasize on the importance of including the HCI area in computer science programs. However, the undergraduate programs of the three public universities in Costa Rica teaching computer science or any related major do not include any course on HCI [7, 8, 15]. Hence, the ECCI has considered it is very important to be pioneer in this field in the country, offering a basic course to create a new generation of software developers aware of the role of human beings in the success of any software application and the importance of their participation within the software development process.

Five strategies to introduce the topic of HCI into a computer science curriculum have been proposed [3]: 1) to cover HCI topics in required and elective courses, for example, programming and software engineering courses, 2) to initiate with a course exclusively dedicated to HCI (for example, a Masters course or an elective undergraduate course), 3) when the first dedicated course proposed in strategy 2 has consolidated, to initiate a dedicated course at the other level, 4) to simultaneously initiate HCI courses at the Bachelor and Masters levels, and 5) to offer several elective courses on HCI. At the ECCI, we decided to follow a combination of strategies 2 and 3, starting with a dedicated undergraduate course, and a semester later teaching a dedicated Masters course. This allowed the professor to become familiar with the HCI topic and improve the course before it is taught at Masters level, especially important since, in general, graduate students expect professors to have deep knowledge of the topic. Additionally, [3] also recommends to have two independent courses, one for undergraduate and another for Masters.

4. CHARACTERISTICS OF STUDENTS

Around 90% of undergraduate students registered in the HCI course are seniors with full time dedication to study, 24 years old on average, taking three or four courses simultaneously, with no professional experience, and with basic knowledge of English. Their lack of professional experience makes them not to highly value the importance of users for the success of a software system. Undergraduate students know their classmates and have previously worked together. They like teamwork and can easily find classmates to form a working team. Most of them already took Software Engineering I and are currently taking Software Engineering II. Topics covered on Software Engineering I include software project planning, cost estimation, requirement analysis, and high level design. Topics covered on Software Engineering II include detailed design, implementation, testing, and maintenance [9]. HCI topics are not explicitly covered in any software engineering course.

On the other hand, graduate students who take the HCI course have two or three years of working experience. Most of them work in software development related tasks. They have a full time job and take one or two Masters courses.

Their job has higher priority than their studies. Masters students are 26 years old on average. They practice English in their working places and frequently travel abroad, which make them miss three or four lessons per semester. Additionally, students do not know most of their classmates, since they belong to different college generations or obtained their undergraduate degrees at any other university. For all these reasons, graduate students find it hard to form a team to work with and prefer individual work.

Despite the difference on average ages of the two student groups is not very significant – only 2 years-, there are notable differences in the behavior of both groups. For example, undergraduate students are quiet in class and seldom ask any question, whereas graduate students share their personal experiences with the group and ask many questions, which promotes more discussion and opinion exchange.

Graduate students have higher expectations about the course they take and are more critical, due to their higher professional maturity and higher Masters course fees.

The author has characterized both groups of students after nine years of teaching experience. This information was very valuable when designing the curriculum, the methodology, the evaluation schema, and the class dynamics of the two HCI courses.

5. Course Curricular Design

Topics covered in a course are a key aspect in the learning process. Graduate and undergraduate courses on HCI are designed as introductory and the learning objectives, in terms of what students are able to do after having taken the course, are as follows:

1. Identify the human factors which are pertinent when designing the human-computer interaction of any interactive software system.
2. Design and prototype the human-computer interaction of an interactive software system, in such a way that user needs are satisfied in an effective and efficient manner.
3. Conduct an usability evaluation of a human-computer interface.

Because graduate courses are divided into two sub-courses, the corresponding laboratory course has its own learning objectives:

1. Conduct usability and accesibility requirement analysis and engineering.
2. Develop paper prototypes and storyboards describing the prototype behavior.
3. Analyze tasks executed by users from a usability and accesibility point of view.
4. Design a software prototype based on usability and accesibility requirements and task analysis.
5. Improve an interface based on evaluation results.

Neither graduate students nor undergraduate students have previously taken a course on HCI. Hence, we decided to cover the same topics on both courses, giving different emphasis on practice, research and theory.

HCI is a multidisciplinary area. Apart from knowledge on software engineering, topics such as human factors, sociological and anthropological aspects, and ergonomics, among others, are required [1]. Both courses were designed based on contents of a book on HCI published by the Association of Human-Computer Interaction (AIPO for its name in Spanish, Asociación de la Interacción Persona Ordenador) [1]. This book presents the topics suggested by ACM/SIGCHI [6] in a complete and easy-to-understand way. Most books on HCI have been written in English. Having a text book originally written in Spanish is advantageous for students, since this is their mother language. Additional didactic materials are necessary. Due to their character of introductory courses, both the undergraduate and the graduate courses cover the following topics:

1. Human factors: human skills and limitations, cognitive process
2. Metaphors, styles and paradigms: use of familiar objects, forms of interaction between a person and computer
3. Usability evaluation
4. Human-computer interaction design: prototypes and task analysis
5. Interaction devices
6. Accesibility
7. Software product internacionalization
8. Standards and guides for the design of human-computer interaction

6. EVALUATION SCHEMATA

A notable difference between the undergraduate course and the graduate course presented in this paper is the evaluation schema. This difference is consequence of the differences between the two student groups, such as working experience and time availability, and the number of credits assigned to the courses (4 for undergraduate and 6 for graduate). However, both evaluation schemata make emphasis on practice.

As shown on Table 1, there are common elements in both evaluation schemata, such as the presentation of design patterns, but some significant differences exist too. For example, homeworks represent 40% of the final grade of the undergraduate course, whereas only 10% of the graduate course. At the undergraduate level, homeworks are an important evaluation instruments, since they give the professor the opportunity of assigning relatively long research, reading, and practical homeworks, necessary to complement concepts presented in the classroom. At the graduate level, homeworks are evaluation instruments too, but because most students have a full time job, it is better not to overload them with many long individual assignments. In this case, homeworks are short and focused on very specific topics. Research is then promoted making students write a research paper, based on a broad literature review, in which a new idea is presented. Students are expected to review at least twelve different bibliographical references, such as books and journal papers, in order to support what they assert on their research papers.

Table 1: Evaluation Schema

Undergraduate Course		Masters Course	
		Theory course:	
Presentation of a design pattern	5%	Presentation of a design pattern	5%
Heuristic evaluation of an interface	20%	Heuristic evaluation of an interface	25%
Design of an interface	25%	Presentation of a journal paper	5%
• Requirement analysis		Research paper	45%
• Development of a paper prototype		Presentation of the research paper.	10%
• Development of a *storyboard*		Homeworks	10%
• Task analysis			
• Development of a software prototype		**Laboratory course (practice)**	
• Cognitive walkthrough evaluation		Design of an interface:	
Presentation of the interface defined	10%	*Part 1*	50%
Homeworks	40%	• Requirement analysis	
		• Development of scenarios	
		• Focus group evaluation	
		• Development of a software prototype	
		• Development of a storyboard	
		Part 2	50%
		• Thinking aloud evaluation of the paper prototype	
		• Task analysis	
		• Development of a software prototype	
		• Cognitive walkthrough evaluation	
		• Performance measurement evaluation	

More details about some of the evaluation instruments shown on Table 1 are described in the following sections.

6.1 Design of an Interface

The assignment of designing an interface is the most important evaluation instrument, since it allows students to put into practice the theory. For most of them, this is the first time in their lives they seriously think on the impact of their interaction design decisions on users' performance. Design activities are followed by usability evaluation activities. In spite of being a common element on both evaluation schemata, graduate students are expected to produce a result with a higher degree of usability than undergraduate students. The professional experience of graduate students has given them the opportunity of getting in contact with software system users and their daily problems, which is an advantage when compared with undergraduate students.

Even when it would be possible to allow students to choose the software system they will design, we suggest proposing several software systems to be developed. All suggested systems should have the same degree of difficulty and complexity and have a relatively limited functionality, so that students set most of their effort on designing interaction instead of understanding functionality. Additionally, when software systems to be developed are mobile applications or require designing specialized hardware devices, students are challenged to research more.

6.2 Heuristic Evaluation

The heuristic evaluation is a common element on both evaluation schemata. The heuristic evaluation is a group assignment, with three or four students participating on each group. Students have to plan and conduct a heuristic evaluation of a software system. According to [12], conducting a heuristic evaluation of a badly designed software system makes students become aware of the importance of investing time in designing the interaction Another suggestion for choosing the software system to be evaluated is to select one not complying with the most frequently used standards and guides. Conducting a heuristic evaluation is important because students experience what users feel when using software systems. Most software developers never use the software they create and do not understand what users complain about.

Each group should prepare a short plan before conducting the evaluation. This plan must contemplate a description of the heuristic evaluation methodology, a profile of the software system users, characteristics of the evaluators (the students) and the evaluation environment (hardware and software), and templates to be used. If possible, there should be several different evaluation environments. For example, if the software system is a Web application, then different operating systems and browsers (e.g. Opera and Internet Explorer) should be used.

Conducting a heuristic evaluation should include individual and group activities. The professor can provide a guide on conducting the heuristic evaluation to help students understand what they are supposed to do. As an example, we use the following guide:

- Individual activity: each student in the group evaluates the software system and identifies problems and positive findings. For future reporting, each problem and positive finding is supported by a screenshot.
- Group activity: based on individual evaluations, the group prepares a list of positive findings.
- Group activity: based on individual evaluations, the group prepares a combined group list of problems, highlighting which problems were found only by one student and which were specific for one particular browser.
- Individual activity: each student takes the combined group list of problems and assigns severity ratings to each problem (1 to 4 integer scale). The scale used for rating is as follows:
  [1] Negligible problem: it does not have to be fixed unless there is enough time.
  [2] Low importance problem: fixing it is not very important.
  [3] Severe problem: fixing it is important.
  [4] Catastrophic problem: it must be fixed.
  Any other scale may be used, as far as students describe it on their final report.
- Group activity: based on individual severity ratings, the group calculates the average severity ratings for each problem and sorts problems into descending order (on average severity).

The first time the HCI course was taught, we allowed students to choose the software system to be evaluated, whereas the second time we decided to choose it ourselves, in order to guarantee similar software system complexity degree and design quality for all students. We also provided the format to be followed to prepare the final evaluation report. The structure of final evaluation report used by the authors is as follows:

1. Software system functionality. Students have to describe the software system functionality based on user manuals and the evaluation process itself.
2. Evaluated points and their importance. Students identify the set of heuristic principles used to conduct the evaluation and explain why they are important.
3. Positive findings. Every software system interface has positive aspects. Students must find at least three.
4. Problems found in the software system. Describe the severity rating scale used. Show individual and average severity ratings.
5. Problem analysis. Select the five most severe problems, analyze them, and discuss how they could be fixed. Use bibliographical references to support suggestions.

6.3 Presentation of a Design Pattern

Master students present a journal paper. In order to motivate them to research more on a specific topic, students must look for additional bibliographical materials to complement the paper they present. The schema used to evaluate presentations covers aspects such as presentation organization, audiovisual material, and personal performance. Effective oral communication is a very important aspect in the area of HCI. The evaluation schema highlights aspects which are important to develop communication skills (Figure 1).

a. Organization – 40 points
  a. Introduction (8 points)
    i. Identifies the topic. (1-4)
    ii. Provides an overview of the contents of the presentation. (1-4)
  b. Topic development (20 points)
    i. Defines the meaning of what is being said. (1-5)
    ii. Explains concepts which the audience should know in order to totally understand the presentation. (1-5)
    iii. Describes the topic showing deep knowledge and self confidence. (1-5)
    iv. Highlights the most important aspects of the topic. (1-5)
  c. Expresses her own opinion about the ideas presented on the paper. (5 points)
  d. Conclusions: concludes with comments relating all that has been previously said. (7 points)

b. Audiovisual Material – 25 points
  a. Use of audiovisual material (15 points)
    i. Slides not overloaded with text (1-7)
    ii. Ortography (1-8)
  b. Use of at least one external bibliographical source per person (10 points)
    i. Source reputation – e.g.., videos, book, journal paper, interview. (1-5)
    ii. Source citation (1-5)

c. Personal Performance – 35 points
  a. Speech time: between 20 and 30 minutes per person in the group (5 points)
  b. Punctuality: 1 to 5 minutes late: 4; 6 to 10 minutes late: 3; 11 to 20 minute late: 2; 21 to 29 minutes late: 1; 30 minutes late or more: 0. (5 points)
  c. Presentation flow (10 points)
    i. Appropriate hand gestures and visual contact (1-5)
    ii. Body movement (1-5)
  d. Language (15 points)
    i. Pronunciation and entonation (1-5)
    ii. Volume (1-5)
    iii. Avoidance of jargon and words which might cause confusion or offend the audience (1-5)

Figure 1. Schema for Evaluating Paper Presentations

6.4 Research Paper

When the undergraduate course evaluation schema is compared with the graduate theory course evaluation schema (Table 1), it can be noticed that a very important difference is that more emphasis on research is set on the graduate course. In fact, at the Masters level, all students write a technical paper, which represents 45% of the final grade of the theory course. Students in the professional option have the possibility of writing the research paper on a topic related to the development of the assignment of design of an interface, such as how design decisions where taken and how problems were solved. However, they can also choose a topic on which they have a special interest. Students in the academic option may choose a research topic related to their thesis research.

7. SCHEDULE

Table 2 shows the proposed schedule for the graduate course, with details about topics covered weekly in class and activities. The distribution of topics is the same for the undergraduate course, but activities such as research paper presentations are not considered and assignment due dates slightly vary. In general, one topic is covered in one week. However, due to their importance or complexity, some topics such as human factors, design and prototypes, accesibility, and graphical design are developed in two weeks.

Table 2: Schedule of Topics and Activities for the Graduate Course

Week	Topic	Activities, presentations, and assignment due dates
1	Introduction / Human factors	
2	Human factors	
3	Metaphors	
4	Styles and paradigms	Pattern presentations
5	Interface evaluations	Evaluation technique presentations Presentation: "Methods for inclusion Employing think aloud…"
6	Design and prototypes	Presentation: "Relating HCI and software engineering" Presentation: "Sketching interfaces" Presentation:" Task Analysis and human computer interaction"
7	Design and prototypes	Pattern presentations
8	Accesibility and universal design	Visit to the library to know about software and hardware for people with special needs Pattern presentations
9	Accesibility and universal design	Pattern presentations Presentation: "Adapting the Interaction in a Call Center System" Presentation:" User Needs Analysis and requirements engineering"
10	Internationalization	Speech about vehicle interfaces Due date for Part 1 of the laboratory project Presentation: "Cultural differences on attention and perceived usability" Presentation: "Cross cultural interface design strategy"
11	Graphical design	Due date for the heuristic evaluation of a software system Presentation: "Web Design Interface"
12	Graphical design	Speech about interfaces for e-learning Presentation: "Usability and open source software"
13	Standards and guides	Speech about groupware interfaces Presentation: "An introduction to 3-D user interface design"
14	Course summary	Presentation: "Human Computer Interaction and Security" Presentation: "Metrics for Evaluating Human Computer Interaction Systems"
15	Research paper presentations	Due date for the research paper
16	Research paper presentations	Due date for Part2 of the laboratory project

8. LESSON DEVELOPMENT

Undergraduate students receive two 100-minute lessons every week, generally early in the morning. On the other hand, graduate students receive only one 200-minute lesson per week, usually in the evening. Students at both levels attend the same amount of hours, but it is definitely different to attend and pay attention for almost four hours. Graduate students have worked the whole day before they attend the lesson. They can easily get tired and bored. This represents a great challenge for the professor.

Most of the time, the professor lectures undergraduate students. Slides based on the AIPO book [1], other books [11, 13], and journal and conference papers are used to support the lecturing process. The professor assigns short practical problems to be solved in the classroom. Students present design patterns complemented with actual examples. The professor can cover material on the slides because students do not ask many questions.

On the contrary, graduate students participate much more in the classroom. Slides are available for them too, but lecturing for 200 minutes is boring. These two factors together suggest that it is better to plan the lesson as a combination of participative activities and short summarizing lectures. If students have previous knowledge about a topic covered in class, activities such as discussions or practical problems related to the topic can be solved by students working in groups. This allows students to generate their own knowledge. Additionally, group activities developed in the classroom allow graduate students to meet classmates and know them better, which additionally helps them to choose the team they will work with on other assignments. The professor closes the lesson with a short summarizing lecture useful to provide a theory base and review main concepts and results obtained in class activities.

It is not possible to follow this participative approach to cover all topics, because some are new for students. For example, human factors, which include psychology and sociology aspects, are relatively unknown to both undergraduate and graduate students. In such a case, the professor lectures in order to create a sound base and a common language to be used all the semester long.

As shown on Table 2, the schedule of the graduate course includes external speakers, who are professionals with sound knowledge on a HCI topic. Topics such as interfaces for e-learning or for vehicles are new to most students and are useful to create a broader overview on HCI. Undergraduate students are invited to these presentations.

Both undergraduate and graduate students visit the university library, where they receive a speech and a demonstration on software and hardware used by people with special needs, such as blind people (week 8 on Table 2). This visit is important in order to create awareness of the importance of taking accesibility into account when designing software systems.

9. Findings

Over two semesters, a total of 48 undergraduate students and 22 graduate students took the corresponding HCI course. The passing rate is 100% for the undergraduate course and 95% for the Masters course. At the end of the semester, a 5-question questionnaire is filled up by students in order to assess their level of satisfaction. Figure 2 shows the results for the question about whether the course objectives were reached. Questionnaire answers show that 68% feel the objectives were fully achieved and 32% believe they were partially achieved. Some of them expected more emphasis on graphical design and more practice on usability evaluation and interface design.

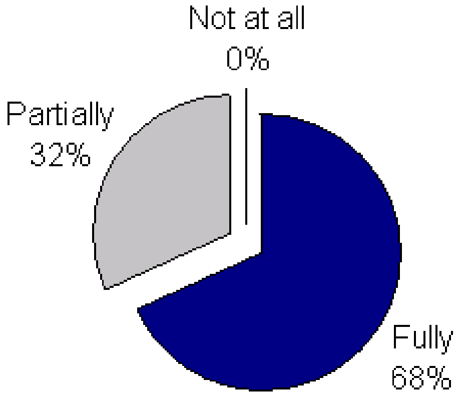

Figure 2. Students´ Opinion about whether the Course Objectives Were Reached

On the other hand, 57% of students believe the knowledge acquired in the course will be very useful for their professional performance and 43% believe it will be useful to some extent. This means that the HCI course has a positive impact on their professional practice.

Students were asked about the most important concept they learned. Since it is an open question, many different answers were received, but 53% of students referenced in their answers the importance of user participation during the development process, and the concepts of usability and accessibility.

Overall, we feel comfortable with the level of satisfaction of students, but we are aware there are many aspects which need to be improved. Students suggested adding more practice on usability evaluation and interface design, and including architectural models.

10. Future Work

Currently we are planning to teach the graduate course once again. Following students´ suggestions, this time we give more emphasis on interaction design and evaluation. The next step is to design a second course to allow students to study in depth some HCI topics. As we expect in the future many undergraduate students will take the undergraduate HCI course, we plan to change the graduate course curricular design, in order to offer students the opportunity of taking a more advanced course. However, because students from all universities in the country apply to our Masters program, we cannot ignore this situation and have to design a course starting from basic HCI concepts.

11. Conclusions

In this paper we have described our first experience on designing and teaching two introductory courses on HCI, one for undergraduate students and one for graduate students. The strategy of initiating an elective undergraduate course was very positive, since it allowed the professor to gain sound grip on the area. Due to differences among undergraduate and graduate students and among undergraduate and Masters courses, evaluation methodology differences were implemented. Previous knowledge about the characteristics of students taking the courses is a key factor in identifying those differences.

References

[1] Abascal, J., Aedo, I., Cañas, J., Gea, M., Gil, A.B., Lorés, J., Martínez, A.B., Ortega, M., Valero, P. y Vélez M. *La interacción persona-ordenador*. Lleida, España: Asociación Interacción Persona Ordenador (AIPO). 2001.

[2] ACM and IEEE Computer Society. *Computing Curricula 2001 Computer Science*. Report of the ACM/IEEE-CS Joint Task Force on Computing Curricula. 2001. http://www.computer.org/education/cc2001/final/index.htm.

[3] Carey, J., Galletta, D., Kim, J., Te'eni, D., Wildemuth, B. y Zhang, P. The Role of Human-Computer Interaction in Management Information Systems Curricula: A Call to Action. *Communications of the Association for Information Systems*. Vol. 13, No. 23, (2004), pp. 357-379.

[4] Consejo Universitario UCR. *Reglamento de Régimen Académico Estudiantil*. 2001. URL: http://www.cu.ucr.ac.cr.

[5] Cox, A. *Diagnóstico y perspectiva de los estudios de posgrado en Costa Rica.* Digital Observatory for Higher Education in Latin America and the Caribbean. UNESCO, OPES 18/2004, 2004. URL: http://unesdoc.unesco.org/images/0014/001404/140485s.pdf

[6] Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G. y Verplank, W. *ACM SIGCHI Curricula for Human-Computer Interaction*. Technical Report. Nueva York, Estados Unidos: ACM Press. 1996.

[7] Instituto Tecnológico de Costa Rica. Plan de Estudio Bachillerato en Ingeniería en Computación. *Instituto Tecnológico de Costa Rica*. URL: http://www.itcr.ac.cr Cartago, Costa Rica.

[8] Instituto Tecnológico de Costa Rica. Plan de Estudio de Maestría en Computación con Énfasis en Ciencias de la Computación. *Instituto Tecnológico de Costa Rica*. URL: http://www.itcr.ac.cr Cartago, Costa Rica

[9] Jenkins, M. Enseñando ingeniería de software en un programa de Ciencias de la Computación. *Memorias del X Congreso Iberoamericano de Educación Superior en Computación* (X CIESC). Montevideo, Uruguay. 2002.

[10] Lorés. J., Granollers, T. y Aguiló, C. An Undergraduate Teaching Experience in HCI at the University of Lleida. *Proceedings of The First Joint BCS/IFIP WG13.1/ICS/EU CONVIVIO HCI Educators Workshop* (Limerick, Ireland, 23-24 Marzo). 2006. URL: http://www.idc.ul.ie/hcieducators06/Procs/lores.pdf

[11] Kent, N. *The Psychology of Menu Selection Designing Cognitive Control at the Human/Computer Interface.*. Westport, CT, United States: Ablex Publishing Corporation, 1991.

[12] Peslak, A. A Framework and Implementation of User-Interface and Human-Computer Interaction Instruction. *Journal of Information Technology Education*, Vol. 4, (2005), pp. 189-205.

[13] Tidwell, J. *Designing Interfaces*. First Edition, Sebastopol, California: O´Reilly Media, Inc. 2005.

[14] Tucker, A:B. *Computing Curricula 1991. Report of the ACM/IEEE-CS.* Report of the ACM/IEEE-CS Joint Curriculum TaskForce. Estados Uniddos: ACM Press and IEEE Computer Sociedy Press. 1991.

[15] Universidad Nacional. Plan de Informática con grado de bachillerato. Universidad Nacional. URL: http://www.una.ac.cr Heredia, Costa Rica.