International Journal of Engineering Education Vol. 29, No. 3, pp. 666–673, 2013
Printed in Great Britain

0949-149X/91 $3.00+0.00

Teaching Human-Computer Interaction through developing Applications in Collaboration between Academy and Autism Organizations*

ANDRÉS MEJÍA FIGUEROA and REYES JUÁREZ-RAMÍREZ
Autonomous University of Baja California - Calzada Universidad 14418, Parque Industrial Internacional Tijuana, C.P. 22390 B.C., México. E-mail: mejia.andres@uabc.edu.mx, reyesjua@uabc.edu.mx

Human-Computer Interaction is a discipline that is getting more interest nowadays, not only because the proliferation of computer science but also due the proliferation of new technological devices that require user-friendly interfaces. Furthermore, new technology is enabling us to attend needs of special groups with specific impairments, such as autistic users. Most efforts have been made in order to integrate curricula for this discipline; also most efforts have been made in the teaching context of these themes. However, it is necessary to continue improving teaching practices, integrating research and good practices from real environments. In this paper we present some experiences in teaching Human-Computer Interaction and Usability Engineering in conjunction with Software Development. Our teaching experiences involve real projects in collaboration with a specialized organization in autism. This teaching environment allows students to interact with end users, making sense of how useful is applying user interface design principles in order to facilitate the life of users with special capabilities.

Keywords: collaboration between academia and autism organizations; user-centered design; teaching Human-Computer Interaction.

1. Introduction

"*Understanding HCI will allow the interface designer to produce interfaces that are usable by everyone, extending the impact of computing and communication to a diverse set of users within many domains, academic disciplines, and outside demographics*" [1].

As an engineering oriented discipline HCI pretends not only efficiency and reliability in interfaces [1] but also to develop usable interfaces for the user's comfort. Thimbleby [2] supports this assumption, stating that HCI motivates and empowers students to make a valuable and lasting contribution to the world. This must be one of the main motivations of HCI practitioners, generating comfort for human beings' necessities through usable interfaces.

From the engineering point of view, it is important to train HCI students to solve problems by using procedures and analytical methods that supports and extends usability engineering practices [1]. Furthermore, HCI students must be trained with the user-centered design approach, involving interaction with real users in order to fit user necessities.

Many proposals have been made for teaching HCI in the academic context; they go from theoretical classes to very practical approaches. Thimbleby [2] suggests that in order to develop usable interfaces as part of HCI courses, the better teaching approach is this: "*teaching is about making learning possible*" [3]. HCI is concerned with the user experience; almost any HCI issue begs an analogy to teaching and learning. In this case, Thimbleby [2] argues that "*HCI itself suggests that HCI teaching should provide an immediate opportunity to act, based on real tasks, should prevent mistakes, and be brief.*" Also argues that as a consequence "*a significant part of any HCI syllabus should be user learning, a topic that makes a useful counterpoint to students learning HCI, and is an opportunity for students to be taught learning and learning skills explicitly.*"

In a collaborative environment for developing systems all the participants can learn: students, end users, and other stakeholders. For example, in [2] is stated that "*not just to students but also to users who have to decide whether to achieve results quickly or whether to learn more advanced features of a system so they can do even better, but by delaying immediate results.*"

Some authors such as [4] argue that "*standard approaches within computer science need to be augmented and that new models of education can aid us in producing students with broad competence in the design of computer systems for human use.*" Based of this, we suggest to integrate the software engineering process with HCI and usability engineering in order to have usable interfaces based on a systematic approach. As we can see in Fig. 1, we initiate the development process with a requirements gathering phase, where engineering students visit end users in order to collect their necessities. After that, a user and task analysis is performed in order to specify special characteristics for the user

* Accepted 15 January 2013.

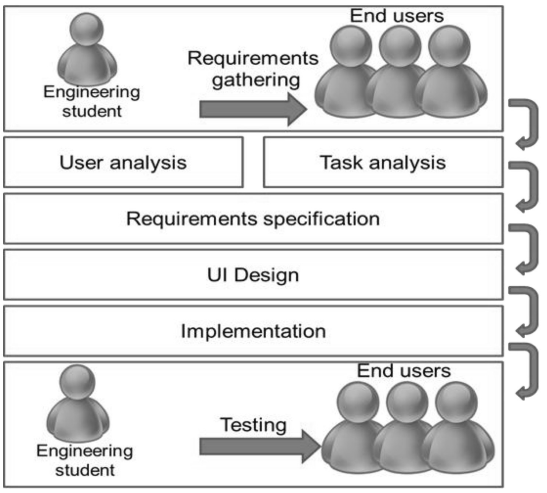

Fig. 1. Integrating software engineering process and HCI practices.

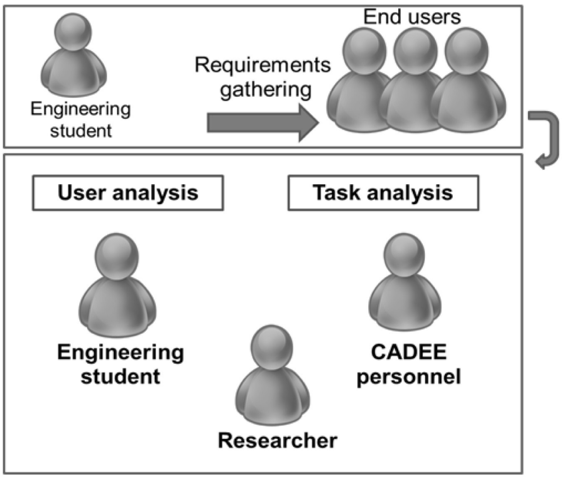

Fig. 2. User and task analysis an interdisciplinary approach.

interface. Next, the remaining phases of software development are executed in the traditional order.

It is important to emphasize that the user and task analysis is performed in an iterative way, interacting the engineering students, CADEE personnel and researchers (Fig. 2) in order to get a more specific set of specifications for the UI.

This document is organized as follows. Section 2 states the hypothesis that guides this research. Section 3 presents some related works. Section 4 presents a description of the curricula block focused on software development. Section 5 describes the collaboration environment between our research group and schools for special education. Section 6 describes briefly part of the user profile oriented to special capabilities related with autism. Section 7 describes the development process of the application and the results and effects of the project on our students. Section 8 contains the conclusions and future work.

2. Establishing a hypothesis to guide the teaching approach

Taking into account that our objective is to have more usable interfaces for the comfort of humans facilitating their life, we propose a hypothesis that guides our pedagogical approach for teaching HCI:

> Hypothesis: *Students make a better sense of human aspects to be considered in user interface design if they observe real situations of people with special capabilities; this contributes to optimize the usability attributes.*

As we see in most universities, students don't give the proper importance to the HCI practices and fail to use or resist using those practices in other classes or projects. We hope that with having firsthand experience dealing with a user, especially users with special capabilities, the students will have a better understanding of the importance of user centered software design.

3. Related works

Usability and human-computer interaction are important aspects for new technological devices. Recently, new characteristics of technological devices are covering special needs of users with disabilities. New technology is designed by novel professionals in product design and software development; based on this, it is important to educate these professionals in topics of usability engineering and human-computer interaction. Most research works have been realized in the context of teaching HCI. Previous works goes through course integration, teaching methodologies used, practical aspects, and project-based learning techniques used. Next we describe some related works.

In Aberg [5], he noticed that the HCI and usable systems courses had suffered bad course evaluations. In order to investigate the causes, focus groups were formed. The results showed that students perceived the course as trivial and lacked any real motivation to use the HCI and usability practices presented during the course. In order to get students more interested and motivated in HCI a course redesign was implemented with a focus on implementing a working prototype with a real application tied to a research project. A new evaluation of the course showed great improvement including the student's attitude towards HCI practices.

Plimmer et al [6] propose that in order to show computer science students how other disciplines contribute to HCI, they organized the course so that HCI students will learn the contributions of other disciplines through peer teaching as a part of a group project. By organizing teams with students with complementary skills (such as a student with high technical skills, other with designer skill, for example) knowledge transfer would occur and a more usable final software would be created as a result.

McCrickard et al [4] present a pedagogical approach based on case methods in order to react to challenges that have been observed in HCI education. This study provides insights into challenges that can be expected in the employment of case methods, student learning outcomes, and considerations for HCI curriculum planning. They emphasize on broad recommendations for improving integration of HCI professional practice, research, and education.

Greenberg [7] presents a schema for an undergraduate course where HCI is considered as the integrated process of design, implementation and evaluation. This work emphasizes on developing student skills to apply HCI in practical situations. The teaching methodology is based on theoretical lectures, video demonstrations and practical cases dealing with creating graphical user interfaces.

All of these works achieved great results in teaching HCI in an effective manner improving the student's attitude toward HCI practices. Not only do we set out and try to change students perception of HCI by integrating HCI practices into the development life cycle of projects and other courses, but also we try to make them socially aware of the impact those practices can have in the daily lives of other users, mainly users with disabilities.

4. The curricula block for software development

In our university, as part of the curricula of our undergraduate program in computer engineering, we have a set of courses oriented to software development: Requirements Engineering (REN_ID), Software Analysis and Design (SAD_ID), Human-Computer Interaction Design (HCI_ID), Software Engineering (SEN_ID), and Software Quality Assurance (SQA_ID).

In the REN_ID course the student learns about the process of eliciting, documenting and maintaining the requirements for software being developed. In the SAD_ID, as the name states, the students learn the proper way to analyze, document, and design software using UML and other documentation techniques. The HCI_ID course the students learn the importance of designing good interfaces and usable systems. In the SEN_ID course, the students learn all the phases of the software development cycle, remarking the phases such as: requirements gathering, analysis, design, implementation, and testing. Also, this course involves lessons on agile methodologies, emphasizing the interaction with clients and the end users. The SQA_ID course is focused in teaching methodologies such PSP and TSP and quality models such as CMMI and MoProSoft, as well as qualitative and quantitative quality evaluation.

The SEN_ID is an integrative course because it converges all the previous courses. Based on this, some it involves the development of a complete software system.

This curricula block is based in two important aspects: useful outcomes and real projects. Techniques from HCI and UE are taught for all the software development stages in an iterative cycle in order to try to achieve more usable software.

5. Collaborating in an interdisciplinary environment

In the semester 2012-1 we established collaboration with "Voz del Autismo" and CADEE, a Civil Association and a School for children with autism, respectively, in Tijuana, México. Firstly CADEE visited our research group reaching for software applications to assist the learning and therapy processes of children with autism.

CADEE has a group of professionals in psychology, special education, and physiotherapy. Also mothers of children are assisting the teaching and therapy activities. GIIPIS is our research group focused in software engineering, HCI and usability engineering as research areas. The members of our group teach in the undergraduate computer engineering program.

During the academic terms 2012-1 and 2012-2 we visited CADEE frequently. Our engineering students visited their installations, taking a look of the groups of children with autism. Undergraduate student analyzed children's behavior during therapy and teaching classes. As a result of these visits, students gather requirements specification to build software applications as course projects and extra course projects. Graduate students gathered information in order to integrate the autistic user profile.

In the other hand, personnel from CADEE visited our installations at the university, explaining students the children behavior and special needs, and helping us to define the autistic user profile. This interaction with CADEE contributed greatly in the instruction of students in topics such as gathering real user requirements.

6. The user model for autism

One of our current research projects in our research group GIIPIS is the generation of user models

Table 1. Proposed user model for autistic children

Attribute	Value
Language	Yes; No
Response to sounds	Yes; No
Visual contact	Yes; No
Identity	Yes; No
Interest to environment	Yes; No
Attends Instructions	Yes; No
Decision making	Yes; No
Tolerance to sounds	High; Medium; Low
Activity	High; Normal; Passive
Tolerance to change	Yes; No
Temper	Yes; No
Learning Style	Visual; Kinestesic

considering all the characteristics of a user in order to make more usable systems [8]. We have previously proposed an integral generic user model that allows adapting the interface based on the user's physical, psychological, cognitive and demographic characteristics [9, 10].

For this paper, we need to create a specific user model for autistic children in order to design more suited interfaces for them. The user model was determined via interviews with the parents, teachers and psychologists of CADEE and observation of the children. The user model obtained is as follows. The "Attribute" column indicates the user characteristic, while the "Value" column presence or lack of that characteristic (Table 1).

7. Designing an App for autistic users

In our collaboration with autism organizations, CADEE asked us a need for a mobile software application in hopes to help their children develop social and communication skills since therapy takes very long time, considering that new technologies, such as tablet PC's, seem to have make great strides in developing those skills, since they permit the children to communicate with little time using those devices.

They had used a program called Proloquo2Go (Fig. 3) for the iPad, having great success in getting children to communicate. Proloquo2Go is a software application for communication, easy to use, that allows the user to construct full sentences by selecting from different elements classified in different categories, all with illustrations associated with the element or category. Also the software is able to read the sentence and permitting the speech impaired user to communicate his wishes.

Our students took on the task of developing their own similar communication software in Spanish for the Android OS trying to achieve an overall more usable software, by applying a user centered design process and usability engineering practices.

In order for our students to develop an appropriate manner our version of the software, we separated the tasks needed for the analysis, design and development of the software depending on the course they were taking at the moment.

7.1 Organizing the application development

The students from the REN_ID course were in charge of conducting the necessary techniques (interviews, observation, use cases, among others) in order to obtain the requirements, and developing use cases, making special emphasis in using usability engineering techniques such as task analysis and user analysis.

In the case of the HCI_ID course, students, which most were also taking the SAD_ID course, were in charge of: obtaining the user profile (in conjunction with the students from REN_ID), developing the prototypes based on the use cases obtained by the requirement engineering students, proposing different user interfaces for the software and conduct some usability testing with the children to improve the interface.

The user profile was obtained with the help of the parents, teachers and psychologist of the children, giving special insight on their behavior and characteristics via interviews and visits to the CADEE installations.

Having the user model we obtained as a guide, the students were able to develop better use cases for the implementation of the system.

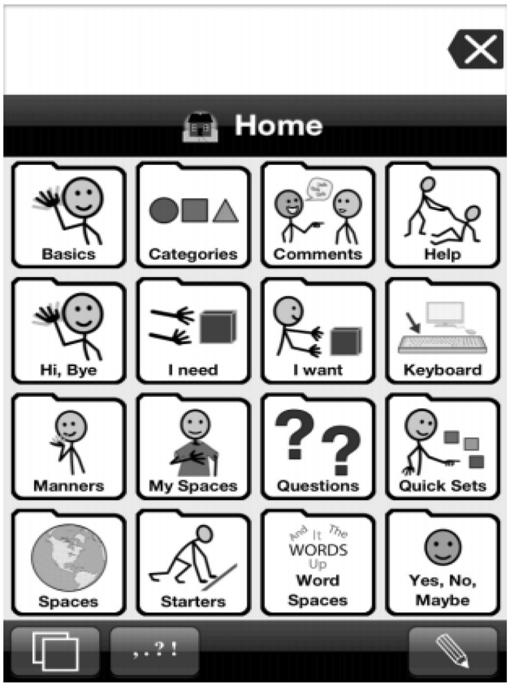

Fig. 3. Proloquo2Go for the Ipad.

7.2 Establishing relation between the prototype and the autistic user model

The relationship between the user's characteristics and the use cases is as follows (Table 2).

As a result we obtained a prototype of the software (Fig. 4). In essence it has the same basic functionality as Proloquo2Go, but it differs in that it has a different interface with improved navigability by having all the elements needed to the user with a scroll screen, with each category being displayed on screen and each having its own mini scroll while in Proloquo2Go there are several screens containing the contents of different categories, forcing the user to go back and forth between several screens to form a sentence. In addition to the navigation aspect of the software, the students also implemented a basic grammatical structure to the sentences, so the software may correct the user if he or she inputs several elements of the sentence in an order that is not grammatically correct.

Table 2. Use cases affected by the user's attributes

Attribute	Affected Use Case
Language	Create sentence, Speak sentence
Response to sounds	Create sentence, Speak sentence
Visual contact with another person	None
Identity	None
Interest to environment	Create sentence; Speak sentence
Attends Instructions	None
Decision making	None
Tolerance to sounds	Create sentence, Speak sentence
Activity	None
Tolerance to change	Create sentence; Speak sentence; Erase sentence; Navigate between groups and elements
Volatile Temper	None
Learning Style	Create sentence; Speak sentence; Erase sentence; Navigate between groups and elements

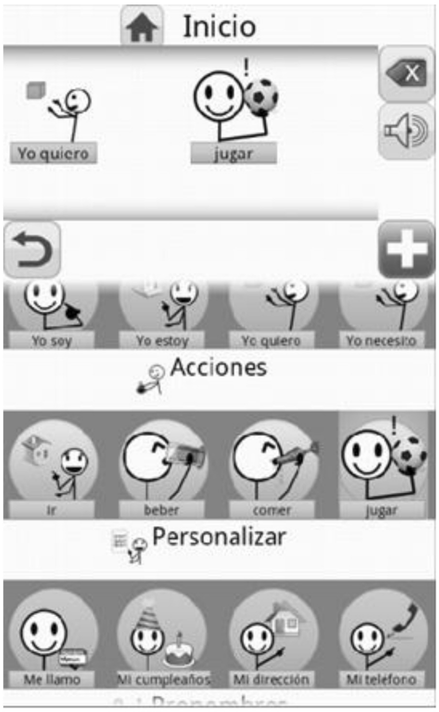

Fig. 4. Prototype software presented.

7.3 Usability testing of the prototype

The main development team comprised of five students, and were tasked in conducting an usability evaluation using Nielsen's usability heuristics [11] (Table 3) for both Proloquo2Go and the software they developed in order to determine if we achieved an overall more usable system in this current prototype. They applied the heuristic evaluation to the use cases that were implemented and compared the results between Proloquo2Go and the software they developed.

The consensus between the evaluators was that the interface of Proloquo2Go still had better usability elements, such as a help guide which our software was still lacking in this current version, and more customizable options. Those additions are being considered for the next iteration of the development cycle. The heuristic results are as follows, as shown in tables 4 and 5, where "Yes" is used where the software abides by the heuristic and "No" where it didn't.

Also, we are currently trying to program user centered usability testing to determine if the navigation improvements and grammatical function offer better overall usability to the children.

7.4 Response from the students

After the design and development of the first's prototypes, a small three question survey was given to the students to try to see if their perception of the importance of HCI and usability practices had changed. A positives answer (Yes), could sig-

Table 3. Usability heuristics by Nielsen

Tag	Heuristic
H1	Visibility of system status
H2	Match between system and the real world
H3	User control and freedom
H4	Consistency and standards
H5	Error prevention
H6	Recognition rather than recall
H7	Flexibility and efficiency of use
H8	Aesthetic and ,minimalist design
H9	Help users recognize, diagnose, and recover from errors
H10	Help and documentation

Table 4. Proloquo2Go heuristics results

Use case	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
Create sentence	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Erase element	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Read sentence	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Add element	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 5. Prototype heuristics results

Use case	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
Create sentence	No	Yes	No	No	No	No	Yes	Yes	No	No
Erase element	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Read sentence	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No
Add element	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No

Table 6. Survey presented to students

Tag	Question	Possible answers
Q1	When tasked with developing the application for autistic children, did it spark your interest?	Yes/No
Q2	When visiting our user (autistic children) and witnessing their needs first hand, did it change your perspective on the importance of user-centered software development?	Yes/No
Q3	Did you find yourself motivated in making a lasting contribution to the user during the development of the project?	Yes/No

Table 7. Results of survey

Question	Answer	Number of students	Percentage
Q1	Yes	25	83
	No	5	17
Q2	Yes	25	83
	No	4	13
Q3	Yes	29	97
	No	1	3

nify a change of attitude toward HCI practices. The survey is as follows in Table 6.

7.5 *Results and discussion*

A total of 30 students answered the survey. The results were as shown in Table 7 and Figs 5, 6, and 7.

With this joint project with students from different courses focused on HCI techniques and practices, the students felt the impact of those practices in all the stages of the software development cycle. In the last experiences, students showed profound interest in making more usable software specially when directly observing, interacting and getting to know their users, in this case autistic children, taking into consideration all the user's characteristics all the way through the development cycle, confirming our hypothesis in this case. Even several students were touched by the focus that was given to the course and the tool we were developing since, as it turned out, several of the students had a friend or family member with some level of autism.

This new found interest in the students went

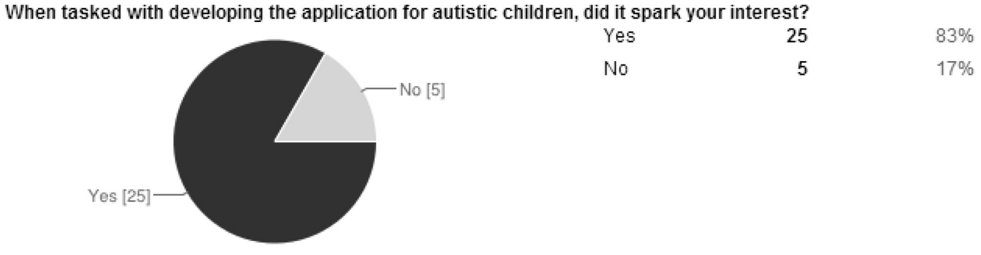

Fig. 5. Question 1 results.

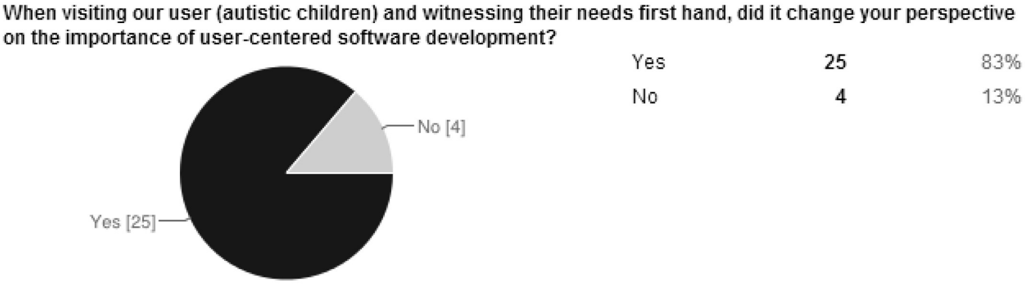

Fig. 6. Question 2 results.

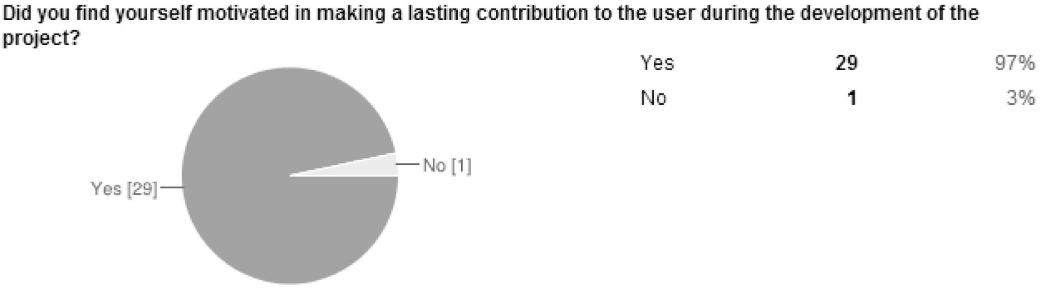

Fig. 7. Question 3 results.

beyond their current course, teachers in other courses reported students applying the techniques they were using on their courses without the teacher asking them or suggesting they use them.

8. Conclusions and future work

Cooperation between different courses with a goal in mind proved to be an excellent way for the student to adopt a more user centered approach when developing software, especially when they directly saw the impact of those techniques they learned to the user. They displayed a more willing attitude to use those techniques in other courses and seem to develop a more humanistic approach to the software development related courses.

Some of the students even continue to develop the application on their free time hoping to distribute the software for free to any person that might find it useful, while others take part in other software projects with a social focus.

We hope they continue to see the importance of HCI long after they graduate and start a change in our local software industry.

References

1. D. S. McCrickard, C. M. Chewar and J. Somervell, Design, Science, and Engineering Topics? Teaching HCI with a Unified Method, *SIGCSE 2004 Proceedings of the 35th SIGCSE technical symposium on Computer science education.* Norfolk, Virginia, United States of America, March 3–7, 2004, *ACM SIGCSE Bulletin*, **36**(1), March 2004, pp. 31–35.
2. H. Thimbleby, Teaching and learning HCI, *UAHCI '09 Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Addressing Diversity. Part I: Held as Part of HCI International 2009*, San Diego, California, United States of America, July 19–24 2009, Springer-Verlag 2009, pp. 625–635.
3. P. Ramsden, Learning to teach in higher education, 2nd ed., RoutledgeFarmer (Taylor & Francis), London and New York, 2003.
4. T. Winograd, What can we teach about human-computer interaction? *CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, Washington, United States of America, April 1–5, 1990, ACM, pp. 443–448.
5. J. Aberg, Challenges with teaching HCI early to computer students, *Proceedings of the fifteenth annual conference on Innovation and technology in Computer science education*, Ankara, Turkey, June 28–30 2010, ACM, pp. 3–7.
6. B. Plimmer and R. Amor, Peer teaching extends HCI learning. *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education,* Bologna, Italy, June 2006, *ACM SIGCSE Bulletin*, **38**(3), pp. 53–57.
7. S. Greenberg, Teaching human computer interaction to programmers, *Interactions*, **3**(4), 1996, pp. 62–76.
8. G. Fischer, User modeling in human–computer interaction. *User modeling and user-adapted interaction*, **11**(1), 2001, pp. 65–86.
9. S. Inzunza, A. Mejia, R. Juarez-Ramirez and M. Gomez-Ruelas, Implementing user-oriented interfaces: From user analysis to framework's components, In *Proceedings of the 2011 International Conference of Uncertainty Reasoning and Knowledge Engineering (URKE),* Bali, Indonesia, *IEEE*, 2011, **1**, pp. 107–110.
10. A. Mejía, R. Juárez-Ramírez, S. Inzunza and R. Valenzuela, Implementing adaptive interfaces: a user model for the development of usability in interactive systems, In *Proceedings of the CUBE International Information Technology Conference*, Pune, India, ACM 2012, pp. 598–604.
11. J. Nielsen and M. Kaufman. *Usability Engineering*, Academic Press (Elsevier), United States of America, 1993, pp. 115–155.

Andrés Mejía. Graduated from the Autonomous University of Baja California (UABC, in Spanish) in 2007 receiving his bachelor's degree in Computer Engineer with a specialty in Software Engineering. Currently finishing his M.S. Degree in Computer Science with a focus in Software Architecture and Human-Computer Interaction under the direction of professor Reyes Juárez. Since 2010 he teaches part time at the School of Chemical Sciences and Engineering, at UABC, Tijuana.

Reyes Juárez-Ramírez was graduated from the Ensenada Center for Scientific Research and Advanced Studies (CICESE, in Spanish), receiving his M.S. Degree in Computer Science in October, 2000. He obtained his Ph.D. in Computer Science on June, 2008 from the UABC.

Since 2002, Dr. Reyes Juárez-Ramírez has taught fulltime at the School of Chemical Sciences and Engineering, at UABC, Tijuana. Since 2005, he has been an active evaluator in the accreditation process of engineering undergraduate programs, which is regulated by the Counsel for the Accreditation in Engineering Teaching (CACEI, in Spanish). Since 2009 hi's a member of the Researchers National System (SNI, in Spanish) in México.

His areas of interest in research are Formal Aspects of Software Engineering, Metrics for Software Quality, Process Models, Human-Computer Interaction, Usability Engineering, and Agile Methodologies for developing Mobile Applications. Another area of his interest is concerned with the teaching of these disciplines. He has twice received the "Best Paper" award (2005, 2007) as first author in International Conferences sponsored by the International Institute of Informatics and Systems, and two more times as second author.