CUPI2 COMMUNITY
Promoting a Networking Culture that Supports the Teaching of Computer Programming

Jorge Villalobos, Nadya Calderón and Camilo Jiménez
Department of Systems Engineering, University of Los Andes, Bogotá, Colombia

Keywords: Online Community, Computer Programming Education, Teaching Programming Support, Knowledge Management, Active Learning.

Abstract: Difficulties in computer programming education have prompted the need to tackle the teaching and learning of programming from alternative pedagogical approaches. However, modern engineering education demands more than simply working around the students' learning process. Others who play important roles in academia face substantial challenges as they support the management of knowledge and the improvement in teaching within computer science (CS) departments. As a possible solution to these challenges, an emerging online community culture has applied an effective strategy that can guide people working toward common goals. To support teachers and CS departments in charge of computer programming education, this paper presents a social network developed with active learning approaches: the Cupi2 Community, a set of people, policies, resources, contributions, technological mechanisms, and interaction strategies that promote the generation of collective knowledge and offer continuous support to CS educators involved with active learning approaches.

1 INTRODUCTION

Following the guidelines of (ACM, 2001), courses in computer programming consist of three basic courses (CS1, CS2, and CS3) in most computer engineering and computer science (CS) programs. Some of them (CS1, and usually CS2) are core courses taught in other engineering programs as well as math and physics departments. Thus, computer programming courses typically have a large number of students with a variety of personal and professional interests and backgrounds.

Worldwide, difficulties of teaching and learning in computer programming courses have been a recurrent topic for the last twenty years. Traditional strategies in teaching have been primarily focused on covering programming fundamentals rather than on generating abilities relevant to graduates for their myriad professional domains. As a result, the development of abilities to understand and abstract problems, analyze abstractions, model solutions, and build those solutions using technological tools have have been left aside, expecting that writing many programs will lead to their development (Woodley, 2007).

We consider that it is important to recognize separate challenges and difficulties from the perspective of three actors involved in computer programming courses: students, teachers, and institutions, specifically CS departments. First, students have been the focus of recurrent issues relating to their frustration and lack of motivation, reflected in frequent comments such as "I do not feel like I belonged," "classes were unfriendly," or "classes were boring" (Biggers et al., 2008). Furthermore, students must learn to program, much as they learn to write: they need to understand the intention, receive detailed feedback, and rewrite and receive more feedback. Instead, traditional approaches to teach programming confront students to lectures explaining general algorithms and grad them by results of programmed functional tests (Woodley, 2007).

Secondly, teachers, often with different academic profiles, have typically developed their own personal strategies, but they must plan classroom lectures and laboratory practices that fit the approaches adopted in the CS courses as means for dealing with student difficulties. In this way, as they must design complementary learning objects that support these approaches, they must also develop activities such as

Villalobos J., Calderón N. and Jiménez C. (2009).
CUPI2 COMMUNITY - Promoting a Networking Culture that Supports the Teaching of Computer Programming.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 78-85
DOI: 10.5220/0001977000780085

planning lectures and designing resources. Such activities are generally developed individually by each instructor leaving team discussion as an informal practice. Moreover, there is a poor culture around networking practices between teachers as well as there is a lack of implanted reflection mechanisms that encourage teachers to share results of their implemented strategies or practices or to share their learning objects and materials.

Finally, CS departments are confronted with the need to manage knowledge generated in programming courses and support the massive admissions each semester. For example, in the last four years at the University of the Andes, CS courses have comprised 6,798 students distributed in 341 sections, consisting of 35 sections of CS1 and 20 sections of CS2 each semester. These numbers indicate that about 1,200 students from eight different programs are enrolled in every class. The instruction of these courses generally involves more than 45 different teachers, including graduate students that practice teaching as part of their professional training. Hence, knowledge management has grown extremely complex, particularly in light of the provisional permanency of graduate students that instruct CS courses and the lack of mechanisms to capture their experiences.

Clearly, the problems faced by students, teachers, departments, and CS education in general require some creative solutions. Since 2004, the University of Los Andes has developed a set of methodologies, resources, and experiences that support the learning process, called the Cupi2 project (http://cupi2.uniandes.edu.co). The objective of Cupi2 is to find new approaches to teaching computer programming and to support student learning. The project is currently being applied in more than 25 higher education institutions in Colombia. Evaluations of the Cupi2 project, however, have identified important challenges related to teachers and CS departments.

This paper presents the Cupi2 Community, an online social network developed to promote networking practices among computer programming teachers and to support relevant processes related to teaching programming under active methodologies: the teachers' adaptation process to the new methodologies introduced by Cupi2; the process of planning the instruction of CS courses; the design of the required learning resources and the process of supporting the instruction of these courses; and a structured process of capturing feedback from teachers' experiences.

The community provides support, policies, and tools for each of these processes. The community also provides mechanisms that enable members of the CS community to share and reuse learning resources and to exchange knowledge and experiences around the teaching of computer programming. It also empowers departments with the ability to manage knowledge and to gain experience and facilitates the tracking and evaluation of evolutive results.

The general structure of this paper is as follows: Section 2 provides an overview of specific problems confronted by students, teachers, and CS departments within the computer programming teaching and learning. Section 3 details our solution to challenges faced by both teachers and CS departments where the Cupi2 community is depicted as a social networking culture supporting computer programming teaching. Section 4 reports the results of three years of application of the project inside our community, and Section 5 discusses related work and then concludes.

2 CUPI2. A PEDAGOGICAL APPROACH TO TEACHING PROGRAMMING

Evidence on high rates of failure and students dealing with low motivational issues related to a sense of a lack of purpose and low grades (Jenkins, 2001) reveal the need to face programming from alternative approaches. Although researchers have proposed various tools and strategies to overcome these problems (Gearailt, 2002) (BlueJ, 2008) (Robocode, 2008), none of them have identified workable solutions. We have tried simple and direct solutions (using different textbooks, teaching alternative programming languages, reordering subjects, developing classes in computer labs) but none have resulted in effective improvements.

The foundation of the Cupi2 project is a pedagogical model based on four main components that allow the construction of a balanced solution (figure 1).

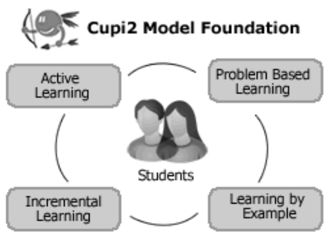

Figure 1: Cupi2 pedagogical model.

First, *Active Learning* engages students to take on the main role in their learning process. Thus, teachers act less as instructors and more as promoters of activities that ensure the generation of relevant abilities to analyze problems and model solutions. Students have to do more than just listen: they must read, write, discuss, and engage in solving problems. More importantly, they must be actively involved in such higher-order thinking tasks as analysis, synthesis, and evaluation (Bonwell, 1991). Secondly, they are expected to confront problems developed from motivational issues through *Problem Based Learning*. Such learning requires that students continually face problems that reflect real world challenges. In this way, concepts on a subject are explained through their relation with specific parts of these problems

The third element in the model is the inclusion of *Incremental Learning*, whereby students are able to generate abilities and acquire knowledge distributed in several levels. Cupi2 structures 18 levels from the beginning of CS1 to the end of CS3. The levels allow the instruction of CS concepts from the fundamentals to gradually more complex structures. In this way, students continuously reinforce knowledge and abilities developed incrementally through each level. The last component is the Learning by Example approach, in which students have access to examples of best practices and common solutions. The Cupi2 project currently provides more than 150 examples to students.

The basis of the learning evaluation model is the four components. During each of the 18 levels, students work around a specific problem (guideline problem) in which they apply the subject matter of each level. In each case, students receive an incomplete application. Depending on the complexity of the guideline problem, what they receive is more or less finished. Their goal is to design and develop the missing parts until they obtain a fully working application. In addition, students also present a written test and develop a laboratory practice in which they use knowledge and abilities that they have obtained up to that current level. Both a written test and laboratory practice accompany the guideline problem.

Experience and results around Cupi2 model have been exposed in (Villalobos, 2006). The project has been awarded the 2007 Colombian Informatics Award by the Association of Colombian Computer Engineers (ACIS) based on the quality of its learning objects, the amount of materials built to support the learning process, and the impact on the academic environment. The university has tracked the relevant indicators of success, leading to the following conclusions:

- The number of students who fail computer programming courses has declined by 49%.
- Evaluations of computer programming courses made by students have improved by more than 21%.
- The average grade in computer programming courses has increased by more than 11%.

2.1 Challenging Teachers

The introduction of active approaches in Cupi2 model has generated different necessities to the other actors involved in the learning experience. Teachers require the planning of their lectures and laboratories using different learning resources and fostering collaboration with fellow teachers. Thus, the Cupi2 approach has answered these challenges by designing learning sequences.

A learning sequence (figure 2) is an ordered set of activities followed to achieve specific learning objectives of a subject. In Cupi2, a sequence is designed after the objectives of a specific level and it uses the level's guideline problem. This sequence helps the teacher to structure several classroom lectures and labs with activities and working materials related to each of the activities. It also motivates teachers to keep records of its own experiences on designed activities.

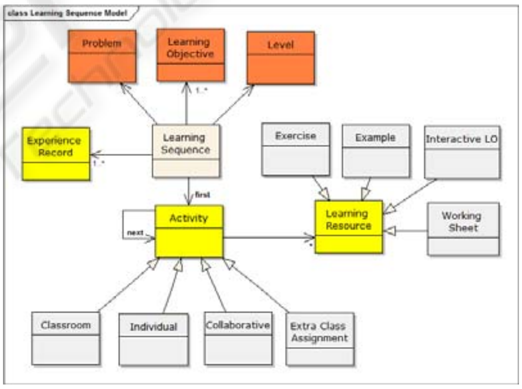

Figure 2: Learning Sequence Model.

In order to provide feedback opportunities and generate an active climate to students, teachers plan learning sequences composing activities of different kind such as: classroom activities, extraclass activities, collaborative, individual, laboratory practices and homework asignments. Each of these activities is supported with an appropriate set of learning resources.

Many important advantages arise from facing the instruction of subjects as structured sequences. First, teachers always keep the learning objectives of the level in mind while preparing their classes. Secondly, they can always perform self-evaluation of experiences obtained from their designed activities.

The result is a continuous contribution to self-improvement and experience that they can share with other teachers, improving their overall practice. Thirdly, since teachers are free to design or reuse their own set of activities and learning resources, learning sequences reflect personal strategies. Finally, the time invested in planning classes and building resources is incrementally reduced as teachers reuse learning objects and sequences of their own or other teachers.

2.2. Teaching Improvement and Knowledge Management in CS Departments

Regardless of how effective new approaches to teaching may be, university administration must take steps to establish a suitable climate for change before significant change can take place (Rugarcia, 2000). The heterogeneous nature of personal profiles and personal strategies and the short-term permanency in departments of some instructors calls for more permanent support.

CS departments must ensure their mature growth as well as offer the required support to their faculties. Hence, departments (as organizations) must develop abilities and strategies to generate, manage, reinforce, and transfer collective knowledge among its faculty members since they represent the main conduit of learning in their departments. When faculty members coordinate actions effectively, they enable and promote knowledge transfer inside their institutions not only to students but to the organization itself (Espejo et al., 1996).

Characteristics behind online community work appear as an adaptable, affordable, and innovative proposal to tackle issues around teachers' adaptation and knowledge management. A proper community culture guided by clear policies allows members' interaction, provides feedback mechanisms, and ensures the collection of contributions from its members. In a teaching environment, this community culture offers advantages to ensure permanent support and knowledge appropriation to teachers.

3 THE CUPI2 COMMUNITY

While an online community can be understood and viewed from a number of diverse perspectives (e.g., sociology, technology, e-commerce, and so forth), those who have been guided by practice and experience have identified four principal elements of an online community: 1) individuals who interact, 2) a shared purpose, 3) policies as protocols or rules that guide interaction among these individuals, and 4) computer systems that support and mediate interaction (Preece, 2006).

The Cupi2 online community promotes a culture of networking and social interaction between computer programming teachers challenged with new methodologies to instruct their subjects. A set of policies have been established to guide the members work and a robust technological infrastructure has been developed.

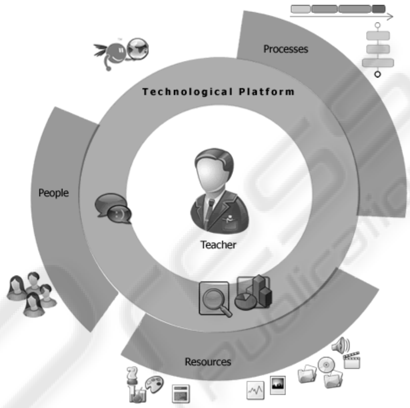

Figure 3: Cupi2 Community Structure.

Figure 3 shows the structure of the individuals, the policies (as teaching processes), the technological platform, and the resources inside the Cupi2 online community. Teachers interact with processes to get them involved with active approaches and to contribute feedback from their experiences. In this way, instructors find processes of mentoring or an introduction to the community in which the basics about rules, platform, systems, and navigation are discussed. It also includes a process that assists with plans for the instruction of their courses. In particular, teachers can find strategies, resources, and tools related to the design of learning sequences for the different levels in their courses. Finally, it outlines a process that guides teachers on how to contribute feedback and reflection of their experiences. The technological platform supporting our community provides mechanisms of interaction among members, search engines for resources, learning sequence editors, and statistics reporting.

3.1 Members of the Community

Members of the Cupi2 Community are computer programming teachers from different CS departments in higher education institutions around Colombia. All members have included active Cupi2 approaches in their curricula; therefore, challenges from interactive learning approaches have become a main subject in their departments. Even though the community provides support for diverse teaching processes, teachers are invited to participate in the community without being forced to follow any of them. They are able to participate in any process according to their specific needs as well as interact directly with other services provided independently. A team of systems' administrators and moderators ensure proper service to participants in the Cupi2 community.

3.2 Community Resources and Contributions

In our teaching environment community, knowledge is represented as a set of resources and experiences contributed by members on the network. The collection and storing of these resources in a structured fashion enables the community to grow and mature. Stored resources are available to search, reuse, and evaluate continually. Inside the Cupi2 community, these resources and working materials have been categorized according to the function they support and the types of standard elements. Accordingly, we built repositories containing resources on three main categories: Learning objects and working resources, portfolios of teachers' experiences and finally statistical information and tracking indicators.

The statistical information repository stores relevant information that can be used to assess the impact and evolution of the project. Similarly, the portfolios' repository keeps records of how teachers are planning the instruction of courses and information about successful and unsuccessful practices or strategies encountered on their experience. Thus, the repository serves as a resource where teachers share learning sequences and feedback documents.

When designing learning sequences, teachers produce and/or reuse two types of resources: activities and learning objects. The activities designed to be executed in the classroom contain exercises, working sheets or videos as complementary resources. Teachers can add extra assignments that students can work on during extra-class time. These assignments contain interactive learning objects that provide students with feedback while they interact with them. Laboratory practices are supported by case studies, tutorials, demos and more interactive learning objects.

The repository of learning objects contains all these resources that support activities. The resources are classified by the kind of work they support. Hence, teachers can find and contribute 1) worksheets, 2) laboratory exercises, 3) interactive learning objects, 4) audiovisual presentations, 5) videos and animation, 6) mind maps with concepts of the courses, 7) tutorials, and 8) exams.

For instance, during 2008-II, one of the member teachers planned to introduce OO concepts and Java during five classroom lectures, two laboratory practices, and two exam sessions (one to be developed in a laboratory and a written one).

During the first and second class sessions, students were assigned a set of individual and peer activities with worksheets in a case study of an application to manage employees' information. The students had to complete the assignments between sessions. After the two classes, students attended a laboratory session in which the teacher planned an exercise that required students to interact with Eclipse IDE, supported by a set of available tutorials. During the next two lectures, the teacher also planned to present some slides and proposed a group discussion. He attached a short video to this activity, presenting differences between the OO concepts of object and class. During the second laboratory, students had to implement extensions to the employees' application in order to practice what they had studied during the class. In the last class session, the teacher proposed working on a summary asking students to review mind maps of these subjects. Finally, the teacher evaluated the progress of the students by administering two exams. Some of these activities were found in other teachers' sequences, available for others to reuse. At the end of the instruction, the teacher reported results on the statistical repository. He also evaluated his own experience and shared the results by writing reports of each activity.

3.3 Governance

As online behavior must be regulated, we defined a set of policies that would guide user interactions and enable a structured navigation across our community. These policies were formulated as: 1) a set of structured processes that guide what members should do during different stages of the instruction process in their courses and 2) a set of rules that regulate communication and members' interaction.

Main access to the community is provided through a web portal (http://cupi2.uniandes.edu.co) composed by sections with different permission levels (public, registered teachers, students, and moderators). The first interaction with the Cupi2 community takes place in a section in which non-registered members can consult online public content. However, to join the community, individuals must register. Once they have registered and their academic affiliation has been verified, members can access and contribute different resources described above.

3.4 Technological Platform

The Cupi2 community platform is a set of web-based systems, developed to provide services related with learning design, resource management, member interactions, and statistical data collection. All the applications are integrated and supplemented continually so that they support the needs of the different processes. New functional requirements are continuously developed to enhance current services or to include new ones. During the past four years, complete platform systems have been designed and developed at the University of the Andes.

Systems supporting the community include the following: 1) a virtual working space manager in which teachers can structure documents and resources with a specific purpose and scope. For example, a virtual space is built with documents related to a complete course while a second is dedicated to a session or even a specific subject. 2) A resource search engine for locating documents and resources of different types. The search allows several criteria to filter results. The engine includes searching over already designed learning sequences. 3) A Mind Maps navigator: a special resources navigator that allows one to search XML structured mind maps describing CS1, CS2, and CS3 concepts.

In order to support the design of learning sequences, systems also include: 4) the learning sequences editor. This editor provides services that assist the structuring of activities and resources. Figure 4 illustrates the system used to structure activities for several classroom lectures (N7 C1 to N7 C4) and laboratory sessions (N7 L1 and L2) on a CS2 course. The sequence is designed to help teachers meet their objectives on subjects of sorting, searching, and automatic testing. Since teachers must describe activities and attach supporting resources, the editor is connected to the resource search engine. The platform also includes 5) forums as our interaction system. Teachers post comments, general questions and suggestions in structured forums available in the community site; and 6) a statistics reporter that keep track of the metrics registered by teachers during the instruction of their courses over the years.

Finally, the system provides a set of administrative tools that support administrators and moderators of the community. These tools appear in tasks such as the assignment of assistants to courses or the update process of news displayed online.

Figure 4: Resources search engine in the learning sequences editor system.

4 METRICS OF A GROWING COMMUNITY

Last three years of community experience have shown that the Cupi2 Community has been growing and improving. To highlight the growth of the structural elements of the project and their impact, we will describe relevant metrics tracked on five categories until November 2008. First, we show the size of the community in terms of the number of current members and the total number of teachers and students that have become involved since 2005 (Table 1); second, we show the size of the community measured in learning resources, which are the result of contributions made by teachers and in-house developments (Table 2).

Table 1: Size of the community (membership).

Total number of universities joined	30
Currently active members	157
Total teachers involved since 2005	198
Students involved with Cupi2 since 2005	6,798

Table 2: Size of the community (learning resources from teachers’ contributions and development).

Examples	78
Laboratory exercises	106
Working sheets	821
Tutorials	15
Videos , animation and demos	172
Interactive learning Objects	33
Mind Maps	108
Exams (written tests and laboratories)	2,959
Data structures	52

Next, we measure the size of the community in terms of software development. Almost a half million LOC's and more than a thousand documents have been developed to provide examples, interactive learning tools, exercises, tutorials, and support tools for both students and teachers (Table 3). After that, we show the size of the community in terms of experience records and teaching portfolios (Table 4). After the instruction of 341 sections of CS courses, the community was able to offer almost a thousand records of experiences and a wide set of learning sequences that are now used as a guide for new members in the community. Finally, we describe community access in numbers (Table 5). Since most of the member universities are located in Colombia and the resources and Cupi2 portal are currently in Spanish, we determined the impact of the community on this set of members.

Table 3: Software metrics of resources developed in our university.

Examples, laboratory exercises and interactive learning objects LOCs.	321,036
Examples, laboratory exercises and interactive learning objects: Test LOCs.	82,963
Public documents describing examples, exercises, functional requirements, and UML models from developed resources.	1,138
Developers' support tools	10

Table 4: Size of the community (experience records and teaching portfolios).

Number of learning sequences contributed by teachers.	434
Teachers' documents of experiences.	974

Table 5: Community impact and use. Statistical visits to the Cupi2 portal during 2008.

Country	Average monthly visits in 2008	Total reported visits for 2008
Colombia	16577	156884
Spain	1302	13154
USA	685	6695
Others	167	4364

5 RELATED WORK

Several online social networks have appeared in recent teaching environments, both from academia and the private sector. In academia, we can find the Curriculum Access System for Elementary Science (CASES). This system is an online environment designed to support new elementary science teachers as they learn to teach inquiry-oriented science effectively (Davis, 2004). CASES is based on three principles: First, teachers must reflect on their teaching; second they must use instructional resources; and finally, they must interact in a supportive community grounded in the study of practice. In this way, elements that support these principles such as inquiry-oriented unit plans, online teacher journals, and online discussion spaces are incorporated in an online learning environment. Together with other resources, each of these elements are also present in the Cupi2 Community. In this case, each one is designed according to the Cupi2 pedagogical model, not the inquiry science pedagogy model suggested in CASES.

Another social network in the teaching environment is described in Hmelo-Silver et al. (2005), which presents the Elementary and Secondary Education Project (eSTEP). This project is a problem-based learning online environment for teacher education grounded in three critical elements: A learning sciences conceptual model, a learning planning/design component, and a connection between the first two elements. Thus, the eSTEP system provides video cases of classroom practice, online learning science hypertext, and a collaborative problem-based whiteboard. This whiteboard is the central learning tool in the environment and represents the focus of negotiation in face-to-face problem-based learning in eSTEP. Just as the whiteboard supports the teacher learning process, learning sequences, interactive learning objects, and examples support teacher education in Cupi2 Community. Each of these elements is complemented by processes and tools that follow and guide the members of our community through not only their individual learning process but also the teaching process.

Several online networks for teaching environments are also found in the private field. These networks often focus on social elements of a community such as sharing and connecting rather than directly supporting the process of learning how to teach. An example of such an online network is the TeachingToday website (TeachingToday, 2008). As the developers of the website are aware that online communities provide a wealth of opportunities for educators to share teaching strategies, ideas, and best practices, they developed an online warehouse of free tips, tools, and resources that are easy-to-use and pedagogically sound. These elements can also be found in the National Geographic Education Network (NatGeo, 2008). This community includes information about professional development opportunities such as workshops and online events that members can attend. One of the most promising online networks developed by the private sector is Yahoo Teachers (Yahoo Teachers, 2008). This network, designed by and for teachers, provides a venue in which teachers from all fields can create, modify, and share standards-based curricula.

6 CONCLUSIONS

In this paper, we have described the main results of the Cupi2 project after four years of innovation in teaching programming. We have exposed the needs of teachers and CS departments taking part in the active methodologies of Cupi2. For this reason, we have built and presented the Cupi2 online community. The community promotes a networking culture among teachers that allows the generation and appropriation of collective knowledge and at the same time provides effective support strategies to all its members. This work provides evidence that Cupi2 has led to relevant changes in departments, reflected by less invested in teacher planning and effective introductory faculty programs. The most relevant impact metrics and growing numbers have been presented as evidence of the work supported by the Cupi2 community and its importance to computer programming teachers. We are continually designing plans to reach more universities in Colombia and developing exercises, examples, tutorials, and services inside our technological platform.

ACKNOWLEDGEMENTS

The Authors would like to thank contributions made by teachers working on project Cupi2 especially to Milena Vela. We would also like to thank the team of graduate students who supported our work as well as to Diego Leal from CIFE.

REFERENCES

Science final report, available at: http://www.computer.org/education/cc2001/final/index.htm. (Accessed October 31th 2008).

Biggers M., et al, 2008. *Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers.* Proceedings of the 39th SIGCSE technical symposium on Computer science education. Portland, USA 2008.

BlueJ. *The Interactive Java Environment*, available at: http://www.bluej.org/. (Accessed May 28th 2008).

Bonwell C, Eison J., 1991. *Active Learning: Creating Excitement in the Classroom* . Jossey-Bass publishers.

Davies E., Smithey J., et al, 2004. *Designing an online learning environment for new elementary science teachers: supports for learning to teach.* Proceedings of the 6th international conference on Learning sciences. Santa Monica, California 2004.

Espejo R., et al., 1996: *Organizational Transformation and Learning: a cybernetic approach to management.* John Wiley & Sons. 1996.

Hmelo-Silver C., et al, 2005. *From parallel play to meshed interaction: the evolution of the eSTEP system.* Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years! Taipei, Taiwan 2005.

Gearailt, A., 2002. *Teaching with Java: Using Java to increase active learning in programming courses.* Proceedings of the inaugural conference on the Principles and Practice of programming, Dublin, Ireland 2002.

Jenkins T., 2001. *The motivation of students of programming*. Proceedings of the 6th annual conference on Innovation and technology in computer science education, Canterbury, United Kingdom 2001.

NatGeo. *National Geographic Education Network – Workshops, Forums and more*, available at: http://www.ngsednet.org/. (Accessed November 5th 2008).

Preece J., 2006: *Online Communities. Designing Usability, Supporting Sociability*. Wiley Editorial. Second Edition 2006.

Robocode. *The open source educational game*, available at http://robocode.sourceforge.net/. (Accessed May 28th 2008).

Rugarcia A, Felder R, et all., 2000: *The Future of Engineering Education I. A Vision for a New Century*. Available at: http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/Quartet1.pdf

TeachingToday. *Teaching Tips, lesson plans and more*, available at: http://teachingtoday.glencoe.com/. (Accessed November 05th 2008).

Villalobos J., Casallas R., 2006. *Teaching/Learning a First Object-Oriented Programming Course outside the CS Curriculum.* 10th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts.

Woodley M., Kamin S., 2007. *Programming Studio: A course for improving programming skills in undergraduates*. Proceedings of the 38th technical symposium on computer science education. Kentucky, USA 2007.

Yahoo Teachers. *Yahoo Teachers! Your passion, our technology*, available at: http://teachers.yahoo.org/. (Accessed November 5th 2008).