A playful approach for learning Intelligent Systems in Engineering

Carlos A. Iglesias, Mercedes Garijo and Francisco Santiago
Departamento de Ingeniería de Sistemas Telemáticos
E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid
28040 Ciudad Universitaria s/n Madrid (Spain)
Email: {cif,mga}@dit.upm.es, frsantiago@alumnos.upm.es

***Abstract*—This paper presents a ludic environment for learning intelligent systems, based on programming soccer teams in the Robocup competition. The environment developed by the faculty group follows the rules of the RoboCup league, incorporating facilities for programming intelligent soccer players using rules. This article describes the pedagogical environment and the teaching experience since 2001 in the Telecommunications Engineering School from the Polytechnic University of Madrid in the subject of fifth course "Laboratory of Intelligent Systems". In conclusion, the approach has promoted the recreational interests and participation of students, that should face practical resolution of complex issues such as managing coordination, noisy environment, as well as dealing wih proactive and reactive behaviours.**

I. Introduction

Changing the curriculum for the European Space of Higher Education is a challenge regarding how the learning process of students is managed.

In this article, we present our experience with a playful approach in the soccer domain. In addition, we present a software library developed for teaching purposes.

The RoboCup competition [9] is an international project to promote artificial intelligence, robotics and related fields to be held since 1997 and with several categories, for teams of programs and machines (robots) to play soccer.

The RoboCup competition offers a very interesting environment for experimentation with a complex simulator, able to perceive and act in real time. It provides an excellent resource for experimenting with distributed decision-making and coordination techniques in a competitive environment.

The article shows how a ludic learning strategy can result in an incentive and stimulus for learning and, at the same time, this strategy can facilitate the understanding of complex concepts, through a combination of experimentation and teamwork in a complex domain.

The learning environment Robocup GSI [5] has been used in the Telecommunications Engineering School of the Universidad Politécnica de Madrid since 2001 in the subject of Intelligent Systems Laboratory. In particular, it is used in the last assignment, which is the largest and where students can select from a range of available assignments about multiagent systems.. The assignment using the library-Robocup GSI has been selected every year by at least one third of the students, although it is an assignment that requires great dedication.

The rest of the article is structured as follows. The section II makes an introduction of the Robocup competition. Then the section III presents the teaching strategy followed and the skills we want students to acquire. Next section IV presents the functionality offered by the library GSI-Robocup. Then section V we present the lessons learnt based on our teaching experience. Finally, section VI presents the general conclusions of the article and our lines of work.

II. The Robocup competition

The Robocup competition [1] is an international project to promote artificial intelligence, robotics and related fields. The competition takes place since 1997 in various categories, where the objective is to program intelligent systems that play soccer.

One of the categories of the RoboCup is the simulation league, which has leagues in two and three dimensions.

This article focuses on the league simulation in two dimensions. The simulator of the RoboCup is a distributed system whose architecture is shown in figure 1. The central element is the simulator (RoboCup soccer server) [2], which acts as a server and performs the soccer simulation. Along with the server, it is distributed a program that displays the game in two dimensions (RoboCup soccer monitor), and there are other monitors available in three dimensions.

Communication between the simulator and the clients is done via sockets UDP / IP, making it easier for players to be programmed in any language. Players on teams acting as clients, once connected to the server, receive and send sensory information to control each player commands (move a distance in one direction, shoot with a certain strength, etc.)..

The simulator can use a manual or automatic trainer, and defines rules on the duration of the game, outside game, or number of parties and duration of each. Furthermore, define rules of conduct, to prevent massive sending of orders to the server

Agents have three types of sensors: auditory, visual and corporal. The ear sensor transmits the audio messages of the coach, the referee and other players on your team. The

978-1-4244-5386-3/09/$26.00 ©2009 IEEE

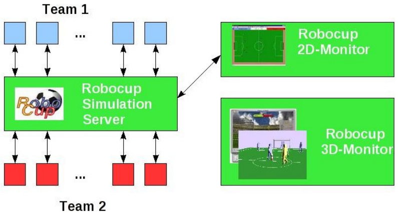

Fig. 1. Robocup Simulator Architecture

simulator limits the hearing of the players to avoid excessive use of this facility and receive messages only if both are less than a threshold. The visual sensor periodically reports on objects the player sees as well as their relative position to the player.

Objects can be the goal line, other players and the ball. The player also has a vision that depends on his position in the field and the position of his head. In order to introduce noise, the simulator introduces uncertainty on the position of distant objects. The sensor provides information on the physical fitness of the player, including the quality and breadth of vision, speed and current direction, the direction of the head and body with respect to the power of the player. The simulator decrements this power as the player makes physical exercise, which limits their speed. In order to include the unexpected movement of objects as in the real world, noise is introduced in the model of moving objects. Players can accelerate to a position, shoot, turn the body or head, or send messages. The goalkeeper is a special player who can catch the ball, and has specific simulation parameters.

III. The game strategy to facilitate the acquisition of skills

In the Spanish university we are in a phase change [4], [1] with the adaptation of the Spanish university system to the European Space of Higher Education. One of the most important issues is the change of approach in the orientation of education toward a model more focused on student learning and competences acquisition.

Since the Robocup addresses a domain so popular and attractive, this has facilitated the Robocup has been previously applied to the teaching of intelligent systems and robotics [10], [7]. Depending on the competences to be acquired, its pedagogical use can be different, obviously, covering topics such as robotics with physical elements, programming languages like C or Java, or, as is our case, declarative programming based on rules.

The RoboCup offers an experimental environment that encourages the acquisition of basic cognitive competences. The problem requires to handle the complexity to receive and operate in real time and also the decision-making process is distributed, and offers facilities for coordination in a competitive environment. The skills to be able to think in this environment requires the application of concepts from multi-agent systems and, in our case, also from rule-based declarative programming.

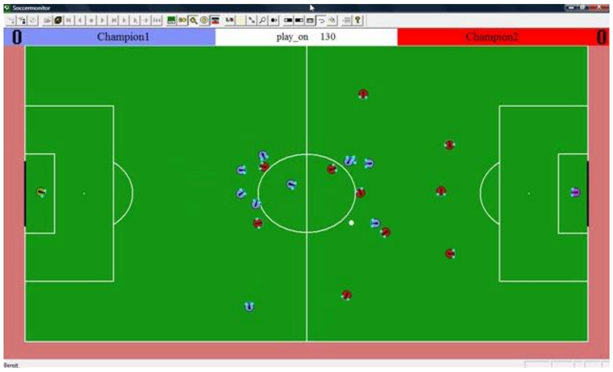

Fig. 2. Sample of a simulation

In our case, we have used the Robocup for acquiring specific competences of the degree, as well as transversal competences. Specific skills which have been considered are:

- Capacity for analysis and synthesis of distributed systems and services in real time.
- Ability to implement, debug and program a distributed service.
- Ability to build fault tolerant distributed systems capable of self-reorganisation.
- Reasoning ability with rule-based declarative programming.
- Ability to define coordination mechanisms between telematics systems.

From the perspective of the transversal competences, the assignment seeks to acquire the following competences:

- Instrumental competences
  - Capacity for analysis and synthesis. The system is complex, and the student only knows the reasoning of his own team. Furthermore, it is complex to determine the overall behaviour of the team from the decisions of individual players.
  - Organisational skills and planning.Students, working in teams, must organise themselves to determine what strategy should follow themselves to win the other team.
  - Oral and written communication. The assignment is evaluated with a written report and a public presentation of the results, so that all students learn from the work of others.
  - Knowledge of foreign language (English). The original manuals are in English.
  - Troubleshooting. During the assignment, since it is an environment that simulates reality, and introduces

noise in the communications, and in addition, messages may be lost, students should address these problems and define strategies for addressing them.

- Personal competences
  - Teamwork. The assignment is done in teams, benefiting from the sharing of effort and creativity of its members.
- Learning competences
  - Autonomous learning. Although guidelines have been developed to facilitate learning, students should learn for themselves based on these guidelines.
  - Creativity. In our view, reinforces the learning fun and allows the acquisition of this competence. There is not a soccer coach to please everyone, and strategies for a winning team are numerous (definition zonal player, usage of coordination messages, etc.).

Students are organised into teams. The fact that the domain is known, and has a large number of soccer fans, promotes the fun aspect of learning, serving as a highly motivating element for students. This aspect favours encouraging creativity among students and teamwork. In addition to solving the problem, which is complex, it is necessary not only know the theory (know), but putting it into practise (know-how) as well as consulting the literature available on the Internet.

IV. The library GSI-Robocup

GSI-Robocup library was created to provide an environment that simplifies the programming of intelligent agents for the RoboCup, providing a programming environment that allows them to abstract from problems they have learnt in previous courses, such as:

- Programming of the connection to the server, and management of UDP sockets.
- Programming a parser for complex messages that the server sends to the players.
- Programming and modeling of the objects in the field.
- Programming a model of relative positions of the Robocup, which requires making a geometric processing of information received.

In our case, we wanted to reinforce rule-based declarative programming, as this is usually one of the major difficulties encountered by students, who are accustomed to imperative programming.

Althought the library abstracts from many aspects, as happens in real environments, students may have to inspect these features when the system does not meet their expectations. This can be done through an integrated log system.

The library GSI-Robocup [11], [5], [3], [8] allows to program soccer players using the rule-based declarative programming language Jess [6].

The main functions offered by the library are:

- Management of communications with the server UDP Soccer Server, managing the establishment of connections, as well as receiving and sending messages.
- Analysis of messages received from the server, updating the knowledge of the player and the environment, and obtaining the absolute position of objects using geometric calculations.
- Integration of reasoning module programmed with the declarative language Jess that allows reasoning about the objects of the Java environment.
- Management of threads to receive and process messages, so messages are not lost.
- Management of threads to run the game. Although the agents can be launched as separate processes, they initiate as threads for efficiency.

GSI-Robocup library allows students to focus on programming planners using Jess. This layer updates the working memory with the information received.

For illustrative purposes, table 1 shows an example of a rule turnToMate to turn towards a mate: if the game is active (in “play_on”), and the player has the ball near and far other mates, the player turns towards a mate.

Table 1 Example of rule turnToMate

```
(defrule turnToMate
   (reactivation)
    (playMode (realPlayMode "play\_on"))
    (ball (distance ?ballDistance)
          (direction ?ballDirection))
    (position (id "p teamMate 1")
              (distance ?mate1Distance)
    (direction ?mate1Direction))
        ; ball near and far the teammate
    (test (and (<= ?ballDistance 1)
               (> ?mate1Distance 1)
         ;body does not point to him
               (>= (call Math abs
                 ?mate1Direction) 5)
          )
    )
=>
  ;we turn
   (turn ?mate1Direction)
)
```

V. Teaching experience

GSI-Robocup library has been used in the Telecommunications School of the Universidad Politécnica de Madrid since 2001 in the subject of Intelligent Systems Laboratory. In particular, it is used in the last assignment, which is the largest. In this assignment, students can choose among several alternatives for developing a multiagent system. The GSI-Robocup has been selected every year by at least one third of the groups, even though it is a assignment that requires great dedication. Each year, teams are exceeding our expectations. Students are able to develop teams which exhibit different behaviours (forward, defender, goalie, ...) and also develop dynamic strategies (if I'm losing more aggressive, to pass between partners, ...). Moreover, even though they have previously developed an expert system with Jess, there is a notable improvement in the style of programming in this assignment, because of the impossibility to predict the order in which the rules will be executed.

The main difficulties encountered by students were:

- Difficulties in debugging. For many students, this project is their first contact with the programming of distributed systems without a centralised log. While the RoboCup offers a (em) logPlayer that allows playback of recorded game, it is necessary to see why rules are activated or not. In addition, we have to consider the noise introduced by the simulator.
- Difficulties with the processing of messages. Analysis of messages on the server is complex and time consuming, and transport is not guaranteed, which results in loss of messages.
- High consume of resources (CPU and memory).

To overcome these problems, the platform has been redesigned, optimising the processing of messages, and implementing agents to be managed as threads. There have been extensive tests with JUnit in order to reduce the required memory and processing requirements. Furthermore, in order to accelerate learning, tutorials have been developed in the e-learning tool Moodle. In addition, students can consult the teams developed by students in the previous years.

VI. Conclusions and Future Work

Our experiences with the library GSI-Robocup were satisfactory and showed that motivates the students, especially soccer fans, who are a large majority.

From a pedagogical point of view, the Robocup domain allowed to teach the main notions of intelligent agents, such as perception and actuation with the environment, communication and coordination with other agents and planning. Regarding coordination between agents, it is common that students try to make an explicit coordination based on direct communication, but since the simulator introduces noise, it is common that the students explore more sophisticated coordination schemas such as organisational coordination based on roles.

This year we have worked mainly on improving the documentation and creating examples and tutorials in order to accelerate the learning of the platform, as well as in its optimisation.

Currently, we are working on an on-line competition system where students will be able to submit their teams and receive the results.

Acknowledgement

The authors would like to thank all students who have contributed to the development of the platform, Álvaro Martínez [11], Roberto Somolinos [8], Luis Manuel Cruza [3], Victor Sanchez and Pablo Muñoz, and the students of Laboratory Intelligent Systems for their suggestions during these years.

References

[1] Real decreto 1393/2007 which regulates university studies in spain. boe 260 on 30/10/07, october 2007.

[2] M. Chen, E. Foroughi, F. Heintz, Z. Huang, K. K. Spiros Kapetanakis, J. Kummeneje, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, and X. Yin. Users manual. robocup soccer server for soccer server version 7.07 and later. Technical report, The Robocup Federation, Aug. 2002. Available at http://sserver.sourceforge.net/docs/manual.pdf.

[3] L. M. Cruza. Aplicación web de composición de agentes inteligentes para la competición robocup. Master's thesis, E.T.S.I. Telecomunicación, Univ. Politécnica de Madrid, 2006.

[4] M. de Miguel Díaz. daptación de los planes de estudio al proceso de convergencia europea, programa de estudios y análisis destinado a la mejora de la calidad de la enseñanza y de la actividad del profesorado universitario. October 2004.

[5] Grupo de Sistemas Inteligentes (GSI), Departamento de Ingeniería de Sistemas Telemáticos, E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid. Web page of gsi robocup.

[6] Sandia National Laboratories. Web de Jess. The Rule Engine for the Java Platform.

[7] E. Sklar, S. Parsons, and P. Stone. Robocup in higher education: A preliminary report. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *RoboCup*, volume 3020 of *Lecture Notes in Computer Science*, pages 296–307. Springer, 2003.

[8] R. Somolinos. Desarrollo de un entorno java-jess de composición de habilidades de agentes inteligentes para la competición robocup. Master's thesis, E.T.S.I. Telecomunicación, Univ. Politécnica de Madrid, 2003.

[9] The Robocup Federation. Web page of robocup competition.

[10] J. Vidal and P. Buhler. Teaching multiagent systems using RoboCup and biter. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 4(2), 2002.

[11] Álvaro Martínez Reol. Desarrollo de un sistema multiagente integrando jess y java para la robocup. Master's thesis, E.T.S.I. Telecomunicación, Univ. de Valladolid, 2000.