



## An adaptive and innovative question-driven competition-based intelligent tutoring system for learning

Pedro J. Muñoz-Merino\*, Manuel Fernández Molina, Mario Muñoz-Organero, Carlos Delgado Kloos

Department of Telematics Engineering, Universidad Carlos III de Madrid, Av. Universidad, 30, E-28911 Leganés, Madrid, Spain

### ARTICLE INFO

**Keywords:**  
Competition  
Education  
Intelligent tutoring system  
Software tool

### ABSTRACT

ISCARE (Information System for Competition based on pRblem solving in Education) is a new and innovative intelligent tutoring system that we have designed and implemented. This tool allows the competition among students for improving their learning process in a course. The tool takes some ideas from the Swiss-system widely used in chess and adapts them to the educational area. The competition is based on different tournaments and rounds. In each round, students are assigned in pairs of two, which compete one against another, and each pair receives different questions that students have to solve in a limit of time. Students can see their partial ratings after each round and their final rating after a tournament. A lot of knowledge from different disciplines was used to design, and implement this system, as ISCARE includes different functionality such as the students' registration into the system, the creation of tournaments, the registration and assignment of students to tournaments, the management of each tournament life cycle (started, in execution, finished, etc.), the addition of the different exercises to tournaments, the calculation of pairs of students for each round with different algorithms, the assignment of exercises per round and pair, the scorings of the students per round and tournament, the management of the students' ratings, or the visualization of information. This paper presents the ISCARE intelligent tutoring system, describing its different options, menus, or functionality as well as its architecture and the specific modeling to achieve the desired features.

© 2012 Elsevier Ltd. All rights reserved.

### 1. Introduction

The use of Intelligent Tutoring Systems (ITSs) in education has increased considerably over the last years. An ITS is an expert system that helps students and teachers in the learning process. ITSs can be accessed by teachers and students through the Web or as a stand-alone application. An ITS is usually divided into four modules (Wenger, 1987): the interface module (what teachers and students interact with in the system), the expert module (the specific contents), the pedagogical module (the different strategies of teaching), and the user model (some data to represent students and teachers in the system, so that different contents and strategies can be adapted to the different profiles). ITSs are software applications that must enable and manage the desired learning processes, and make convenient decisions about different issues such as the materials that are shown at each moment for different profiles, or motivational and emotional aspects.

One type of ITS is based on the competition among students in their learning process, but there are only a few of these

competition-based ITSs. Maybe, this limited number of competition-based ITSs is a consequence of the different arguments against competition, which are described by different authors, e.g. (Kohn, 1986). But there are other studies that argue about the benefits of competition in education, e.g. (Yu, 2001). From our point of view, the success of the introduction of competitive tools in the learning process depends on the specific competitive strategies applied, as stated in Ediger (1996).

In this direction, we have developed and implemented a system for students' competition in education that tries to include some of the best practices to increase student learning in the process. The designed and implemented competition-based system is called ISCARE (Information System for Competition based on pRblem solving in Education), and it includes a couple of new and innovative features with respect to existing competition-based systems for learning, as it is analyzed in Section 2.

One of the first ideas taken for the design of ISCARE was the application of the Swiss-system used e.g. in chess tournaments. The Swiss-system divides a tournament in different rounds. In each round, participants are paired, forming matches of one against one. Students are paired with classmates of their similar knowledge level in that moment. Participants can win (1 point), lose (0 points), or draw (0.5 points) in each round. All participants play the same number of games (one per round) even if they lose in several

\* Corresponding author. Tel.: +34 91 624 6233; fax: +34 91 624 8749.

E-mail addresses: [pedmume@it.uc3m.es](mailto:pedmume@it.uc3m.es) (P.J. Muñoz-Merino), [manuferna@gmail.com](mailto:manuferna@gmail.com) (M. Fernández Molina), [munozm@it.uc3m.es](mailto:munozm@it.uc3m.es) (M. Muñoz-Organero), [cdk@it.uc3m.es](mailto:cdk@it.uc3m.es) (C. Delgado Kloos).

games. There are partial ratings after each round in which students are rated according to their performance up to that round, and a final tournament rating. The Swiss-system has been adapted in ISCARE taken into account an educational competition perspective. ISCARE supports different rounds in which students participate, and matches are assigned according to the students scorings up to that round, so that students with a similar knowledge level can compete one against another. In addition, students cannot only receive points to increase their scoring depending on the comparison with their opponents, but also depending on their problem solving performance in that round.

ISCARE has to take care of a lot of tasks for implementing the desired competition that can produce a learning benefit for students. For example, ISCARE has to make the student pair assignments for each round (trying to pair students with a similar scoring up to that moment), select the different possible problems for each round, adapt the different problems to the different pairs (trying to assign problems with a difficulty level according to the students pairs knowledge levels), track the different students' progress during the tournament and their statistics (such as topics in which a student failed more), show the different problems to the students in a synchronized way, show the opponents and own information during the round competition, manage a tournament life cycle (start, execution, finished, etc.), manage the different tournaments (creation, deletion, assignment of students to different tournaments, etc.), or enabling teachers to upload different types of problems for students before a tournament starts and making possible their reuse.

Several of the presented tasks in the ISCARE system cannot be done efficiently by human teachers, without the intervention of computers. Teachers would have to devote a lot of time to execute the algorithms to pair the different students, adapt the different problems for different pairs, give different pieces of papers for every student which are adaptive depending on previous round results, make the different scoring calculations, manage the different rounds and tournaments, etc. In addition, these are complex operations that can usually lead to teachers' mistakes because of the high number of algorithmic operations to perform. Nevertheless, an intelligent system such as ISCARE can make it quickly, efficiently and without errors because of the high computing capacity of ITSs. In order to achieve it properly, a lot of considerations from different expertise areas must be taken into account, which includes: a useful interface for teachers and students, a complete and proper modeling for all the educational and management functionality, selection of the proper algorithms (for adaptation of problems, adaptation of pairs, scoring calculations, etc.), or the selection of the architecture.

This paper shows a detailed description of the implemented ISCARE system to show all their features and the differences with respect to existing systems. Furthermore, the paper shows how the functionality is implemented as a specific modeling. Section 2 explains related works and compares existing competition-based ITSs for education with respect to our solution, emphasizing the novelties of our ISCARE system. Section 3 explains the architecture of ISCARE. Sections 4–6 are devoted to the explanation and description of the different features of ISCARE: Section 4 is about the management of users and tournaments in the system, Section 5 is about the description of a specific round, while Section 6 is about the different ratings and information of the users in the system. Finally, Section 7 summarizes our conclusions and directions of future work regarding the ISCARE tool.

## 2. Related work and innovation of the ISCARE system

There are different experiences of the use of competition (Verhoeff, 1987) and games (Amori, Naicker, Vincent, & Adams,

1999; Squire & Jenkins, 2003) in education. Although competition and games are different concepts, they have a relationship and most games introduce a certain degree of competition (against the computer or other opponents). Therefore, a lot of games in education also contain a competitive part. In any case, both games and competition provide a motivational and enjoyment component that can improve the learning process. There are multiple types of games and competitions, each one with different features, and the selection of a proper combination of features is what can make the difference to improve the learning experience.

There are games and competitions in education that do not need a computer support while there are others that need the use of computers because a stronger calculation and operation base is required. In the line of games and competitions without a computer support, there are different approaches such as for learning software development using a card game (Carrington, Baker, & van der Hoek, 2004), cyber defense trying to prevent network attacks (Conklin, 2006), some computer engineering problems through puzzles (Behrooz, 2009), data structures using the competition between programmed codes (Lawrence, 2004), different implemented programs comparing the size/speed of each one (Striegel & Rover, 2002) or programs built based on a template and using game theory and competition (Burguillo, 2010).

The previous commented methodologies do not require a computer based support as there is not neither personalization nor adaptation, course materials are limited, competition is based on an easy methodology, do not require complex scorings calculations, etc. But when more complex operations are needed, then ITSs are required. Therefore, the execution of such tasks would not be adequate and feasible without using computers that can make complex calculations and decisions. Most of the competition systems in education are based on the resolution of a set of questions or challenges. In this direction, several authors presented works related to competition systems, each one with some differences in the presentation or proposal of the questions:

- Lin, Young, and Hung (2008) show a crossword game for learning English.
- Pfothenhauer, Gagnon, Litzkow, and Blakesley (2009) present the “Cool-it” game for learning cryogenic principles. The game is based on a Matlab simulator that challenges students to do a set of actions and compares the students' provided solution with respect to the ideal one, giving students the degree of achievement.
- Chen, Liao, and Chan (2010) show a system with a graphical environment with questions for learning Chinese. Players are replaced by virtual pets to reduce negative emotions.
- Chang, Yang, and Yu (2003) present the Joyce system, which represents a table board and each participant advance depends on the resolution of quizzes.
- Regueras, Verdu, Verdu, and de Castro (2010) evaluate an experience with the QUEST system, in which students interact with a set of questions and they are rewarded depending on the time to answer and the comparison with their opponents.
- Silva and Madeira (2010) propose a collaborative system for competition based on a set of automated evaluation questions.
- Liu and Liu (2005) show a pedagogical agent for learning Chinese with different participant profiles. Participants can test their knowledge in different language skills with different challenges.
- Sindre, Natvig, and Jahre (2009) show a graphical adventure with different rooms enabled or not depending on the student progress on different challenges.

One of the most important differences of our ISCARE system with respect to the commented systems is the competition

methodology for education, which is based on the Swiss-system in ISCARE, but adapted to education, introducing the required modifications. In other educational systems, all students compete at the same time against all other, having a global ranking such as in Liu and Liu (2005), Regureas, Verdu, Verdu, and de Castro (2010), Sindre et al. (2009), or Behrooz (2009); using a system like the Champions League (Silva & Madeira, 2010); challenging another player that is selected by the own student such as in Chen et al. (2010) or Chang et al. (2003). But in our ISCARE system, according to the Swiss system, there are different rounds, all the students play in all rounds (there is not the concept of elimination), and the pairings for each round are selected by the system trying to match students with a similar knowledge level (similar number of points in the scoring), and students compete one against another in pairs of two. The adaptation of the Swiss-system for an educational environment makes a motivational impact on students as they can make progress in the specific round and tournament rankings, having challenges with other students of their similar level.

Other differences of ISCARE with respect to the presented systems, which make ISCARE innovative, are the following:

- *The adaptation implementation.* ISCARE provides an innovative adaptive methodology in which the different pair assignments are adapted for each round trying to minimize the difference of points between the different students' pairings. In addition, the different assigned problems for each pair are adapted so that the difficulty level of the materials can be according to the students' knowledge levels, but being the same for students of the same pairing. There are different adaptive tutors for education that adapt the educational materials to students such as the commented in Wang, Wang, and Lin (2010) or Chua, Liaob, Chenc, Lind, and Chen (2011), and there are even adaptive tutors in education that personalize questions in an assessment such as the commented in Guzman and Conejo (2005) or Feng, Heffernan, and Koedinger (2006). But these systems do not implement any competition. ISCARE presents adaptation of questions and pairings in a competitive environment in a way that is not done by other competitive systems.
- *The scoring system.* A student has different scorings assigned (on-live round, round, tournament, and global). The scoring of a student does not only depend neither on his/her performance on the resolution of the different proposed problems nor on the comparison with his/her opponent, but it includes a combination of both.
- *On-live visualization of the competition.* The student can see on-live his/her own progress for a round (number of points, time left, or questions left) as well as the opponent on-live progress for the same round with the same parameters. In this way, there is an on-live challenge with the opponent.
- *Complete tournament management.* The concepts of tournament and round life cycles are introduced. In this way, tournaments and rounds pass through different states activated with different conditions, which makes easier the competition management.
- *Complain with a subset of the IMS-QTI (Question and Test Interoperability) specification (IMS QTI & Test Interoperability Specification, 2005).* As far as we know, ISCARE is the first competition tool that is compatible with the IMS-QTI specification, which enables the reuse and interoperability between different question designers and systems.

### 3. Architecture of ISCARE

The ISCARE system is based on a typical client-server architecture. The server listens to different HTTP requests for clients in a specific port. The server can make different actions, one of them is to access to a defined data base which can be in another different

machine from the server. The server makes the processing of the different client requests, and sends the different HTTP responses to the clients. The HTTP responses include HTML materials. The clients are browser applications and the server can be e.g. Tomcat. Oracle was the initial data base in which was implemented, but the application can be migrated to other data bases in the future.

The ISCARE system implementation follows the MVC (Model View Controller) design pattern in which the system is divided into the data component and business logic using this data (Model), the user interface (View) and the part of the application that waits for user actions and calls the proper business logic methods (Controller). Fig. 1 shows the different subsystems of ISCARE, which are the following:

- *User pages.* This subsystem includes the different web pages that students and teachers can browse. These web pages show the information that is returned by the actions' subsystem.
- *Help.* It is composed by web pages that contain all the information about the use of the tool to help the users. These web pages are mainly statics, so there is no need to call the actions' subsystem. The user pages subsystem and the help subsystem compound the menu subsystem.
- *Forms.* This subsystem contains the different classes to represent the data that needs to be sent from the menu subsystem to the actions subsystem. In this way, users' requests can be received correctly.
- *Control operations.* It contains the logic to process each type of request from the users, and it can forward the request to the correct page destination of the menu subsystem, once that the required data have been loaded through the data management subsystem. The different possible operations can be classified into general operations and operations for tournaments, problems or listing.
- *Data management.* This subsystem includes the different actions to change and retrieve data from/to the data base. It can only be accessed by the control operations subsystem.

Fig. 2 shows the data model of the application with an Entity Relationship diagram. There are two types of users in the application: students and teachers. Each profile has different associated functionality. Teachers are mainly in charge of the management

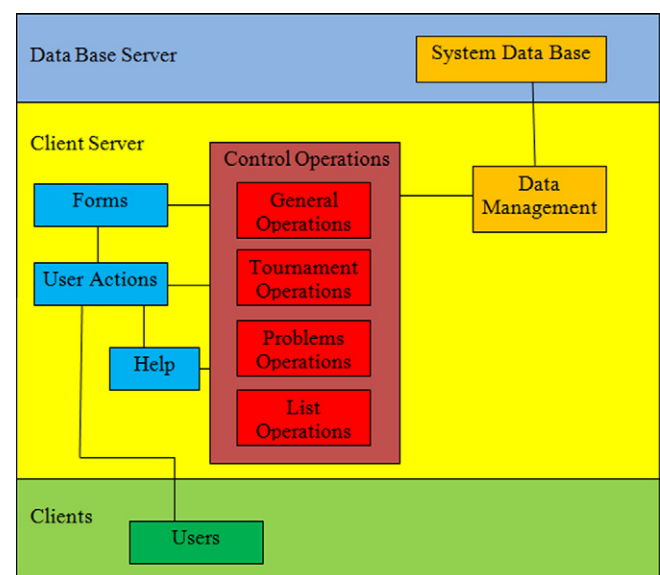


Fig. 1. Subsystems of ISCARE.

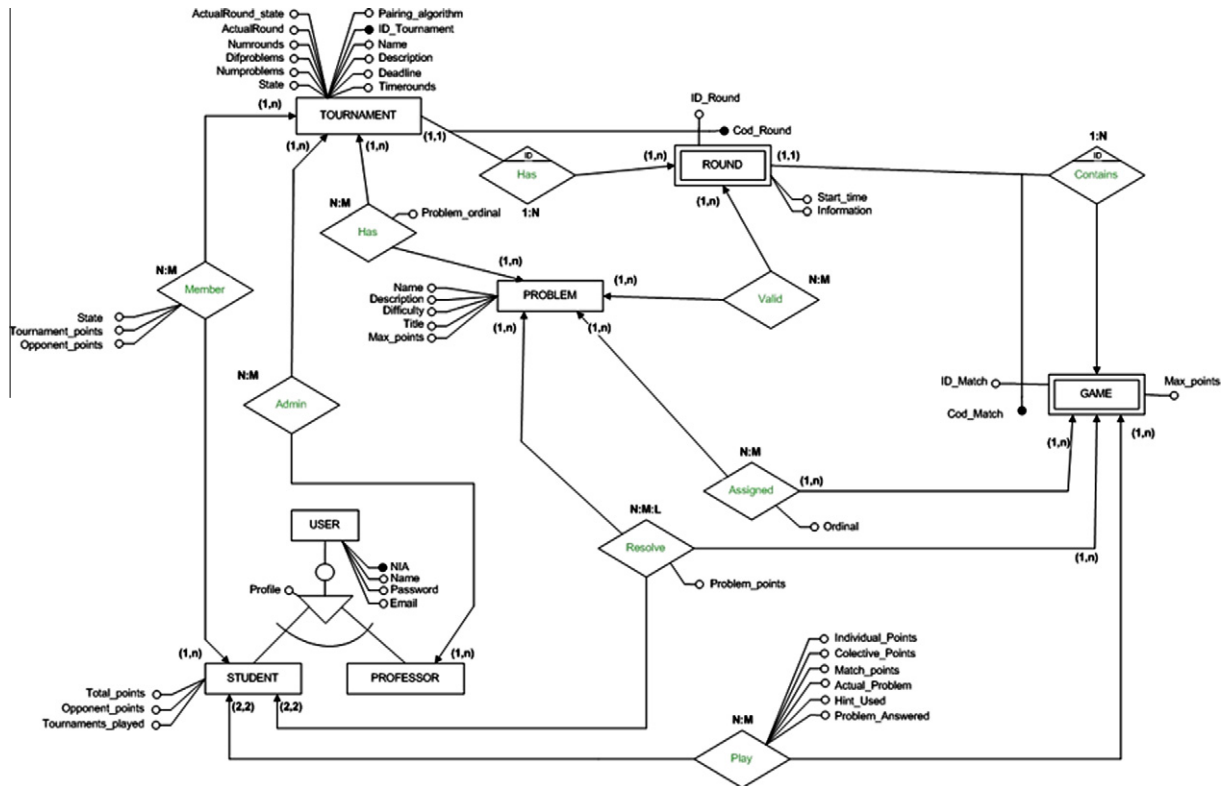


Fig. 2. Data model representation.

and administration of the application tournaments. A teacher can view the tournaments he/she has created and he/she is the only person that can be in charge of their administration. While students can request for registration in tournaments that were created by teachers, and latter on, students will participate in the tournaments, answering questions that are proposed by teachers for each round. As a tournament participant, a student has a set of points in the tournament at each round, and a global scoring in the tool, which is not related to a specific tournament but to their global performance in the different tournaments.

Among the teachers' possibilities, there is one to add problems to tournaments so that problems are assigned to tournaments with a number that identifies such problem in a specific tournament. In addition, a problem can be marked as only valid for specific rounds if the tournament is enabled with this feature (*difproblems* attribute). Later on, as rounds are running, students are assigned in pairs of two to compete, and the assigned problems to each pairing are selected by the tool. The assignment in pairs is done applying the defined pairing algorithm for the tournament and using the PLAY relationship, so that there is a historical storage of pairings and their results. The assignment of problems per each pairing is done with the ASSIGNED relationship, which associates the problems for each game in a specific order of visualization. Once all the assignments are done and the round has started, the RESOLVE relationship is used to store the different scorings of each student for each problem assigned in that round. In addition, the PLAY relationship is updated to show the individual scores that have been obtained by each student up to that moment.

4. Management of users and tournaments

In this section, we describe the main aspects related to the management of users and tournaments in the ISCARE system. There are two different profiles in the system: students and teachers/administrators. Depending on the user profile, there are different

options available in ISCARE. The tournament is the high level concept in which the competition is based. Students compete in tournaments that are configured by teachers.

This section is organized in the following subsections: Section 4.1 explains the details of the user accounts including the registration, the user account data details, and the initial menu options for students and teachers. Section 4.2 is devoted to the creation of tournaments, the different parameters associated to tournaments, and the request of tournaments by students. Finally, Section 4.3 is focused on the tournament life cycle (the different states of a tournament, the different students and teachers' views about the different tournaments, and the addition of students to tournaments by teachers).

4.1. User accounts

Fig. 3 shows the initial web page where users can access into the ISCARE platform with its login and password.

Users can proceed with their registration into the system, and they must fill in the form of Fig. 4 in order to introduce the different parameters that define a user into the system.

Depending on the user profile, the initial ISCARE menu will be different. Fig. 5 shows the students initial menu, while Fig. 6 shows the teachers/administrators initial menu. We can see that the initial options are different depending on the profile, as the students' actions are different from the teachers' ones, e.g. a student must request for registration in a tournament, but teachers must create new tournaments, and approve these students' registrations. We can also see a toolbar menu that is common to students and teachers with the following options: return to the home initial page, options for editing or removing a user account, help about the tool, and a logout button.

When registered users access into the system, they can edit their personal information or remove their account into the system through the interface of Fig. 7.



Fig. 3. User login initial page.

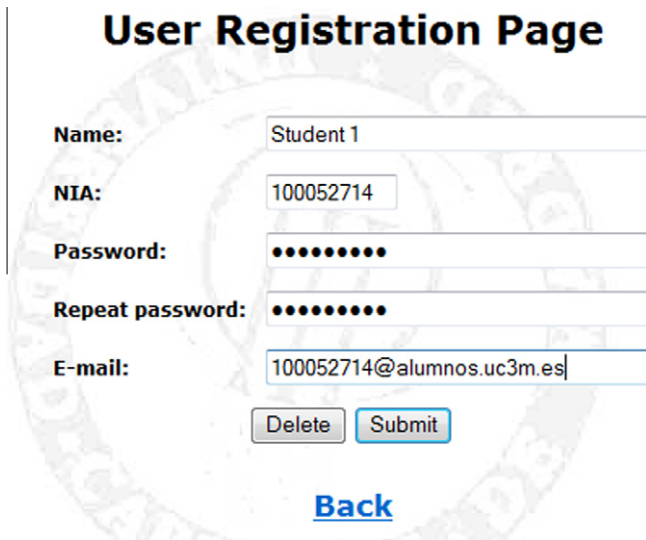


Fig. 4. User registration page.

4.2. Creation of tournaments and students requests for tournaments

Tournaments must be created by teachers. Students can request for participation in several tournaments. The final approval of the students that participate in a tournament is done by teachers.

Teachers can create a new tournament with the correspondent “Create new tournament” option of the teachers’ initial main menu. A form as the one of Fig. 8 must be filled in by teachers. In the example, a tournament related to a driver’s exam is proposed. A tournament name must be provided, as well as a description, and a deadline for students to request participation for that tournament. Students’ requests for participation are closed after the deadline. Next, the number of rounds for a tournament must be introduced (an integer number), the limit of time per round (in minutes), the number of problems that will be generated to each student per round, and whether the same problems can be repeated among different rounds or not (each round with different problems).

Students compete one against another in each round. Each student competes against as many opponents as the total number of configured rounds (5 in this example). Until one round does not finish, the next round cannot start. In each round, there is a configured maximum limit of time (in this case 20 min) in which a student must solve the configured number of problems (10 in the example). Depending on the commented configuration, the presented problems for a specific student can be repeated among different rounds in the tournament or not. For each round, the different pairings among students (2 by 2) must be done. The way to pair the students can be done in the tool using different implemented algorithms. The specific algorithm to use is selected as a tournament option and is applied for all the tournament rounds. Let be  $N$  the total number of students that participate in a tournament, and let  $R$  the number of rounds of that tournament. All the different pairing algorithms of the tool are the following:

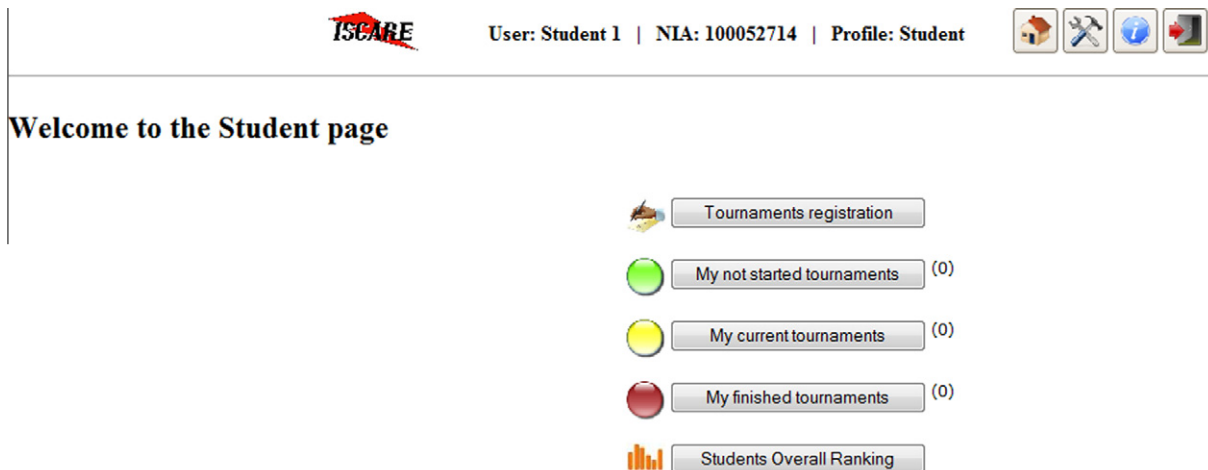


Fig. 5. Initial students' menu.

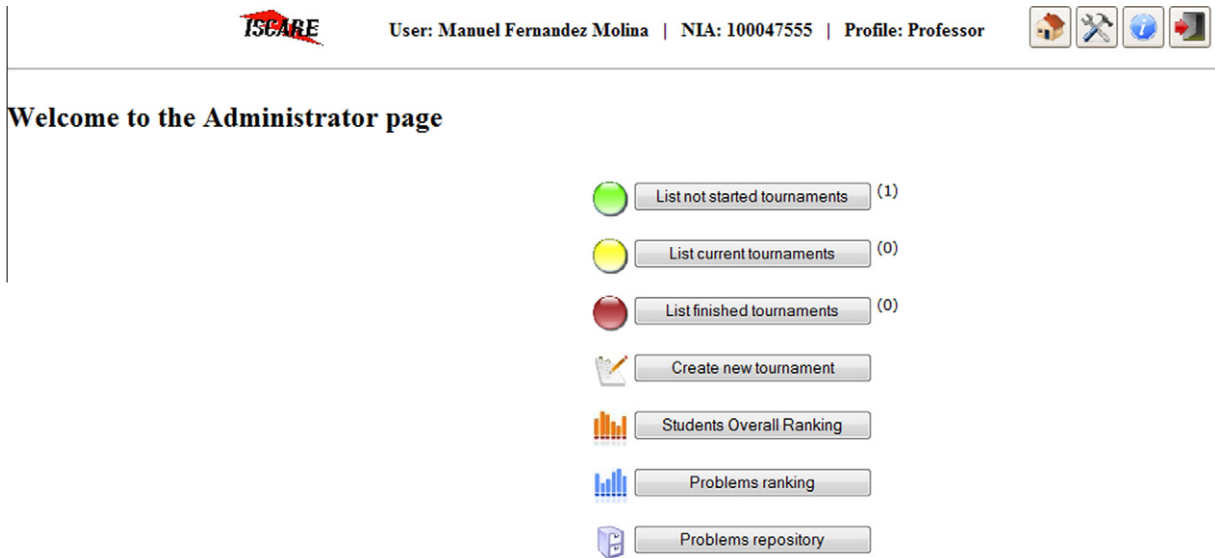


Fig. 6. Initial teachers/administrators' menu.

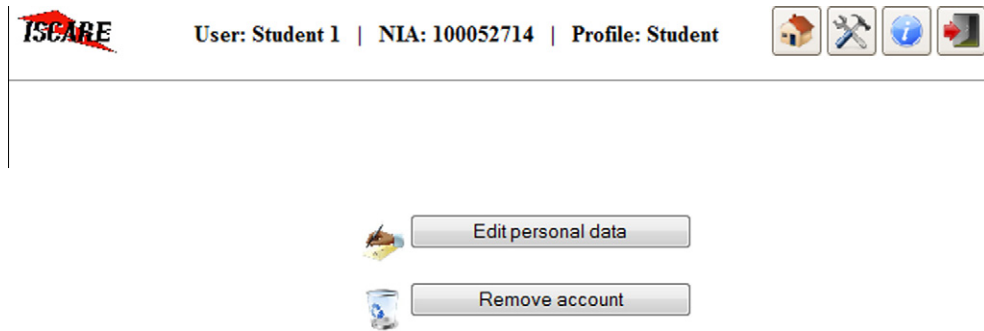


Fig. 7. Edition and removal of the personal account.

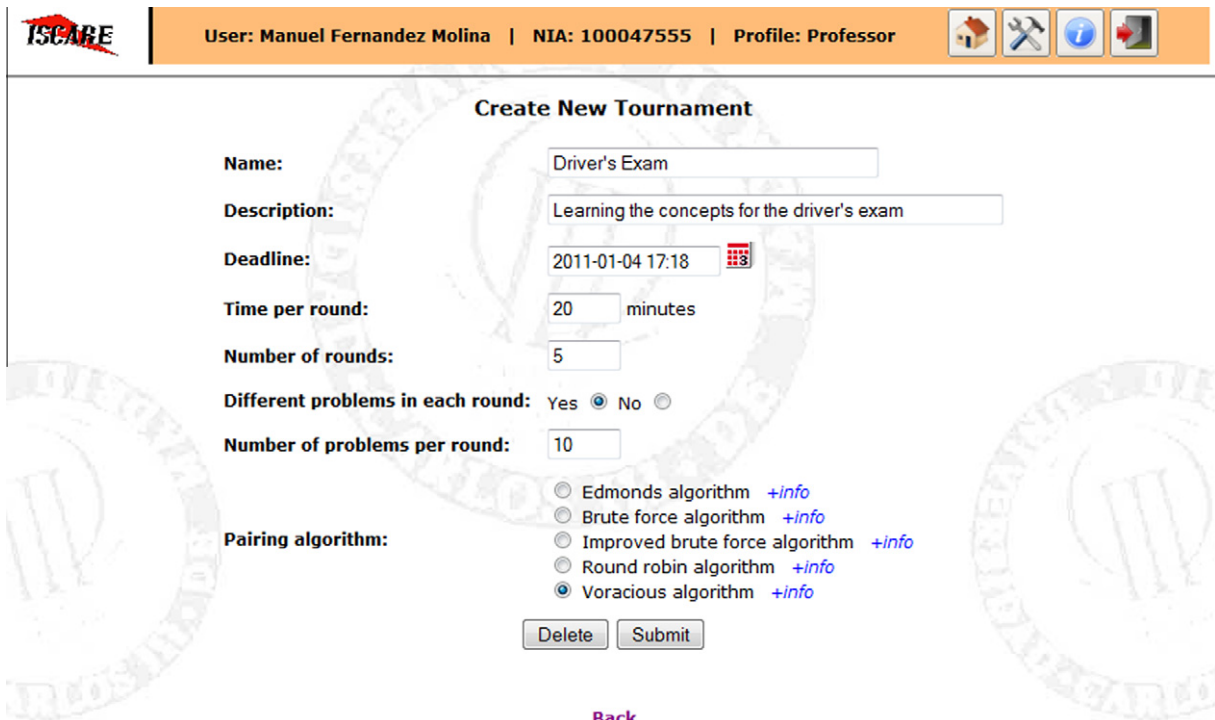


Fig. 8. Screen for the creation of a new tournament.

- *Round Robin*: This algorithm does not minimize the scoring distance between the proposed pairings. The algorithm builds a matrix of students in both axis, and shifts the different students' pairings for each round without any complex calculation. The pairings are randomly selected in the different rounds, without having to make any calculations about students' scorings. This algorithm implies a minimum time of calculation. The algorithm works well when  $N$  is quite similar to  $R$ , because there is no need of extra calculations, and the tournament is quite similar to a league in which all students compete against all the other participants.
- *Voracious*: This algorithm does not minimize the scoring distance between the proposed pairings. This algorithm tries to pair those students that have least possibilities to play with, based on previous round pairings taking into account that pairings cannot be repeated.
- *Edmonds*: This algorithm tries to find the global minimum distance between the nodes of a graph. The global distance is the sum of the distances of the nodes 2 by 2 according to the established pairings. The distance between two nodes is represented as the scoring difference between two students. This algorithm finds an optimal solution without repeating pairs if the number of students is greater than the number of rounds.
- *Brute Force*: This algorithm explores all the pairing possibilities to find the one that minimize the scoring difference between each students' pairing. It can be only applied if  $\log_2(N) > R$ . But this is a typical condition that is usually satisfied in most tournaments.
- *Brute force improved*: This algorithm works quite similar to the brute force algorithm, but avoids exploring such branches that are not necessary because of impossible pairing conditions. This algorithm improves the execution time with respect to the pure brute force one, and it can only be applied if  $\log_2(N) > R$ , as in the pure brute force case.

Table 1 shows a comparative among different existing algorithms for pairing (some of them implemented in the ISCAR system).

In addition, students can request registration for tournaments that were created by teachers, but only before the configured deadline. Students can request such registration through the "Tournaments Registration" option of the students' initial menu. Next, a list of current created tournaments in the system, for which the registration deadline has not expired, is shown (in this case only one, as indicated in Fig. 9). A student can select as many tournaments as he/she wants to request for registration. When a student pushes the tournament registration request button, then the student requests registration for a couple of tournaments, which will disappear from the list. Therefore, it is not possible for a specific student to request registration for the same tournament more than once.

**Table 1**  
Comparison among different pairing assignment algorithms.

Algorithm	Are pairings optimized?	Recommended situations	Implementation complexity	Computational complexity
Round Robin	No	$N$ close to $R$	Low	$O(n)$
Brute force	Yes	$\log_2(N) > R$	Medium	$O(n^2n)$
Evolutionary	No	Any	High	–
Voracious	No	Any	Medium	$O(n^2)$
Applied	No	Any	High	$O(n^4)$
Swiss-system				
Hungarian	Yes	Any	High	$O(n^3)$
Edmonds	Yes	Any	High	$O(n^4)$

A teacher/administrator can also set some global parameters that are common to all the application, using the variables option. Some of these options are related to tournament parameters while others are not. Fig. 10 shows this screen with the different global parameters. The ones that are not related to any tournaments are the limit of time in which if there is no user activity then the system logs out automatically, the SMTP configuration server (sends emails with registration welcome messages for new users), and the maximum size for bulk uploads (the maximum total size of a couple of problems that are added to the system in one transaction, this is to avoid problem files that are too big and can damage the system). Furthermore, there are other parameters which are related to all tournaments: the percentage of random problems (this is the percentage of problems that are selected randomly among the possible candidates in a moment, while the rest or problems are selected according to the students' knowledge level and the problems' difficulty level), the behavior with one student pairing (if there are an odd number of students in a tournament, then one of them is not paired at all for each round, and the decision can be to assign all the possible scoring for that student in that round or to make the student to compete with a fictitious student, which will obtain a 0 in its scoring for that round), and the default pairing algorithm (this is the one that is selected by default when a teacher creates a new tournament, but it can be modified later).

#### 4.3. Tournament life cycle

Each tournament has a life cycle, passing through different states. These states have an established order. If a tournament passes from one state to another, then it cannot back to a previous state. The different states for a tournament are the following:

- (i) *Not Started*. In this state, the tournament has been created and its parameters set by a teacher according to the instructions of previous Section 4.2. In the not started state, students can request for registration in that tournament until the configured deadline date. But a tournament can be in a not started state, and students cannot request for registration in that tournament in case the deadline date has expired. When teachers select listing all not started tournaments, then a screen as Fig. 11 is obtained. Fig. 11 shows the list of present not started tournaments with their features. Teachers can delete a not started tournament, or edit its parameters.


In addition, teachers can approve or revoke members for that tournament with the correspondent button when a specific tournament has been selected. Fig. 12 shows that menu, where there are three columns: on the left, all the students that have requested for registration in that tournament (in this case 16 students requested for registration in the tournament); on the center, the students that have been revoked by teachers (only one in this case); and on the right, the students that have been approved by teachers (15 students in this case). With the approval, revoke, and delete buttons, a teacher can accept or reject the different students' requests for participation in that tournament.





The students' view for the listing of not started tournaments is a bit different. They can access through their "My not started tournaments" option. First, students have not all the teachers' options available, but can only visualize the tournaments. Moreover, only tournaments that have been requested by the own student are the ones that are listed. Fig. 13 shows a listing example. A student can also see the state of his/her request with the color flag depending on the teachers' decision: approved, revoked, or requested

**Request registration in tournaments**  
**List of created tournaments**

	Id	Name	Description	Deadline	Numrounds
<input checked="" type="checkbox"/>	21	Driver's Exam	Learning the concepts for the driver's ex	2011-01-04 19:23	5

Fig. 9. Students' request for subscription in tournaments page.



**User: Manuel Fernandez Molina | NIA: 100047555 | Profile: Professor**

---

**Application variables**

- **Max. session inactive interval:**  min (you'll have to re-login to take effect)
- **Percentage of random problems:**  % (other will be selected according to its difficulty and students rank)
- **SMTP Configuration:**
  - SMTP Server:
  - From field:
  - Subject field:
- **Max. size for bulk uploads:**  Mb
- **Behavior with 1 student pairings:**
  - They'll take all the collective score, no matter what they do.
  - They'll be faced with a fictional student who get individual score of 0.
- **Default pairing algorithm:**
  - Edmonds algorithm
  - Brute force algorithm
  - Improved brute force algorithm
  - Round robin algorithm
  - Voracious algorithm

[Back](#)

Fig. 10. Application variables page.

(ii) *Started*. From the list of not started tournaments, a teacher can select a tournament and start the tournament with the “Start Tournament” option (Fig. 12). Once a tournament is started, then it is not possible to add neither more problems nor students. Therefore, a teacher must be sure that there will not be more participants for that tournament, and all the tournament related problems have been included,

because there is no possibility to go back to the previous state. A teacher can see the list of all current started tournaments through the correspondent menu option, while a student can only see the ones in which he/she participates (“My current tournaments” option). Next, students and teachers can select one of their current tournaments in the list and enter the specific actions for that tournament. For



**ISCARE** | User: Manuel Fernandez Molina | NIA: 100047555 | Profile: Professor

### List of not started tournaments

	Id	Name	Description	Deadline	Time per round	Number of rounds	Problems per round	Problems added	Different problems	Outstanding requests	Students approved
<input type="checkbox"/>	1	LAO	Shell Script, Makefile, Procesos	2010-12-24 12:32	20	5	10	0	✓	0	0
<input checked="" type="checkbox"/>	21	Driver's Exam	Learning the concepts for the driv	2011-01-04 19:23	20	5	10	0	✓	1	0

Buttons: Delete, Edit, Aprove/Revoke members, Add problems, Start tournament, Select all, Clear selection

Fig. 11. List of not started tournaments page.

example, a teacher can configure each specific tournament round, activate the pairings generation, activate the assignments of problems to students, or enable each round competition.

- (iii) *Finished.* A started tournament finishes when all the tournament rounds take place. A teacher can view all the finished tournaments (“List finished tournaments” option), while a student can view his/her finished tournaments in which he/she has been enrolled.

## 5. Description of a competition round in the system

For the execution of a round in the ISCARE system, teachers need to upload the candidate problems that students must solve during that round, set the specific round parameters (such as the time in which the round competition will take place), enable the calculation of the different pairings for that round and the

assigned problems to solve for each pairing, and start the round. Students must view the round information to know the time for the round competition and the opponent for each round. Students must enter into the round menu at the configured time in order to compete with its opponent for that round. During the round competition, students must solve a set of assigned problems for that pairing. Section 5.1 is about the management of the problems in the ISCARE system, Section 5.2 focuses on the information for each round, Section 5.3 is about the generation of the different pairings for each round and the assignment of problems to pairings, while Section 5.4 is about the on live competition between students in a round.

### 5.1. Management of problems

There are three methods to upload problems in the ISCARE system for a specific tournament. These are the problems that students

### Aprove/Revoke members to current tournament




**Tournament: 21**

Requested members Students: 0		Revoked members Students: 1		Aproved members Students: 15	
NIA	Name	NIA	Name	NIA	Name
		<input type="checkbox"/>	29	<input type="checkbox"/>	14
				<input type="checkbox"/>	15
				<input type="checkbox"/>	16
				<input type="checkbox"/>	17
				<input type="checkbox"/>	18
				<input type="checkbox"/>	19
				<input type="checkbox"/>	20
				<input type="checkbox"/>	21
				<input type="checkbox"/>	22
				<input type="checkbox"/>	23
				<input type="checkbox"/>	24
				<input type="checkbox"/>	25
				<input type="checkbox"/>	26
				<input type="checkbox"/>	27

Buttons: Aprove, Revoke, Delete, Select all, Clear selection

Fig. 12. Page for the approval and rejection of students' requests to tournaments.

### My not started tournaments

Legend:  = Requested |  = Approved |  = Revoked


Id	Name	Description	Deadline	Numrounds	Request state
21	Driver's Exam	Learning the concepts for the driver's ex	2011-01-04 19:23	5	

Fig. 13. List of not started tournaments for a student.

have to solve in the different round competitions. One method is uploading a single problem from a file, another one is uploading a set of problems from a comprised file that contains the different problems, and the third one is uploading problems using a repository of previous uploaded files in the system, instead of making it from files on a local computer.

Fig. 14 shows a form for teachers in which they can add single problems into a tournament. This form can be accessed through the list of not started problems, clicking the “Add problems” link. The different parameters to configure when adding a new problem into the system are the following:

- The upload type corresponds to the three commented types of uploading problems. In this case, it is selected from a single file.
- The valid for rounds option permits to select the different rounds in which the problem can be used for competition. There is a select button for each round (five rounds for this tournament), and the teacher selects the rounds in which this problem can be presented to students as part of the competition. For example, the problem of Fig. 14 can be used in four rounds (rounds 1,2, 4, and 5). Therefore, it is not possible that ISCARE generates this problem in round 3.
- The name of the problem.
- The description of the problem.

- Difficulty. The teacher must select the difficulty of the problem in a scale from 1 to 5. This difficulty is used by the algorithm to assign this problem to students’ pairings in order to try to adapt the students’ knowledge level to the problems difficulty.
- Path: This is the path of an XML file that represents the problem to upload.

Teachers can also select the “Bulk Upload” option in the “Upload Type” parameter. In this case, the parameters are the same as the “Single File” option, but several problems can be uploaded at once instead of only one problem. For this reason, the path is not an XML file but a ZIP file, which contains a couple of XML files that represent the problems to upload in the bulk upload. As it was presented, there is a maximum size for bulk upload configured in the general options of the tool.

The “Add from repository” option allows searching problems to add to the current tournament from an internal repository of the ISCARE system, instead of using an external to the system ZIP or XML file. Fig. 15 shows the searching menu. The search can be done with keywords of the name or title, but also with the difficulty level of the problem. Bellow, once the search is done, there is a list of the problems that are according to the searching criterions, and next some of these problems can be selected to be added to a tournament. The problems that are in the repository are those that

### Add problems to current tournament

#### Tournament info

Tournament: 21 Name: Driver's Exam

#### Problem data

**Upload Type**  Single file  Bulk upload  Add from repository

**Valid for rounds**  1  2  3  4  5

**Name**

**Description**

**Difficulty** (Very Easy)  1  2  3  4  5 (Very Difficult)

**Path (\*.xml)**

Fig. 14. Page for adding problems to a tournament.

Fig. 15. Problems repository search for adding problems to a tournament.

have already been included in the ISCARE system previously, e.g. in previous tournaments.

The problems' repository can also be accessed from the teachers' main menu, but with the purpose of downloading some repository problems to obtain XML files that represent the problems. There is a search menu similar to the one of Fig. 15 for this option, but including an option to download the selected problems, or to delete problems from the repository.

A teacher can upload different problems for a tournament combining the three commented methods. With independence of the three options selected, there is a list, below of the search menu, of the current problems that have been added for that tournament until the moment. Fig. 16 shows an example of the problems added until some moment. From Fig. 16, teachers can see all the information for each problem (e.g. the rounds in which each problem can be applied) and teachers can delete the problems they desire, in case some problem should be redefined, or was added by mistake.

## 5.2. Edition and visualization of rounds information

From the list of not started tournaments, a tournament can be selected and started. In that moment, the tournament passes to the list of current tournaments. From the list of current tournaments, the tournament can be selected and teachers can select the "Enter Tournament" button to enter that started tournament. Once a teacher enters a tournament, the teacher accesses the "manage current tournament" menu (Fig. 17) and the rounds information options (Fig. 18).

From the "manage current tournaments" screen (Fig. 17), a teacher can view the different configured parameters of a tournament (name, description, number of rounds, number of problems per round, time per round, pairing algorithm used to pair the students, and if different problems per round are allowed). In addition, a teacher can view the total number of students that have been approved in that tournament, the total number of uploaded problems for that tournament, the number of the present round in the tournament, and the current round state (a green flag if the round has not been started, a yellow one if the round is in execution, and a red one if the round has finished). When a round is in execution,

then the time left option is enabled, indicating the time left for the round to end.

Teachers can select the "Edit rounds info" button to define some specific information for each round of the tournament. A screen like the one of Fig. 19 appears when this button is selected.

In the "Edit rounds information" screen, the different rounds of the tournament can be edited. For each round, a teacher must set the time when the round will take place and the description of the round. Students can view this information of the rounds that is set by teachers.

Students can see the information about a present tournament through the "My current tournaments" option, and then a screen as Fig. 20 appears.

## 5.3. Generation of round pairings and assignment of problems to pairings

For each round, ISCARE must calculate and generate the following adaptive issues:

- *The round pairings.* The system make pairs, grouping students two by two each round for  $N$  students. There should not be any repetitions of pairings among different rounds in a tournament. The pairings are done by ISCARE according to the selected pairing algorithm.
- *Assigning of problems to each pairing.* For each pairing, different problems must be assigned and generated according to different conditions. First, the only candidate problems are the ones that were marked as valid for a specific round. Next, a percentage of problems (depending on the commented application variables) are selected randomly among the candidates, and the rest are selected taking into account each pairing students' scorings and the difficulty level of the problem (which was provided by the teacher). The algorithm tries to assign problems with a difficult level that is close to the pairing student knowledge level.

A teacher can view the different pairings for a round and the problems assigned to each pairing, using a screen like the one of

Description:

Difficulty: (Very Easy)  1  2  3  4  5 (Very Difficult)

Path (\*.xml):

The problems have been uploaded successfully

### List of uploaded problems in this tournament

Problems uploaded: 31

<input type="checkbox"/>	Id	Nº	Name	Title	Difficulty	Valid for rounds
<input type="checkbox"/>	13	13	Pset	Precauciones con niebla	3	1-2-3-4-5
<input type="checkbox"/>	14	14	Pset	Distancia de seguridad	3	1-2-3-4-5
<input type="checkbox"/>	15	15	Pset	Heridos en accidente	3	1-2-3-4-5
<input type="checkbox"/>	16	16	Pset	Advertencias acústicas	3	1-2-3-4-5
<input type="checkbox"/>	17	17	Pset	Estacionamiento en autopista	3	1-2-3-4-5
<input type="checkbox"/>	18	18	Pset	Marcha atrás	3	1-2-3-4-5
<input type="checkbox"/>	19	19	Pset	Pasos de peatones	3	1-2-3-4-5
<input type="checkbox"/>	20	20	Pset	Tomar alimentos	3	1-2-3-4-5
<input type="checkbox"/>	21	21	Pset	Incorporación a la circulación	3	1-2-3-4-5

Fig. 16. List of uploaded problems in a tournament.

## Manage current tournament:

**Id:** 21      **Name:** Driver's Exam

**Description:** Learning the concepts for the driver's exam

**Different problems per round? :**  (Yes)

**Problems per round:** 10      **Problems uploaded:** 31

**Number of rounds:** 5      **Number of students:** 15

**Present round:** 1      **Present round state:**

**Time rounds:** 20 min      **Time left:** ---

**Pairings Algorithm:** Voracious algorithm

Fig. 17. Management of a started tournament.

Fig. 18 (the different students of the different pairings which are matches, and the problems assigned to each match). If there is an odd number of students, then there is one without a pair (match 1 in Fig. 18), and the ISCARE behavior depends on the configured parameters (a pairing with a fictitious student or that student obtains all the points of the match directly).

For the first round, these calculations are done quickly without further intervention of the teacher. If a round has not finished, neither the next round can be started nor their associated pairings generated (the next rounds are unavailable). When the previous round has finished, then the "Generate next round" button of the present round is enabled (see Fig. 17) but not the "Start round"

button. When the "Generate next round" button is pressed, then the pairings for the next round are calculated and the problems for each pairing are assigned according to the selected options in the system. In addition, after this step, the "Generate next round" button is not enabled, but the "Start round" button is again available. When a teacher starts a round, then the round is in execution during the configured time for a round. During this time, the different students can enter into the system and compete solving the different problems. After this configured time, the round passes to the finished state. Fig. 21 explains a round life cycle with its different states and different buttons or actions that enable the next state of a round.

## Rounds information:

Please select the round you want to view:

Round 1

View round ranking

View round results

Id	Match		Problems to solve
	Student 1	Student 2	
1	14	---	7,30,9,23,6,17,4,13,11,18
2	15	16	24,9,29,1,3,26,21,30,5,18
3	17	18	21,12,24,20,27,2,4,13,5,19
4	19	20	28,17,20,26,15,4,30,2,23,21
5	21	22	23,17,2,1,14,10,25,9,20,4
6	23	24	29,22,16,12,26,9,18,31,15,20
7	25	26	14,20,6,1,4,5,2,22,10,23
8	27	28	15,13,26,27,30,21,28,29,19,16

Fig. 18. Assignment of students' pairings and problems for each match in a round.

## Edit round's information

**Note:** Remember that the rounds information is only to report the students the approximate hours of which will begin rounds, but the rounds will have to be started manually by the professor that administrates the current tournament.

Also remember that will only be displayed those rounds that have not been started yet

### Tournament info

**Tournament:** 21  
**Name:** Driver's Exam  
**Description:** Learning the concepts for the driver's exam

#### Rounds information

	Id	Start time	Information
<input checked="" type="checkbox"/>	1	2011-01-02 18:43:04	--- Empty ---
<input type="checkbox"/>	2	2011-01-02 19:03:04	--- Empty ---
<input type="checkbox"/>	3	2011-01-02 19:23:04	--- Empty ---
<input type="checkbox"/>	4	2011-01-02 19:43:04	--- Empty ---
<input type="checkbox"/>	5	2011-01-02 20:03:04	--- Empty ---

#### Edition Form

**Id**

**Start time**  
 2011-01-02 18:43:04

**Information**

Fig. 19. Edition of a round information.

#### 5.4. Round competition between pairs

When a teacher starts a round, next students can enter into the round to answer the different assigned problems for this round in the established limit of time. An example of students view in this phase is shown in Fig. 22. On the top, students can view the

identification number of the tournament and the round number in which they are competing.

On the right, a student can view the time left for this round to finish (when time is over, then the screen of Fig. 22 disappears). A student can view its own information and the opponent's information on live during the round. This information includes its

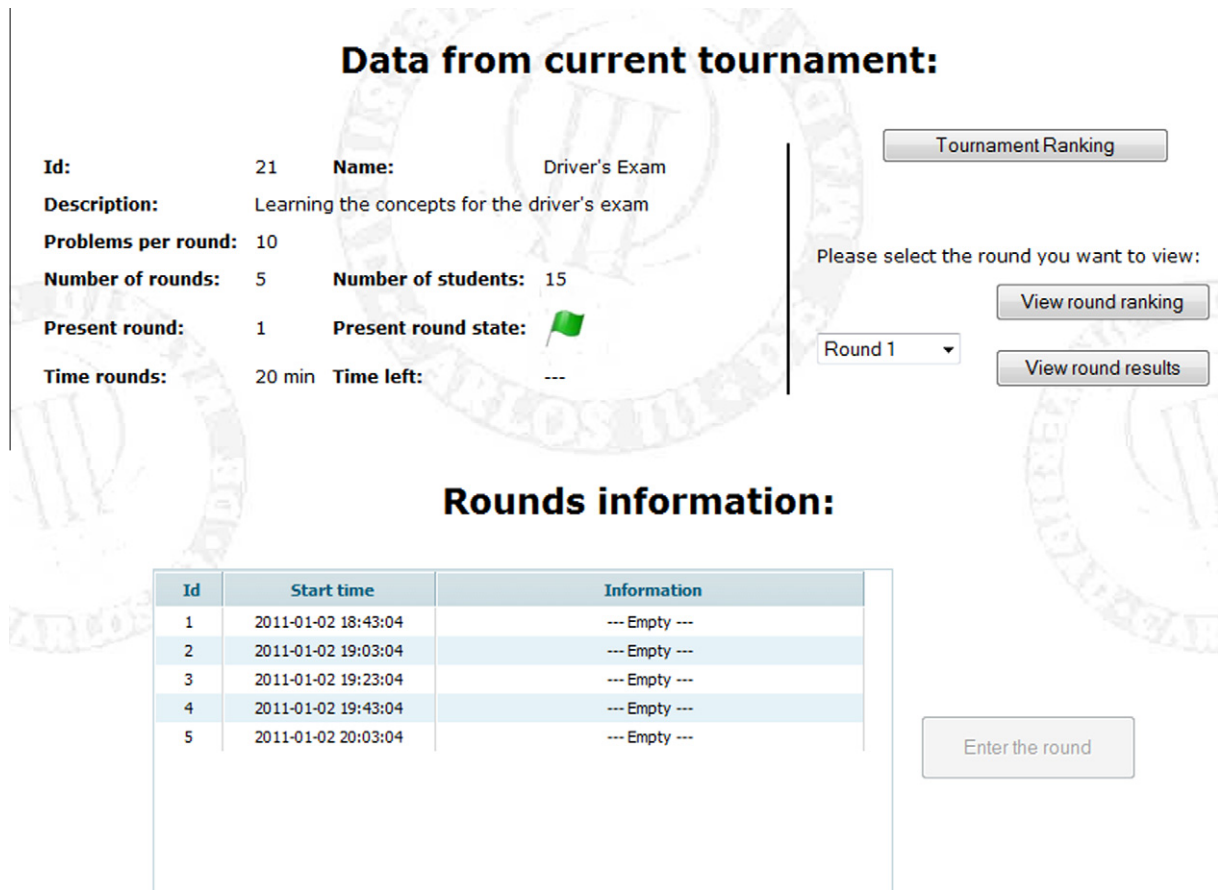


Fig. 20. Data from current tournament.

name, NIA identifier, average score (this is an average of the student's scoring in all the finished tournaments without including the present tournament), tournaments points (it is in the X/Y format, where X is the number of points of the student in that moment, and Y is the maximum number of possible points in the tournament until that moment), internal round points (it is in the T/Z format, where T is the number of points of the student in that moment in this round, this is with the number of answered problems until that moment, and Z is the maximum number of possible points in this round), and resolving (this indicates the number of problems that have been solved by this student in a specific round).

It is important to note that, from the point of view of the global scoring of a tournament, the points that can be obtained from a round is a maximum of 2, but the internal points of a round during the students' solving of problems is a different concept with a different scale. The maximum number of internal points for a round is not fixed but depends on the specific scorings assigned to the different problems for a specific students pairing. The maximum internal score points for a round (this is Z) is the same for a student and its opponent for that round, and is the sum of all the specific problem maximum scores assigned for that round (which are visualized one by one by as the different problems are presented to the student). In the same round, different pairings can have different maximum internal round points depending on the problems assigned. But for the tournament points, a round can score a maximum of two points for all the students of all the pairings. The calculation of the round points for a tournament depends on the internal round points, but also on the comparison of the student with its opponent. The specific calculation will be presented in Section 6.

On the left (Fig. 22), the present problem to solve is presented. A student can see the problems title, maximum score and the type of problem. The problem can contain images and text. Three types of problems are supported in the ISCARE tool: multiple choice (several options and only one is correct), multiple response (several options and several of them are correct) or fill in the blank (a correct word must be written in the selected space). A student can answer the problem using the "Submit Answer" button. Finally, a student has to select the "Next Problem" button to go through the next question.

**6. Information about the users and problems in the system**

When time for a round is over, the round finishes and ISCARE calculates a round assignment of points for all students. The final

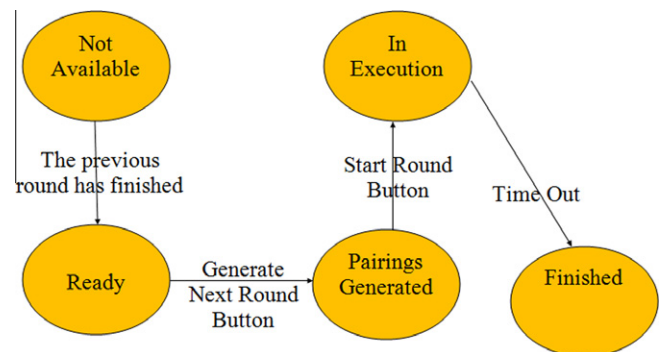


Fig. 21. Round life cycle.

**Tournament 21 - Round 1**  
**Resolving problem 1 of 50**

**Problem data:**

- Title: Pasos a nivel
- Problem max. score: 1.0
- Type: choice

---

**Problem:**

Question

- Answer 1
- Answer 2
- Answer 3

Time left: 58 min, 6 seg

**My information:**

- NIA: 56
- Name: Diego
- Average score: 79.0%
- Tournament Points: 0.0 / 2.0
- Round points: 0.0 / 50.0
- Resolving: 1<sup>st</sup> problem

**Opponent information:**

- NIA: 57
- Name: Maria
- Average score: 90.5%
- Tournament Points: 0.0 / 2.0
- Round points: 0.0 / 50.0
- Resolving: Not started yet

Fig. 22. A problem view during the competition in a round.

students scoring for the tournament related to a round is in the interval  $[0,2]$  and it is calculated as the sum of these two terms:

- *Individual points.* A scoring in the interval  $[0,1]$  depending only on the student's performance on the different problems. Internally, the round points can have an undefined maximum number of points that is the sum of the maximum points for each of the assigned problems for that pairing. The obtained student result points out of this maximum is scaled to the  $[0,1]$  interval.
- *Collective points.* A scoring in the interval  $[0,1]$  depending on the comparison of the student's result in the round with respect to its opponent. The result in a match can be a clear victory (1–0), close victory (0.75–0.25), draw (0.5–0.5), close defeat (0.25–0.75) or clear defeat (0–1).

Students and teachers can view the different match results of a finished round of a started or finished tournament, and an image

like the one of Fig. 23 is presented, in which the final scoring is in the interval  $[0,2]$  and is obtained and presented for all students and matches, separating individual and collective points.

In addition, students and teachers can view the different ratings after each round of a started or finished tournament, this is the obtained points by all students until the end of each round, and the ordered list of students according to their global tournament points until that moment. An example can be seen in Fig. 24. The maximum possible points for a student after a round  $R$  is  $2 * R$ . If there is a draw for several students according to the global number of points, then there is an additional calculation to avoid the draw: scorings of the opponent students in previous rounds are taken and summed, and students are rated better in case of draw if this sum is greater, because this can imply that he/she played with students of greater level.

The commented options can be accessed through the started or finished tournaments links, selecting the proper round. Moreover,

**Tournament: 2 - Torneo de prueba**  
**Results for the Round N° 1 on 2008-09-11 20:20:46**

Match	NIA	Name	Round results			Name	NIA
			Ind. Pts	Collective Pts	Ind. Pts		
1	56	Diego	0.2	1.0 / ---	---	---	---
2	57	Maria	1.0	0.75 / 0.25	0.8	Patricia	61
3	59	Andres	0.8	0.25 / 0.75	1.0	Victor	60
4	58	Angel	0.8	0.5 / 0.5	0.8	Pedro	70
5	63	Jesus	0.4	0.25 / 0.75	0.6	Jose	64
6	62	Alejandra	0.8	0.75 / 0.25	0.6	Luis	73
7	66	Raquel	0.4	0.75 / 0.25	0.2	Sergio	67
8	65	Francisco	0.4	0.25 / 0.75	0.8	Quique	75
9	68	Ana Isabel	1.0	1.0 / 0.0	0.2	Jose Luis	69
10	71	Manuel	1.0	1.0 / 0.0	0.4	Pablo	72
11	74	Esperanza	1.0	0.75 / 0.25	0.8	David	76

Fig. 23. Results of a round per pairings.

**Ranking for the Tournament: 2 - Torneo de prueba**

Note: This results are provisional because the tournament is still in progress...

Max. points for the current tournament: 10.0

Pos	NIA	Name	Email	Points achieved	Player average	Opponents average
1º	57	Maria		9.05	90.5%	52.3%
2º	56	Diego		7.9	79.0%	49.0%
3º	71	Manuel		6.2	61.999%	64.9%
4º	76	David		6.05	60.5%	57.2%
5º	75	Quique		5.6	56.0%	47.1%
6º	59	Andres		5.6	56.0%	46.2%
7º	74	Esperanza		5.5	55.0%	62.0%
8º	68	Ana		4.8	48.0%	46.9%
9º	54	Jose		4.8	48.0%	41.9%
10º	70	Pedro		4.7	46.999%	46.8%
11º	72	Pablo		4.65	46.5%	46.1%
12º	66	Raquel		4.55	45.5%	49.0%
13º	60	Victor		4.45	44.499%	63.0%
14º	58	Angel		4.4	44.0%	55.8%
15º	65	Francisco		4.05	40.5%	43.5%
16º	61	Patricia		3.95	39.5%	53.6%
17º	53	Jesus		3.85	38.5%	42.5%
18º	67	Sergio		3.65	36.5%	45.125%
19º	62	Alenjandra		3.6	36.0%	43.4%
20º	73	Luis		3.45	34.5%	42.0%
21º	69	Jose Luis		3.2	32.0%	41.625%

Fig. 24. Students' rating screen.

teachers can see the global students' ranking that is an average of the students' participation in the different tournaments, which is in the interval [0,100].

In addition, teachers can view the global problems' ranking, which gives information about the problems that were easier to answer by students and the ones that were more difficult, showing the number of times that a problem was answerer, the correct answers, or the variance.

## 7. Conclusions

This work presents a detailed description of our ISCARE tool, which is an ITS that enables competition among students in an educational environment. The paper shows the different features of the tool and how they are innovative with respect to previous educational competitive systems. With the description, it is shown the specific innovative modeling of the tool, including its different features, how they are easy to use, the algorithm selection or the selected architecture.

The ISCARE tool includes problem solving capabilities with a specific scoring calculation, artificial intelligence capabilities (for the adaptation of the different pairings for each round, the adaptation of problems depending on the students knowledge level, the on-live visualization of matches, or the different ratings), and management capabilities (with the defined life cycles for tournament or rounds).

As future work, we are planning to perform experiences in the classroom with this tool in order to compare learning gains obtained when using this tool with other situations, test the motivational impact of the tool or measure the enjoyment enrollment produced by ISCARE.

## Acknowledgements

Work partially funded by the Learn3 project TIN2008-05163/TSI and the EEE project TIN 2011-28308-CO3-1 within the Spanish "Plan Nacional de I+D+I", and the Madrid regional community project eMadrid S2009/TIC-1650.

## References

- Amori, A., Naicker, K., Vincent, J., & Adams, C. (1999). The use of computer games as an educational tool: Identification of appropriate game types and game elements. *British Journal of Educational Technology*, 30(4), 311–321.
- Behrooz, P. (2009). Motivating computer engineering freshmen through mathematical and logical puzzles. *IEEE Transactions on Education*, 52(3), 360–364.
- Burguillo, J. C. (2010). Using game theory and competition-based learning to stimulate student motivation and performance. *Computers & Education*, 55, 566–575.
- Carrington, D., Baker, A., & van der Hoek, A. (2004). It's all in the game: Teaching software process concepts. In *ASEE/IEEE frontiers in education conference FAG* (pp. 13–18).
- Chang, L., Yang, J., & Yu, F. (2003). Development and evaluation of multiple competitive activities in a synchronous quiz game system. *Innovations in Education & Training International*, 40(1), 16–26.
- Chen, Z., Liao, C. C. Y., & Chan, T. (2010). Learning by pet-training competition: alleviating negative influences of direction competition by training pets to compete in game-based environments. In *IEEE international conference on advanced learning technologies* (pp. 411–413).
- Conklin, A. (2006). Cyber defense competitions and information security education: An active learning solution for a capstone course. In *Hawaii international conference on system sciences* (pp. 1–6).
- Chua, H., Liaob, M., Chenc, T., Lind, C., & Chen, Y. (2011). Learning case adaptation for problem-oriented e-learning on mathematics teaching for students with mild disabilities. *Expert Systems with Applications*, 38(3), 1269–1281.
- Ediger, M. (1996). Cooperative learning versus competition: Which is better? *Journal of Instructional Psychology*, 204–300.
- Feng, M., Heffernan, N. T., & Koedinger, K. R. (2006). Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In *International conference on intelligent tutoring systems* (pp. 31–40).



- Guzman, E., & Conejo, R. (2005). Self-assessment in a feasible, adaptive web-based testing system. *IEEE Transactions on Education*, 48, 688–695.
- IMS QTI, Question and Test Interoperability Specification (2005). <<http://www.imsglobal.org/question/>>.
- Kohn, A. (1986). *No contest: The case against competition*. Boston: Houghton.
- Lawrence, R. (2004). Teaching data structures using competitive games. *IEEE Transactions on Education*, 47(4), 459–466.
- Lin, C., Young, S. S., & Hung, H. (2008). The game-based constructive learning environment to increase english vocabulary acquisition: Implementing a Wireless Crossword Fan-Tan Game (WiCFG) as an example. In *IEEE international conference on wireless, mobile, and ubiquitous technology in education*.
- Liu, Z., & Liu, Z. (2005). Building an intelligent pedagogical agent with competition mechanism to improve the effectiveness of an educational game. In *Workshop on educational games as intelligent learning environments*.
- Pfotenhauer, J. M., Gagnon, D. J., Litzkow, M. J., & Blakesley, C. C. (2009). Designing and using an on-line game to teach engineering. In *Frontiers in education conference, W2C* (pp. 1–5).
- Regueras, L. M., Verdu, E., Verdu, M. J., & de Castro, J. P. (2010). Design of a competitive and collaborative learning strategy in a communication networks course. *IEEE Transactions on Education*.
- Silva, B., & Madeira, R. N. (2010). A study and a proposal of a collaborative and competitive learning methodology. In *IEEE eng. education conference* (pp. 1011–1018).
- Sindre, G., Natvig, L., & Jahre, M. (2009). Experimental validation of the learning effect for a pedagogical game on computer fundamentals. *IEEE Transactions on Education*, 52(1), 10–18.
- Squire, K., & Jenkins, H. (2003). Harnessing the power of games in education. *Insight*, 3, 6–33.
- Striegel, A., & Rover, D. T. (2002). Problem-based learning in an introductory computer engineering course. In *Frontiers in education conference, FIG* (pp. 7–12).
- Verhoeff, T. (1987). *The role of competitions in education*. <<http://www.olympiads.win.tue.nl/loi/loi97/ffutwrl/competit.html>>.
- Wang, C., Wanga, D., & Lin, J. (2010). ADAM: An adaptive multimedia content description mechanism and its application in web-based learning. *Expert Systems with Applications*, 37(12), 3639–8649.
- Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational approaches to the communication of knowledge*. Los Altos, CA: Morgan/Kaufmann Publishing Co..
- Yu, F. Y. (2001). Reflections upon cooperation-competition instructional strategy: Theoretical foundations and empirical evidence. *The National Chi Nan University Journal*, 5(1), 181–196.