Teaching HCI: A Challenging Intercultural, Interdisciplinary, Cross-Field Experience

Cristian Rusu and Virginia Rusu

Pontificia Universidad Católica de Valparaíso, Av. Brasíl No. 2241,
Valparaíso, Chile
cristian.rusu@ucv.cl

Abstract. The extraordinary development of the Human-Computer Interaction (HCI) is still poorly reflected in South America, with the notable exception of Brazil. Our interest in HCI came from practice, as we have an extensive experience in software development. We are teaching HCI in a Chilean university (Pontificia Universidad Católica de Valparaíso) since 2003. Teaching HCI was a highly challenging intercultural, interdisciplinary, cross-field, but very rewarding experience. It was an intercultural experience, as we were born, raised, educated, with work experience both as professionals and professors in an East-European ex-communist country (Romania), but we taught HCI in a Latin-American country (Chile). Moreover, we did it in English, for Spanish speaker students. It was an interdisciplinary experience as HCI is a highly interdisciplinary science itself. It was a cross-field experience, as it allowed us to build a bridge between theory and practice.

Keywords: Human-Computer Interaction, Computer Science Curricula, Intercultural, Interdisciplinary.

1 Introduction

The *Special Interest Group on Computer-Human Interaction* (SIGCHI) of the *Association for Computing Machinery* (ACM) defines *Human-Computer Interaction* (HCI) as a discipline concerned with the design, evaluation, implementation of interactive computing systems for human use, and with the study of major phenomena surrounding them [24]. Basically, HCI tries to make people's experience with computers more productive, more time-efficient, and more pleasant.

The importance of HCI education for software professionals should be evident and well understood, when designing *Computer Science* (CS) programs, at all levels. It sounds good in theory, but is difficult in practice. Unfortunately the professors involved in CS programs usually consider HCI to be a secondary matter. This is probably a reflection of the well known conflict between HCI specialists and software engineers [8]. It has serious consequences for the training of the software professionals, as they will focus on the inner part of the software systems, ignoring the importance of the outer part (the interface). What will we get then? Top models dressed in poor clothes, and not in the "haut couture" creations they deserve!

The extraordinary development of the HCI field, both as theory and practice, is poorly reflected in South America, with the notable exception of Brazil [1], [2]. This

T. Ishida, S.R. Fussell, and P.T.J.M. Vossen (Eds.): IWIC 2007, LNCS 4568, pp. 344–354, 2007.

unfortunate "rule" is followed by Chile. There is an obvious lack of HCI courses in Chilean CS programs. These courses are rather exceptions, due to enthusiasts, and are usually optional, not compulsory subjects.

We are teaching HCI as optional subject in undergraduate CS curricula in *Pontificia Universidad Católica de Valparaíso* (PUCV), Chile, since 2003. A major step forward was the introduction of HCI as compulsory subject into the curriculum of a Master Degree (MD) program in CS (in PUCV), in 2006. It was a major "battle" that we won.

2 Teaching HCI: An Intercultural Experience

Heaving a previous teaching experience in Europe (Romania), teaching in a different environment, a Latin-American country (Chile), in a language that we did not study formally (Spanish), was quite a challenge. Comparing the two systems was particularly interesting [15], [16], [18].

The problems of higher education in CS are, generally speaking, similar in any country. The education system has to develop in harmony with the systems used abroad, taking advantage of the experience of others. There are no "best" or "worst" systems. The experience of other countries should be selectively used, adapting it to the realities of the own country, and to the policy and strategy of a particular university. Exchanges are always useful and lead to benefits to all levels: personal (for professors and students), universities, and higher education systems.

2.1 Computer Science Education Systems in Chile and Romania: A Brief Comparison

The Chilean and Romanian Higher Education systems are different in many aspects. However, a changing process affected the education system both in Chile and Romania approximately in the same period, the '90s. Changes in Romania were due to the transition from a dogmatic to a liberal regime. Romania will be soon a member of the European Union (January 2007), and all changes to the education system were made in the spirit of the Bologna Process, the main purpose being the integration into the *European Higher Education Area* [21]. On the other hand changes in Chile were part of the general process of transition from dictatorship to democracy.

The education in CS in Romania has a tradition of more than 30 years. Nowadays the education system in CS has components at all levels, from elementary school to graduate programs. There are no special high schools of *Informatics* (CS) in Chile. There certainly are classes of Informatics, at all levels, from primary schools to high schools, but not special, systematic programs in Informatics, like in Romania. There are technical-oriented high schools, but not exclusively Informatics-oriented. The technical-oriented high schools are not elite schools, like Informatics high schools in Romania.

There are three traditional branches at undergraduate and graduate levels in Romania: *Informatics* (more theoretical-oriented), *Computer Science* (more technical-oriented), and *Business Informatics* (Informatics applied in Economy). The higher education in CS in Chile is quite heterogeneous. Various academic units are offering programs in Informatics, as a response to the market demand. The response is rather empiric. The traditional three branches that exist in Romania are not present in Chile. Undergraduate programs in Computer Science are generally called *Informatics Engineering* (*Ingeniería Informática*). There are two traditional types of programs:

- *Ingeniería Civil Informática* (12 semesters),
- *Ingeniería de Ejecución en Informática* (8 semesters).

A third type of program is gaining popularity lately, a 10 semesters program (*Ingeniería Informática*). The differences between *Ingeniería Civil Informática* (*Civil Informatics Engineering*), and *Ingeniería de Ejecución en Informática (Execution Informatics Engineering)* are quite obvious, but it is more difficult to establish the specificity of *Ingeniería Informática*.

Usually all Chilean undergraduate programs last an average of 2 semesters more than the similar programs in Romania. The explanation consists mainly in the higher complexity of the fundamental science programs (Mathematics, Physics, Chemistry, Biology) in the Romanian high schools, load that usually passes to undergraduate level in Chile. For instance, teaching Calculus in the Romanian high schools is a rule, but it is usually omitted in the Chilean high schools.

The Chilean curricula are usually more flexible than the Romanian ones. Students have open possibilities to plan their study program, according to their abilities and specific interests. Romanian curricula only recently became more flexible, due to the requirements of the Bologna process and the adopting of the *European Credit Transfer System* (ECTS).

The Romanian high education system includes both *theoretical* (lectures) and *practical* activities (seminars and/or laboratories), for all courses. The only teaching activities in Chile are lectures. There are additional practical activities ("*ayudantías*"), usually in the case of the compulsory courses. These are taught by students, not by professors, so their quality might be questionable.

All programs that run in Chile are taught in Spanish. There are no English programs in Informatics, neither undergraduate nor graduate. Some programs do not include the study of English as foreign language. Teaching CS courses in English is even less common. Students have a quite low level of (applied) English knowledge, and this is a major handicap when graduates want to work abroad.

There are many undergraduate programs in CS in Chile, but not a proportional number of graduate programs. There is a lack of MD programs, and especially PhD programs. The rather small number of MD programs that are currently running usually last from 3 to 4 semesters. The few PhD programs in CS last 8 semesters. There are two kinds of MD programs:

- "*científico*" – research oriented,
- "*profesional*" – technical oriented.

The PhD programs in Chile are not exclusively research-oriented, as in Romania. They include formal courses, for at least 4 semesters, then a graduation exam, that leads to the preparation of PhD thesis (PhD candidate status).

Our teaching experience in Chile takes place in *Pontificia Universidad Católica de Valparaíso* (PUCV), a prestigious and long tradition university, which is considered one of the best universities not only in the Central area (Viña del Mar – Valparaíso), but all over Chile. The university was founded in 1925, being the first university ever opened in the 5th Region of Chile (Valparaíso Region). It continuously developed itself, as an academic and research establishment, having nowadays more than 12,000 students, enrolled in over 65 undergraduate and over 50 graduate programs (*diplomado*, *magister*, and *doctorado*).

The *Center of Computer and Information Sciences* (*Centro de Ciencias de Computación e Información*) of (Pontificia) Universidad Católica de Valparaíso was created in 1972. Later, in 1981, it was incorporated into the Faculty of Engineering, and became *Informatics Engineering School* (*Escuela de Ingeniería Informática*) in 1982. An 8-semesters undergraduate program of Informatics Engineering started the same year, being the first program of the new academic unit. Fifteen years later, in 1997, a 12-semesters undergraduate program was initiated. The two programs are presently running, *Ingeniería de Ejecución en Informática* (8 semesters), and *Ingeniería Civil Informática* (12 semesters), with over 650 undergraduate students. An MD program in *Informatics Engineering* (*Magíster en Ingeniería Informática*) runs in Escuela de Ingeniería Informática of PUCV since 2006. It started with 19 students, a rather large number, which certainly exceeded the expectations.

2.2 The HCI Course in a Computer Science Undergraduate Program

Once we discovered HCI as a well established science, we were tempted to share our experience with our students. As there were no HCI subjects in CS curricula in Pontificia Universidad Catolica de Valparaiso, Chile (PUCV), the only way to do that was to propose an HCI course as optional subject. We did that for the very first time back in 2003. It was a double challenge, as it was our first experience in teaching HCI, and we did it in English, not in Spanish, which is the official language in Chile (and in PUCV). The experience was highly successful, so we repeated it every year. We never intended to teach an English course; English was just a tool, but not a goal. The course was designed from the very beginning as a CS course. However, we intended to also offer an opportunity to practice applied technical English [19].

Table 1. Number of students of HCI undergraduate course

Year	No. of students	Language of teaching
2003	15	English
2004	33	Spanish
2005	26	English
2006	27	English

When we tried to teach the same course in Spanish, the interest among students was even higher, as only a relatively small number of students have enough knowledge of English to participate in HCI classes taught in English (see Table 1). This second version of our HCI course was, however, not a simple translation of the course previously taught in English. We had to redesign the course, especially due to the rather poor Spanish references. So we came to realize the disparity of HCI bibliography in English and Spanish. We do not have yet the necessary resources to offer two simultaneous versions of the HCI course, one in English, and another one in Spanish. We decided to teach only the English version, since it offers a unique opportunity to our students, being the only subject taught in English.

The structure that we are using for the undergraduate HCI course is as follows:

- *The field of Human-Computer Interaction,*
- *The nature of Human-Computer Interaction,*
- *Computer system and interface architecture,*
- *Usability,*
- *Interaction design,*
- *Web design.*

The most important area of HCI, from a software developer's point of view, is the user interface design and development. This mainly involves user-centered design and task analysis, dimensions of interface variability, dialogue tools and techniques, interface design, interface implementation, interface quality and evaluation methods.

Our course mainly focuses on Usability, which is practically its core [9], [11], [12]. We think usability is the most important attribute of the software products, and the course deals with topics like:

- *Usability Paradigms and Principles,*
- *Usability Engineering,*
- *Usability Evaluation,*
- *Usability in Practice.*

The ISO 9241 standard defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Designing for maximum usability should be the main goal of the software design. There cannot be software without software users! That is why we consider usability as the most important software attribute. Two main problems occur: (1) how can an interactive system be developed to ensure its usability? and (2) how can the usability of an interactive system be demonstrated or measured?

We strongly believe in a user-centered approach for interface design and software development [3], [4], [10]. A user-centered development methodology differs from traditional software engineering methodologies in three key areas: (1) it is user centric, not data centric, (2) it is highly interdisciplinary and draws knowledge from many areas, and (3) it is highly iterative and involves as much testing and revision as possible.

The main changes that we have done since 2003 were into the theory-to-practice balance. Our belief is that we have to prepare HCI practitioners in CS undergraduate programs. Students have to prepare a final HCI project (preferable a group project), consisting in the development of user interface prototypes.

Students' feedback was excellent; they really enjoyed the HCI course. This is proved by the increasing number of the students that are choosing the course (from a wide range of available courses), the increasing number of students that are choosing HCI related subjects for their final (graduation) thesis, and the contact that they are maintaining with us over the time.

We supervised the first undergraduate thesis developed in PUCV in the usability field, back in 2003. Since than, more and more students are choosing a large range of HCI or HCI-related subjects for their graduation thesis (see Table 2).

Table 2. Undergraduate HCI thesis supervised in PUCV

Year	No. of thesis	Subjects
2003	1	• Methodology for Usability Evaluation of Software Systems
2004	1	• Usability Evaluation by Formal Methods
2005	1	• Tool Prototype for Usability Evaluations
2006	5	• Usability on E-learning Platforms • Interaction Modeling Methodologies in Software Development • Accessibility in Software Systems for Users with Mental Disabilities • 3D Interface for Operating Systems • Usability and Communicability of Software Systems

2.3 The HCI Course in a Computer Science Graduate Program

A major step forward was the introduction of HCI as compulsory course into the curriculum of the MD program in CS, a program that PUCV offers. We consider that the introduction of the HCI as a compulsory course into the new curricula was a major "battle" we won. Some of the arguments that helped us win this battle were:

- the successful experience of teaching HCI as optional course in undergraduate program,
- the interest of the students,
- the feedback of the ex-undergraduate students (now software professionals) that have graduated the HCI course,
- the demand of the potential MD students.

The last two arguments were proved by an inquiry that was made during the preparation of the MD program curriculum. Surprisingly, the software professionals seem to be more convinced of the importance of HCI than our own colleagues (professors of CS programs). The belief comes from practice, from their professional experience. We find this extremely important, as it proves the awareness of the importance of HCI practices. This is stronger and more important than any bureaucratic argument that may occur when designing CS curricula. It is particularly encouraging for the Latin American context.

The new MD HCI course represents a new challenge. Potential students are coming either from professional environment (being experienced software professionals) or from school (recently graduating a CS program, without practical experience). The audience is more heterogeneous than in the case of the undergraduate HCI course: junior professors, software professionals, and (still) undergraduate students.

Obviously, the new course has to be different. It has to include some HCI basics (for students without prior experience in HCI), without entirely repeating the content of the undergraduate HCI course. In addition, it has to bring the concepts to a higher level. We have to focus on preparing both HCI practitioners and researchers in CS graduate program. We are building the new course over the structure of the old undergraduate one, but with emphasis on different topics. The teaching process is more personalized, tanking advantage of the reduce number of students in a graduate program (19, comparing to approximately 30, in undergraduate programs).

We intend to keep the main focus on usability, and usability evaluation, but we introduce new topics, as elements of semiotic engineering that can help to bridge the gap between HCI practitioners and software engineers [6]. Besides classical evaluation methods we will also introduce a communicability evaluation method [5], [13]. Regarding the infrastructure, a usability laboratory becomes necessary, and it is already under construction.

2.4 Adapting to Changes: A Subjective Experience

Adapting ourselves to a new education system was not as difficult as we thought, probably because we experienced huge changes during the transition process in Romania. This made us more adaptable to new environments and new challenges.

One of the challenges that we had to deal with was a language that we were not familiar to (Spanish). We never formally trained in Spanish; we learned it step by step, in everyday situations. In fact, we started teaching classes in Spanish after only a few weeks of submerging ourselves in a Chilean environment. Both Romanian and Spanish are Latin languages, similar in many aspects, but this has advantages and disadvantages. The advantage was that speaking Romanian and having knowledge of French was quite easy to understand Spanish, Italian or Portuguese. The disadvantage was that having to adapt ourselves to Spanish in a very short period, we had to give emphasis to quantity instead of quality. This caused us serious problems later, as we rapidly acceded to a relatively acceptable level, but it was tremendously difficult to improve it. The lack of formal training and the (chronic) lack of time for properly studying a (still) foreign language (even if a familiar one) had long term consequences. It is particular bothering for perfectionist people as professors usually are!

We had the wrong idea that Spanish is the same, all over the world. Obviously we were wrong! Moreover, Chilean Spanish is quite different from whatever else. The difference is more difficult to deal with in human contacts, as the professor - students relationship is.

Teaching a HCI course in English, for Spanish speakers, was an even bigger challenge, as English is not our native language. The major difficulty was (and it still is) the (very) heterogeneous level of English that our students posses. As we like to improvise, to motivate the students to interact, to be active listeners, it was twice as difficult as we thought, both for our students and for ourselves.

We had to deal with a different environment, a different country, and different idiosyncrasies. Life gave us many lessons, and not all of them were pleasant experiences. Maybe the most valuable lesson was that a certain feeling of European superiority (“Chile is the end of the world!”) is completely wrong. Fortunately, we never had that feeling. Moreover, we were sometimes embarrassed by the feeling of inferiority that we perceived in the Chilean society (“Oh, Europe…!”).

As professors, maybe the biggest challenge was to adapt ourselves to a higher education system that does not explicitly include practical academic activities (seminars, laboratories), like the Romanian system does. We had to include all these activities during the regular classes’ schedule, which was quite a challenge. We didn’t count anymore on the support of an assistant professor. We didn’t even have a tutor student (*ayudante*), as optional courses do not include such activities. As we believe that a HCI course at undergraduate level should put a special emphasis on practice, it was difficult to include as many practical activities as we would.

Apart from the negative side that the Chilean education system may have, it also has a positive side: it forces students to push themselves, to solve problems without (much) external help, it builds stronger personalities. Students are more “aggressive”, more interrogative, they “don’t buy” everything that the teacher says, which is far more interesting for a professor.

Graduate students were a new challenge, as they represent a larger spectrum of academic, professional, and even HCI backgrounds. Some of them are our former HCI undergraduate students, but most of them are having the very first (formal) contact with HCI. We had to personalize the graduate HCI course more, to offer both HCI basics and advanced topics.

3 Teaching HCI: An Interdisciplinary Experience

HCI is a highly interdisciplinary area, involving mainly Computer Science (application of design and engineering of user interfaces), Psychology (application of theories of cognitive processes and the empirical analysis of user behavior), Sociology/Anthropology (interactions between technology, work, and organization), and Industrial Design (designing of interactive products). HCI focuses on Computer Science, and belongs mainly to Computer Science area. Other disciplines serve as supports. HCI uses supporting knowledge on both machine and human side. Actually HCI may be considered as a combination of Science, Engineering and Design.

We may think interdisciplinary at least at two levels: (1) students level, and (2) inter-academic units’ level.

The HCI course is very appealing for the students as it offer them a whole new world. They have a solid background in CS courses, but the curricula are definitely neglecting the most important counterpart of the interactive software systems, the users! In order to support interdisciplinary and to make connections with other subjects of CS curricula, students are encouraged to choose HCI project subjects that can help them in improving some other software development projects.

Interdisciplinary at the inter-academic units’ level is still an intention. It is necessary, it is important, it can help giving different points of view, it can support the interdisciplinary research, but it is quite a challenge in practice. There may be many reasons, but at least in our case (and the case of our university, PUCV), the major challenge is the lack of time, the overload agenda of the professors.

4 Teaching HCI: A Cross-Field Experience

Our interest in getting better user interfaces came from practice, as we have a rather extensive previous experience in software development [20], [22], [23]. Later on, we discovered HCI as a well established field of CS. So, we made our way from practice to theory, and we had the natural impulse to get back to practice. We tried to do this in our HCI classes [17].

Teaching HCI every year offered us the possibility to improve the course, and to adapt it to the necessities of local software companies, mainly based on the feedback received from ex-students. We kept the general structure of the course more or less the same, but we changed the weights of different topics.

We gradually increased the weight of practical activities, and we came to focus more and more on teaching the students how to put the HCI theory into practice. That is why we are trying to include as many practical exercises as possible. For example, students have to perform at least 3 – 4 heuristic evaluations during the semester. They also have to practice brainstorming sessions and user interviews, among others. They are always encouraged to interact, to expose their ideas, to make proposals, to analyze and evaluate them.

During the development of their HCI project, students have to apply the usability concepts, to cross-evaluate the prototypes, and to improve them based on the evaluations they performed. They have to highlight the changes and the improvements they have made, in public presentations. As the undergraduate HCI course is open for two different programs, we strongly encourage the formation of mixed teams, including students that belong to different programs. Even if they don't have yet (much) working experience, they bring different points of view, which definitely enrich the proposed solutions.

Ex-students are keeping contact with us after graduation. They offer a valuable feedback from their practical experience as software developers. They are trying hard to apply in practice what they learned in HCI classes. Even if it is difficult, and sometimes discouraging, they are actually developing, step by step, an awareness of the importance of HCI practices into their professional environment.

There is a whole debate over the conflict between HCI specialists and software engineers. Many authors analyzed the integration of the usability issues in CS curricula at different levels [14]. We perfectly agree about the existence and the relevance of the conflict, but fortunately it is easier for us to solve it, as we teach both Software Engineering and HCI. Of course, this is an exception, and the problem should be formally solved when (re)designing CS curricula. Having the mentioned privilege, we are trying to take full advantage of it, always highlighting the strong relationship that should exist between the Software Life Cycle and the Usability Engineering Lifecycle.

There are many arguments to support the introduction of HCI topics early in the CS curricula [7]. Introductory CS courses focus on systems, ignoring users most of the time. Once again we are fortunate enough to be able to informally influence the situation, because we are also teaching the very first introductory course in CS. So it is up to us to stress from the very beginning the importance of the user over the system.

5 Conclusions

A CS curriculum has to be dynamic, to adapt itself to the continuous IT changes. We consider HCI as a basic part of the formative process of the future software professionals. Changing curricula is usually a long and bureaucratic process. Improvements are faster and much easier to implement by offering new or redesigned optional subjects.

We proposed an HCI course as optional subject back in 2003. The experience was highly successful, so we repeated it every year, in CS undergraduate programs. As we strongly believe in a user-centered approach, we are mainly focusing our HCI course on usability. We try to improve the course each year, and to adapt it to the necessities of local software companies, mainly based on the feedback received from ex-students. We gradually increased the weight of practical activities, and come to focus more and more on teaching the students how to put HCI theory into practice.

A major step forward was the introduction of HCI as compulsory course into the curriculum of the MD program in CS, a program that PUCV offers. We focus on preparing HCI practitioners in undergraduate programs, but also on preparing both HCI practitioners and researchers in graduate programs.

Surprisingly, the software professionals are more convinced of the importance of HCI than our own colleagues (professors of CS programs). This proves an awareness of the importance of HCI practices into the software professional environment, and is particularly encouraging for the Latin American context.

Teaching HCI was a highly challenging intercultural, interdisciplinary, cross-field, but very rewarding experience. It was an intercultural experience, as we were born, raised, educated, with work experience both as professionals and professors in an East-European ex-communist country (Romania), but we taught HCI in a Latin-American country (Chile). Moreover, we did it in English, for Spanish speaker students. It was an interdisciplinary experience as HCI is a highly interdisciplinary science itself. It was a cross-field experience, as it allowed us to build a bridge between theory and practice.

Acknowledgments. Our thanks to *Pontificia Universidad Católica de Valparaíso* (Chile), which considers internationalization a priority, and has made possible the experience described here.

References

1. Baeza, R., de Souza, C., Rivera, C.: Enseñanza de Interacción Humano-Computador en Latinoamérica. In: Proceedings of I Jornadas de Trabajo sobre Enseñanza de HCI, Puertollano, Spain, pp. 21–31 (2005)
2. Collazos, C., Vivas, N., Ramírez, M.: La enseñanza de HCI en Colombia: Un trabajo multidisciplinario basado en competencias. In: Proceedings of XXXI Conferencia Latinoamericana de Informática (CLEI 2005), Cali, Colombia, pp. 51–58 (2005)
3. Constantine, L., Lockwood, A.: Software for Use. ACM Press, New York (1999)
4. Dix, A.: Human-Computer Interaction. Prentice Hall, Englewood Cliffs (1998)

5. de Souza, C.: The Semiotic Engineering of Human-Computer Interaction. MIT Press, Cambridge (2005)
6. de Souza, C.: Semiotic Engineering: bringing designers and users together at interaction time. Interacting with Computers 17(3), 317–341 (2005)
7. Faulkner, X., Culwin, F.: Integration of usability issues within initial software development education, ACM SIGCSE Bulletin. In: The proceedings of the thirtieth SIGCSE technical symposium on Computer science education SIGCSE '99, vol. 31(1), pp. 296–300. ACM Press, New York (1999)
8. Folmer, E., van Welie, M., Bosch, J.: Bridging patterns: An approach to bridge gaps between SE and HCI. Information and Software Technology 48(2), 69–89 (2005)
9. Ivory, M., Hearst, M.: The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys 33(4), 470–516 (2001)
10. McCracken, D., Wolfe, R.: User-Centered Web Site Development: A Human-Computer Interaction Approach. Prentice-Hall, Upper Saddle River (2004)
11. Nielsen, J.: Designing Web Usability. New Riders Publishing, Indianapolis (2000)
12. Nielsen, J.: Usability Engineering. Academic Press, Boston (1993)
13. Prates, R., de Souza, C., Barbosa, S.: A method for evaluating the communicability of user interfaces. ACM Interactions 7(1), 31–38 (2000)
14. Rozanski, E., Schaller, N.: Integrating Usability Engineering into the Computer Science Curriculum – A Proposal, ACM SIGCSE Bulletin. In: Proceedings of the 8th annual conference on Innovation and technology in computer science education ITiCSE '03, vol. 35(3), pp. 202–206. ACM Press, New York (2003)
15. Rusu, C.: Computer Science Higher Education in Chile. In: Creative Mathematics, North University of Baia Mare, Romania, pp. 121–128 (2003)
16. Rusu, C.: Estudio comparativo entre los sistemas de enseñanza superior en informática entre Rumania y Chile, Final Report of Project No. 209.731/2003, Pontificia Universidad Católica de Valparaíso, Chile (2004)
17. Rusu, C.: Human-Computer Interaction. Lecture Notes, Pontificia Universidad Católica de Valparaíso, Chile (2006)
18. Rusu, C., Rusu, V.: El sistema de enseñanza en informatica en Rumania, Ingenerare. Revista Facultad de Ingeniería, vol. 15, pp. 54–60 (2003)
19. Rusu, C., Rusu, V.: Human-Computer Interaction from theory to practice, VIII Congreso Iberoamericano de Informática Educativa, Costa Rica (accepted paper, November 2006)
20. Rusu, C., Rusu, V.: Proiectarea unei interfeţe utilizator în Microsoft Excel folosind Visual Basic for Applications. Lucrarile seminarului de creativitate matematica, North University of Baia Mare, vol. 10, pp. 149–166 (2001)
21. Rusu, V., Rusu, C.: El Espacio Europeo de la Enseñanza Superior entre Teoríay Práctica. In: Rivera, S., Nuñez, J. (eds.) Experiencias Docentes en Ingeniería, Argentina, vol. 1, pp. 157–166 (2006)
22. Rusu, V., Rusu, C.: Ergonomia Software in birotică. Buletinul Ştiinţific al Universităţii de Nord Baia Mare, Romania, Fascicola ŞTIINŢE ECONOMICE, Seria A, vol. 21, pp. 144–146 (2004)
23. Rusu, V., Rusu, C.: Intefata utilizator pentru functii de analiza bursiera. In: Economia contemporana. Prezent si perspective, pp. 303–312. AGIR Publishing House, Bucharest (2004)
24. Rusu, V., Rusu, C.: ACM SIGCHI Website: (ACM's Special Interest Group on Computer-Human Interaction) http://sigchi.org