FC Portugal - High-level Coordination Methodologies in Soccer Robotics

Nuno Lau and Luis Paulo Reis
University of Aveiro
Portugal
University of Porto
Portugal

Abstract

This chapter presents the research performed in the context of FC Portugal project in the areas of agent architectures, coordination methodologies, coaching and agent development tools. FC Portugal's research has been integrated in several teams that have participated with considerable success in distinct RoboCup leagues and competitions. The chapter includes a brief description of the main competitions in which FC Portugal has participated with focus in the simulation leagues and related challenges. It also presents some of the developed techniques and results achieved in controlled experiments. These results, together with the impressive record of results achieved by FC Portugal teams in RoboCup competitions show that the techniques developed can significantly improve any soccer robotics team performance.

1. Introduction

This chapter will be structured around the main contributions of the FC Portugal project in the field of Soccer Robotics.
Initially, the environment of the RoboCup Simulation League Competitions (2D and 3D Simulators, Physical Visualization and Nanogram), and also of the Middle-Size League will be presented, as they form the base where the contributions have been applied.
The main research goal of FC Portugal project is the development of a formal model for the concept of team strategy for a competition with an opponent team having opposite goals, general enough to be instantiated to various dynamic competitive domains. The formal model enables the design of an agent architecture suitable for RoboCup simulation league agents and a world state model capable of storing the information needed for an intelligent agent to play soccer. It has been applied in Simulation (2D, 3D, Coaching and Physical Visualization) and Real Robot RoboCup Soccer Leagues (middle-size, small-size and legged leagues) allowing for a flexible and structured strategy instantiation. It has also been applied to other domains such as the RoboCup Rescue.

Source: Robotic Soccer, Book edited by: Pedro Lima, ISBN 978-3-902613-21-9, pp. 598, December 2007, Itech Education and Publishing, Vienna, Austria

The project research focus is also concerned with developing general decision-making and cooperation models for soccer playing. Cooperation mechanisms include the Situation Based Strategic Positioning and Dynamic Positioning and Role Exchange mechanisms.

Situation Based Strategic Positioning (SBSP) mechanism is used to calculate the strategic positionings of all agents in the team according to the game situation. The agent autonomously calculates its base strategic position, adjusting it according to the ball position and velocity, situation and player type strategic information. This mechanism enables the team to move similarly to a real soccer team, covering the ball while remaining distributed along the field allowing for a cooperative positioning among autonomous agents.

The Dynamic Positioning and Role Exchange (DPRE) enables players to exchange their positionings and player types in the current formation if the utility of that exchange is positive for the team. Positioning exchange utilities are calculated using the distances from the player's present positions to their strategic positions and the importance of their positionings in the formation on that situation.

Communication languages and protocols, to convey the most relevant information at the right times to players were also developed. Research was also focused on intelligent control of players' sensors to achieve maximum coordination and world state accuracy.

Coaching is a very important research topic in Soccer Robotics. Coach Unilang – a general language to coach a (robo)soccer team was developed to enable high-level communication between a coach agent (or human coach) and a soccer robotics playing team. Our FC Portugal coach conveys strategic information to players, while keeping their individual decision autonomy, by using this language.

FC Portugal is also very concerned with the development of agent evaluation and monitoring tools like our offline client methodology, that permits the reiteration of the execution of the agent without real-time constraints allowing a precise insight into the agent's reasoning; WstateMetrics, that evaluates the accuracy of world states, and is used to assess the development of the intelligent sensors and communication protocols capabilities; and Visual debugger used to graphically, and in a very intuitive way analyze the reasoning of all the agents in the team in a integrated manner.

While most of these techniques have been developed in the context of simulated soccer robotics, some are already applied in real robot teams and other domains.

The chapter presents the evaluation of the developed techniques performed in controlled experiments and also in real competitions where teams that resulted from the presented research have participated.

The rest of the chapter is structured as follows. Section 2 presents the RoboCup initiative focusing in the rules of the competitions where FC Portugal research has been applied and tested. Section 3 is focused on some of the aspects of the architecture of FC Portugal agents used in most of our teams. In section 4 the developed coordination methodologies are presented while section 5 presents and gives directions into our coaching research. Section 6 is devoted to the principles and implementation of our multi-agent development tools and section 7 presents the results of some controlled experiments that were performed to test the coordination methodologies. Finally, section 8 draws the conclusions.

2. RoboCup International Initiative

RoboCup is an international initiative that aims to motivate the research on multi agent systems and intelligent robotics (Kitano, H., et al. 1997). Every year RoboCup organises scientific meetings and world robotic competitions in the fields of robotic soccer and robotic search and rescuing. Some competitions use simulated environments while others use real robots.

In the real robots competitions there are several leagues including the middle-size, small size, four-legged (based on AIBOs from Sony) and humanoid soccer leagues. These competitions, besides having different rules concerning robots and field dimensions differ in autonomy of the robots and robot construction details. While in the middle-size and legged league, robots are autonomous and sensors are mounted on the robots, in the small-size league a single agent may decide and send commands to every robot of a team. The agent of the small size league typically receives information from a camera positioned above the field to detect the robots and ball positions. The middle-size league will be explained in more detail, as some of the coordination methodologies developed by FC Portugal are currently being transferred to this league. In simulated environments there are several competitions that are detailed in the following sections.

2.1 Middle-Size League

In the middle size league two teams of at most 6 real autonomous robots play soccer in a 18x12m field. Robots height is limited at 80cm, the radius is limited at 50cm and the weight is limited at 40kg. Robots are completely autonomous, although they are allowed to communicate each other, and all sensors must be mounted on the robots. The environment is color marked i.e. the field is green, the lines are white, goals are painted in blue and yellow and the ball is orange. Robots must be black except for the markers of each team that must be cyan and magenta.

The mechanical and electrical/electronics solutions found to build the robots play a very important role in the final efficiency to play soccer. Also the vision subsystem is critical for the final performance of the robots. Games are very active and interesting and the top teams exhibit some very interesting coordinated behaviour.

2.2. Simulation 2D League

RoboCup Simulation League is one of the 3 leagues that started the RoboCup official competitions in 1997. In fact a demonstration of the soccer simulator used in this competition had already been done during the pre-RoboCup 2006. The view of the RoboCup Organizers is to concentrate the research in this league at the top-level modules of the soccer robotics problems: the high-level decision and the coordination of teams of, possibly heterogeneous, robots. Over the years the 2D simulator has evolved, including new features and tuning some others, but the core architecture of the simulator is the same as the one used in 1997.

In the 2D simulation league a simulator, called soccerserver (Chen et al., 2007), creates a 2D virtual soccer field and the virtual players, modelled as circles (Fig. 1). The simulator implements the movement, stamina, kicking and refereeing models of the virtual world. The models in the simulator are taken from real robots (like the differential drive steering) and from human-like characteristics (like the stamina model).

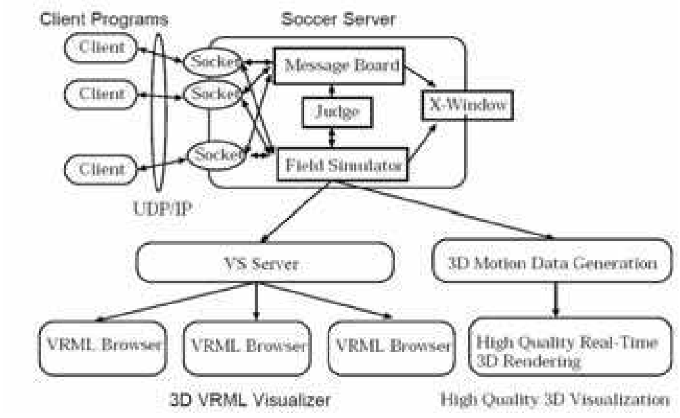


Fig. 1. 2D simulator architecture from (Kitano, H., et al. 1997)

Teams must build the software agents that control each of the 11 virtual robots and also a coach agent. The control of the agents is performed by sending commands to the simulator. The main commands are *dash(dpower)*, *turn(tpower)*, *kick(kpower,kangle)*, *tackle(tangle)* and *catch(cangle)* (used only by the goalie). The simulator implements several virtual sensors for each robot and sends the measures of these sensors to the robots periodically. The most important sensor is the vision sensor, but there's also a sense body sensor (that informs the player of its stamina and own speed) and a hearing sensor. Sensory data is in general subject to noise or to some other type of pre-processing that precludes the agents from knowing the exact value of the measures.
The simulation advances in steps of 100ms, meaning that every 100ms the positions and velocities of every player and of the ball are updated by the simulator. Some of the sensory data (like some modes of the vision sensor) is sent to the agents with a different period than that of the simulation update.
Each agent controls only one virtual robot and must coordinate its efforts to make its best contribution for the teams' goals. It is important to note that the knowledge of the various agents about what is happening at a certain moment is not identical, due to noise and restrictions on several sensors (like the angle of vision or the cut-off hearing distance). Also the environment is very dynamic with the opposite team controlling their robots to oppose the teams' goals.
The coach agent is a special kind of agent that receives, from the simulator, the positions of all players in the field and of the ball without noise. However, the coach as severe limitations on its communication with field agents, that make it impossible to control the field robots using the coach information. The coach may have a very significant impact on team performance by giving advice to the field players and using it to perform high-level tasks like tactic analysis and selection or opponent modelling.
Visualization of the games is assured by an independent application that communicates to the simulator to receive the players and ball positions. Fig. 2 and Fig. 3 show two possible visualizations of the games in the 2D simulation league. All the 3D features of the viewer in Fig. 3 are not modelled in the simulator but inferred by the viewer for better attraction.

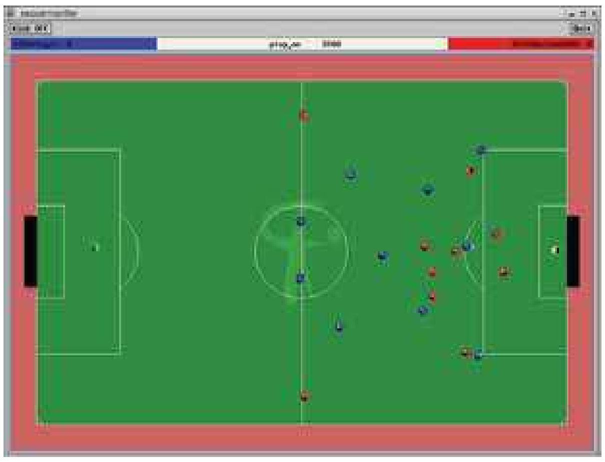


Fig. 2. 2D Simulation League Traditional Viewer

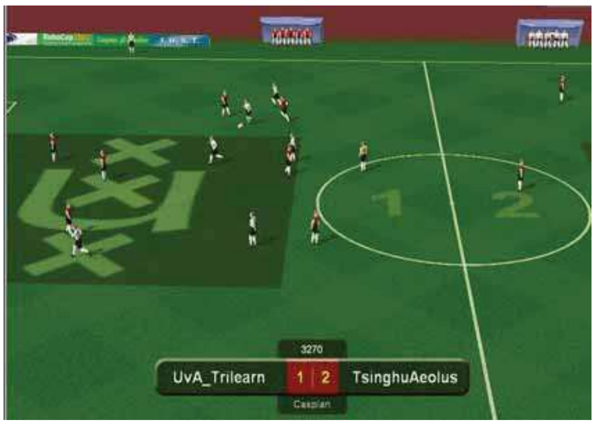


Fig. 3. 2D Simulation League Viewer with 3D displaying capabilities (Sedaghat M. & al. 2003)

2.3. Simulation 3D League
The first version of the 3D simulation league simulator was made available to the RoboCup community during January 2004. The proposal of the 3D simulator had the following objectives:

- Replace the 2D environment of previous simulator with a 3D environment;
- New, more realistic, physics model;
- Simulation results should not be dependent on available computational power or on the quality of network resources.

The differences between the new 3D simulator (Obst, O. & Rollman, M., 2005) and the 2D simulator (Chen, M. et al., 2007) used in previous RoboCup competitions, and in our previous research, are very significant.

Similarly to the 2D simulator, the simulation environment of the RoboCup 3D Simulation League is based on a client-server model. The simulator is the server and agents and visualization tools are the clients. The simulator creates the virtual environment (soccer field, markers, goals, etc.) where agents live, sends sensory information to the agents, receives their actions and applies the virtual physics model in order to resolve positions, collisions and interactions with the ball. Each team plays with 11 agents that must cooperate to score as much goals as possible while not allowing the other team to score.

The development of the 3D simulator used available open-source tools extensively. It uses the SPADES (Riley, P. 2003, Riley, P., 2003a) framework for the management of agent-world communication and synchronization, ODE (Smith, R., 2006) for the physical model, expat (Expat XML Parser, 2004) for XML processing, Ruby (Ruby 2004) for scripting language support and boost (Boost, 2004) for several utilities.

The 3D simulation server is implemented above a platform called SPADES (System for Parallel Agent Discrete Agent Simulation) (Riley, P., 2003). SPADES is a middleware system for agent-based distributed simulation. It aims to provide a generic platform to run in multi-computer systems. It implements the basic structure to allow the interaction between agents and a simulated world so that the users do not have to worry about communication and synchronization mechanisms such as sockets, addresses, etc.

SPADES' main features are:

- Agent based execution – support to implement sensations, thinking and actions.
- Distributed processing – support to run the agents applications on many computers.
- Results unaffected by network delays or load variations among the machines – SPADES ensure that the events are processed in the appropriate order.
- Agents can be programmed independently from the programming language – the agents can be programmed in any language once it provides methods to write/read to/from Pipes.
- Actions do not need to be synchronized in the domain – the actions of the agents can take effect at varying times during the simulation.

SPADES components are organized in a client-server architecture. The Simulation Engine and the Communication Server are provided by SPADES; while the Agents and the World Model are built by the user and run upon the formers.

The Simulation Engine is a generic piece of software that provides abstractions to create specific world models upon it. Agents may run in the same computer of the Simulation Engine on in remote computers linked to the network, in this case a Communication Server must be running in the remote computer. The World Model module must be running in the same computer of the Simulation Engine. This module specifies the characteristics of the environment where the agent will live.

SPADES implements what it calls the sense-think-act cycle in which each agent receives sensations and replies with actions. That means that an agent is only able to react after receiving a sensation message. The agent is also capable of requesting its own sensations, but the principle remains - a sensation must always precede an action. In order to allow actions between "normal" sensations, SPADES provides an action called *request time notify* that returns an empty sensation and after receiving it the agent is able to respond with actions. For example, if an agent received a sensation at cycle 100 and wants to produce an action at cycle 110, and if the next sensation will only arrive at cycle 120, the agent can ask to receive a time notify message at cycle 110 and just reply with the desired action after receiving it.

Fig. 4 depicts the sense-think-act cycle and the time where each of its components runs. From A to B a sensation is sent to the agent. After receiving the sensation (from B to C) the agent decides which actions will be executed; then from (C to D) the actions are sent do the server.

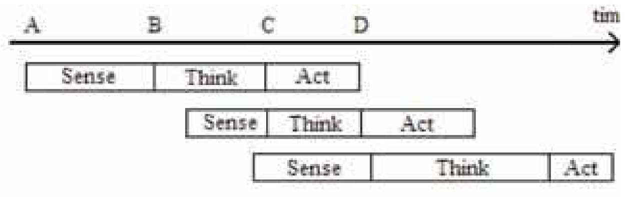


Fig. 4. SPADES Sense-Think-Act Cycle

In many agents, the sense, think and act components may be overlapped in time (like in Fig. 4). There is just one restriction: The thinking cycles for one agent cannot be overlapped. This constraint makes sense, since just a single processing unit is used per agent, and thus, just one sensation at time can be processed.

As stated before the simulator runs upon the SPADES, and uses ODE to calculate the physical interactions between the objects of the world. The graphical interface is implemented using OpenGL.

The 3D simulation server (Obst, O & Rollmann, M. 2005) allows twenty two agents (eleven from each team) to interact with the server in order to play a simulated robotic soccer game. Each agent receives sensations about the relative position of the other players and field goals and other information concerned with the game state and conditions. The information about the positioning of the objects in the world is given by the vision sensor that initially allowed the agent to receive visual information in 360 degrees but changed in 2006 to a 180 degrees angle of vision. The agents have the shape of a sphere (Fig. 5). Replying each sensation an agent sends actions like *drive* or *kick*. Driving implies applying a force on the body with a given direction and kicking implies applying a force on the ball radially to the agent. Each sensation is received on every 20 cycles of the server and each cycle takes 10 ms.

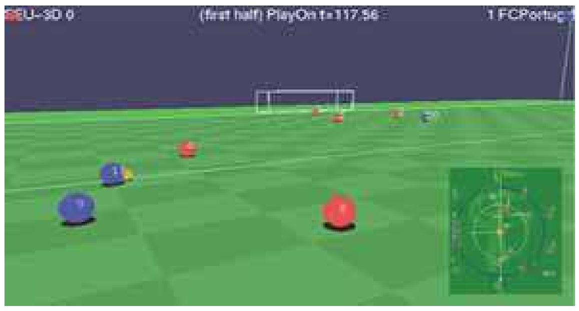


Fig. 5. 3D Simulation League Match – Sphere Robot Model

Each sensation takes 10 cycles to reach the agent (send delay) and actions sent by the agent take 10 cycles to reach the simulator. Hence a sent action starts to take effect at the time the next sensation is sent by the server as it is shown in Fig. 6.

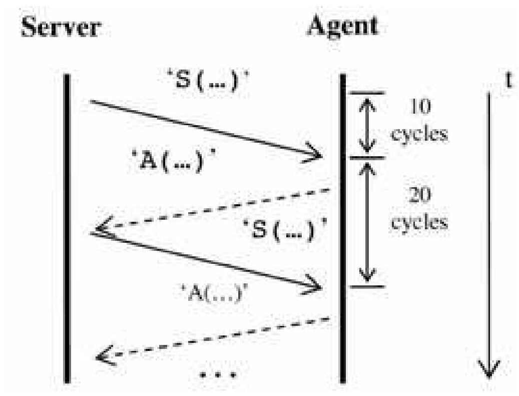


Fig. 6: Server-Agent Protocol in 3D simulator

In 2007 the robot model used in the 3D simulator changed from the sphere to a humanoid model. Also the SPADES support was abandoned and a new simpler type of timer was developed. The humanoid simulator only allowed games of two against two, but it is expected that in the future the number of robots for each team will increase. Figure 7 shows an image of the RoboCup 2007 3D Simulation Final.

Fig. 7. 3D Simulation League Match – Humanoid Robot Model

2.4. Microsoft Soccer Robotics Challenge

As part of its efforts to promote their recent Robotics Studio (Microsoft, 2007), Microsoft developed a simple soccer simulator and sponsored a new demonstration league at RoboCup 2007. Microsoft Robotics Studio uses the AGEIA PhysX physics-based 3D engine (AGEIA, 2007) to create simulated but realistic environments for robotics research while

making it easy to move the developed controllers to real robots. The AGEIA PhysX enables developing rich immersive physical gaming environments with features such as explosions; and characters with complex, jointed geometries.
Initially, it was supposed that teams would work with an early version of the simulator supporting only wheeled robots so that developers could have a chance to become familiar with the simulator and could work on team strategy and high-level coordination. However, Microsoft fastly developed a reasonably stable simulator capable of using legged robots. This was the simulator used in RoboCup 2007 competition. However, focus of the competition had to be moved from coordination to vision, low-level robot control and navigation, since the simulator was not stable enough to accommodate more than two players by team.
The new robot soccer simulation developed, included a 3D simulated soccer field and refereeing services, as well as support enabling different simulated robots to be used as soccer players. A simulated four-legged robot player, called RobuDog (Fig. 8), from Robosoft, was developed to be the base of the competition (Robosoft, 2007). Robosoft also previewed its hardware-based RobuDog robot, stating that the code developed for use in the simulated competition can be used directly on this new physical robot.

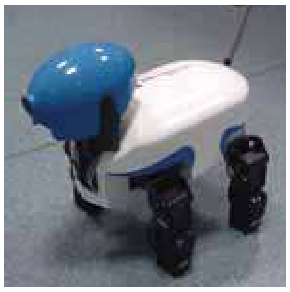


Fig. 8. The Robudog Real Robot (Robosoft, 2007)

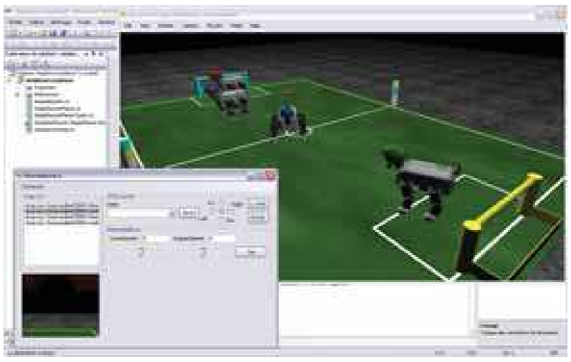


Fig. 9. The Robudog Simulated Robot and Micrsoft Robotics Soccer Challenge Field (Robosoft, 2007)

The new league was quite successful in RoboCup 2007 – Atlanta (Fig. 9). In the future it will become more clear that Microsoft is not limiting itself to just simulation. They intend to work closely with robot manufacturers to develop real soccer playing robots with software development based on the Microsoft Robotics Studio, making an impact in robotics over the next years. Microsoft gained support from a number of robot manufacturers including Kuka, Lego Systems, iRobot, SRI International, Yujin Robotics, Coroware, Parallax, Robosoft, etc. Thus, support for this new league seems well secured and new, more challenging rules are being developed for next year.

2.5. Physical Visualization League

The Physical Visualization (PV) Soccer League is a new RoboCup league where small real robots, called Eco-Bes, play soccer on top of a virtual field with a virtual ball, thus using the concept of augmented reality (Mackay, W. & Gold, R., 1993). Augmented Reality is an environment that includes both virtual reality and real-world elements (Milgram, P. & Kishino, F., 1994). Most augmented reality research uses a processed video which is augmented with virtual elements. The Augmented Reality at Physical Visualization League can improve the simulation, adding virtual elements that surround the real player. An example applied in that soccer league is a virtual leg with the ability to kick and to dribble the virtual ball and vision to perceive the world state (Azuma, 1997).

The Physical Visualization League offers a very interesting challenge for teams since several research challenges are included in this setup (Vision Based Self Localization, Data Fusion, Real-Time Control; Decision and Cooperation). The simplicity of this setup compared with the small-size league, makes it very interesting for educational and demonstration purposes (RoboCup, 2007a).

The Eco-Be is a very small vehicle remotely controlled by infrared commands, currently handmade. As presented in Fig. 10, it is composed by two step motors (1), a li-ion polymer battery (2), a control board (using an 8bit PIC18 family processor) (3), an Infrared Sensor (4) to receive its movement commands and an aluminium body (5). The robot can not send any kind of messages and its position is determined by an external camera (Yanagimachi, S. & Guerra, R., 2007). Each robot has a configurable ID, which can be freely changed. The robot can use each motor individually and each motor has three different speeds available. While using the fastest speed, all resources are drained from the controller, disabling the reception of new commands during this fast movement.

The kit furnished by Citizen contains a set of 20 robots with one charger, two AC/DC adapters and one infrared transmitter.

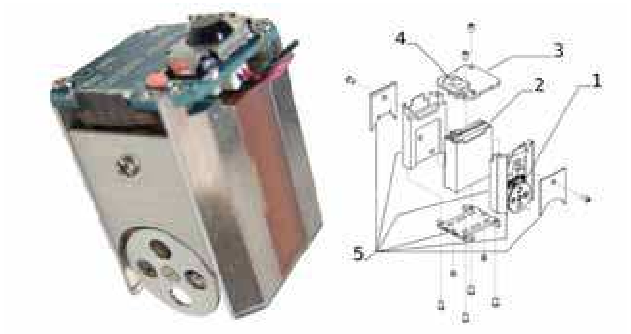


Fig. 10. The Eco-Be Robot (Yanagimachi, S. & Guerra, R. 2007)

The Eco-Be has a simple protocol of communication to control its movement. The robot's IR sensor recognizes a command through the modulation of the flash light as a square wave at 40 kHz with an on-to-off ratio of 50%. The period of Signal/Space allows the translation into valid logical states.

The LIRC software (Linux Infrared Remote Control) encodes the correct sequence of flash lights to compose a valid string to the robot (LIRC, 2007). LIRC is an open source software capable to decode and send infrared signals from and to the common serial port. It receives commands via socket and sends them to the infrared device driver. A specific communication protocol is followed in order to send commands to the robot. The transmitter uses a public circuit which receives a signal from the serial port and polarizes the infrared LED accordingly. The robot's Infrared Protocol accepts a string of 12 bits as a valid word. The first 5 bits identify the destination robot. The next three bits command the left motor and the following three command the right motor. The last bit is used for parity.

The Physical Visualization Sub-League started in 2007 focused on augmented reality, on its practical simple environment, and on demonstrating that the possibilities of the Eco-Bes are not limited to robot soccer. The league format had three competitions: demonstrations, technical developments and soccer tournament (RoboCup, 2007a). The main rules and league format details are shown as follow.

The final rules applied at RoboCup 2007 defined that each team plays with just two players in the soccer games. The number of players was fixed at only two players by the league committee due to technical limitations of the platform in this initial stage of development. Each soccer match takes ten minutes and each team has an optional break time of two minutes.

The PV League main part, the soccer tournament, is organized in two stages: the group stage and the elimination rounds. In the group stage, like in real soccer, each team is awarded 3 points winning, 1 for a drawing and 0 for a losing. In this group stage each team plays once with the other teams in the same group and the top two teams move forward to elimination rounds.

The elimination rounds have extra time of five minutes in case of a draw in the regular time. If no team scores a goal, teams alternately start a sequence of five penalty kicks. In this penalty kicks the attacker has to score a goal, the ball starts at middle field and the attacker a little behind. There is no limit to the number of times each player can touch the ball however the penalty kick is over if the ball exits trough the back line or after thirty seconds without a goal. If after five penalty kicks for each team the score is even, a new series of five penalties without goalies starts. Finally if the game is still tied a coin is flipped to decide the winner.

Rules of the tournament disallow communication between the teammates in such a way that it is solely up to the server to provide players' with sensory data.

The soccer tournament was organized with two groups of five teams playing the round-robin model. The best four go to the semi-finals, and, then, the winners pass to the finals.

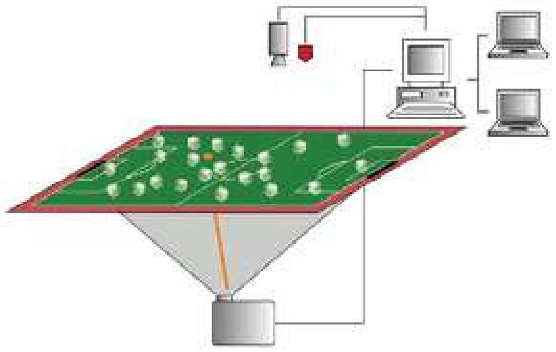


Fig. 11. PV League Schematic Setup (RoboCup, 2007a)

Prior to any game the camera is calibrated to recognize the field and the robots' markers. Each robot is identified by an individual marker that is recognized by the vision system. Each robot then receives an ID that corresponds to its marker and thus may be controlled by an autonomous agent running on a separate PC. Fig. 11 presents the field Schematic Setup.
A server application is responsible for controlling the socket connections with the clients, the monitor, the camera and the communication with the infrared USB transmitter. It has a file that defines the field size and the positions of each goal and each flag pole. For each instant, the server just has to identify the robots using their color markers. For the other elements, the server uses the positions in the configuration file. The server, then, compiles all information and, for each robot, it sends the relative positions (polar coordinates) of all elements in the field, including the ball and the other robots.
An associated monitor application uses the same information to draw the field and project it at the display where the robots play. The server sends the robot's absolute position to the monitor that projects them in the screen with a considerable precision. The monitor also shows the virtual ball.
Fig. 12 depicts a match with all the necessary elements. The field's background, lines, goals and poles are the passive elements drawn by the monitor. The marks below the robots and the ball is repeatedly updated, this way when the robots are over the screen monitor, the setup transmits the sensation the robots are really controlling the virtual ball with their physical movements.

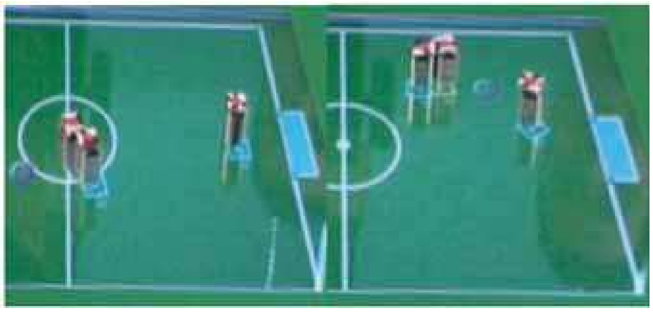


Fig. 12. Snapshot from a PV-League official match

Developing new low level skills for physical robots in virtual environments is a very challenging task as it was clear in RoboCup 2007, first edition of the PV-League. Methods for navigation, ball dribbling, passing, shooting and for goalie positioning were the basis for having a successful team.
In RoboCup 2007 it was also clear that: this new competition was clearly both viable and successful, the hardware platform can have a bright future, where multi-robot research can be directed to new and diverse scientific goals and finally that the league is very attractive to a wider audience.

2.6. Nanogram League

The RoboCup Nanogram competition challenges researchers to construct microscopic robots that compete against each other in soccer-related agility challenges. These robots measure a few tens of micrometers to a few hundred micrometers in their largest dimension and have masses ranging from a few nanograms to a few hundred nanograms.
The playing field consisted of a set of insulated interdigitated electrodes, across which an AC waveform can be applied (Donaldt, B.; Levey, C. ; McGray, C.; Rus, D. & Sinclair M. 2003), (Donald, B; Levey, C.; McGray, C.; Paprotny, I. & Rus, D., 2006). This waveform provides both the electrical power and the control instructions for the micro robots through a capacitive coupling. A digital camera placed on top of the field captures the action through a microscope sending it to the robot's control systems.

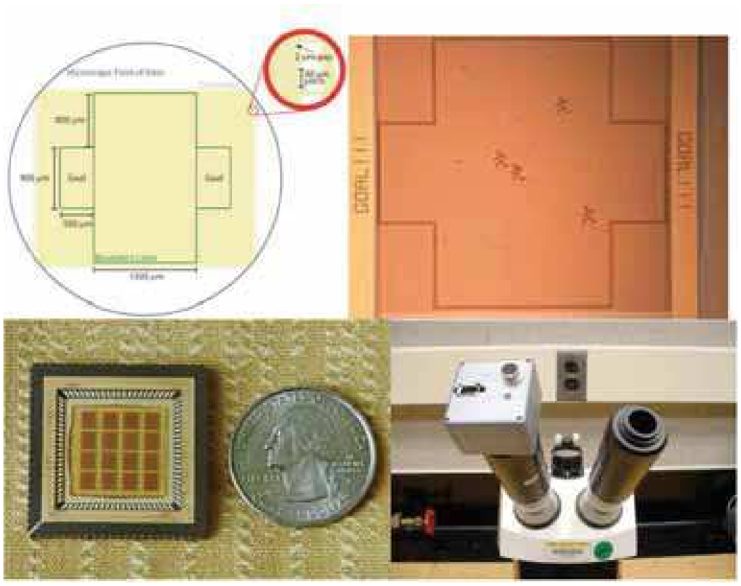


Fig. 13. Nanogram League Field (top), comparison of 16 fields with a coin (bottom-left) and the league Visualization Equipment (bottom-right) (McGray, C.; Arai, F.; Jacoff, A.; Tadokoro, S. & Gaitan, M., 2007)

For the first demonstration competition held in Atlanta in 2007, the contest consisted of three compulsory exercises: The 2 Millimeter Dash; the Slalom Drill; and the Ball-Handling Drill.

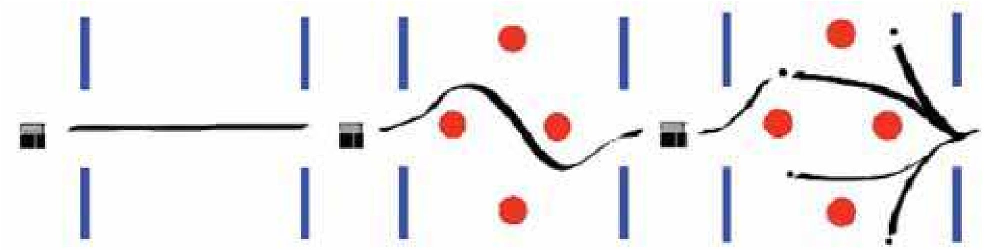


Fig. 14. The 2 Millimeter Dash; the Slalom Drill; and the Ball-Handling Drill (McGray, C.; Arai, F.; Jacoff, A.; Tadokoro, S. & Gaitan, M., 2007)

In the 2 Millimeter Dash, each microrobot must sprint across the playing field from one side to the other. The micro-robot begins with its entire structure behind one of the goal lines, and then it must dash until the first point of its structure crosses the opposing goal line. Each team will be allowed three trials being the winner determined by the best of the three.
In the Slalom Drill, the path between goals was blocked by simple objects (representing inanimate "defenders") that the micro-robot should avoid as it goes from one goal to the other. Each team was allowed three trials counting for scoring the best of the three.
The Ball Handling Drill was a more soccer-like challenge. The objective was for the robot to dribble as many balls as possible into the goal within three minutes, also avoiding the inanimate "defenders". The balls consisted of thin-film discs of silicon nitride, with dimples on their base to enable easy sliding along the field of play. Also, each team had three trials for this ball handling drill, with balls and defenders placed in different locations for each trial.
The league was also quite successful in its first appearance in RoboCup 2007. ETH Zurich (Swiss Federal Institute of Technology) won all the competitions, achieving the 2 Millimeter Dash in 316 milliseconds, the Slalom Drill in 583 milliseconds and scoring three goals in the Ball Handling Drill.

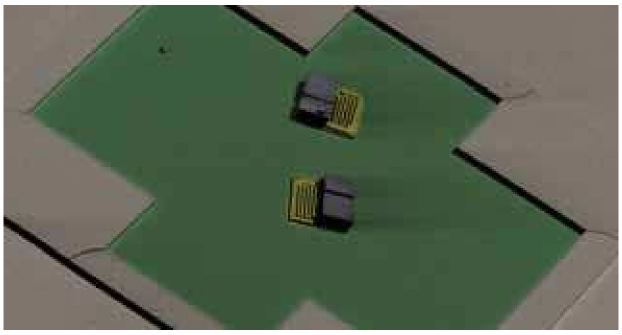


Fig. 15. Two IRIS ETH Zurich microrobots on the Nanogram playing field (McGray, C.; Arai, F.; Jacoff, A.; Tadokoro, S. & Gaitan, M., 2007)

In future years of competition, it is predictable that the competition should become more complex and the environment less structured, to encourage progress in robot cooperation, 3D manipulation, on-board power supplies, integrated logic and sensing, locomotion over rough terrain and robust operation in dirty, wet, and volatile environments (McGray, C.; Arai, F.; Jacoff, A.; Tadokoro, S. & Gaitan, M., 2007)

3. Agent Architecture and Knowledge Structures

To enable a team to perform cooperative multi-agent tasks, like playing simulated soccer, in a partially cooperative, partially adversarial environment a lot of knowledge is needed. Also, agents must have a world state representation as updated and as accurate as possible. Whenever the domain becomes more complex, knowledge importance is even greater. This is the case in multi-objective, partially cooperative and adversarial domains in which agents have limited perception and action capabilities. For this type of domains we argue that to correctly perform cooperative tasks, agents should include knowledge at three levels:

- Individual action execution;
- Individual decision-making;
- Cooperation.

Knowledge for executing actions is concerned with the specific commands needed to perform a given low-level skill. Individual decision-making knowledge is concerned with the way agents choose the action to execute. Knowledge for cooperation is concerned with tactics, situations, dynamic formations, roles, dynamic plans and communication protocols (Reis, L.P. & Lau, N., 2001). Representation structures for this type of multi-level knowledge are one of our research goals. The world state representation must remain updated so that it may be used to effectively decide the individual and cooperative actions to perform. The following methods are used to update our agent's world state information: visual perception analysis; communication; and action prediction.

4. Coordination Methodologies

In RoboCup past editions, teams with the best decision making mechanisms were very successful. In RoboCup 2001, the champions (Tsinghuaeolus (Yao, J.; Chen, J.; Cai, Y., Li, S. 2002) also champions in 2002) and vice-champions (Brainstormers (Riedmiller, M. & Merke, A., 2002)) did not use the coach agent and trusted mainly on their better low-level skills and well-tuned individual decision-making mechanisms. In RoboCup 2004 STEP won the competition mainly due to their fast dribbling ability. So, besides having a configurable strategy and flexible coordination mechanisms (Lau, N. & Reis, L.P. 2002), well-tuned individual decision making mechanisms are still very important in RoboCup simulation league (Teixeira, C.; Lau N. & Reis, L.P. 2004).

How to define roles based on standardized agent behavior characteristics for the RoboCup simulated soccer domain is one of the problems that has been tackled. To improve the flexibility of our team, agents are able to switch their relative positions (for a given formation) and roles (that define agent behavior at several levels), at run-time, on the field. We have proposed, and continually developed, Situation Based Strategic Positioning (SBSP) mechanism (Reis, L. P.; Lau, N. & Oliveira, E. C. 2001) that may be used to dynamically spatially position a team using different flexible formations for different situations. This mechanism is based on the distinction between active and strategic situations (Reis, L.P. & Lau, N. 2001). If an agent is not involved in an active situation then it tries to occupy its strategic positioning that change according to the situation of the game. Situation is a concept on a high-level analysis of the game (attacking or defending for example). SBSP was one of the main innovations of FC Portugal and is now used directly or as the base for the positioning systems of many simulated soccer teams and being used in some middle-size league teams.

4.1. Strategical Coordination

CMUnited brought the concepts of formation and positioning to RoboSoccer (Stone, P. & Veloso, M.) (1999; Stone, P., 2000) and used dynamic switching of formations as well. FC Portugal extended these concepts and introduced the concepts of tactics and player types. FC Portugal's team strategy is based on a set of tactics to be used in different game situations and a set of player types (Fig. 16).

Tactics include several formations used for different game specific situations (defense, attack, goalie free kick, scoring opportunity, etc). Formations are composed by eleven positionings that assign each player a given player type and a base strategic position on the field.

One of the most significant features is the clear distinction between strategic situations (when the agent believes that it is not going to use an active behavior soon) and active situations (ball recovery and ball possession). In strategic situations, players use a SBSP mechanism (presented in section 4.2). For active situations---ball possession, ball recovery or game stopped---decision mechanisms based on the integration of real soccer knowledge are used.

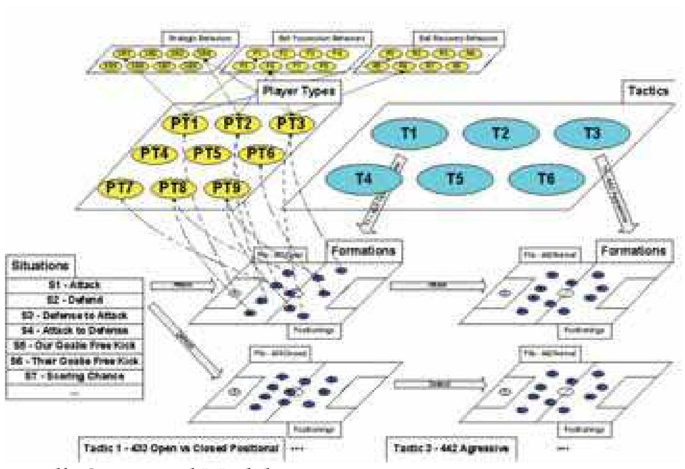
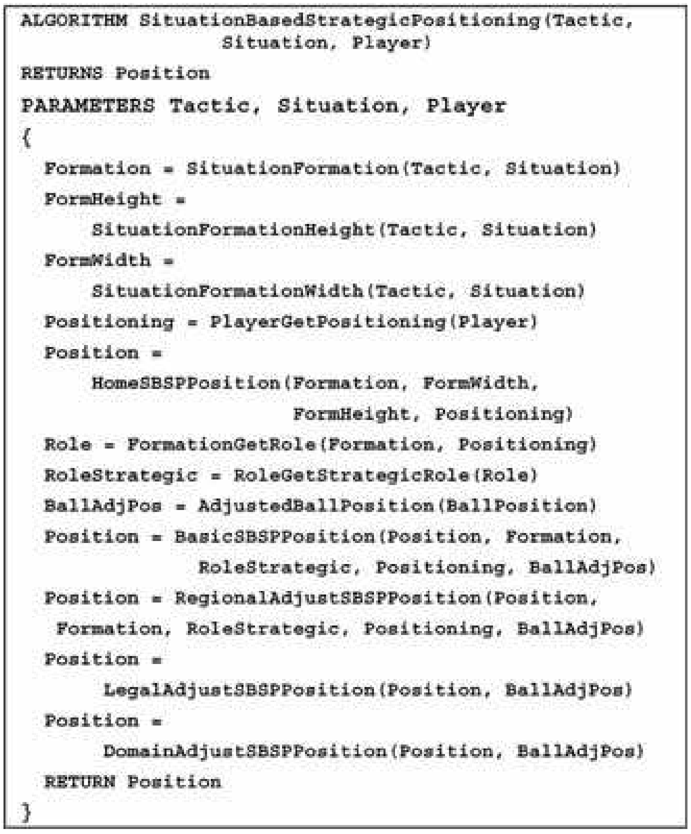


Fig. 16. FC Portugal's Strategical Model

4.2. Situation Based Strategic Positioning

Situation Based Strategic Positioning (SBSP) mechanism (Reis, L.P. & Lau, N. 2001; Reis, L. P.; Lau, N. & Oliveira, E.C. 2001) is used for strategic situations (in which the agent believes that it is not going to enter in active behavior soon). To calculate its strategic positioning, the agent analyzes which is the game situation. Then the agent calculates its base strategic position in the field in that formation, adjusting it according to the ball position and velocity, situation and player type strategic information. The result is the best strategic position in the field for each player in each situation. Since, at each time, only a few players are in active behavior (conducting the ball or trying to recover the ball) most players are close to their strategic positionings. SBSP enables the team to move similarly to a real soccer team, covering the ball while the team remains distributed along the field.



```
ALGORITHM SituationBasedStrategicPositioning(Tactic,
                 Situation, Player)
RETURNS Position
PARAMETERS Tactic, Situation, Player
{
  Formation = SituationFormation(Tactic, Situation)
  FormHeight =
      SituationFormationHeight(Tactic, Situation)
  FormWidth =
      SituationFormationWidth(Tactic, Situation)
  Positioning = PlayerGetPositioning(Player)
  Position =
      HomeSBSPPosition(Formation, FormWidth,
                       FormHeight, Positioning)
  Role = FormationGetRole(Formation, Positioning)
  RoleStrategic = RoleGetStrategicRole(Role)
  BallAdjPos = AdjustedBallPosition(BallPosition)
  Position = BasicSBSPPosition(Position, Formation,
               RoleStrategic, Positioning, BallAdjPos)
  Position = RegionalAdjustSBSPPosition(Position,
   Formation, RoleStrategic, Positioning, BallAdjPos)
  Position =
       LegalAdjustSBSPPosition(Position, BallAdjPos)
  Position =
       DomainAdjustSBSPPosition(Position, BallAdjPos)
  RETURN Position
}
```

Fig. 17. SBSP – Situation Based Strategic Positioning

4.3. Dynamic Positioning and Role Exchange

The Dynamic Positioning and Role Exchange (DPRE), and Dynamic Covering (Reis, L.P. & Lau, N. 2001), was based on previous work from Peter Stone (Stone, P. 2000) which suggested the use of flexible agent roles with protocols for switching among them. The concept was extended and players may exchange their positionings and player types in the current formation if the utility of that exchange is positive for the team. Positioning exchange utilities are calculated using the distances from the player's present positions to their strategic positions and the importance of their positionings in the formation on that situation.

```
ALGORITHM DynamicPositioningExchange(WorldState,
                              Situation, Positionings)
RETURNS
        Positionings(TeamSize)
PARAMETERS
        WorldState, Positionings[TeamSize], Situation
{
FOR PL1 = 2 TO TeamSize-1 DO
  FOR PL2 = PL1+1 TO TeamSize DO
   IF PositionValid(PL1) AND PositionValid(PL2) THEN
   {
    Dist11 = Distance(Position(Pl1),SBSPPosition(Pl1))
    Dist22 = Distance(Position(Pl2),SBSPPosition(Pl2))
    Dist12 = Distance(Position(Pl1),SBSPPosition(Pl2))
    Dist21 = Distance(Position(Pl2),SBSPPosition(Pl1))
    Adeq11 = PosAdequacy(Pl1, Positioning[Pl1])
    Adeq22 = PosAdequacy(Pl2, Positioning[Pl2])
    Adeq12 = PosAdequacy(Pl1, Positioning[Pl2])
    Adeq21 = PosAdequacy(Pl2, Positioning[Pl1])
    Util = ExchangePositions(DPREMode, Situation,
         Dist11, Dist22, Dist12, Dist21,
         Adeq11, Adeq22, Adeq12, Adeq21,
         PosImportance(Positioning[Pl1]),
         PosImportance(Positioning([Pl2])
    IF Util > ThresUtil(Situation) THEN {
         Aux = Positionings[Pl1]
         Positionings[Pl1] = Positionings[Pl2]
         Positionings[Pl2] = Aux
         }
    }
RETURN Positionings
}
```

Fig. 18. DPRE – Dynamic Positioning and Role Exchange

4.5. Intelligent Perception and Communication

The Communication model of the simulation league is restricted by the available bandwidth and uncertainty of message delivery. In 2002 the communication rules have changed: bandwidth has been constrained (messages with maximum length of 10 bytes), but uncertainty on delivery can now be reduced. Also, a new form of visual communication (enabling players to point to regions of the field) has been introduced. We identify four potential research areas in the definition of a communication protocol:

- What to communicate?
- When to communicate?
- Who should be heard at each time?

• How received messages will affect player's behavior?

Previous approaches have used message contents, mainly, to share world state knowledge between players (Stone, P. 2000), to communicate useful events/opportunities and to enhance cooperation (Reis, L.P. & Lau, N. 2001). With the reduction in message size, teams must carefully select which information should be conveyed. We have extended our ADVCOM principle: "Communicate only when you have something important so say" (Reis, L.P. & Lau, N. 2001) with "Communicate only what is important", measuring the importance of each piece of information through utility metrics based on the current situation and on estimated teammates knowledge. This idea was also been extended with a Situation Based Communication framework (Ferreira R.; Reis, L.P. & Lau, N. 2004). In RoboCup simulation league, if two or more players talk in the same simulation cycle, the simulator delivers only one of the messages to the other players. Teams have dealt with this restriction either by assuring that only one player talks in each cycle or by allowing several players to talk (Reis, L.P. & Lau, N. 2001). We believe that the first approach is not sufficiently flexible and reliable and thus have used the second approach (Reis, L.P. & Lau, N. 2001). In our protocol every player estimates the importance of his knowledge to the rest of the team through utility measures and only communicates when its communication utility his higher than the others or is above a threshold.

In the simulation league agent's visual perception is obtained through controllable sensors. Players may control their visual quality, the sensibility angle and the position of their neck relatively to the agent's body. We have developed intelligent perception mechanisms, namely, SLM - Strategic Looking Mechanism (Reis, L.P. & Lau, N. 2001). SLM decides the direction a player should look, in each cycle, maximizing the predicted world state update value from that perception.

5. Game Analysis and Coaching

In RoboCup simulation league, the online coach agent has a global vision of the field (without errors) gathered from the soccer server (Chen et al., 2007). The coach agent is able to analyze the game and send high-level commands to his team in order to improve the team global behavior.

In previous work (Reis, L.P. & Lau, N., 2002) we have explored different ways of implementing the coach agent. Different coaching architectures have been implemented and compared (Fig. 19), including the division of the coach agent into one assistant agent (capable of gathering game statistical information and opponent modeling information) and a principal coach (that uses the information provided by the assistant coach in order to decide the best tactic to be used by the team at each moment in the game). Other coaching architectures were also explored, including the subdivision of the coach functions by the players in order to have a completely distributed coaching behavior.

We have proposed Coach Unilang – a general language to coach a (robo)soccer team (Reis, L.P. & Lau, N., 2002). The development of translators from Coach Unilang to Clang has provided us with a very useful tool to test team behaviors and to participate with success in the Coach Competition.

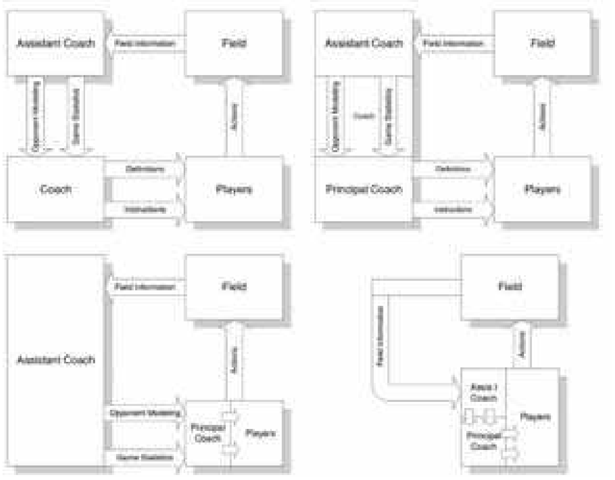


Fig. 19. Coaching Architectures of COACH UNILANG

6. Agent Analysis and Debugging Tools

Soccer Playing Agents have the following characteristics: autonomy, reactivity, pro-activity and social ability. Although agents take their decisions autonomously, the developer should be able to motivate the use of "good" actions and deprecate the use of "bad" actions. Furthermore the environment of the simulation leagues is uncertain, dynamic, real-time, heterogeneous, partially cooperative, and partially adversely. Agents take a large amount of decisions in very short time and understanding their decisions at the same time they are executing is a very complex task. Having these considerations in mind, we feel that offline analysis of agent's decisions, i.e. examining agent reasoning after execution, without real-time pressure, is the best way of making in-depth analysis of agent's reasoning.

Our debugging methodology is based on the following principles:

- Offline debugging;
- Visual debugging;
- Superimposed real environment and agent physical knowledge;
- Feature-focused debugging;
- Information structured in layers of abstraction with different detail levels.

In the context of the development of FC Portugal agents we have identified the following features that should be object of independent selection by the developer during a debugging session:

- Communication;
- Ball Possession;
- Ball Recovery;
- Synchronization;
- Low-level skills;
- Opponent Modeling.

The selection of each of these features, although independent, is not exclusive. The developer can activate one or more of the above listed features at each moment providing him with maximum flexibility in the selection of the information he finds relevant.
During the course of the development of FC Portugal agents, the following development tools have been implemented:

- Visual Debugger used to analyze the reasoning of agents (Reis, L.P. & Lau, N., 2001).
- Team Designer that enables the graphical definition of soccer strategies;
- Offline client methodology;
- WstateMetrics that evaluates the accuracy of world states;

Evaluation by domain experts using graphical tools is another methodology that has been used to fine tune our team.

6.1. Visual Debugger

The main debugging tool of FC Portugal is called Visual Debugger (Fig. 20). Its implementation is based on CMUnited99 layered disclosure tool (Stone, P., Riley, P. & Veloso, M., 2000) and the soccerserver logplayer application. CMUnited layered disclosure tool included the possibility of synchronous visualization of the game (using soccermonitor) and of one of the players reasoning (at several levels of abstraction) saved in action logfiles. We have integrated the two applications (logplayer with layered disclosure and soccermonitor) in a powerful team debugging tool and added the visual debugging capabilities and real and believed world-states superposition.

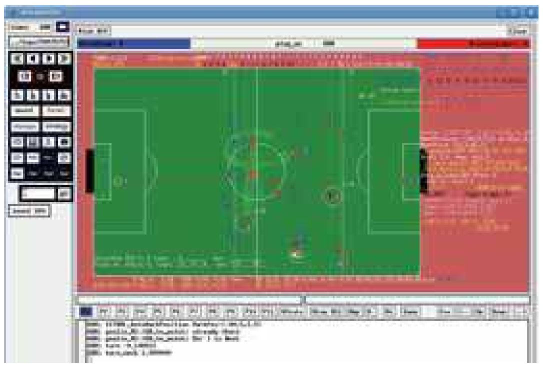


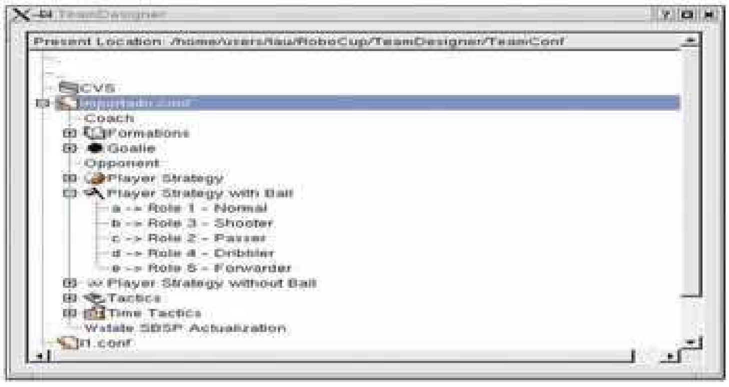
Fig. 20. The Visual Debugger window

Visual information is much more easily handled than text information. Soccermonitor (Chen et al., 2007) includes the possibility of drawing points, lines and circles over the field, but this functionality is not reported as being used by other teams. This soccermonitor feature was exploited, modified and extended. Agents can present their reasoning in graphical way using a simple API that includes functions like:

```
LogDrawLine(level, xy_i, xy_f, color);
LogDrawCircle(level, xy_center, rad, color);
```

This way the agent knowledge can easily be superimposed (and compared) with real data taken from the simulator logfile. This allows the developer to understand why the agent decided in a particular way and to focus developer attention on the most relevant features that need to be improved.
The developer can navigate over the pre-recorded game back and forth, examining the reasoning of each of the 11 players in the team. The level of detail of the information retrieved by the visual debugger may be controlled by the developer.

6.2. Team Designer
Team Designer application includes a tactics editor, a game statistical analysis tool, an offline coach and an online coach. The tactics editor allows the definition of the whole strategy to use during a given game: tactics, formations, individual player decision, strategic positioning features, etc. may all be changed in a friendly and safe way. Some of the tactic parameters are defined by direct manipulation of their graphic view. This is the case for the definition of players' home positions inside a formation and for the definition of new situations using the integrated offline coach.
The creation of a new strategy may use features from previously saved strategies, through the selection of which items are interesting to merge (Fig. 21).


Present Location /home/users/lau/RoboCup/TeamDesigner/TeamConf

```
CVS
[illegible]
  Coach
  Formations
  Goalie
  Opponent
  Player Strategy
  Player Strategy with Ball
    a -> Role 1 - Normal
    b -> Role 3 - Shooter
    c -> Role 2 - Passer
    d -> Role 4 - Dribbler
    e -> Role 5 - Forwarder
  Player Strategy without Ball
  Tactics
  Time Tactics
  Wstate SBSP Actualization
f1.conf
```

Fig. 21. Defining a tactic using Team Designer

The analysis tools gather game statistical information that is shown to the user and sent to the online coach. The statistics include ball position in several field matrixes, shoots and passes by field areas, ball loss and recovery positions, etc (Reis, L.P. & Lau, N., 2002).

6.3. Offline Client
In some situations the action log files saved by players are not sufficient to understand what really happened in a certain situation. A finer debugging degree can be achieved by employing our offline client tool.
The principle of the offline client is that we can repeat the execution of an agent over exactly the same setting of a previous game without the intervention of the server and without real-time constraints. Then we can use a normal debugger (like gdb) to examine the contents of all variables, set breakpoints, etc. at the execution situations we want to analyze.

A special log file, that records every interaction with the server and the occurrence of timer signals, must be generated to use the offline client. If an agent has probabilistic behavior some more information might be needed. The offline execution of an agent is achieved through a stub routine that reads the special log file. Player's behavior is maintained, as it is not affected by the substitution of this stub routine. The execution of a normal debugger over this offline client permits complete control of the execution flow of the agent reasoning.

7. Results and Discussion

The experiences performed to validate the approach used eight of the historical best known 2D soccer simulation teams. FC Portugal base team, using a single, well-tuned tactic, was tested against these teams using different coordination methodologies.

The first experiment was performed by performing 10 soccer 2D simulated games each of the 8 opponents selected, using five distinct positioning systems. A total of 10*8*5=400 games were performed for this experiment. The positioning systems used were:

- PACTS (Simple Active Positioning) – No strategic positioning. Agents always assume active positioning (ball possession or ball recovery active behavior).
- PACTF (Active Positioning with Static Formation) – Similar to PACTS but all agents posses a default static formation. If no active action is sufficiently good, agents return to their base positions.
- SPAR (Positioning by Attractions and Repulsions) – Positioning based on Stone et al. algorithm (Stone et al., 2000). Players are attracted by the ball, repelled by teammates and attracted by opponents if they have the ball or repelled otherwise.
- SP (Simple Strategic Positioning) – Using only one situation and one dynamic formation.
- SBSP (Situation Based Strategic Positioning) - Using situations for attack, defense, goal kicks, corners and throw-ins.

The analysis was made by automatically calculating, using the tool previously described, the goals scored and conceded in each game, the number of games won, draw or loosed by the team, the number of shoots made and conceded and global ball possession statistics in three field regions (attack, middle field and defense). The results achieved are shown in Table 1.

Analysing the results achieved it is easy to conclude that the best results are achieved using SBSP positioning. Another conclusion is that a good positioning system like SBSP enables the team to achieve better results against the best teams. The difference between the SP and SBSP positioning systems is clearly visible mainly against the best opponents (TSI and KB) while has low or almost no influence against the weakest opponents.

The second experiment was performed in order to evaluate the usefulness of the DPRE – Dynamic Positioning and Role Exchange mechanism. Ten games were performed against each of the eight selected opponents with and without using DPRE. The results achieved in the 10*8*2= 160 games conducted, are summarized in Table 2.

By analysing the results achieved, the main conclusion is that DPRE has greater impact against the best opponents. Although the number of shoots of the opponents does not increase significantly, the number of conceded goals increases significantly by not using DPRE. By analysing some of the real games in more detail it may be concluded that in some situations by not executing appropriate positioning exchanges in defense "holes" appear in the team defense letting the opponents score easily.

Opp. Teams / *Results*	*TSI*	*KB*	*UVA*	*YOW*	*FCM*	*FA*	*ATT*	*CMU*
PACTS (Simple Active Positioning)								
Goals (Scor.-Conc.)	**0-65**	**0-22**	**1-38**	**6-19**	**0-34**	**2-2**	**14-6**	**21-0**
Wins–Draws–Loses	**0-0-10**	**0-2-8**	**0-0-10**	**1-3-6**	**0-0-10**	**2-6-2**	**8-1-1**	**10-0-0**
Shoots	**1-86**	**2-36**	**2-72**	**13-37**	**2-61**	**5-16**	**26-14**	**37-16**
Attack–Middle-Def.	**8-58-34**	**9-61-30**	**14-44-42**	**16-54-30**	**8-47-45**	**14-58-28**	**38-46-16**	**42-46-12**
PACTF (Active Positioning with Static Formation)								
Goals (Scor.-Conc.)	**0-48**	**1-18**	**2-46**	**10-16**	**4-24**	**4-3**	**24-4**	**32-0**
Wins–Draws–Loses	**0-0-10**	**0-3-7**	**0-0-10**	**3-4-3**	**0-0-10**	**3-5-2**	**10-0-0**	**10-0-0**
Shoots	**2-75**	**3-32**	**5-89**	**21-39**	**6-48**	**6-20**	**39-12**	**54-8**
Attack–Middle-Def.	**12-56-32**	**10-58-32**	**18-47-35**	**22-50-28**	**14-55-31**	**24-50-26**	**44-42-14**	**50-43-7**
SPAR (Positioning by Attractions and Repulsions)								
Goals (Scor.-Conc.)	**0-36**	**2-12**	**1-42**	**8-14**	**5-25**	**6-2**	**33-3**	**45-0**
Wins–Draws–Loses	**0-0-10**	**1-4-5**	**0-0-10**	**2-4-4**	**0-0-10**	**4-5-1**	**10-0-0**	**10-0-0**
Shoots	**2-55**	**8-26**	**12-76**	**16-42**	**6-46**	**8-12**	**52-10**	**71-5**
Attack–Middle-Def.	**10-54-36**	**13-52-35**	**19-41-40**	**25-45-30**	**16-41-43**	**29-45-26**	**50-37-13**	**58-34-8**
SP (Simple Strategic Positioning)								
Goals (Scor.-Conc.)	**9-6**	**12-1**	**30-7**	**91-0**	**81-1**	**145-0**	**209-0**	**236-0**
Wins–Draws–Loses	**4-3-3**	**7-3-0**	**8-1-1**	**10-0-0**	**10-0-0**	**10-0-0**	**10-0-0**	**10-0-0**
Shoots	**23-12**	**22-6**	**42-18**	**132-3**	**108-10**	**196-1**	**264-0**	**256-0**
Attack–Middle-Def.	**31-51-18**	**48-33-19**	**38-37-25**	**60-30-10**	**53-34-13**	**68-29-3**	**74-25-1**	**77-22-1**
SBSP(Situation Based Strategic Positioning)								
Goals (Scor.-Conc.)	**10-3**	**19-0**	**32-6**	**98-0**	**76-1**	**158-0**	**204-0**	**246-0**
Wins–Draws–Loses	**6-2-2**	**7-3-0**	**9-1-0**	**10-0-0**	**10-0-0**	**10-0-0**	**10-0-0**	**10-0-0**
Shoots	**16-4**	**35-2**	**45-16**	**146-0**	**102-3**	**209-0**	**253-0**	**273-0**
Attack–Middle-Def.	**34-50-16**	**40-41-19**	**40-35-25**	**63-25-12**	**55-32-13**	**71-27-1**	**72-26-2**	**75-24-1**

Table 1. Results Achieved Using Different Positioning Systems

Opp. Teams / *Results*	*TSI*	*KB*	*UVA*	*YOW*	*FCM*	*FA*	*ATT*	*CMU*
Without using DPRE – Dynamic Positioning and Role Exchange								
Goals (Scor.-Conc.)	**8-10**	**10-3**	**24-12**	**76-0**	**65-6**	**126-0**	**187-0**	**225-0**
Wins–Draws–Loses	**2-4-4**	**6-3-1**	**6-2-2**	**10-0-0**	**9-0-1**	**10-0-0**	**10-0-0**	**10-0-0**
Shoots	**18-14**	**16-7**	**34-21**	**108-2**	**89-15**	**170-1**	**232-0**	**243-0**
Attack–Middle-Def.	**27-50-23**	**40-36-22**	**34-39-27**	**55-34-11**	**47-33-20**	**62-33-5**	**69-28-3**	**78-21-1**
Using DPRE – Dynamic Positioning and Role Exchange								
Goals (Scor.-Conc.)	**9-6**	**12-1**	**30-7**	**91-0**	**81-1**	**145-0**	**209-0**	**236-0**
Wins–Draws–Loses	**4-3-3**	**7-3-0**	**8-1-1**	**10-0-0**	**10-0-0**	**10-0-0**	**10-0-0**	**10-0-0**
Shoots	**23-12**	**22-6**	**42-18**	**132-3**	**108-10**	**196-1**	**264-0**	**256-0**
Attack–Middle-Def.	**31-51-18**	**48-33-19**	**38-37-25**	**60-30-10**	**53-34-13**	**68-29-3**	**74-25-1**	**77-22-1**

Table 2. Results Achieved with and without using DPRE

8. Conclusions

FC Portugal research on coordination methodologies enabled the definition a model for the strategy of team for a particular soccer game. This model is then implemented over SBSP and DPRE, enabling the team to perform in a very efficient way, as shown by our controlled experiments and competition results, and also in a real-soccer like manner. The development tools that enabled the tuning of the cooperative team behaviour and individual decision and individual skills mechanisms have been presented. These tools are based on general principles that can be of applied to the development of several types of agent in different domains.

FC Portugal achieved more than 15 awards in international RoboCup competitions. These awards included wining the World championships of the 2D simulation league (in 2000), Coach Competition (in 2002) and 3D simulation league (in 2006) and several European championships (including RoboCup Rescue). Its participation in the very recent World Championship of the Physical Visualization league resulted in the 2nd place (in 2007). The collaboration of FC Portugal members with CAMBADA and 5DPO Middle-Size teams resulted in a technology transfer from the simulation league to the real robots leagues, as, for example, the coordination model of the CAMBADA team (showed in the RoboCup 2007 free challenge) directly based in SBSP.

9. References

AGEIA (2007), AGEIA PhysX , http://www.ageia.com/physx/

Azuma, R. (1997), A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments* Vol. 6, No. 4 (August 1997), pp. 355 – 385.

Boost (2004), Boost C++ Libraries, http://www.boost.org

Chen, M;, Foroughi, E.; Heintz, F.; Huang, Z.; Kapetanakis, S.; Kostiadis, K.; Kummeneje, J.; Noda, I.; Obst, O.; Riley, P.; Steffens, T.; Wang, Y. & Yin, X. (2007), RoboCup Soccer Server Manual, http://downloads.sourceforge.net/sserver/manual.pdf (2007). The RoboCup Official Web Site, http://www.robocup.org/

Donaldt, B.; Levey, C. ; McGray, C.; Rus, D. & Sinclair M. (2003), Power Delivery and Locomotion of Untethered Micro-Actuators, *Journal of Microelectromechanical Systems*, Vol. 12, No. 6 (2003), pp. 947-959

Donald, B; Levey, C.; McGray, C.; Paprotny, I. & Rus, D. (2006), An Untethered, Electrostatic, Globally Controllable MEMS Micro-Robot, *Journal of Microelectromechanical Systems*, Vol. 15, No. 1 (2006), pp. 1-15

Ferreira, R.; Reis, L.P. & Lau, N. (2004), Situation Based Communication for Coordination of Agents, Proceedings of Scientific Meeting of the Portuguese Robotics Open 2004, Reis, L.P. (Ed.) pp.39-44, ISBN 972-752-066-9, April 2004, FEUP Edições, Porto, Portugal

Expat (2004), The Expat XML parser, http://expat.sourceforge.net

Kitano, H. et al. (1997), Robocup: The Robot World Cup Initiative, Proceedings of 1st International Conference on Autonomous Agent (Agents97), 1997, The ACM Press, Marina del Ray.

Lau, N. & Reis L.P. (2002), FC Portugal 2001 Team Description: Configurable Strategy and Flexible Teamwork, In: *RoboCup-2001: Robot Soccer World Cup V*, Birk, A.; Coradeshi, S. & Tadokoro, S. (Ed.), Springer Verlag, Berlim, LNAI 2377

LIRC (2007), Linux Infrared Remote Control, http://www.lirc.org/

Mackay, W. & Gold, R. (1993), Computer augmented environments: Back to the real world, *Commun. ACM*, Vol. 36, No.7 (July 1993), pp.24--26,

McGray, C.; Arai, F.; Jacoff, A.; Tadokoro, S. & Gaitan, M. (2007), RoboCupSoccer – Nanogram Competition – White Paper, on line, available at http://www.eeel.nist.gov/812/nanogram/white_paper.pdf

Microsoft (2007), Microsoft Robotics Studio, http://www.microsoft.com/robotics.

Milgram, P. & Kishino, F. (1994), A Taxonomy of Mixed Reality Visual Displays, *IEICE Transactions on Information Systems* ,Vol. E77-D, No. 12 (December 1994), pp. 1321-1329

Obst, O. & Rollmann, M. (2005), Spark - A generic simulator for physical multi-agent simulations. *Computer Systems: Science & Engineering*, Vol. 20, No. 5 (September 2005)

Reis, L. P.; Lau, N. & Oliveira, E. C. (2001), Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents, In: *Balancing Reactivity and Social Deliberation in Multiagent Sytems: From RoboCup to Real Word Applications*, Hannenbauer, M.; Wendler, J. & Pagello, E., (Ed.), pp. 175-197, Berlin Springer-Verlag LNAI 2103.

Reis, L.P. & Lau, N. (2001), FC Portugal Team Description: RoboCup 2000 Simulation League Champion, In: *RoboCup-2000: Robot Soccer World Cup IV*, Stone, P., Balch, T., & Kraetzschmar, G., (Ed.), pp. 29-40, Berlin Springer-Verlag, LNAI 2019.

Reis, L.P. & Lau, N. (2002), COACH UNILANG - A Standard Language for Coaching a (Robo)Soccer Team, In: *RoboCup-2001: Robot Soccer World Cup V*, Birk, A.; Coradeshi, S. & Tadokoro, S. (Ed.), pp. 183-192, Springer Verlag, Berlim, LNAI 2377

Riedmiller, M. & Merke, A. (2002), Karlsruhe Brainstormers - a reinforcement learning approach to robotic soccer II, In: *RoboCup-2001: Robot Soccer World Cup V*, Birk, A.; Coradeshi, S. & Tadokoro, S. (Ed.), Springer Verlag, Berlim, LNAI 2377

Riley, P. (2003), SPADES: System for Parallel Agent Discrete Event Simulation, *AI Magazine*, Vol. 24 No. 2, pp. 41–42.

Riley, P. (2003a), SPADES for Parallel Agent Discrete Event Simulation, http://spades-sim.sourceforge.net/

RoboCup (2007a). Physical Visualization Sub-League Official Web site. http://wiki.cc.gatech.edu/robocup/index.php/Physical_Visualization_Sub-League

RoboCup (2007a), Site of Osaka University's Physical Visualization League: http://er04.ams.eng.osaka-u.ac.jp/ecobe-robocup/

Robosoft (2007), RobuDOG simulator for the Robocup http://www.robosoft.com

Ruby (2004), Ruby Home Page http://www.ruby-lang.org

Sedaghat M. & al. (2003), Caspian 2003 Presentation Description,

Smith, R. (2006), Open Dynamics Engine v0.5 User Guide, http://opende.sourceforge.net/,

Stone, P. & Veloso, M. (1999), Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork, *Artificial Intelligence*, Vol. 110, No. 2, (June 1999), pp. 241–273.

Stone, P.; Riley, P. & Veloso, M. (1999), CMUnited-99 source code, 1999, Date Accessed: July 2007, Accessible from http://www.cs.cmu.edu/~pstone/RoboCup/CMUnited99-sim.html.

Stone, P. (2000), *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*, MIT Press, ISBN: 0262194384.

Stone, P.; Riley, P. & Veloso, M. (2000), Layered Disclosure: Why is the agent doing what it's doing?, Proceedings of Fourth Int. Conf. on Autonomous Agents (Agents 2000), 2000, Barcelona.

Teixeira, C.; Lau N. & Reis, L.P. (2004), FC Portugal 2003 Shoot Evaluation Based on Goalie Movement Prediction, Proceedings of Scientific Meeting of the Portuguese Robotics Open 2004, Reis, L.P. (Ed.) pp.149-155, ISBN 972-752-066-9, April 2004, FEUP Edições, Porto, Portugal

Yanagimachi, S. & Guerra, R. (2007), Citizen Micro Robot´s Reference Manual. Osaka University. http://er04.ams.eng.osaka-u.ac.jp/ecobe-robocup/files/robot-manual.pdf

Yao, J.; Chen, J.; Cai, Y., Li, S. (2002), Architecture of TsinghuAeolus, In: *RoboCup-2001: Robot Soccer World Cup V*, Birk, A.; Coradeshi, S. & Tadokoro, S. (Ed.), pp. 491-494, Springer Verlag, Berlim, LNAI 2377