

A Study and a Proposal of a Collaborative and Competitive Learning Methodology

Bruno Silva
DSI/ESTSetúbal
Instituto Politécnico de Setúbal
Setúbal, Portugal
bsilva@est.ips.pt

Rui Neves Madeira
DSI/ESTSetúbal
Instituto Politécnico de Setúbal
Setúbal, Portugal
rmadeira@est.ips.pt

Abstract— Competition is evident throughout our society, lives and work. “It transcends time and place, as well as all manners of people”. In this paper, we present a study and a framework proposal for a mixed collaborative-competitive learning environment, applied to a programming course in a University level engineering program. While neither collaboration, nor competition, is inherently good or bad in supporting the learning process, the way teachers employ these strategies in classrooms determines their value in preparing “soon-to-be” professionals. We believe this mixed-based learning approach best serves students as they are able to achieve academic success both in working with others and on an individual basis. Besides the traditional theoretical lessons and laboratory practices of the course, the integration of the framework automates all the support and evaluation processes. It implements both an individual competitive setting and a collaborative one, where intergroup competition occurs, by means of two interconnected modules. We discuss our results and findings in its preliminary use, concluding that the framework promises to have a good impact in the way we teach some programming courses in our Institute.

Keywords-competitive learning; collaborative learning; programming language innovation; automated evaluation framework.

I. INTRODUCTION

Competition is evident throughout our society, lives and work. “It transcends time and place, as well as all manners of people” [1]. In this matter, Universities play an essential role in preparing students for their professional lives, and courses should be adapted to an increasingly competitive and challenging world outside of them. Therefore, although different teaching/learning methods and styles exist nowadays, we emphasize, and describe in the current work, teaching as a mixed collaborative-competitive setting.

While neither collaboration nor competition is inherently good or bad in supporting the learning process, the way teachers employ these strategies in classrooms determines their value in preparing “soon-to-be” professionals [2]. An effective classroom must have the right mix of cooperative learning and competitive learning (along with individualized learning) [3]. This mixed-based learning approach best serves students as they are able to achieve academic success, both in working with others and on an individual basis, enhancing learning

opportunities for students pursuing a professional career [4]. These two learning techniques do relate to the workplace of our times. Often in the business world people work on a variety of team settings, thus, collaborative activities in a classroom can prepare students for the experience of their professional lives. With these activities, students may also engage in learning because they enjoy cooperating with others. Similar to a collaborative structure, employing competitive learning has a complementary potential to engage students. Once they are faced against each other, their competitive instincts can encourage them to increase their commitment towards the learning process. Even students who initially may not be inspired by the subjects may begin to be interested once they have to compete. We also believe that this is particularly useful for students who are not prone to be competitive, since they should be confronted with this reality as soon as possible.

This work presents a study and a methodology proposal for a collaborative and competitive learning setting. It is based on a computational framework currently being under development for a programming course on a Polytechnic Institute engineering program. This methodology relates to learning in competition with others, achieving individual goals with or without collaboration from peers [5, 6]. Besides the usual theoretical lessons and laboratory practices, the integration of the framework in the course automates all the support and evaluation processes. It implements two competitive settings instantiated by the Champions League (CL) module and the Automated Test and Group Ranking (ATGR) module. The first module targets an individualized type of competitive learning, where students are selected to participate in a tournament with a group stage and a knock-out phase, similarly to the well-known UEFA Champions League. The later module aims at an intergroup competitive setting, in which balanced student groups are formed, competing among them. A third module is predicted to be implemented in the future, where the actual programs coded by the students compete in an arena, but its scope of applicability isn't as broader as the other two – only in artificial intelligence-like programming courses – and will not be further discussed in this paper.

The paper is structured as follows: in Section II we report some related work and studies with competitive and collaborative learning, while the methodology and supporting framework are presented in Section III. In the following

section, we discuss our findings and present student opinions for this methodology, based on empirical and questionnaire results. Finally, in Section V, overall conclusions are taken and future work is presented.

II. RELATED WORK

Concrete, reflective, abstract, experiential and active learning are some terms used to describe alternative pedagogical methods that fit different learning styles. Hence, there are several learning styles proposals and mixed applications described in literature [3, 7]. Obviously, learning styles are individual preferences and tendencies that influence the learning process [8]. Johnson and Johnson [3] refer a close relationship between learning styles and attitudes towards learning, including motivation to learn, involvement in learning activities, attitudes towards teachers and self efficacy. Students learn better if contents are presented to them through a medium that matches their preferred learning style [3, 7, 9]. Moreover, independently of the area of knowledge, recent innovations in pedagogical techniques have led to the introduction of new instructional methods in the classroom. These innovations include, for example, educational and on-line games [10, 11], asynchronous instruction (e.g., email, electronic bulletin boards and podcasting) and computer-based teaching [12]. Therefore, the different learning approaches are basically applied by teachers according to their experiences, the ambience where they find themselves teaching and the concepts they want to pass through.

Based in our experience in teaching programming courses, we have found that a mixed collaborative and competitive method is an adequate approach for such courses. Competition has been defined as “a social process that occurs when rewards are given to people based on the basis of how their performances compare to the performances of others doing the same task or participating in the same event” [13]. Competitive learning holds both positive and negative aspects. Lam et al. [14] found that competition had a positive impact on performance goals and learning motivation in the classroom. Also, it has been stated that “learning by losing” was a valuable process for students preparing themselves for professions where working under pressure was necessary [15]. On the other hand, some handicaps related to competitive learning have been identified in literature, like high anxiety levels, self-doubt, selfishness and the interference with the capacity to solve problems [3]. Nonetheless, these issues can be attenuated if one establishes different competing environments, being one of them the competition between groups of students, during which elements of a group cooperate in achieving personal goals, hereby introducing collaborative learning. Collaboration, or cooperation, has been defined as “a social process through which performance is evaluated and rewarded in terms of the collective achievement of a group of people working together to reach a particular goal” [13]. As with competition, collaboration also yields positive and negative aspects. Duetsch [16] suggested that collaboration embodies positive interdependence and found that college students solved more problems in a collaborative environment than students in a competitive environment. The main problem that we identify with collaborative learning is the possibility of

an uneven engagement in learning by individual students within a group, translated into uneven contributions for the group's final grade. Nevertheless, with our methodology, the individual merit of a student can be assessed in the individual competitive setting.

Programming courses offer good conditions for the implementation of a competitive environment, based on both collaborative and individualized learning. Structuring a problem-based competitive project, wherein collaborative student teams compete against each other, is a way by which teachers can ensure both collaborative and competitive learning opportunities for students [4]. These methods have also previously been applied to other engineering classrooms [17, 18, 19].

Previously to the work presented in this paper, some experiences of learning through competition and collaboration were carried out in our Institute. The implementation of a market based on software agents was the main goal of a curricular innovation experience in an advanced course on Distributed Artificial Intelligence [20]. The task was prepared to enhance the learning experience through collaboration and competition among the students. The project involved the creation of a community of agents capable of modelling a market of goods and competences. For each producer agent (with reasoning capabilities implemented by a group of students), the goal was to maximize its production capabilities and earn more money, by means of competition with other producer agents. This proved to establish a rich and challenging collaborative-competitive environment for the students, supported by their opinion and developed works. In another case of competitive learning evaluation, groups of students were asked to implement a representation of the horse game, the game operators (piece movements) and a search algorithm. The horse game is a simple game that finds its origin in the chess game and is played by two individuals on a ten per ten square boards. Essentially, the groups of students would play against each other and the first tree ranked groups of the tournament would benefit from a bonus translated into a higher subject grade. The introduction of the competitive element in the course motivated the groups to care and look for better implementations of their programs. By testing them against each other, they automatically started to explore new heuristics and to improve their algorithm's implementation to get better results in the competition.

As for related frameworks, *Mooshak* [21] is the only one we know about and is interesting because it presents some similarities with modules and functionalities of our proposal. *Mooshak* is originally a system for managing programming contests on the Web and its basic features include automatic judging of submitted programs, answering to clarification questions about problem descriptions, re-evaluation of programs, and tracking printouts. The system was originally intended for contests, but it is increasingly being used in programming courses, used to give instant feedback on practical classes, to receive and validate assignments submissions and to partially evaluate and mark assignments.

However, in competitive programming field there are several examples on the web like *Robocode* [22], *Top Coder* [23] and *ProjectEuler.net* [24]. *Robocode* is a game where a programmer codes the behaviour of a robot that is put into battle with others around the world. *Top Coder* is more targeted for professionals, since competition happens with programmers solving problems for customers and getting paid for the job in case they win. Finally, *ProjectEuler.net* is composed of hundreds of mathematical problems in which computer programming skills are required to solve the majority of them. By solving one problem it will expose the programmer to a new concept that allows him to undertake a previously inaccessible problem. So, a determined participant will slowly, but surely, work his way through every problem [24]. We find this interesting because the idea is similar to our “hints” mechanism in the proposed framework (next Section).

III. METHODOLOGY AND FRAMEWORK

This section details the methodology proposed, highlighting the main principles, the supporting computational framework’s architecture and the development of the two competitive modules. The followed learning methodology implementation is explained along with the framework’s architecture and modules description.

A. Main Principles Requirements

When implementing a new methodology and designing a framework like the one presented with this work, we first establish guidelines for the design. Some principles requirements that must be regarded in the conception of our proposal are the following:

- a) To implement different competitive learning settings (competitive and collaborative), but with a strong interdependence between them;
- b) To insure flexibility with regard to time and location, the framework should be available in a permanent manner, i.e. anywhere, anytime;
- c) To present teaching and assessment materials in a language that is simple, clear, direct, adequate and attractive to the target for which the courses are intended;
- d) To create auto-evaluation functionalities, in the form of evaluative exercises and tests that allow students to evaluate their own progress;
- e) To continue introducing a variety of competitive components, such as games and challenges, to stimulate the student to engage with the learning solution;
- f) To allow, in case of failure, the repetition of the studying process as well as the practice and auto-evaluation, yet with a different set of exercises;
- g) To use the information obtained from evaluation moments and every time students use the system to define measures that allow guaranteeing the existence of an adequate component of support and feedback from the system and from teachers;

- h) To devise adequate analysis that allows teachers to become familiar with students who are really learning from this methodology.

B. Overall Architecture

The framework supports a methodology divided in two different competitive settings. As mentioned in the Introduction, these settings are mapped into two modules of the architecture (Fig. 1) – the Champions League (CL) and the Automated Test and Group Ranking (ATGR).

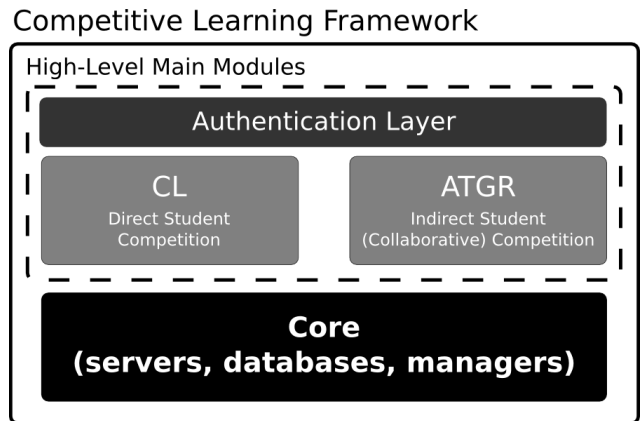


Figure 1. Overall architecture of the presented framework.

The CL module implements an individualized type of competitive learning, while the ATGR module is related with intergroup competition where collaboration occurs within groups. An Authentication Layer, for controlling the access to the system, is an independent module because it depends on the information system used at the Institute. This way, we do not keep duplicate information about users, but instead use the information already available in the information system, e.g. student's number, name, course, number of enrolments, etc. It is only necessary to map the users’ unique identification on that system to the information stored in this learning framework.

These high-level modules are strongly interdependent, not only because they are used in conjunction to evaluate and to teach the students, but also because the results taken from the CL are used to configure groups in the ATGR module. The groups of students are generated upon results from the CL to ensure homogeneous grouping, maximizing the chance of winning for all groups. Homogeneous grouping allows the groups to be as evenly matched as possible to provide a challenging environment for competition. We discuss this grouping methodology in Section IV. Therefore, the output from a module is used as input on other modules, despite being used in an indirect way. This means that the output data from a module must be stored in the core of the system.

The core is composed by the necessary servers, like web and database servers, and the manager modules, which are used to configure the system to a learning case. This configuration is made by advanced users, e.g. teachers with an administrator course role, who determine exactly how to apply the proposed methodology through the supporting framework.

In addition, the framework also integrates contents modules for the lessons and for the students study, being a complement for the competitive learning. These modules have a strong emphasis on multimedia contents with audio, video and animations. They are presented through web interfaces and a multimedia interactive DVD-Rom.

C. The Champions League (CL) Module

The first main module implements a competition where the students are selected to participate in a tournament similar to the well known UEFA® football Champions League (CL), with a group stage and knock-out rounds. The architecture for this module includes four main sub-modules (Fig. 2).

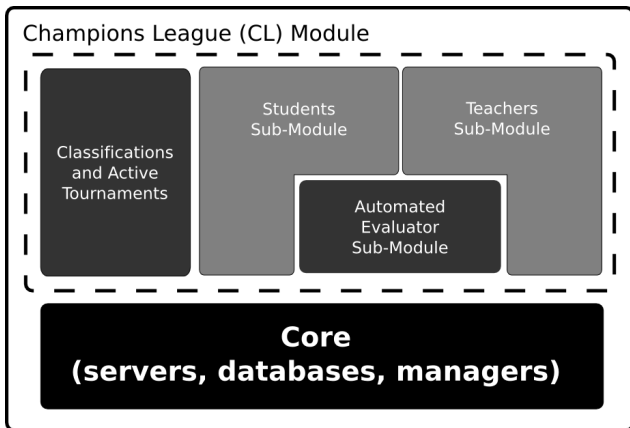


Figure 2. Detailed architecture of the Champions League (CL) module.

One of them is the manager of Classifications and Active Tournaments (CAT). All the information is gathered from the corresponding database. The CAT module is a Web-based application where it is possible to get information about past and present competitions. Students have access only to data related with competitions where they have participated. Teachers can export results from competitions designated as assessments, which are configured according with that. In short, it presents tournaments classifications and matches results, detailing the obtained goals for each student player, and statistics that can be obtained through a set of queries.

The module related with teachers integrates all the functionalities necessary for the correct management of the competitions. One teacher can add new exercises, with the corresponding type descriptors, to the database and configure CL tournaments, which may be of the formative or assessment type. Through this sub-module, teachers access the Automated Evaluator (AE) sub-module, configuring it according to the class of students they have in mind to compete. Additionally, this module has access to another sub-module integrated in the CAT module. Through it, teachers can determine which and how results and statistics data should be presented to students, with the possibility to decide if they are made public.

The AE module crosses the submitted results (programs resolutions) with the configuration made by teachers and the exercises descriptors taken from the database to evaluate how many goals were achieved by each student in the scheduled

match. Each exercise has a set of goals to be achieved that are specified by a teacher when submits them to the database.

For the competition, students use the laboratory computers to implement the proposed code for each programming exercise (a “match”), where they compete against an opponent and the clock. Using the Students sub-module, students must submit a file with the implemented code to the framework at the final of a match, or when they finalize it. They also have to use this Students module to access their areas in the CAT module.

The AE module automatically evaluates the obtained results of the matches, updating the repository and giving feedback to the participants. Briefly, the result of a match is dependent on the evaluation of each exercise resolution submitted by each student that participates in the match. When a submitted exercise is evaluated, the number of goals obtained by the student is counted and stored. This is important information not only for statistic purposes, but also for course evaluation purposes. If both students obtain the same number of goals, then it is necessary to look up for the timestamp relative to the moment when files were last saved. If this item is also equal then the match result is a draw, resulting in one point for each container, otherwise, three points for winner and nothing for the loser.

The competition can take place for several sessions (lessons, marked days, etc.) and a teacher can configure it by choosing, for example, the starting time of the different group games, the level of the games (exercises to be implemented), the submission type, the competition mode (auto-evaluation/training, assessment, or friendly), among other parameters.

In addition, another sub-module under development will permit that students submit to the database exercises to be implemented by their colleagues in a determined CL competition. This is one of the most interesting functionalities in our methodology, since we think this may improve the learning performance of a student, as he has to find and know adequate exercises for accomplishing this specific task. Although not necessary, this can be a parameter in an evaluation.

D. Automated Test and Group Ranking (ATGR) Module

This module is responsible for automatically evaluating the correct implementation of functionalities within a final course project. The module architecture is depicted in Fig. 3.

One advantage of this module is that it relieves teachers from manually testing each group project, which can be very time-consuming, particularly in courses that easily scale up to hundreds of students. This way, they only have to complement the automatic evaluation by evaluating the quality of the code produced by the groups of students. The second advantage is that student groups may use the system to do periodic checks to their implementations and see if they are doing things right, i.e. if their algorithms are producing the correct results. While doing so, they are competing with the other groups for being the first to achieve correct implementations of the different functionalities/tasks that compose the overall problem assigned

to them. During this process, they are awarded points for a correct implementation, plus bonus points for being the first, second or third groups to achieve each individual goal.

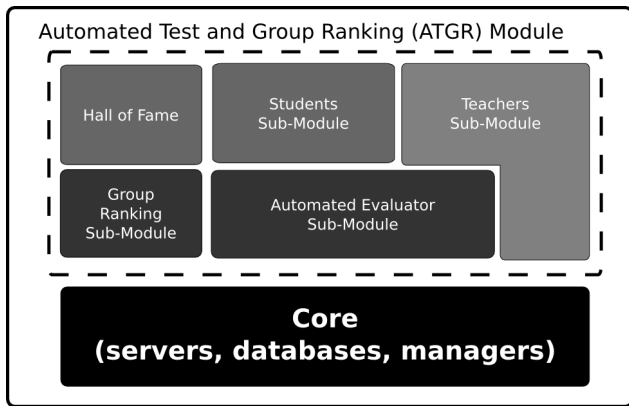


Figure 3. Architecture of the Automated Test and Group Ranking (ATGR) module.

As an incentive, each time a group submits a correct implementation, for a particular functionality, they receive clues (“hints”) for solving other harder functionalities. At the end of the submissions – the deadline – the final ranking table (Hall of Fame) elects a winner group to whom is given a bonus grade, as well as to second and third runners-up. Nevertheless, the ranking table is available to the students throughout the implementation phase as a means of direct competition.

In order to sort out if some kind of plagiarism or code-sharing occurred, all final implementations are submitted to the Moss system [25], a software plagiarism system from Stanford University, upon which a Hall of Shame can be generated. This list is not a priority to us, but in severe cases it must be used and it can even act as a deterrent measure for future course editions, or for other Institute courses.

The front-end of the sub-modules that interact with users are web-based. In order to avoid abuse, teachers are responsible for group information submissions and assign themselves to those groups, using the Teachers module. Through this sub-module teachers also upload the code, which will be used for automated testing of project implementations, to the Automated Evaluator sub-module, in the same way they do for a CL competition. Authenticated groups (any student assigned to the group can do so) submit their implementations code through the Students specific sub-module. After that, they can obtain feedback from the AE sub-module, including the aforementioned clues for solving what is left of the problem.

The Hall of Fame sub-module is responsible for retrieving the information from the database and for generating the rankings table. When the delivery deadline is reached, all the final versions submitted are sent to the Moss server and results can be made public. All data relative to group performance, submission history and rankings are stored in a specific-module database, with modularity in mind, at the core.

IV. RESULTS AND DISCUSSION

In this section, we summarize and discuss our findings and student opinions during this early use of the proposed methodology and supporting computational framework. These experiences were conducted with laboratory classes of an introductory Java programming course. The final grading of this subject is calculated by summing 50% of the exam grade (written exams cannot have a weight inferior to this percentage in our Institute) and 50% of the grade provided by the framework adding its two settings.

A. The CL Pilot Experience

In terms of the competitive learning provided by the CL module, the results obtained with the first application of the methodology were very positive. Students thought it was a very interesting initiative, giving to them some fine moments where they were able to combine the concerns of an assessment with the joy of participating in a game. Not exactly the feelings to expect from an evaluation moment, but they were much nearer of situations from the real World. For teachers, it clearly showed them that better prepared students were the ones obtaining the best results in the final classification.

Pool A				Pool B			
G1	Texugo's Team	6 - 5	Net_Azelha Beta	G1	Rosado	7 - 4	Lello
	Cova Piedade	2 - 5	Java The One		Valongo	4 - 2	Pereira AC
G2	Texugo's Team	4 - 1	Cova Piedade	G2	Rosado	4 - 2	Valongo
	Net_Azelha Beta	4 - 1	Java The One		Lello	3 - 1	Pereira AC
G3	Java The One	0 - 0	Texugo's Team	G3	Pereira AC	0 - 0	Rosado
	Cova Piedade	0 - 3	Net_Azelha Beta		Valongo	0 - 1	Lello
G4	Net_Azelha Beta	4 - 3	Texugo's Team	G4	Lello	2 - 1	Rosado
	Java The One	1 - 1	Cova Piedade		Pereira AC	1 - 2	Valongo
G5	Texugo's Team	4 - 2	Cova Piedade	G5	Rosado	1 - 1	Valongo
	Net_Azelha Beta	4 - 2	Java The One		Lello	2 - 3	Pereira AC
G6	Java The One	0 - 1	Texugo's Team	G6	Pereira AC	0 - 1	Rosado
	Cova Piedade	0 - 4	Net_Azelha Beta		Valongo	1 - 1	Lello

Pools Classifications						
Pool A	W	D	L	GF	GA	Points
Net_Azelha Beta	5	0	1	24	12	15
Texugo's Team	4	1	1	18	12	13
Java The One	1	2	3	9	12	5
Cova Piedade	0	1	5	6	21	1
Pool B	W	D	L	GF	GA	Points
Rosado	3	2	1	14	9	11
Lello	3	1	2	13	13	10
Valongo	2	2	2	10	10	8
Pereira AC	1	1	4	7	12	4

Final Phase				
SF - G1	Texugo's Team	3 - 1	Rosado	
	Lello	1 - 3	Net_Azelha Beta	
SF - G2	Rosado	2 - 4	Texugo's Team	
	Net_Azelha Beta	5 - 2	Lello	
Final	Texugo's Team	3 - 5	Net_Azelha Beta	

Figure 4. First CL edition results. Legend: G (Group), W (Wins), D (Draws), L (Losses), GF (obtained Goals), GA (suffered Goals), SF (Semi-Final).

As an example for an easy understanding, it is possible to analyse a small laboratory class of eight students that took place in the first CL edition. Fig. 4 presents the competition progression with all the obtained results.

First, it is important to summarize how students got placed in the groups/pools and the organization of the competition. They were split into two groups of four students/teams (each student chose a team name to relieve the pressure of competition and evaluation), who played “home” and “away” against each of their pool opponents to decide which two teams

from each pool would advance to the first knock-out round. In these rounds, the last 16 until the semi-finals, teams played two matches against each other on a “home” and “away” basis. In this case, after the groups stage we ended up with only four teams, the best two from each pool. So, the first round was not the last 16, but immediately the semi-finals round, and group winners played against runners-up other than teams from their own group. In this application, it was not necessary to make a draw because the number of participants was limited. The final game was decided by a single match.

The tournament results help to understand the reason why the student with the team name “Net_Azelha Beta” won the competition. Besides being the winner of her pool, this student also “scored” the highest number of goals after six games played, considering that the games/exercises were the same for the two pools. “Net_Azelha Beta” has reached a total of 24 programming goals present in six proposed exercises. It was an excellent result since it was above 95% of the goals represented by the six Java programming exercises. However, the same appreciation can be made for the second best team. Moreover, the course final written exams proved that the semi-finalists students in this CL edition were really the best students of their class. This is an important conclusion because it proves that a competitive learning environment does not prejudice the best students.

After the realization of the competition, with duration of approximately one month and five laboratory lessons, we also carried out a questionnaire survey. It was carried out among all the students in order to determine their satisfaction and opinions regarding the CL competition. The questionnaire was designed to gather information on students’ attitudes towards aspects of the competitive learning they experienced as well as towards some more specific aspects of testing. The questionnaire had six questions:

1. How much do you like it?
2. How useful is it?
3. How much pressure do you feel when participating?
4. How fair is it?
5. How much do you want to maintain this CL format?
6. How functional is the CL module of the framework?

Students had to answer with a number that ranged between 1 (a little) and 5 (a lot) for each question. In Table I we present the mean value of the ratings on each question.

TABLE I. STUDENT’S OPINION QUESTIONNAIRE ABOUT CL MODULE

Students	Questions					
	Q1	Q2	Q3	Q4	Q5	Q6
Testing Class	4,62	4,88	5	4,88	4,75	4,62
All Classes	4,37	4,63	4,82	4,55	4,13	4,67

One row of the Table I has the questionnaire results from the eight students of the previous CL example, while the second row refers to the answers from all students (60) of all classes. We believe that these results traduce perfectly what the CL competitive methodology brought to the course: “more demanding with more fun”.

Moreover, in the classrooms, teachers felt that students were positive about the introduction of this evaluation method and the questionnaire results confirm that informal and empirical conclusion.

B. The ATGR Module Application Example

In regard to the collaborative setting – intergroup competition – the reception by the students was also positive. One of the features they liked the most was the possibility of checking their implementations at home in an autonomous way, rather than just believing that it was correct with their own testing or with the help of teachers.

One of the first things to be set in the ATGR module, besides preparing it for a particular course project, is the composition of the groups. Initially, it is common for students to be somewhat reluctant to being assigned into groups. However, explaining to them that employees in an organization rarely get to choose their co-workers and are often put in situations where they have to work with people they do not know, the students come to understand, accept and benefit from the situation. We first thought in achieving balanced groups, in terms of chance in winning the competition, by pairing the first ranked student in the CL competition with the last, the second with the next-to-last and so on. But soon we found out that this was not the correct approach, because some students would be concern about being paired with an underachiever or that did not have good results in the CL competition. A purely random composition of the groups is preferred [4]. However, upon this random composition process, we used the information gathered from the CL competition, along with our own knowledge about the students, to correct the composition of some groups when we detected that the original ones did not benefit them or the competition. Our main concern is to engage all groups in a healthy competition, even if it happens separately in clusters of those groups.

The competition begins after the groups have been set. One of the interesting things we concluded was that it turns out that it is not beneficial to let the students make all the submissions they want to the framework. This is a negative side-effect that we must take into account for the following courses and competitions. It leads to a kind of frantic development by which they do not reason about what is wrong with their code, but embrace a trial-and-error approach until the system tells that their implementation is correct. In a few cases, it really became a game of fortune. We believe this is not desirable and we chose to limit the number of permitted submissions per-day, e.g. five submissions. This forces the students to use them wisely and also prevents them from reaching a correct algorithm “by chance” in the hurry of being the first ones to implement that functionality.

The students enjoyed the “hints” provided by the system during their submissions. They found them particularly useful

to overcome the harder functionalities implementation as they approached the submission deadlines. Teachers liked the automatic evaluation of the projects since they are able to know beforehand the different levels of completion of each project. This is also particularly useful in giving grades, because students immediately have a tangible way of comparing their expected grades with others. All of this is obtained at the expense of a greater workload on behalf of the teachers during the initial setup of the framework for a particular course, but we believe that it is compensated during all the teaching/evaluating process and provides a rich and challenging environment to students for learning and preparing themselves for their professional lives.

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a study and a proposal for a collaborative and competitive learning methodology, applied to a university level programming course. Due to an increasingly competitive professional world, where at the same time people are put to work with others in teams, we provide a mixed setting between these two learning methodologies with the expectation that it will somewhat prepare them to their professional careers, as well as providing a rich and challenging learning environment.

Opinions gathered from students during the initial use of the methodology and supporting framework were positive and we have discussed our findings and suggestions for such systems. In summary, these preliminary results showed us that this kind of mixed competitive-collaborative learning methodology has the potential to become a solid and reliable approach in learning programming languages. We believe that the developed framework is a valuable tool for the support of both teaching and learning activities. It will have a key impact on how some programming courses are taught in our Institute.

The modules presented for the framework have been implemented, but separately, and are now being integrated and unified in the proposed framework. This obviously leads to some refinements in modules that were independent, particularly in terms of a uniformed user interface, a common authentication layer and database sharing. Depending on the internal success of this framework, we intend to release it as an open-source project so others may benefit from it while, at the same time, helping in the process of perfecting it and/or adding new features.

As future work, a third module that targets artificial intelligence-like courses is planned, whereas the students' actual implementations of some projects will be put into an arena to compete with each other. Later on, we intend to publish another paper describing the technical issues and technologies involved in the computational supporting framework development.

ACKNOWLEDGMENTS

We thank our colleague Cédric Grueau for the great job produced while working with us on the first approach to the proposed methodology.

We also thank the anonymous reviewers for their invaluable comments and suggestions regarding this paper.

REFERENCES

- [1] P. J. Graham. "The competitive emphasis of sport: a need for reflection." Paper presented at the International Congress of Physical Activity Sciences. Quebec City, Quebec, Canada, 1976.
- [2] M. Ediger. "Cooperative learning versus competition: which one is better?" *Journal of Instructional Psychology*, 1996, pp. 204-300.
- [3] D. W. Johnson and R. T. Johnson. "Learning together and alone: cooperative, competitive and individualistic learning." 5th edition. Needham Heights, MA: Allyn & Bacon, 1998.
- [4] S. Attle and B. Baker. "Cooperative learning in a competitive environment: classroom applications." *International Journal of Teaching and Learning in Higher Education*, vol. 19, number 1, pp. 77-83, 2007.
- [5] L. Owens and R. Straton. "The development of a co-operative, competitive, and individualized learning preference scale for students." *British Journal of Educational Psychology*, 50, pp. 147-161, 1980.
- [6] L. Owens and J. Barnes. "Learning Preference Scales: Handbook and test master set." Melbourne: Australian Council for Education Research Ltd, 1992.
- [7] D. A. Kolb and A. Y. Kolb. "Learning styles and learning spaces: Enhancing experiential learning in higher education." *Academy of Management Learning & Education*, 4(2), pp. 193-212, 2005.
- [8] R. M. Smith. "Learning how to learn: Applied theory for adults." Englewood Cliffs, NJ: Prentice Hall, 1982.
- [9] R. F. McNergney, J. Herbert and R. E. Ford. "Cooperation and competition in case-based teacher education." *Journal of Teacher Education*, 45(5), pp. 339-345, 1994.
- [10] O. Shabalina, P. Vorobkalov, A. Kataev and A. Tarasenko. "Educational games for learning programming languages." Third International Conference "Modern (e-) Learning" (MeL 2008), Varna, Bulgaria, June-July 2008.
- [11] R. Neves Madeira, J. Sousa, V. Fernão Pires, L. Esteves and O. P. Dias. "A mobile and web-based student learning system." *WCES-2009 Social and Behavioural Sciences Procedia Journal*, 2009.
- [12] G. Catarino, J. Campos, R. Neves Madeira and C. Luz. "b-3mLearning – A blended multimedia system for mathematics learning. Proceedings of International Conference m-ICTE2006, Vol II, pp. 1181-1185, Formatex, November 2006.
- [13] J. Coakley. "Sport in society: issues and controversies." 4th edition. St. Louis: Times Mirror/Mosbey College, 1994.
- [14] S. Lam and P. Yim and J. Law and R. Cheung. "The effects of competition on achievement motivation in Chinese classrooms." *British Journal of Educational Psychology*, 74(2), pp. 281-296, 2004.
- [15] J. Dettmer. "Competition photography... learning by loosing." *PSA Journal*, 36, June 2005.
- [16] M. Duetsch. "A theory of cooperation and competition." *Human Relations*, 2, pp. 129-152, 1949.
- [17] P. A. Johnson, "Problem-Based, Cooperative Learning in the Engineering Classroom", *Journal of Professional Issues in Engineering Education and Practice*, 1999.
- [18] K. Watson, "Utilization of active and cooperative learning in EE courses: three classes and the results", *Proceedings of Frontiers in Education Conference*, November, 1995.
- [19] M. R. Elara, C. A. Acosta Calderon, C. Zhou, L. Zhang, Y. Yang and T. Yang, "Experimenting Competition Based Learning of Engineering Topics with Soccer Playing Humanoid Robots," 10th Asia Pacific Conference on Giftedness, Singapore, July 14-18, 2008.
- [20] H. Gamboa, B. Silva and J. Filipe. "Modelling a market using software agents – pedagogical experience in a Distributed Artificial Intelligence course." *Proceedings of the IV International Conference on Multimedia and Information & Communication Technologies in Education (m-ICTE2006)*, Formatex, November 2006.
- [21] J. P. Leal. "Mooshak - system for managing programming contests on the Web." <http://mooshak.dcc.fc.up.pt/~zp/mooshak/>, 2009.

- [22] M. Nelson. "Robocode – an open-source educational game." <http://robocode.sourceforge.net/>, 2001.
- [23] Top Coder, Inc. "Top Coder – competition based development." <http://topcoder.com>, 2001.
- [24] C. Hughes. "ProjectEuler.net" <http://projecteuler.net>, 2001.
- [25] A. Aiken. "Moss – A system for detecting software plagiarism." <http://theory.stanford.edu/~aiken/moss/>, Stanford University, 2006.