International Journal of Engineering Education Vol. 28, No. 1, pp. 202–208, 2012
Printed in Great Britain

0949-149X/91 $3.00+0.00

Improving Motivation in Learning Programming Skills for Engineering Students*

JOSÉ M. JEREZ[1], DAVID BUENO[1], I. MOLINA[2], DANIEL URDA[1] and LEONARDO FRANCO[1]
[1] Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga, Campus de Teatinos S/N, 29071 Málaga, Spain. E-mail: {jja, bueno, durda, lfranco}@lcc.uma.es, aim@dte.uma.es
[2] Departamento de Tecnología electrónica. Universidad de Málaga, Campus de Teatinos S/N, 29071 Málaga, Spain.

Engineering degree students sometimes feel that programming courses are not particularly relevant to their main subject and consequently are unmotivated and underperform. In view of this fact therefore, we have realized that in order to motivate them, programming practices must be related to other areas of their degree, in accordance with some principles of the EHEA (European Higher Education Area) process currently being implanted in European universities. This paper presents the work done over a ten year period in which we have tried to motivate the participation of the students studying Telecommunications Engineering in the University of Malaga during their studies. Conceptual maps produced as the result of a coordinated process between several professors in the different areas of the syllabus were used to design the practices and as a result the student success rate has improved while absenteeism has dropped.

Keywords: student motivations; programming courses; European Higher Education Area; conceptual maps for curricula

1. Introduction

Teachers often employ motivational techniques as they can have several effects on how students learn and how they behave towards subject matter [1]. Motivation can direct behavior toward particular goals, lead to increased effort, initiation, persistence, performance, and enhance cognitive processing. A motivated student not only spends more time and effort on learning but also finds the process more rewarding [2–4]. Because students are not always internally motivated, they sometimes need situated motivation that can be found in environmental conditions that the teacher should create. Moreover, a strong correlation has been observed between the degree of student motivation and the acquisition of knowledge and academic performance [5, 6]. In some works the motivation has been achieved using simulators [7].

This work presents the experience of a team of teachers in the design and application of a novel approach to improve the motivation of students in a computer programming course pursuing a Telecommunication Engineering degree at the University of Malaga. In this context, even though acquiring proper programming skills is valuable by itself in an engineering degree, we considered that the learning process could be facilitated if the concepts are introduced within a more practical approach, with exercises related to the topics of other subjects of their degree. An initial study based on a student survey showed that lack of motivation appeared to be the main cause of academic failure. Furthermore, the survey indicated that these students were much more motivated on subjects with greater relevance to their degree curriculum, such as for example radio antennas, or communications. Moreover, the authors found that students were not always aware of the importance and usefulness that computer programming has nowadays in the field of communication technologies.

An initial approach to improve student motivation was first proposed by us in 2001 [9], where the development of practical exercises based on low difficulty games was introduced as an alternative to the traditional methods used when teaching computer programming courses (palindromes, lists, databases, etc.). The aim was to increase student motivation by making the task of solving exercises more fun. This project was developed between 2001 and 2004, with the results demonstrating that, while increased motivation was observed for one group of students (e.g. some were motivated by the use of the games with their families), another group existed of non-motivated students who were not attracted by the development of practical exercises based on computer games. Several authors have also dealt with the problem of student motivation in recent works [2, 3, 10]. In [2], the United States Military Academy added a visual design tool, where the students can test their algorithms. They used visual Java libraries where students could have instant feedback, of their work on graphic problems through testing. In [8] [10] the authors propose the adaptation of the students' personal goals according to their weaknesses or strengths in

* Accepted 9 August 2010.

programming. These personal goals show the students somewhere to aim for 'point to go' that could motivate them. The motivation of the students to take a programming course was analyzed in [3], where aspects such as achievement, enjoyment, learning, programming, aspirations, etc., of those engineering students not motivated to learn programming were taken into account. In recent years, the problem of student motivation in programming courses for non-computer science students has been also analyzed in [11, 12].

In 2004, with the start of the discussions about the future creation of the EHEA [13, 14], the authors studied the application of some guidelines to motivate the students. From a practical point of view, the EHEA is considerably changing the way of teaching from a teacher-centered perspective to a much more practical and student-centered one, where the teacher is considered more of a facilitator or a guide. This change also involves moving from traditional 'chalk and talk' lectures towards more practical classes with increasing student involvement, including debates, exercises and the use of new technologies (web search, virtual spaces, wiki spaces, etc.) to develop the different student abilities required. In relationship to student's motivation, the use of practical exercises and compliance with the core knowledge areas of the degree is included within the EHEA guidelines. To this end, several teachers of the degree were contacted and questioned on aspects of their subjects that could be implemented in a computer programming course. The concept maps technique was then used to analyze the survey results and define a practical framework in which the development of theoretical programming skills could be integrated.

Concept maps represent a technique commonly used by educators and researchers alike, to analyze and diagram relationship among concepts reflecting the importance of hierarchical knowledge structures [16, 17] (the more general the concept, is the closer to the top of the knowledge hierarchy). Many of the studies of concept mapping are done in the context of education literature. Usually, concept maps are reported as having a positive effect [18–20] on the learning process. In fact, some of the issues regard the use of concept maps for a set of tasks, which includes tutoring, improving learning skills as well as evaluation. Therefore, concept maps are used as an educational tool to give students and instructors opportunities to correct common errors or misconceptions about the courses, or are used as tests to assess student learning after the course has been analyzed, or to measure progressive changes in student learning over time [21–23]. Specifically, in the field of engineering, Turns [24] illustrated the use of concept maps for addressing course-level assessment issues (monitoring progress, assigning grades, and exploring students' perceptions of learning in a course).

2. Methodology

2.1 Concept map design

The methodology is based on the use of concept maps as a tool to: 1) integrate specific knowledge of computer programming and general knowledge into the framework of the Telecommunications Engineering degree, and 2) to design practical exercises whose resolution is attractive to students. The implementation was structured in four phases:

1. Design of a concept map for each subject of the degree that could benefit from the knowledge acquired by students on the LCP (Laboratory Course for Programming) subject. A great effort was made by the authors to review the syllabus for all subjects and subsequently select those that could be incorporated into the map. Then, a questionnaire was distributed to the teachers in charge of these subjects, asking them to respond to the question:

   'List three programming concepts that in your opinion a student should know and understand well before attending the course you teach. Furthermore, describe some of the topics of your course that are related to those concepts, establishing the relationships between these topics and concepts.'

   Upon receiving the responses from the teachers, we could see some telecommunications concepts directly related to three of the programming concepts. Additionally, on the same concept map it was possible to see the relationship between the concepts. The teacher contributions were considered an important requisite, and it was possible to guarantee that the relationship between telecommunications concepts, in general, and programming skills in particular were properly established. Table 1 presents a summary of the responses given by the lecturers when asked about the main topics of their subjects in relationship to programming skills. These data was further used to build the map of Fig. 1.

   Next, a detailed description of the links among LCP and Telecommunication concepts (Fig. 1) is provided based on teacher's questionnaire responses:

   - AD/DA converters: the relationship between the input analog level and the digital output in a AD converter can be modeled by using

Table 1. Main topics of the telecommunications subjects related to programming skills, extracted from lecturers' questionnaire

Subject (course)	Main topics
Digital Audio (2nd course)	AD/DA converters. Digital filters.
Communications Lab. (2nd course)	Simulation of communication systems with analog and digital modulations.
Electro-acoustics Lab. (2nd course)	Psychoacoustic experiments. Acoustic noise measurement.
Digital Image Processing Lab. (3rd course)	MATLAB Image toolbox basics. Sampling, quantization, Fourier Transform. Image enhancement. Image restoration and compression.
Digital electronics (1st course)	Boolean algebra. Basic combinational blocks. Basic sequential blocks. VHDL fundamentals.
Communication networks (4th course)	Client-server applications.
Intelligent systems (3rd course)	Expert systems. Auto-organized algorithms. Supervised algorithms.

simple shifts, sums, divisions and round operations. Modeling these circuits would involve the use of control flow statements (*Selection and Iteration sentences*), and a proper use of *Simple data types*.

- Digital filters: FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) are designed as part of the work class by using MATLAB® language. The professor of this subject was mainly concerned on the use of some good programming practices, as the structure and modularity of the code (*Subprograms*), as well as the verification and the debugging of its functionality. In addition, the use of control structures and *Abstract data types* were also pointed as key aspects.
- Image processing: this subject involves the development of basic image manipulation and processing (*Matrix manipulation, Simple data types and Iteration sentences*).
- Psychoacoustics and noise measurement: the psychological correlates of the physical parameters of acoustics accounts for the fact that hearing is a sensory and perceptual event. On the other hand, noise measurement attempts to quantitatively determining one or more properties of acoustic noise. For this purpose, the LCP concepts linked were Modularity *(Subprograms),* control structures and *Arrays*.
- System simulation: this subject aims at simulating different digital and analog modulation techniques used on communications systems. One-dimensional arrays, modularity of the code and simple data types were selected as relevant concepts for the students to be trained on for the formal aspects of the modulation theory.
- Digital electronics: Fundamental of electronics, basic combinational and sequential blocks, basic HDL (Hardware Description Languages, VHDL) and PLD (Programmable Logic Devices) use concepts as Boolean variables (*Simple data types*) and control flow statements.
- Client-server applications: *Abstract Data Types* and *Subprograms* important in those applications to organize the server data and remote methods. *Recursion* is very commonly used for example in JavaScript timers.
- Expert systems: *Matrix and Arrays*, and *Selection Sentences* are used, for example, for the construction of decision trees based models. *Recursion* techniques are applied to the design of searching algorithms in hierarchical networks.
- Supervised and Auto-organized algorithms: *Iteration and Selection Sentences* together with complex data structures (*Arrays and Matrix*) are used to implement neural network learning algorithms.

The next step is to unify all individual concept maps into a global one (see right block in Fig. 1), in which we have the most important telecommunication concepts.

2. Making the concept map for the LCP subject. This concept map included considerably more detail and information than the previous maps in phase one, and was created by teachers responsible for the LCP subject. The map reflects all the programming skills which

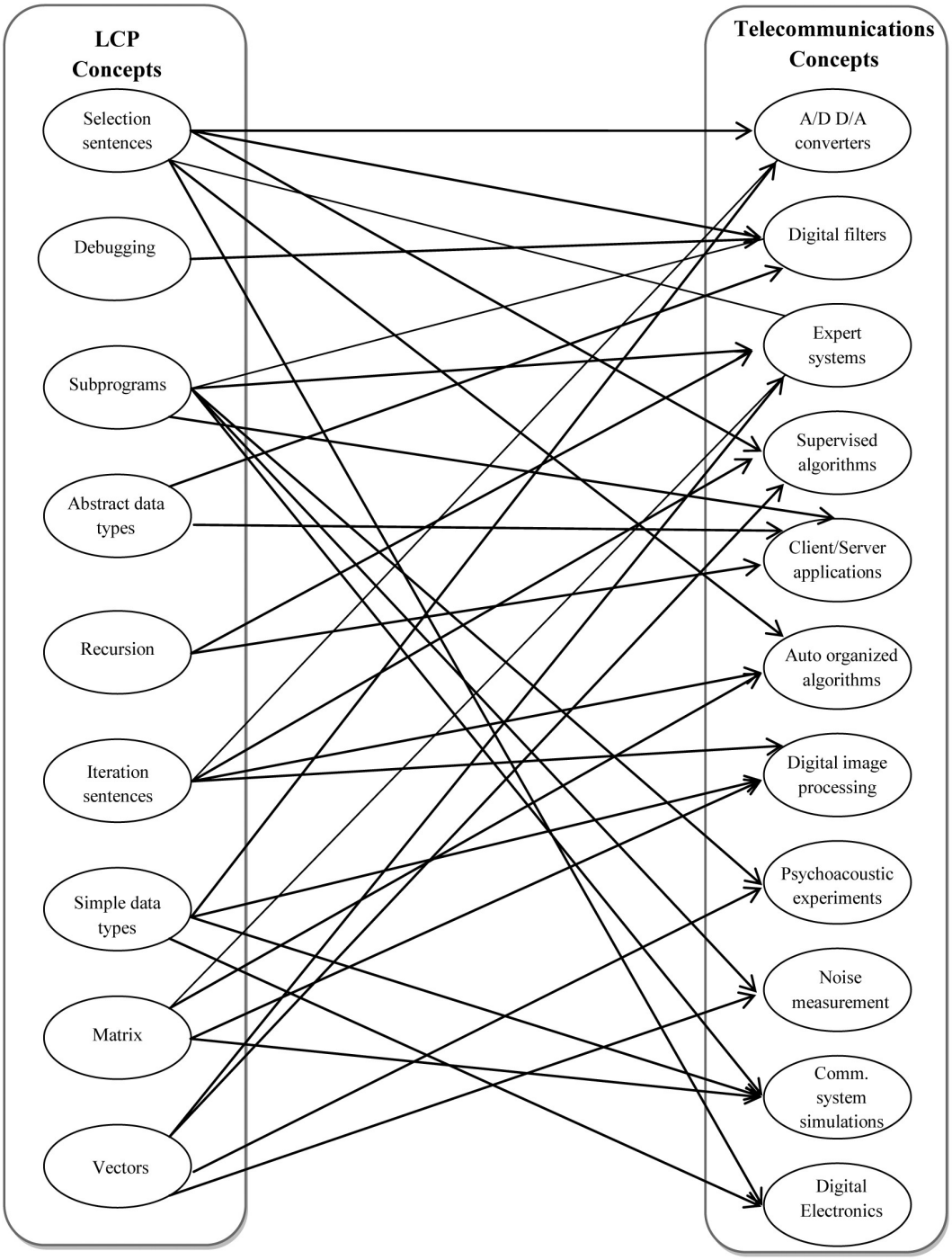

Fig. 1. Relationship table linking topics of the Laboratory Course for Programming with relevant concepts in Telecommunications.

should be acquired by the students: Simple data types, Arrays, Matrix, Selection and Iteration sentences, Subprograms, Recursion, Abstract data types (stacks, lists, trees, queues) and debugging.

3. Unifying both concept maps into just one global map (Fig. 1), in which programming concepts appeared linked to generic concepts for the degree. From this map, LCP teachers would be able to design practical exercises for each programming concept within the framework of generic knowledge for the Telecommunications Engineering degree.
4. The efficacy of the concept map in generating laboratory practices for the LCP course was evaluated.

2.2 Evaluation

The relationships between the two types of concepts are established according to the experi-

ence of the teachers responsible for the subjects involved in this project. However, this does not imply that those relationships were correct, and thus a thorough evaluation of the results should be done in order to readjust the relations between the telecommunications and programming concepts if necessary. The evaluation process takes into account the following aspects:

- Results from surveys given to the students throughout the semester.
- Attendance to classes in comparison to previous years.
- Evaluation of students' abilities to tackle problems related to areas of telecommunication and to provide a correct solution, according to the complexity of the practical exercise.
- Final grades obtained by the students in comparison to previous results.

All these parameters have provided the authors an idea of the proper development of the course following the degree of relationships established on the map. Also, the evaluated parameters allowed readjusting these relationships between concepts to achieve a higher efficacy on further iterations.

3. Results

Different practical exercises were designed using the methodology described in section 2 between 2004 and 2009, which were usually quasi-real world telecommunication applications. Some examples are: 1) Design of an intelligent air controller to avoid collisions between planes using geometric concepts and communication protocols; 2) Implementation of signal processing algorithms, in which the students implemented different audio filters (high pass, low pass, etc) from real audio file in wav format. The exercise involved the programming concepts of selection and iteration sentences, modularity via subprograms and abstract data types, which are incorporated into the second course of 'Digital Audio'. In this subject, the students use Matlab libraries to design digital filters, so the exercise proposed in the LPC subject would complement the knowledge of this discipline (Digital Audio) through implementation of the same filters, but on a lower abstract level and using C++ as the programming language; 3) Developing of image processing algorithms (mean, median and border detection) to process real image file in bmp format; 4) Signal analysis algorithms to process and detect earthquake from seismograph values, or to detect heart attack from an electrocardiogram signal as input file.

The academic results measured as absence and failure rate are shown in Fig. 2. The graph also shows the actual number of students who failed the final test in LCP between academic years 1996 and 2008. The programming languages used to implement the practical exercises in different periods are

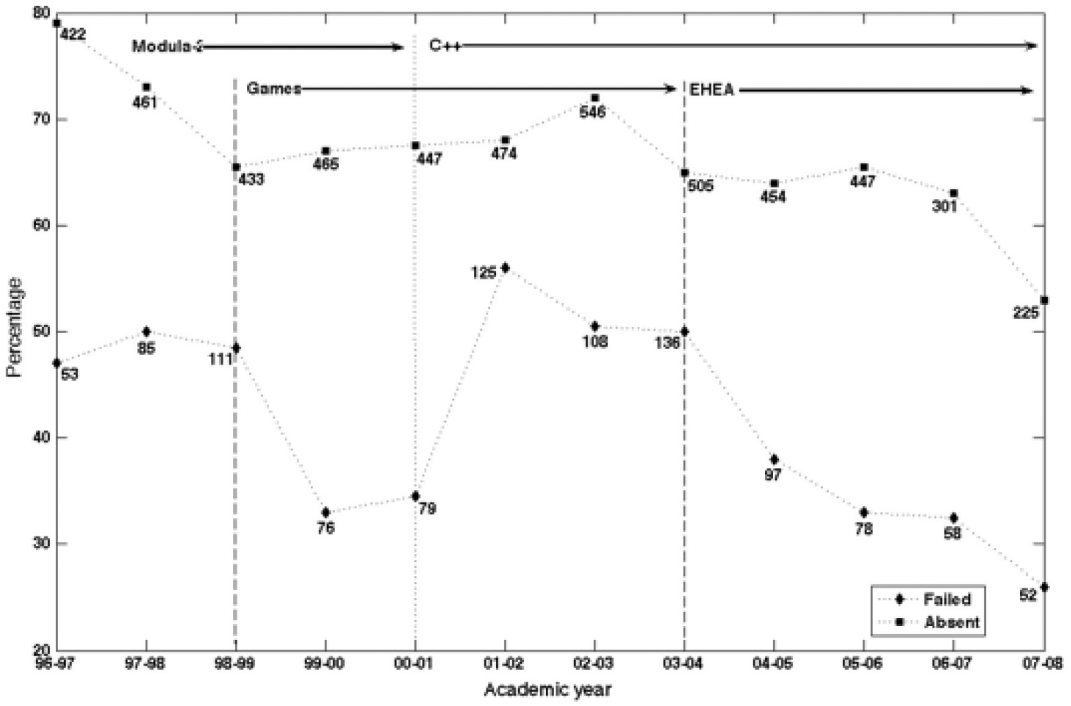

Fig. 2. Exam failure and absence rate for the subject LCP between 1996–2007. The numeric label in marks represents the actual number of students.

also marked in the chart. A change in 2001 in the programming language (from Modula-2 to C++) produced an increase in the failure rate in the following academic year, fact that can be explained by the large proportion of students from previous years who had to learn the new language but failed in the process of adaption.

A first pedagogical methodology change was introduced in 1998, based on the development of practical exercises inspired by computer games, but it did not have a clear positive impact on student performance. From 2003, coinciding with the application of the methodology proposed in this work, there was a clear drop in the failure and absenteeism rates, with the percentage of failed students dropping from 50% in 2003 to 26% in 2008. Furthermore, it is important to note that the percentage of students passing the final exam with 'C' or 'Pass' fell from 86% in 2001 to 58% in 2008, while the percentage of grades 'B' or 'Good' rose from 10% to 30%, and grades 'A' or 'Outstanding' rose from 4% to 12%.

Regarding students' motivation (a feature difficult to be measured but that it can be related to academic performance [5–6]), the lecturers note that the received feedback has improved together with the attitude of the students in the classroom. This perception is corroborated by the fact that more students return to their LCP teachers for further guidance for their final degree project. We note that problems with students' motivation in programming courses for non computer science students have been previously observed in other universities [11, 12], and even if the context of these analyses was different to ours, many similarities can be found and our methodology might be applied and tested in these different environments.

4. Conclusions and future work

This paper presents the work done over a ten-year period in which a group of teachers have tried to improve the motivation of the students of the Telecommunications Engineering degree at the University of Malaga. The work proposes a novel pedagogical methodology to design practical exercises for the subject LCP (Laboratory Course for Programming), based on the use of concept maps as a source of topics around which class exercises are designed. The design was done based on a questionnaire answered by the lecturers of the core subjects of the degree not directly related to programming. From the actual responses, summarized in Table 1, a concept map (Fig. 1) was built and then used for the design of the practices.

The results of the changes on the teaching methodology implemented from the year 2003 and on, measured by the evolution of exam failure and absence rate, lead to a clear drop in the failure and absenteeism rates, with the percentage of failed students dropping from 50% in 2003 to 26% in 2008. Figure 2 also shows that a first implemented change of the programming practices based only on the development of exercises inspired by computer games did not have a clear positive impact on student performance (years 2001–2004).

As a final conclusion, we believe that the proposed methodology constructed from the exchange of information with teachers of other subjects of the degree can be positively used to design programming practices more related to the students' interests, leading to an increase motivation and better academic performance.

For the future we propose to design an automatic system for the generation of practical exercises from a set of programming concepts that will include the implementation of different learning and optimization algorithms to automatically readjust the relationships between concepts.

Acknowledgements—The authors acknowledge support through grants PIE07-083 (Department of Innovation in Education—University of Málaga, Spain) and P08-TIC-04026 (Junta de Andalucía). The project described in this paper has been led jointly by the professors Jose Jerez and David Bueno.

References

1. P. R. Pintrich and D. H. Schunk, *Motivation in education*, Englewood Cliffs, NJ. Prentice Hall, 1996.
2. A. R. Benjamin, G. John and R. Scot, Problem solving through programming: motivating the non-programmer, *J. Comput. Small Coll.*, **23**(3), 2008, pp. 61–67.
3. T. Jenkins, The motivation of students of programming, *SIGCSE Bull.*, **33**(3), 2001, pp. 53–56.
4. J. E. Ormond, *Educational Psychology: Developing Learners*, Fourth Edition, Merrill Prentice Hall, 2003.
5. C. A. Benware and E. L. Deci, Quality of learning with an active versus passive motivational set. *American Educational Research Journal*, **21**(4), 1984, pp. 755–765.
6. K. M. Edmondson, Assessing science understanding through concept maps. In: J. J. Mintzes, J. H. Wandersee and J. D. Novak (eds), *Assessing science understanding. A human contructivist view*, Academic Press, New York, 2000, pp. 15–40.
7. A. García-Beltrán, S. Tapia, R. Martínez y J. A. Jaén, Simulator for a Multi-Programming Environment for Computer Science Learning and Teaching, *International Journal of Engineering Education*, **25**(2), 2009, pp. 221–227.
8. S. Ratnajeevan H. Hoole. Programming Skills in Graduate Engineering Classes: Students from Disparate Disciplines and Eras. *The International Journal of Engineering Education*, **26**(3), 2010, pp. 593–601.
9. D. Bueno, J. Garralón, J. Jerez and A. Maña, *Aprendizaje Lúdico en Laboratorio de Programación*, JENUI 2001, Mallorca, España, 2001.
10. L. Raymond and L. John, *First year programming: let all the flowers bloom*. In: Proceedings of the fifth Australasian conference on Computing education—Volume 20 Adelaide, Australia: Australian Computer Society, Inc. 2003, pp. 221–230.
11. Kurkovsky, S. *Improving Student Motivation in a Computing Course for Non-Majors*. In: Proceedings of the Interna-

tional Conference on Frontiers in Education: Computer Science & Computer Engineering (FECS 2006), Las Vegas, Nevada, USA, 2006.
12. A. Forte and M. Guzdial. Motivation and Nonmajors in Computer Science: Identifying Discrete Audiences for Introductory Courses, *IEEE Transactions on Education*, **48**(2), 2005, pp. 248–253.
13. Official Bologna process website. http://www.ond.vlaanderen.be/hogeronderwijs/Bologna/, Accessed 11 July 2010.
14. The Bologna declaration. http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf, Accessed 11 July 2010.
15. El Espacio Europeo de Educación Superior (EEES). http://www.educacion.es/espacio-europeo-educacion-superior.html, Accessed 11 July 2010.
16. J. D. Novak and D. B Gowin, *Learning How to Learn. New York and Cambridge*, Cambridge University Press, 1984.
17. J. D. Novak, *Learning, creating, and using knowledge: Concept Maps(R) as facilitative tools in schools and corporations*, Mahweh, NJ, Lawrence Erlbaum Associates, 1998.
18. I. M. Kinchin, Using Concept Maps to reveal understanding: A two-tier analysis. *School Science Review*, **81**, 2000, pp. 41–46.
19. I. M. Kinchin. Concept Mapping in biology. *Journal of Biological Education*, **34**(2), 2000, pp. 61– 68.
20. P. G. Markow and R. A. Lonning, Usefulness of Concept Maps in college chemistry laboratories: Students' perceptions and effects on achievement. *Journal of Research in Science Teaching*, **35**(9), 1998, pp. 1015–1029.
21. K. M. Edmondson, Assessing science understanding through concept maps. In: J. J. Mintzes, J. H. Wandersee and J. D. Novak (eds), *Assessing science understanding. A human contructivist view*, Academic Press, New York, 2000, pp. 15–40.
22. J. E. Trowbridge and J. H. Wandersee, How do graphics presented during college biology lessons affect students learning? *Journal of College Science Teaching*, **26**, 1996, pp. 54–57.
23. J. D. Wallace and J. J. Mintzes, The concept map as a research tool: Exploring conceptual change in biology. *Journal of Research in Science Teaching*, **27**, 1990, pp. 1033–1052.
24. J. Turns and C. J. Atman, Concept Maps for Engineering Education: A Cognitively Motivated Tool Supporting Varied Assessment Functions. *IEEE Transactions on Education*, **43**(2), 2000, pp. 164–173.

José M. Jerez finished his PhD studies in computer science in 2003 at the University of Málaga, Spain where he is currently an associate professor. His research interests lie in the areas of computational intelligence, image analysis and bioinformatics. In particular, he is developing prediction software for biomedical problems using artificial intelligence techniques in collaboration with the Málaga university hospital. He has participated in more than 15 research projects funded by international and national boards and he belongs to several reviewing scientific committees. He also has been involved in different education projects related to the introduction of the EHEA in Spanish universities.

David Bueno Vallejo obtained his computer engineering degree (1996) and his PhD (2003) at the University of Málaga, Spain. He is presently working as associate professor in the Department of Computer Science at the same University. He has spent more than twelve year teaching programming at the Engineers Degrees of Telecommunications and Computer Science. His research is related to programming mobile devices, interactive digital TV and recommender systems.

Ignacio Molina obtained his Master's degree in 1994 and his PhD in 2001, in Telecommunication Engineering at the University of Malaga, where he currently works as an associate professor and teaches digital electronics. His research interests lie into the areas of image processing, computational intelligence and bioinformatics. He has participated in more than 10 research projects and authored more than 20 papers in journals and international conferences.

Daniel Urda has recently finished his master studies in computer science at the University of Málaga and is now pursuing a doctoral degree. His research interests include neural networks, bioinformatics and object oriented programming languages.

Leonardo Franco did master and PhD studies in physics at the University of Córdoba, Argentina analyzing the generalization properties of feed-forward neural networks. He has been a postdoctoral fellow at SISSA, Italy and Oxford, U.K. where he became involved in Computational Neuroscience. He is at present at Málaga University, Spain working on neural networks, their applications to biomedical problems and also in computational neuroscience. He has authored more than 40 publications in journals and international conferences and has been involved in two education projects related to the introduction of the EHEA in Spanish universities.