Hierarchical Control in Robot Soccer using Robotic Multi-Agents

José A. Fernández León and Héctor N. Acosta

INTIA Research Institute, Exact Sciences Faculty
Universidad Nacional del Centro de la Provincia de Buenos Aires
Campus Universitario – (7000) Tandil, Bs. As., Argentina
{jleon, nacosta}@exa.unicen.edu.ar

Abstract. RobotCup is an international competition designed to promote Artificial Intelligence (AI) and intelligent robotic research through a standard problem: a soccer game where a wide range of technologies can be integrated [12]. This article shows, in a general way, an architecture proposed for controlling a robot soccer team. The team has been designed with agent concept for robot control in Middle League Simurosot category (FIRA). A brief description of control's architecture is presented. In addition, this paper shows a simple robotic-agent control without an explicit communication of actions to agents.

Keywords: Autonomous Robots, Robot Soccer, Robotic Multi-Agent, Robotic Control

1. Introduction

RobotCup is an international competition designed to promote Artificial Intelligence (AI) and intelligent robotic research through a standard problem: a soccer game where a wide range of technologies can be integrated [12].

In the last decade, agents[1] for RobotCup Software Challenge have gained growing interest among AI researchers [13], [14]. Designing a robot to play football is very challenging because the robot must incorporate the design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, strategic decision making, intelligent robot control, and machine learning [11]. Software agents that control players for RobotCup soccer are called soccer agents and are implemented as clients of soccer servers [15].

This paper aims to define a level of agents to control robot's actions in a varying environment. A level of agents in a hierarchical architecture is based on abstraction of team behavior and enables agents to interact with other agents without explicit communication to find an emergent co-operative behavior [8].

This reports presents a brief architecture description to control a robot soccer team called INCASoT. In addition, this paper shows a simple robotic-agent control without an explicit communication of actions. Each robotic agent in INCASoT is defined and coded according to an individual approach without having an explicit team's concept. In this way, it is sought that the behavior of the team arises from the set of singular actions set and these allow the team to improve its performance over time.

[1] It refers to software agents or robotic agents.

In general, a bottom-up design with strong dedication to experimentation is proposed. The project begins with the outline of the basic functions of agents. Starting from these basic functions are carried out the construction of a model that defines agent's behaviors inside the game field. Then, different control levels are specified referred to the coordination among several robotic agents, to conclude in team strategies or, in other words, the declaration of rules that facilitate the identification of attack and defense opportunities.

Here, a behavior-based approach applied in INCASoT team is emphasized. Specifically, a brief description of team control architecture is presented in section 2. Section 3 shows a description of the simulator. In Section 4, the agent's behavior is described. Interactions among different levels of the proposed architecture are analyzed in section 5. The implementation of INCASoT is described in section 6. Section 7 presents the results obtained in CAFR 2003[2]. Finally, a brief summary is exposed in section 8.

2. The INCASoT Architecture

One of main objectives for INCASoT's development is to build a control and test platform for robotic-agent movements. At present, a hierarchical outline of team control has been adopted in the team (Fig. 1).

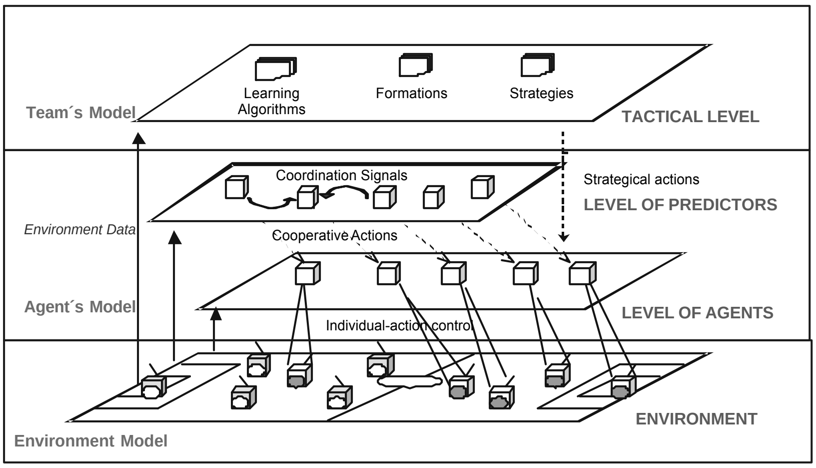

Fig. 1. Hierarchical architecture of robot's control used in INCASoT Team

This outline specifies a modulate architecture divided in levels of functional abstraction. More concretely:

- Environment - It corresponds to the ambient in which robots are acting. This level specifies information coming from game server, in the case of a simulator, when it is executed in virtual form. For example, it provides data about the ball position, the current state of game, orientation and speed of robots.

[2] Campeonato Argentino de Fútbol de Robots. Universidad de Buenos Aires, 2003.

- Level of Agents - This level is responsible for sending control actions to robots allowing them an appropriate acting inside the game environment. Signs or actions toward each robot are related to regular intervals of time. Examples of control signs can be: the speed and the rotation angle so that a robot avoids an obstacle or guide its movement toward the ball. Here, each agent has a state machine of its own [1], [4], which allows individual changes of behavior according to game conditions.
- Level of Predictors - It is the level in charge of detecting collaborative situations with the rest of robots. Therefore, it tries to identify plays in benefit of current tactics. For example, it would identify which ball trajectory would be in N times (or cycles of simulator) to advance towards possible goal situations with or without collaboration from other team partners. This level is composed of different modules, one for each agent, denominated predictors [8], [9].
- Tactical Level - It is made up of several modules related to team's knowledge (e.g. formations), as of attack and defense strategies. It also includes a module looking at the game state, with the purpose of varying strategies and learning from them in real time [5], [6]. Such level has an important role in team acts. In other words, it selects the action course to handle through strategic actions influencing the performances of the agents.

Each level has one or several well-defined functionality with modularity conception. When the predictors level, or the tactical level, cannot contribute with any action on the level of agents, it is not affected since it continues with its normal functionality. Indeed, both tactical and predictors level provide control actions at the level of agents when certain conditions happen in the environment.

3. The Simulated Environment

The simulator provides the environment that supports the competitions among virtual soccer teams, considering an ambient with multiple agents, changing and with answers in real time. Generally, these tools also provide a graphic interface that allows to visualize the development of the game on a screen [17] (Fig. 2).

The soccer in a simulated environment presents an important advantage with respect to soccer implemented by physical robots: allowing researches in learning and cooperation techniques, disregarding the particular inconveniences that can arise from the robots and the physical environment [16]. In this context, it is not necessary to consider problem solving in the robot hardware, or in the communication mechanisms or object recognition.

Fig. 2. FIRA Simulator - RobotSoccer1.4

The simulator presented by FIRA [17] provides a virtual ambient to take a soccer game ahead among non-physical robots. The simulated agents possess the physical characteristics of the robots that participate in the category Middle League MiroSot. Independently of this, each team develops its own strategies.

4. Robotic Agents Behaviors

Due to the fact that agents present individual game behaviors in INCASoT, the architecture proposed does not use explicit aspects of communication among them. Agents only react to what they can perceive from their environment, while there is a superior level in charge for coordinating the set of singular actions to search a particular team behavior [1].

The level of agents is in charge of carrying out a group of tasks linked to the agents behavior, which, in turn, can be divided into under- and high-level behaviors. [3]. Low-level behaviors are referred to actions that an agent communicating to the robot about what it has to achieve in the ambient. For example, robot simple movements can be advancing toward a certain position without considering obstacles.

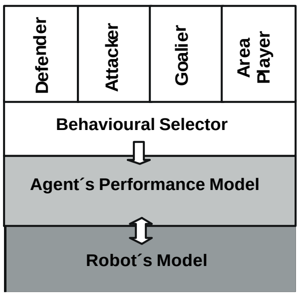

Fig. 3. Agent’s Architecture in Level of Agents

On the other hand, for example, high-level behaviors are referred to select where to move from and which is the best partner to carry out a pass [7], [10]. This division among behaviors of high and low level is found in the module called agent’s performance model inside agent architecture (Fig. 3).

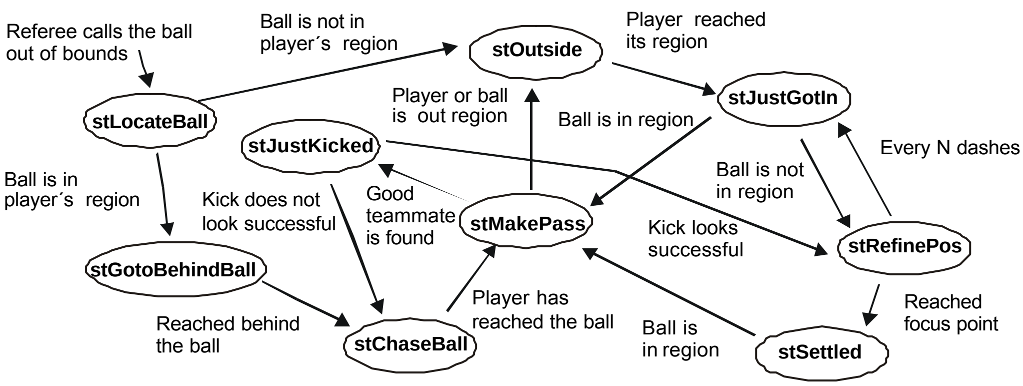

Fig. 4. Finite State Machine used to agent’s control

The conception of the agent's performance model resides in a finite cycle of execution through game states. In such a cycle, different situations are perceived, for example proximity to goal area or to the ball, and it acts according to the derived information of such perceptions [1]. This cycle defines each player behavior as being strongly associated to the concept of regions. In the Fig. 4, the arrows represent transitions between states and the bound text to each arrow specifies the passage condition among them [4].

Therefore, in the agent's architecture can be identify:

- Agent's performance model: it implies a finite state machine divided into two behavior levels, a low-level one, that consists of basic actions inside the environment (for example, kicking and advancing), and another one, with high-level behavior, based on the previous level, to carry out actions of a higher complexity degree (for example, avoiding obstacles and passing the ball) [3].
- Robot's Model: it specifies the characteristics of the robot that an agent should control. In simulated robots, this only refers to characteristic robot variables (for example, quantity of wheels, maximum allowed speed and rotation angle).
- Behavioral Selector: it corresponds to a module separated from the agent performance model. This refers to functional questions linked to the fact that an agent has to present an independent game for any particular role in INCASoT. An agent is associated with a role (for example, attack) when certain ambient conditions require it. In this way, it is sought that each agent has identical functionality than other players but can be specialized whenever faced to punctual game situations. On the other hand, if this selector is part of agent performance model, the agent should identify concrete situations any time. On the contrary, in the hierarchical outline, an independent agent will be informed about roles it has to modify.

Roles inside a multi-agent system assign responsibilities and tasks to members of the same team [8]. An agent with a certain role will have less behavioral options than those with decision-making responsibilities. The ideal thing would be to only have some few possible behaviors for each situation, in such a way that the agent could easily decide what to do.

In synthesis, the roles reduce complexity in multi-agent systems. The reduction of complexity implies that it is necessary to take smaller quantity of decisions. The teams with rules are simpler than those that don't have rules: somehow they are also less flexible, but in a multi-agent systems, an appropriate combination of rules can lead to a very effective, fast and robust system [17].

5. Team Agent Interaction

Each agent is associated with a particular region in INCASoT, which determines the portion of the field covered by each one, and with a focal point in such region [1], [4]. According to that, there are actions that may adjust the team actions in relation with the region. There are actions coming from the tactical level (strategic actions) that adjust, for example, the team formation according to its current situation inside the game [2].

When varying team formation, either by attack or defense formations, each agent will be adapted to a particular situation inside its game area. In this way, there exists an adjustment at team level and at individual level (or agent) to the game by the covering area concept.

It is important to notice that each agent presents autonomy of decision inside the game field but it is influenced by the team game and situations presented inside this. Each agent tries to reach situations related with its role, as well as to cooperate when team objectives are specified (seen as an emergent behavior).

An example of the team control is presented. Suppose that the current team formation is as show in the Fig. 5, which indicate that the team is under conditions of attack (e.g. due it possesses the ball control into the contrary area). It is necessary to clarify that it is considered as ball control, for example, that at least a player is near to the ball and its actions over the ball are visible in the environment (e.g.: make a pass).

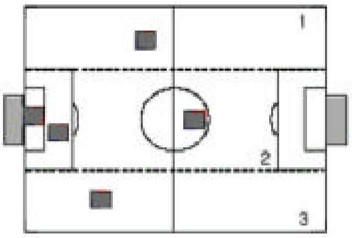

Figure 5 – Strongly-offensive formation

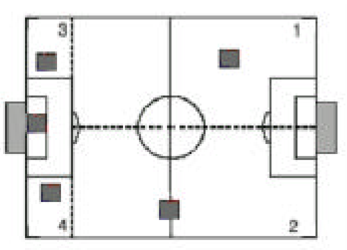

Figure 6 – Strongly-defensive formation

It is possible that in an attack the team loses the control of the ball for a considerable time interval. On other words, the player's actions on the ball are not visible as it was expected in the environment. Therefore, it should be considered by the agent after the robot tries to correct lost of control on the ball.

The tactical level (Fig. 1) monitory actions that happen in the environment, it can decide modify the team´ strategy when specific conditions in the environment appear. It is only carried out varying the team's in INCASoT, for example taking a strongly-defensive formation to affront the opponent attack (Fig. 6).

The variation proposed by the tactical level, can be based in a set of rules defined by an expert in the problem domain (robot soccer in this case). Therefore, the team behavior takes advantage of the knowledge of formations, situations and well-known rules due it makes possible a certain action. An example of a simple rule is: if the ball is inside the contrary area, and exist domain on the ball, then the action will be attack (attack formation).

An important point to remark is that the formation variation should not affect to acting of a player if it presents a special situation during the game. For example, suppose that the tactical level considers to pass from an attack formation to one of defense, when an robot is in front of the ball but without domain on it. This player instead of taking the assigned position for the new formation, it should consider the current position to looking for the ball domain with a defensive role. On this way, although agents consider team's actions, it can relax them according to own situations during the game. A more complex model of formation selection can be carried out by means of statistics taken during the game. For example, considering time of ball retention for the team, score during the game, quantity of times that the team maintains the ball in the contrary area, and quantity of attacks near the own area, etc.

It is important to notice that a rule or a group of them can interfere with others. It is necessary define priorities among rules. For example, if the score is in favor of our team, a possible team's strategy can be take a strongly-defensive formation without considering the side that the ball are. This way, the rules associated with attacks according to the area where the ball is, will not be considered due the few importance of it. When each agent's situation allows it, the Behavioral Selector inside the Agent's Architecture will select the role that the player should carry out according to the characteristics of the proposed formation. This selector considers when to relax an action for another during the game is necessary.

6. Implementation of the Architecture

In the previous sections, different characteristics of the design of a soccer team have been mentioned. For the implemented team it has been opted to work with roles, with varying formations and without explicit communication among the players. The implementation of the team is divided into the following modules:

- The Agent Module is the one in charge to control the robot movements. It is composed from a module with a finite state machine described above (agent player), the roles (agent profiler) and one with routines to control robot (agent control).
- The Team Module is composed from tree modules to control the team (team control), to describe the team's knowledge and other to coordinate several agents (team strategy), and a module with known formations (team formation).
- The Environment Module has a module related with ambient variables (ambient) and one that provides communication with the simulator (main).

The interchange of information among modules is as follows: the agent module provides control's robot methods to the team module, and the last one makes team actions into environment (e.g. strategies).

7. Results

Experiments about the team performance have been made (e.g.: time of ball retention and scoring). The results show that the proposed architecture is valid for robot-soccer team control. However, a major number of tests was carried out in the CAFR 2003 Competition.

The competition involved eight teams from different places [18]. It consisted of a preliminary round of three games, followed by the four advancing teams playing a round playoff. INCASoT presented an acceptable game related with the dynamic of hierarchical control, but there were a number of technical problems during the preliminary and the semifinal rounds. It includes imprecision in primary functions (as intercept the ball).

One of preliminary games was very close against to a very good opponent. INCASoT was tested with the team that will obtain the second position in CAFR 2003. Several problems with forward players precision there was.

On the other hand, one of semifinal game was very problematic. Problems with the simulator caused a bad team performance in the first half of game. In second half, INCASoT presented an acceptable team performance, but problems with the forward players continued. The match was carried out with the team that obtained the first position in the championship. In addition, a high one numbers of infractions happened in these game (mainly penalties in favor of the team opponent). It was caused to have several defenders in the penalty area when an opponent attack occurred.

In these games the opponents demonstrated the dynamic positioning and to very good precision of attackers and goalies. In spite of the problems before enunciated, INCASoT obtained the fourth position in the CAFR 2003.

8. Conclusions

To satisfy the requirement of robot individual control based on agents, and having as further purpose the team general behavior control, hierarchical control architecture has been adopted in INCASoT. For this purpose, concrete modules have been specified inside the architecture, according to the behavior required by agents during the game.

With the objective of analyzing the actuation of the developed team, a series of tests have been carried out. Those tests have been designed to handle several teams with different strategies, according to the rules of the category. Thanks to those experiments, several refinements have been identified for improving a better agent performance.

The results in CAFR 2003 indicate that it is important to obtain a good team performance that has basic functions with a high precision degree (e.g.: to approach to the ball, shooting to the goal, dribbling, and avoidance obstacles). In addition, a strict control to avoid infractions is necessary to obtain a robust Robot Soccer team.

The current focus of INCASot's development is an effective way of making decisions inside the team model, dealing with questions related to learning at the level of agents set and a concrete design of the predictors level inside the agent pattern.

References

1. Acosta, N.; Fernández León, J. A.; Vázquez, M. O. Desarrollo de un Equipo de Fútbol de Robots. 5to WICC 2003. Workshop de Investigadores en Ciencias de la Computación. Tandil, Buenos Aires, Argentina. 2003.
2. Han, Kuk-Hyun; Lee, Kang-Hee, Moon, Con-Kyoung; Lee, Hoon-Bong; Kim, Joung-Hwan. Robot Soccer System of SOTY 5 for Middle League MiroSot. Korea Advanced Institute of Science and Technology (KAIST). Korea. FIRA Robot Congress. 2002.
3. Hu, H.; Kostiadis, K.; Hunter, M., Kalyviotis, N. Essex Wizards 2000 Team Description. University of Essex. Wivenhoe Park, Colchester CO4 3SQ, United Kingdom. 2000.
4. Lerner, Sorin; Poupart, Pascal. RobotCup Software Soccer Team. 304-527B Artificial Intelligence. McGill University, Canada. April 27th, 1998.
5. Matsubara, Hitoshi; Noda, Itsuki; Hiraki, Kazuo. Learning of Cooperative actions in multi-agent systems: a case study of pass play in Soccer. Electrotechnical Laboratory. AAAI-96 Spring Symposium on Adaptation, Coevolution and Learning in Multiagent Systems. 1995.
6. Matsui, Tohgoroh; Inuzuka, Nobuhiro; Seki, Hirohisa. A proposal for Inductive Learning Agent Using First-Order Logic. Nagoya Institute of Technology. Japan. 1999.

7. Veloso, Manuela; Bowling, Michel; Achim, Sorin; Han, Kwun; Stone, Peter. The CMUnited-98 Champion Small-Robot Team. Carnegie Mellon University. Pittsburg, PA 15213. 1998

8. Veloso, Manuela; Stone, Peter; Han, Kwun. Prediction, Behaviors, and The CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control. Carnegie Mellon University. Pittsburg, PA 15213. Autonomous Agent'98. October 1997.

9. Veloso, Manuela; Stone, Peter; Han, Kwun; Achim, Sorin. Prediction, Behaviors, and Collaboration in a Team of Robotic Soccer Agents. Carnegie Mellon University. Pittsburg, PA 15213. ICMAS'98. November 1997.

10. Veloso, Manuela; Stone, Peter; Riley, Patrick. CMUnited-98 champion simulator team. RobotCup98: Robot Soccer World Cup II. Springer Verlag, Berlin, 1999.

11. Duman, H.; Hu, H. Fuzzy Logic for Behaviour Co-ordination and Multi-Agent formation in RobotCup. University of Essex. 2000.

12. Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara, H.; Noda, I.; Asada, M. The RobotCup Synthetic Agent Challenge. International Joint Conference on Artificial Intelligence. 1997.

13. T. R. Federation. RobotCup official site. Since 1998. http://www.robotcup.org/

14. Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, E.; Osawa, H.; Matsubara, Asada, M. The RobotCup Synthetic Agent Challenge 97. Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Volume 1, pages 24-29. 1997.

15. Noda, I.; Matsubara, H.; Haraki, K.; Frank, I. Soccer Server. A tool for research on Multi-agent systems. Applied Artificial Intelligence, 12:233-250. 1998.

16. Santos, J. M.; Castelo, C. C.; Fassi, H. R.; Scarpettini, F. E. Fútbol de Robots: Revisión del Estado del Arte y Desarrollo del Equipo UBASot de Simulación. Tesis de Licenciatura. UBA. 2002.

17. Jo, J.; Smith, R.; Rodwell, A.; Clarke, M.; Truesdell, S. Robot Soccer1.4: 3D Robot Soccer Simulator. Robot Soccer Inc. http://www.fira.com.

18. Departamento de Computación, Fac. Cs. Exactas y Naturales. Workshop del Campeonato Argentino de Fútbol de Robots CAFR 2003. Actas de Trabajos. July 2003.