

Document downloaded from:

<http://hdl.handle.net/10251/64845>

This paper must be cited as:

Guarnizo Marin, JG.; Mellado Arteché, M.; Cheng, YL.; Blanes Noguera, F. (2015).  
Architecting centralized coordination of soccer robots based on principle solution. *Advanced Robotics*. 29(15):989-1004. doi:10.1080/01691864.2015.1017534.



The final publication is available at

<http://dx.doi.org/10.1080/01691864.2015.1017534>

Copyright Taylor and Francis

Additional Information

This is an Accepted Manuscript of an article published by Taylor & Francis in *Advanced Robotics* on 2015, available  
online: <http://www.tandfonline.com/10.1080/01691864.2015.1017534>

## **Architecting Centralized Coordination of Soccer Robots based on Principle Solution**

Jose G. Guarnizo<sup>a\*</sup>, Martin Mellado<sup>b</sup>, Cheng Y. Low<sup>c</sup>, Francisco Blanes<sup>b</sup>.

*<sup>a</sup>Laboratory of Alternative Sources of Energy LIFAE. University Francisco Jose de Caldas. Bogotá. Colombia. <sup>b</sup>Instituto de Automática e Informática Industrial, Universitat Politècnica de València. Valencia. Spain. <sup>c</sup>Humanoid Robot and Bio-Sensing Center, Faculty of Mechanical Engineering, Universiti Teknologi MARA, Malaysia.*

\* Corresponding author: [jguarnizo@udistrital.edu.co](mailto:jguarnizo@udistrital.edu.co) (J. G. Guarnizo), Laboratory of Alternative Sources of Energy LIFAE. University Francisco Jose de Caldas. Carrera 7 No 40-53, Bogotá. Colombia.

[martin@ai2.upv.es](mailto:martin@ai2.upv.es) (M. Mellado), Instituto de Automática e Informática Industrial, Universitat Politècnica de València. Camino de vera, s/n. Ciudad Politécnica de la Innovación. Edificio 8G – Acceso D Planta 3. CP 46022. Valencia. Spain.

[chengyee.low@salam.uitm.edu.my](mailto:chengyee.low@salam.uitm.edu.my) (C. Y. Low), Faculty of Mechanical Engineering, Universiti Teknologi MARA, 40450 Shah Alam Selangor. Malaysia.

[pblanes@ai2.upv.es](mailto:pblanes@ai2.upv.es) (F. Blanes). Instituto de Automática e Informática Industrial, Universitat Politècnica de València. Camino de vera, s/n. Ciudad Politécnica de la Innovación. Edificio 8G – Acceso D Planta 3. CP 46022. Valencia. Spain.

# Architecting Centralized Coordination of Soccer Robots based on Principle Solution

Coordination strategy is a relevant topic in multi-robot systems, and robot soccer offers a suitable domain to conduct research in multi-robot coordination. Team strategy collects and uses environmental information to derive optimal team reactions, through cooperation among individual soccer robots. This paper presents a diagrammatic approach to architecting the coordination strategy of robot soccer teams by means of a principle solution. The proposed model focuses on robot soccer leagues that possess a central decision-making system, involving the dynamic selection of the roles and behaviors of the robot soccer players. The work sets out from the conceptual design phase, facilitating cross-domain development efforts, where different layers must be interconnected and coordinated to perform multiple tasks. The principle solution allows for intuitive design and the modeling of team strategies in a highly complex robot soccer environment with changing game conditions. Furthermore, such an approach enables systematic realization of collaborative behaviors among the teammates.

Keywords: multi-robot systems, robot soccer, strategy, principle solution, architecture.

Robot Cooperation and Coordination < System Design, and Integration

## 1. Introduction

Robot soccer is presented as a multi-robot system that includes uncertain dynamics and hostile environments, where robots are working in a coordinated manner on a real challenge problem [1]. In multi-robot systems the need for multi-robot coordination emerges when a team has to accomplish common goals in its environment, and the efficiency of a multi-robot system depends on how well the robots coordinate [2]. Therefore multi-robot coordination represents a special topic in robot soccer [3], due to the fact that non-coordinated strategies present problems such as fellow team members going for the ball at the same time, obstructing each other or not protecting other zones

of the robot soccer field.

Robot soccer leagues can be classified into leagues with centralized control and those with distributed control. Leagues with centralized control only use one decision-making body and a visual system for location from a remote host computer. The motions of the robots are controlled by the decision-making body, which knows the position of both the ball and the robots on the soccer field [4]. In centralized leagues the assignment of roles and selection of behaviors are realized by the decision-making body, and transmitted to the teammates [5]. Thus, in centralized leagues the motion control of all robots and the perception are centralized [6]. In contrast, leagues with distributed control present fully autonomous robots, without any global perception nor global control, although communication does exist among the teammates [7]. In this kind of leagues the decision about role assignment is distributed among the players, and the selection of behaviors is decided on by the robot itself [8]. In the case of distributed robot soccer teams, however, some research has been focused on the learning behaviors of the agents using techniques such as reinforcement learning to control behaviors [9], learning behaviors and learning to dribble [10]. Other strategies are modeled and implemented using finite state machines [11], in some cases in combination with Petri Net Plans [12]. Some models use only Petri Net Plans involving communications among agents [13] or that introduce an integrated framework for modeling, identification analysis, and execution of robot task plans [14]. Collaborative filtering techniques are also used in leagues with fully distributed control [15].

Research on coordination strategies for centralized robot soccer systems has previously been carried out, some studies focusing on the decision-making system. For example in [16] reinforcement learning is combined with the BDI model from a multi-agent view. In other cases neural networks are used for learning in the decision-making

system, to determine the action that a robot should take in a robot soccer game [17]. In [18], a decision-making system based on the dynamic role transition algorithm is presented, proposing a role task allocation mechanism based on prediction with the area of influence. Artificial immune systems are used in the Role Selection Mechanism and Action Select Mechanism for centralized robot soccer teams [19]. Finite State Machine is used in the decision-making model for a robot soccer team, for the tactical decision [20]. In [21] the decision-making scheme is based on the image processing system. Other works just show strategies for specific collaborative behaviors, for example a shooting cooperation strategy [22] or the goalkeeper strategy [23].

However, these studies did not show full architectures that take into account the several layers that make up a centralized robot soccer team. Other studies [24] present a methodology for designing a strategy based on discrete events, and this strategy takes into account the game conditions but does not contain the architecture for the different layers of the team. In [25] a full strategy for a robot soccer team is presented, although the contribution focuses on reinforcement learning for task assignment to each individual robot.

The functionality, structure and design of a robot soccer team, seen as a complex mechatronic system, results in a highly complex design and requires effective coordination among the different domains that make up the architecture of the design [26]. In this way, a systematic approach based on the specification of a principle solution is presented for the design of strategies for robot soccer teams. This approach allows intuitive specification of team strategies and systematic realization for collaborative behaviors among the robots, starting out with the conceptual design phase and facilitating cross-domain development in a multidisciplinary environment. Although principle solution has already been used for the design of strategies of robot

soccer teams [27], it was designed for decentralized humanoid robot soccer teams, but not in centralized robot soccer environments, which present important differences in their architecture, perception and control.

Even though many robot soccer team strategies have been proposed, most of them were designed for distributed teams, since in the case of leagues with centralized control many studies did not take into account the different layers of the team architecture. In this paper we present the strategy design of centralized robot soccer teams based on a principle solution, which provides a holistic system architecture, with an intuitive role assignation and selection behavior design. This architecture design is compatible with different decision-making systems, or reinforces learning algorithms such as those described above. This strategy is tested using the SimuroSot 5 vs. 5 league simulation platform. This simulator is inspired on the FIRA MiroSot Middle League, a centralized robot soccer league with 5 robots per team and a shared vision system [28].

The paper is organized as follows. In section 2 the structure of the robot soccer team used in this model is presented. In section 3 the strategy is modeled using the principle solution for centralized robot soccer teams. In section 4 the model is validated using the SimuroSot robot soccer simulator. Section 5 provides the conclusions.

## **2. Centralized robot soccer team structure**

The basic architecture of a centralized robot soccer team is shown in Figure 1. It consists in a vision system shared by both teams. The image processing is performed by the team's host computer, players' and ball positions being obtained in a system of (x,y) coordinates. With this information the decision-making body contained in the host computer selects and transmits the robots' behaviors, the communication process being carried out using a radio frequency (RF) communication device. Robots receive and process information, and then adapt behaviors on the soccer field.

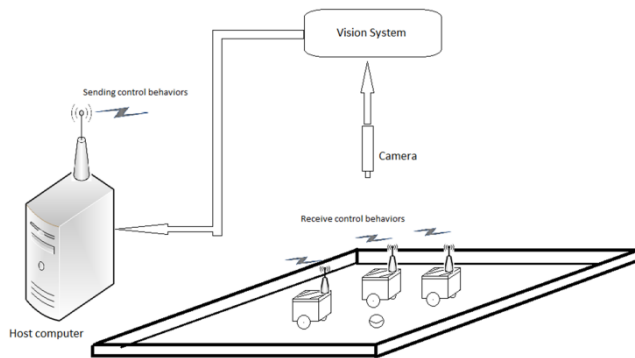


Figure 1. Overall architecture in a centralized robot soccer team.

The strategy presented is designed and implemented using the SimuroSot 3D 5 vs. 5 players simulation platform. In SimuroSot the teams are provided with the positions of the robots and the ball, simulating a top-view vision system. Teams must provide one dll file (programmed in C++) or Lingo archive containing the strategy of the team. This strategy is programmed using only one decision-making body per team.

Each robot soccer team consists of five players. One of them is the Goalkeeper, which is the only static role. The other four roles are active defender, passive defender, supporter and attacker. These roles are assigned depending on game conditions. For role assignment and the selection of behaviors, this strategy is divided into two tactics, each tactic being selected depending on the position of the ball on the robot soccer field. For this purpose the field is divided into two zones, as shown in Figure 2. The defensive zone corresponds to the half zone where the goal and the goalkeeper of the home team are situated, and when the ball is in this zone the defense tactic is selected. The offensive zone is the other half zone where the opponents' goal is located, and if the ball is in the offensive zone, the attack tactic is selected. In Figure 2 the home team is the team where this strategy is implemented, the other is the opponent team.

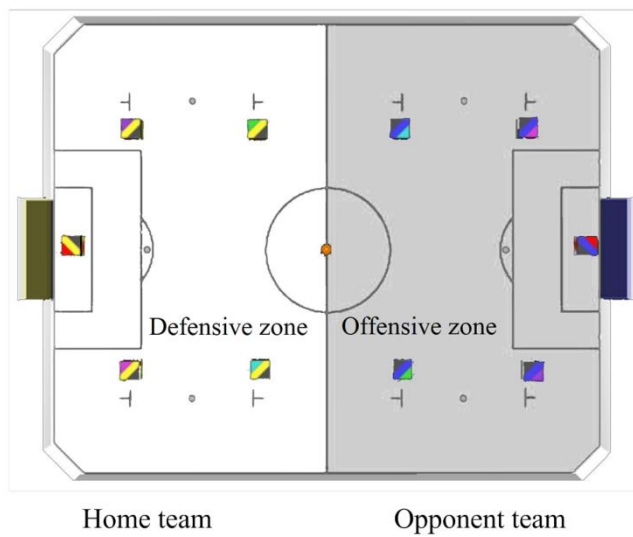


Figure 2. Division of the robot soccer field.

A centralized robot soccer team fits well into the hierarchical structure of complex mechatronics systems suggested in [29]. The lowest layer consists of so-called mechatronic function modules (MFM), which include mechanical structure, sensors, actuators, and local information processing. MFMs are connected by information technology and/or mechanical components that make up autonomous mechatronic systems (AMS), which feature information processing. This information is used to carry out superior tasks such as monitoring, fault diagnosis, and maintenance decisions. In addition, targets for local information processing of the MFM are generated. Multiple AMS connected by information processing form the so-called networked mechatronic systems (NMS).

In a centralized robot soccer environment, the locomotion module (a two-wheeled robot in MiroSot and SimuroSot league) corresponds to the MFM. The whole robot constitutes an AMS, together with the system vision and the host computer. The full team is made up of all the robots, the vision system and host computer (including vision process, strategy and RF communication), and this is the NMS.



### **3. Model of strategy based on a Principle Solution**

For the design of the strategy several aspects must be taken into account [30]. Some of these are the environment, functions, active structure, behavior states, and behavior activities, which are described below.

#### ***3.1 Functions***

Functions correspond to the hierarchical subdivision of the functionality of a centralized wheeled robot soccer team. The main idea of the function is to identify activities and behaviors which must be programmed and taken into account in modeling and designing the team strategy. Function requires coherence between input and output parameters, with the aim of fulfilling a task. These functions are carried out by solution patterns and their concretization. A subdivision into sub-functions takes place until a useful solution pattern appears for the functions.

Figure 3 shows a function hierarchy for a wheeled robot soccer player which belongs to the SimuroSot league. The main function of playing soccer is decomposed into five sub-functions, which are necessary for the full team to play soccer, and in turn four sub-functions are subdivided into lower hierarchy sub-functions. The sub-functions that fulfill the main function are defined as follows.

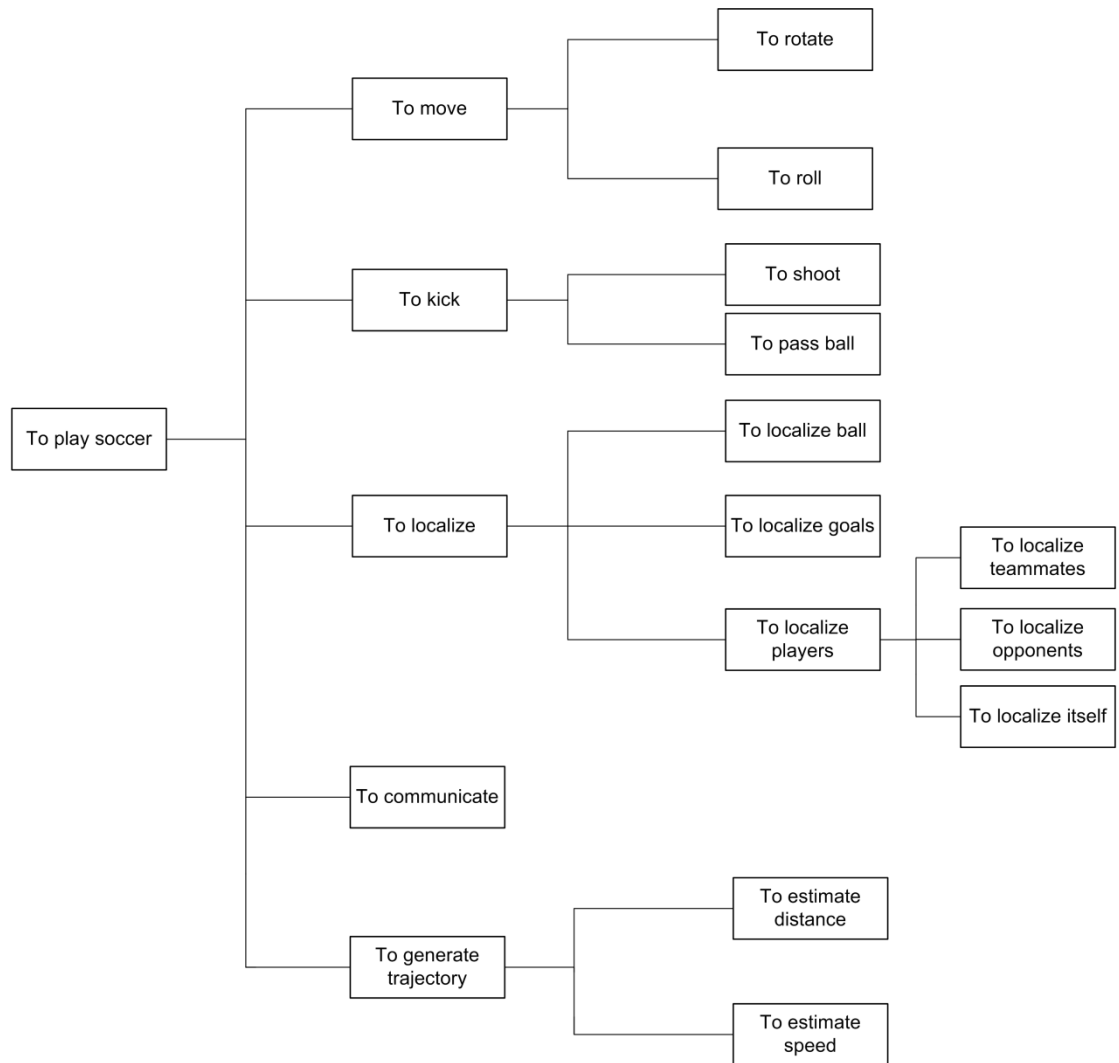


Figure 3. Function and sub-function of a wheeled soccer robot, in a centralized robot soccer team.

### 3.1.1 To move

This function refers to the locomotion tasks. The robot must rotate around its axis, or roll to a particular location on the soccer field. Consequently two sub-functions appear to execute the locomotion tasks. Inverse kinematics of the robots are required, in the SimuroSot league two-wheeled robots are used, the motion control of the robots is carried out from the decision-making body, and transmitted to the robots.

### *3.1.2 To kick*

This function refers to the ability to shoot the ball or pass it to a teammate, thereby generating two sub-functions which depend on the mechanical design and the inverse kinematics of the robot. In SimuroSot the kicking function does not allow hardware modifications to be made to the robots. However, a kicking behavior is designed and carried out.

### *3.1.3 To localize*

This function refers to the ability to localize teammates, opponents, goals and ball on the playing field. As was explained earlier, in centralized leagues this function is performed by a camera above the playing field, which provides the location using a system of coordinates. In SimuroSot the location of the ball and players is supplied by the simulator, the coordinates being provided to the decision-making body.

### *3.1.4 To communicate*

This function refers to the information exchanged between the vision systems and the host computer, and vice versa. In SimuroSot the positions of the players and the ball are supplied by the simulator. Consequently the communication function is integrated into the simulator.

### *3.1.5 To generate trajectory*

This function refers to the path for the movement of the robots. For this purpose the kinematic model is needed. Two important variables must be controlled for this particular strategy, i.e. distance to the target and speed of the robot. This function is carried out through the decision-making body.

### 3.2 Environment

This model describes the surroundings of this system, their relevant influences, and their attributes and characteristics. Figure 4 shows the environment for this strategy.

The system elements consist of a ball, teammates, opponents, goals, host computer, and field zones. The influences between each robot and the environment are related by the influence tables. For example, Figure 4 shows the influences between the robot soccer player and the ball. The influences that trigger a state transition are marked as events. For instance, the distance between ball and player is one influence to assign roles, just as zone location is an influence for selecting tactics, or ball possession is an influence to select behaviors. Influences marked as a non event provide information that allows behaviors to be performed.

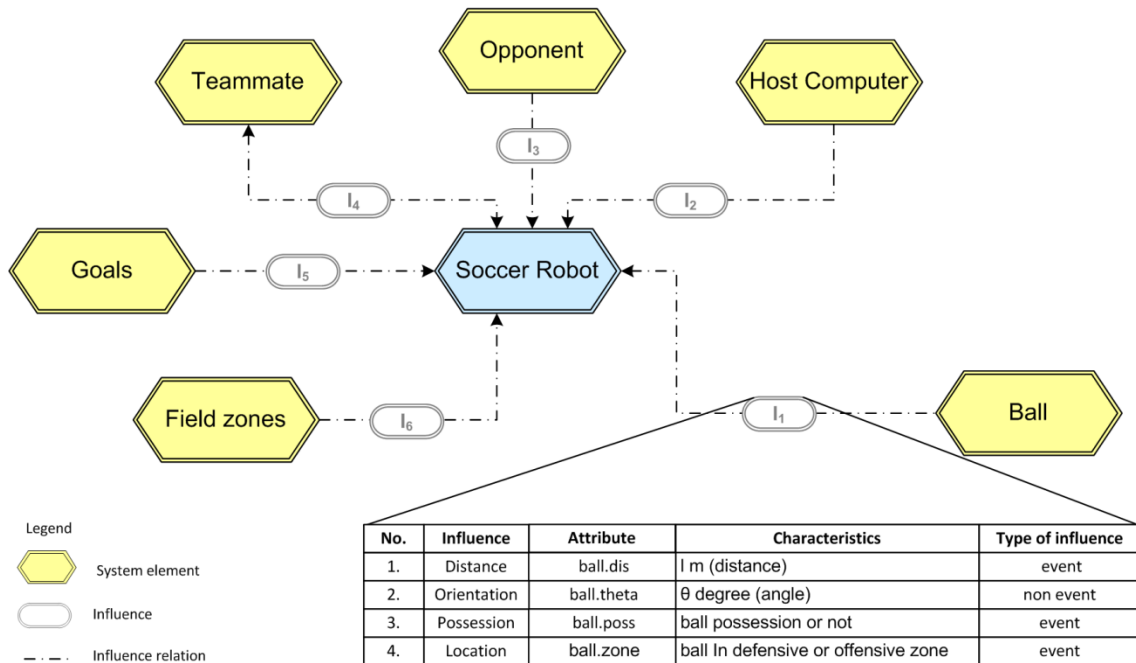


Figure 4. Environment model of centralized robot soccer player. Influence table I<sub>1</sub> between soccer robot and ball is detailed.

Table 1 details the influence tables I<sub>2</sub>, I<sub>3</sub>, I<sub>4</sub>, I<sub>5</sub>, and I<sub>6</sub>.

I <sub>2</sub>				
No.	Influence	Attribute	Characteristics	Type of influence
1.	Initialization of game	player.coordinates	coordinates (x,y)	event
2.	Goals scored	score.team	goals scored by team	non event
I <sub>3</sub>				
No.	Influence	Attribute	Characteristics	Type of influence
1.	Distance	opponent.dis	l m (distance)	non event
2.	Orientation	opponent.theta	Θ degree (angle)	non event
3.	Nearest opponent	opponent.nearest	nearest or not	event
I <sub>4</sub>				
No.	Influence	Attribute	Characteristics	Type of influence
1.	Distance	player.dis	l m (distance)	no event
I <sub>5</sub>				
No.	Influence	Attribute	Characteristics	Type of influence
1.	Distance	goal.dis	l m (distance)	event
I <sub>6</sub>				
No.	Influence	Attribute	Characteristics	Type of influence
1.	Location	player.zone	player in defensive or offensive zone	event
2.	Distance to zone	zone.dis	l m (distance)	event

Table 1. Influence tables.

### 3.3 Active structure

The active structure describes the system elements, their attributes and their interrelationship, with the aim of defining the basic structure of the system. Figure 5 describes an active structure of a centralized robot soccer team, focusing on SimuroSot

League. This structure is used for viewing the system elements and the information flow between them. The modules correspond to software modules for specific applications. Five software modules are identified, four of them in the host computer: perception module, team strategy module, control module, and communication module, and another one in the robot: communication module. Additionally, the sensor and actuator hardware in the MiroSot league is modeled and simulated in the SimuroSot league. The interrelation among modules is related by information flows. Furthermore, the interaction is related as an energy flow. In the case of MiroSot the interaction is sensed by cameras (vision hardware), while in SimuroSot this interaction is embedded in the simulator, without user interaction. The modules are described as follows.

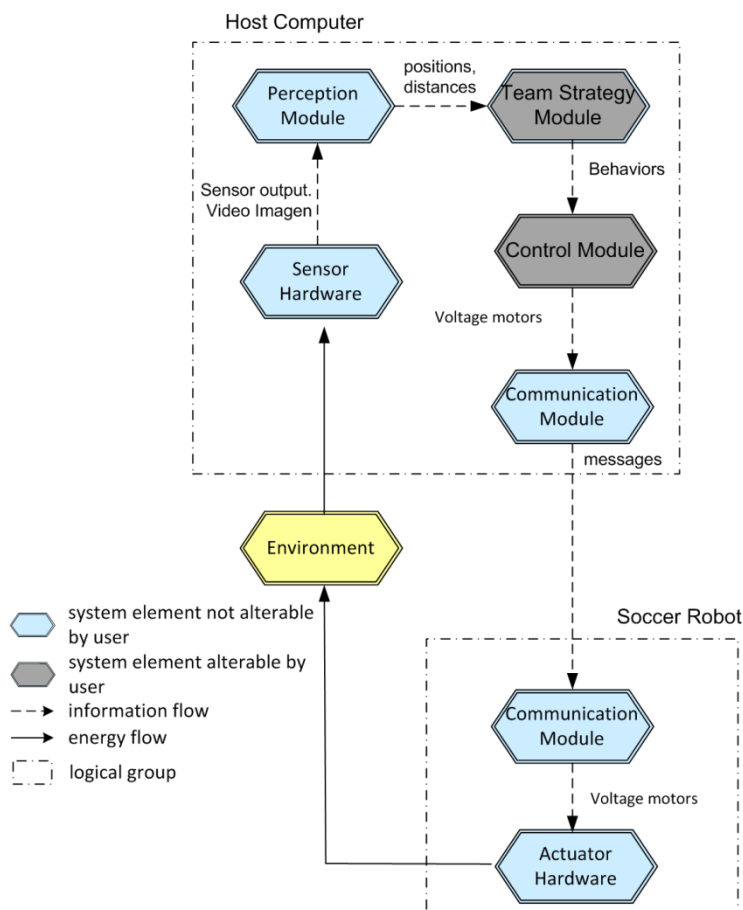


Figure 5. Active structure of a robot soccer player with centralized control, using SimuroSot League.

### *3.3.1 Team strategy module*

This module refers to strategy implementation, selection of tactics, roles and behaviors. In this module the decision-making systems of the strategy are implemented. The information obtained in this module is sent by information flow to the control module.

### *3.3.2 Control module*

In the control module, the information about behaviors must be transformed into actuator signals, based on inverse kinematics of the robots. In both MiroSot and SimuroSot, the inverse kinematics model of the robots must be obtained, in order to calculate the equivalent signal for each actuator, depending on the behavior selected by the team strategy module. This information is sent to the communication module.

### *3.3.3 Communication module*

This module refers to communication and must be further divided into two modules, which are the communication module in the host computer and the communication module in the robot. In the host computer, control signals received from the control module must be sent to the robots using RF. This signal is received by the communication module of the robot, and transmitted to actuator signals of both wheels for the case of MiroSot. In SimuroSot this module is provided by the simulator and is not programmed by users.

## **3.4 Behaviors**

The strategy presented is divided into two tactics. The defense tactic is selected when the ball is located in the defensive zone, and the attack tactic is selected when the ball is situated in the offensive zone. Four roles are assigned depending on game conditions. These roles are active defender, passive defender, attacker, and supporter. Behavior

diagrams are used to assign roles and then to select appropriate behaviors. These consist of two sub-diagrams, one of which is the partial model behavior-state, which describes the envisaged system state, state transitions and events that trigger a state transition. The other one is the partial model behavior-activity, which describes logical sequences of system activities, including all operation and adaptation processes. In this model, operation processes refer to activities which are carried out within a state, adaptation processes refer to activities which are carried out during the state transition. When one event occurs, one adaptation process is triggered. Subsequently, the system takes on a new state and then a new set of operation processes are activated.

In the context of behavioral specifications, the tactics used by the team (defense tactic and attack tactic), roles (goalkeeper, attacker, supporter, active defender, and passive defender) and behaviors (for example, go to the ball, shoot, send a pass, and others), and their interrelations must be described. The behavior-state diagram relates the state-transition between tactics as states, and events that activate each tactic. In addition, behavior-state diagrams also relate the process of role selection, showing roles as states, and the conditions to assign roles as events. Behavior-activity diagrams refer to sequence behaviors that a specific role carries out. In a behavior-state diagram start represents the time when a role is activated by a player, activity corresponds to a behavior to be performed, and alternative corresponds to a conditional which depends on an environment condition to select the next activity from two or more possible options. When an activity is accomplished, the next activity connected by a logical connection is carried out.

Figure 6 shows the behavior-state model for the proposed model. The highest level, which contains two tactics, corresponds to the full strategy of a robot soccer team. The strategy level is divided into two states which are tactics that make up the whole



strategy, in this case they are defense tactic and attack tactic. Once a tactic has been selected, a fitness function is evaluated in each player, and subsequently roles are assigned: this can be seen in Figure 6 as sub-states selected by events. If the ball changes zone, event 1 (E1) or 2 (E2) is triggered, consequently a new strategy is activated and the fitness function is evaluated.

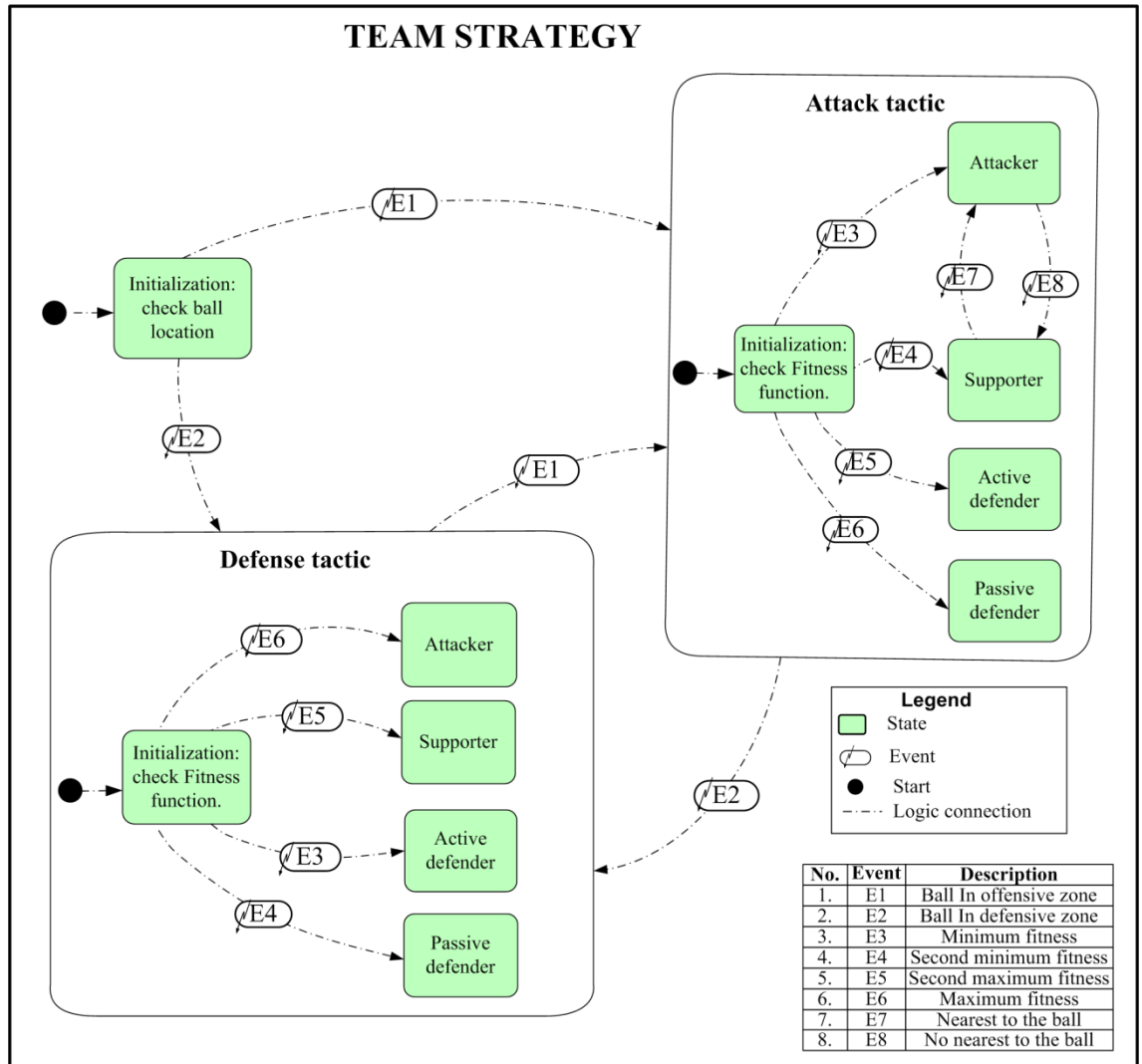


Figure 6. Behavior-state of the proposed strategy for a centralized robot soccer team.

This fitness function is presented in Equation (1):

$$f(t) = k_1 \left( \|x_b(t) - x_p(t), y_b(t) - y_p(t)\|_2 \right) + k_2 \left( \|x_g - x_p(t), y_g - y_p(t)\|_2 \right) \quad (1)$$

Where  $\|x_b(t) - x_p(t), y_b(t) - y_p(t)\|$  corresponds to the Euclidean distance between the ball located at the coordinates  $(x_b(t), y_b(t))$ , and a player located at the coordinates  $(x_p(t), y_p(t))$ .  $\|x_g - x_p(t), y_g - y_p(t)\|_2$  corresponds to the Euclidean distance between the center of the opponents' goal located at the coordinates  $(x_g, y_g)$ , and the player.  $k_1$  and  $k_2$  correspond to two constants such as  $k_1 \geq 0$ ,  $k_2 \geq 0$  and the condition shown in Equation (2):

$$k_1 + k_2 = 1 \quad (2)$$

The importance of  $k_1$  and  $k_2$  in the fitness function presented in Equation (1) is to weight the distances of the player to the ball and to the opponents' goal respectively. Usually the distance between player and ball is significantly lower than the distance between player and the opposite goal, and thus  $k_1$  must have a value that is higher than  $k_2$ , although  $k_2$  is important to provide information about the position of the player, in the case of two players with similar distances to the ball. Many experiments were performed in order to choose the values of  $k_1$  and  $k_2$ . In these experiments robots are placed on the playing field, and different values of  $k_1$  and  $k_2$  are assigned taking into account the conditions shown in equation (2). When there is a tactic transition the new role assignment is accomplished and each role assigned to each player is observed, the designer of the strategy observing whether the role assigned is suitable for playing the game, maintaining ball possession, making passes or blocking opposite passes. On starting with  $k_1 = k_2 = 0.5$ , however, it was observed that the player with the possession of the ball leaves it in order to take on a defensive role, or shows abrupt changes in behavior. In the case where  $k_1$  was very near to 1,  $k_2$  is almost negligible and cooperative behaviors like passes are very rare. By conducting different experiments

with different values it was observed that an adequate team performance was observed with  $k_1=0.75$  and  $k_2=0.25$ .

When the fitness function is evaluated, a new assignation of roles is performed. The activities to be carried out in the sub-states correspond to behaviors executed by roles, a situation that is explained in the behaviors-activity diagrams. This model ensures that the roles of each player change dynamically during the game, and each robot carries out its own behavior while still cooperating with its teammates.

### 3.4.1 Goalkeeper

The goalkeeper is the only static role in this strategy, in both tactics, that is, in defense and in attack. Figure 7 shows the goalkeeper zone, which is an area in front of the goal where the goalkeeper plays. The behavior-activity of this role is presented in Figure 8. When goalkeeper behavior is triggered, the goalkeeper checks its location. If the goalkeeper is in the goalkeeper zone, it localizes and follows the ball in a line running parallel to the goal line. If the goalkeeper is not located in this zone, the robot must become properly located and subsequently checks its location and continues playing.

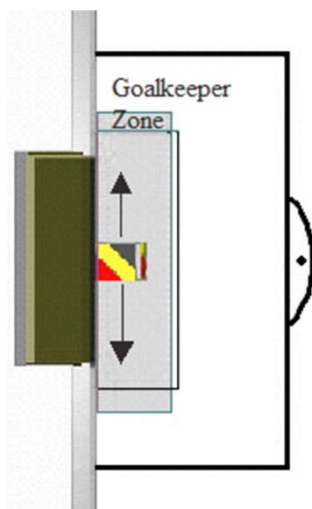


Figure 7. Goalkeeper zone corresponds to the gray area. The goalkeeper follows the ball inside the zone, parallel to the goal line.

Although the goalkeeper strategy presented here has been implemented in this model, any other goalkeeper strategy can be implemented without significantly affecting the strategy proposed.

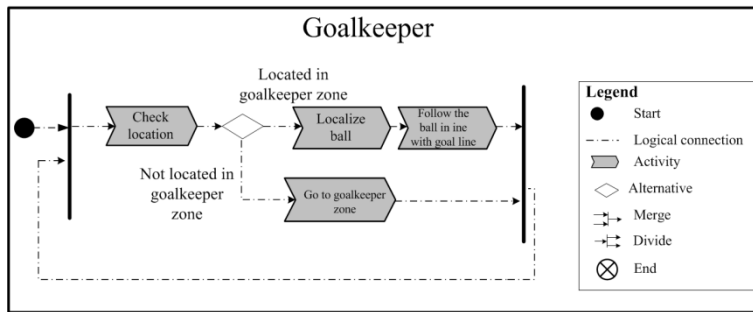


Figure 8. Behavior-activity diagram for the goalkeeper.

### 3.4.2 Defense tactic

The defense tactic is activated when event E2 (ball in defensive zone) is triggered. This tactic is activated with the intention of seeking to intercept and send the ball to the opposite zone. Roles are assigned depending on the values of the fitness function in each player, as shown in Figure 6. The player with the minimum fitness level is assigned with the role of active defender and its behavior-activity is shown in Figure 9. When this behavior is triggered, the robot checks the ball location and subsequently follows the ball. If the robot has ball possession it shoots the ball to the offensive zone and checks the ball location again. If the robot does not have ball possession, it goes to find the ball following the sequence described in Figure 9. Ball possession is defined as the minimal distance between the robot and the ball, where it is possible for the robot to shoot the ball. This value is obtained in an experimental way.

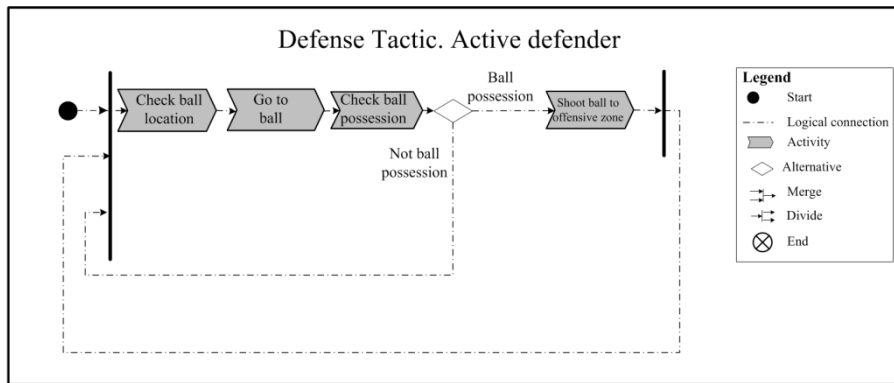


Figure 9. Behavior-activity diagram for an active defender, with defense tactic.

The robot with the second minimum fitness level becomes the passive defender, and its behavior-activity diagram is shown in Figure 10. As can be seen, the robot evaluates the nearest opponent, moving to intercept this opponent player, in order to intercept possible passes directed to the opponent, but always seeking to avoid collisions with it. When the opponent is intercepted, the passive defender again checks for the nearest opponent and continues the behaviors mentioned above.

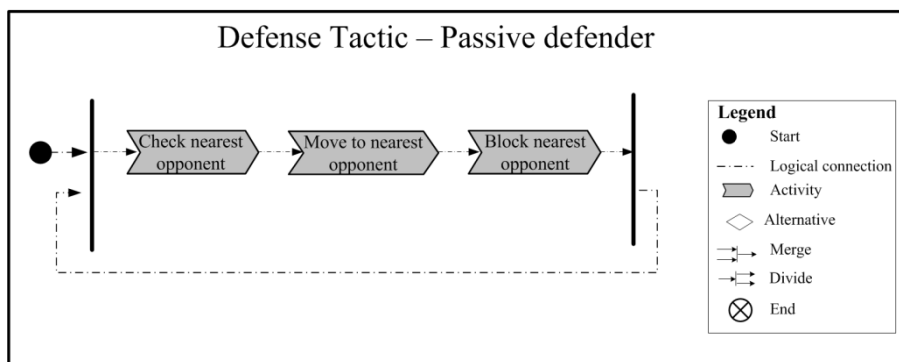


Figure 10. Behavior-activity diagram for a passive defender, with defense tactic.

The robot with the maximum fitness and the second maximum fitness levels are the attacker and supporter, respectively. Their behavior-activity is similar and can be seen in Figure 11. When these roles are activated, the robot checks its respective location, and if it is in the defensive zone, it must go to the offensive zone. If the robot is in the offensive zone, it is waiting for the ball in the case of a counter attack. The only

difference between the two roles is where each robot is placed, the objective being to cover the opponent zone uniformly.

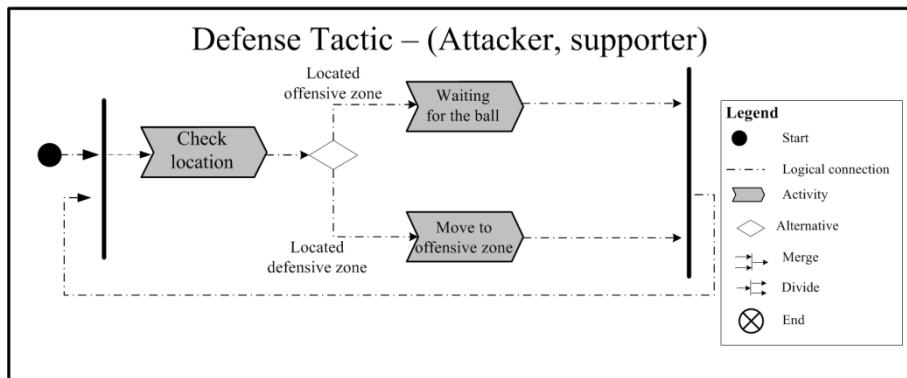


Figure 11. Behavior-activity diagram for attacker and supporter, with defense tactic.

Figure 12 shows an example of assigned roles and their behaviors in a game, when the defense tactic is activated. The goalkeeper follows the ball parallel to the goal line, in the goalkeeper zone, the active defender goes to the ball and shoots to the offensive zone, and the passive defender blocks an opponent player which is trying to intercept the ball. Both attacker and supporter “watch” the ball waiting for a pass.

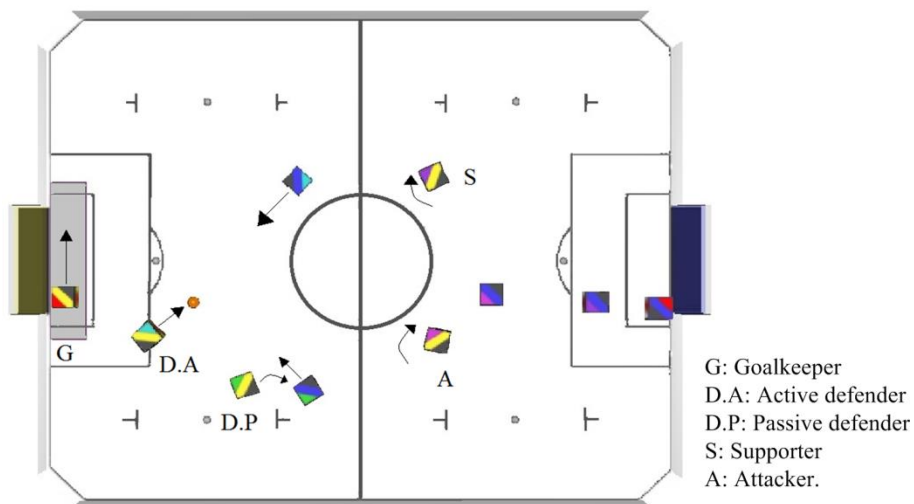


Figure 12.

Example of roles and behaviors for defense tactic.

### 3.4.3 Attack tactic

The attack tactic is activated when event E1 (ball in the offensive zone) is triggered.

This tactic is focused on shooting the ball and scoring a goal. Once a tactic has been

selected, the fitness function is also evaluated for role assignment, as shown in Figure 6.

Players with the maximum fitness and the second maximum fitness values are selected

for the roles of passive defender and active defender, respectively. Each one is assigned

a location within the defensive zone where the robots must follow the ball along a line

running parallel to the goal line. These zones are called the passive defense zone and

active defense zone, respectively, and are shown in Figure 13.

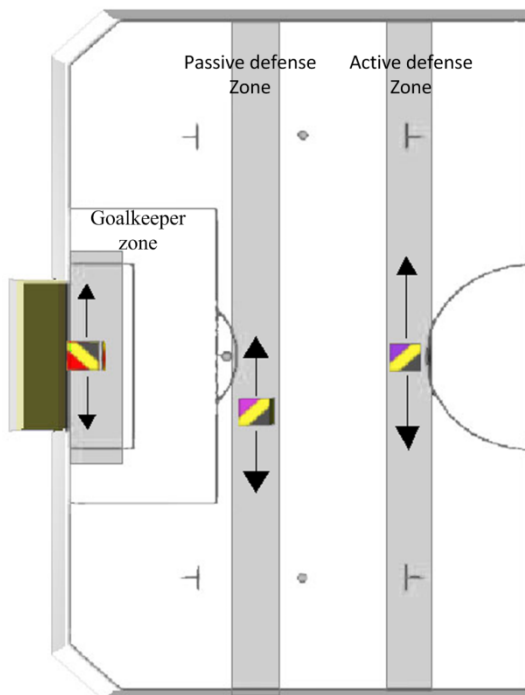


Figure 13. Goalkeeper zone, with the passive defense zone and active defense zone.

The main aim of both defenders is to intercept the ball in the case of a counter attack by

the opposite team. The behavior-activity of both defenders is similar and is shown in

Figure 14. When a defender's behavior (active or passive) is triggered, the robot first

checks its location. If the robot is outside its location, it moves to its respective zone and

then checks its location again. When the robot is inside its zone, it follows the ball in line. The only difference is that the location of the active defender is closer to the halfway line than the location of the passive defender.

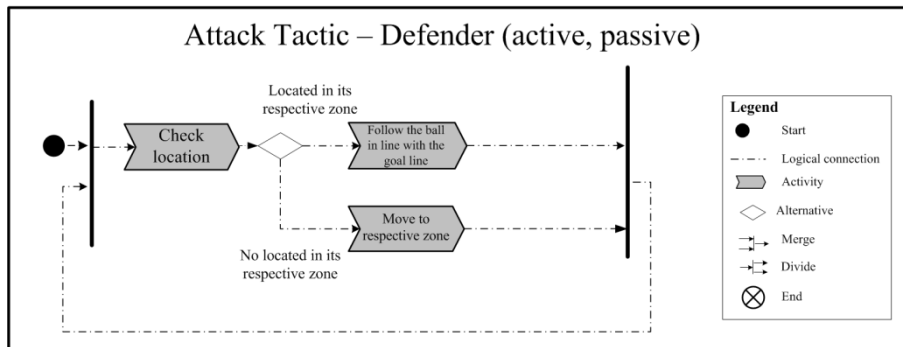


Figure 14. Behavior-activity diagram for the active defender and passive defender, with attack tactic.

Attacker and supporter roles are assigned to players with the minimum fitness and the second minimum fitness values, respectively. However, these roles can be interchanged, as shown in Figure 6. Thus, the attacker is the player that is closer to the ball than its teammates, and the other player becomes the supporter. Figure 15 shows the behavior-activity of the supporter. When the supporter role is triggered, the supporter checks its location. If the supporter is not in the supporter zone, it goes to the supporter zone. When the supporter is properly situated, the robot checks for the ball location and follows the ball parallel to the touchline, subsequently the robot checks the distance to the ball, and if it is nearer than the attacker, this robot becomes the attacker (triggered attacker role shown in Figure 16), but if not, it continues as the supporter and checks its location again.



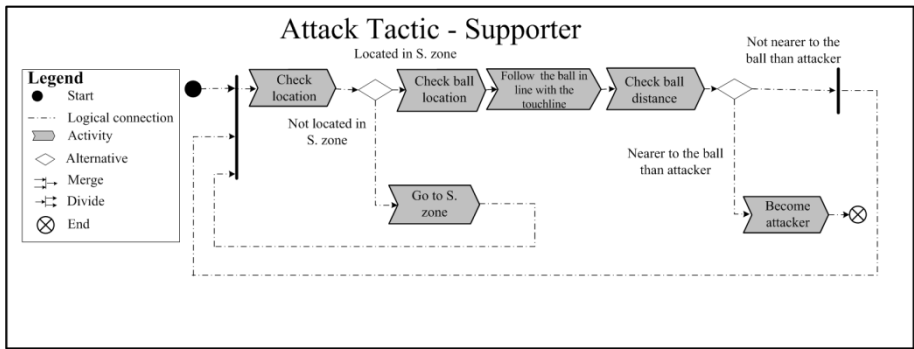


Figure 15. Behavior-activity diagram for a supporter, with attack tactic. In the case of the supporter coming nearer to the ball than the attacker, it goes to the ball and becomes the attacker.

Figure 16 shows the behavior-activity of the attacker. As can be seen, first the attacker checks whether it is nearer the ball than the supporter, event E8 is triggered and this robot becomes the supporter thus triggering the supporter role shown in Figure 15. If it is closer to the ball, then the attacker checks for the ball location and goes to intercept the ball. Subsequently the attacker checks for ball possession. If it has possession of the ball, the attacker shoots the ball at the opponents' goal, and if not, the attacker checks the ball distance again.

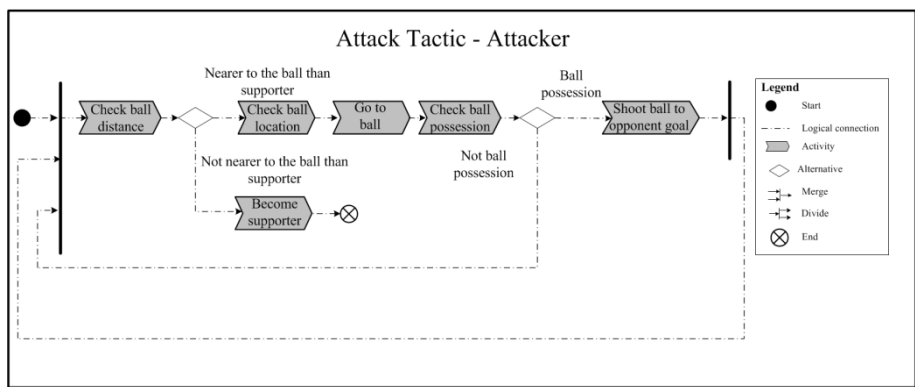


Figure 16. Behavior-activity diagram for attacker, with attack tactic. If the attacker is not nearer the ball than the supporter, then it becomes the supporter.

One example of the roles in the attack tactic is presented in Figure 17. The attacker follows the ball in search of a goal. The supporter follows the ball parallel to the touchline in the supporter zone. Active defender and passive defender follow the ball in their respective zones (passive defense zone and active defense zone), parallel to the goal line.

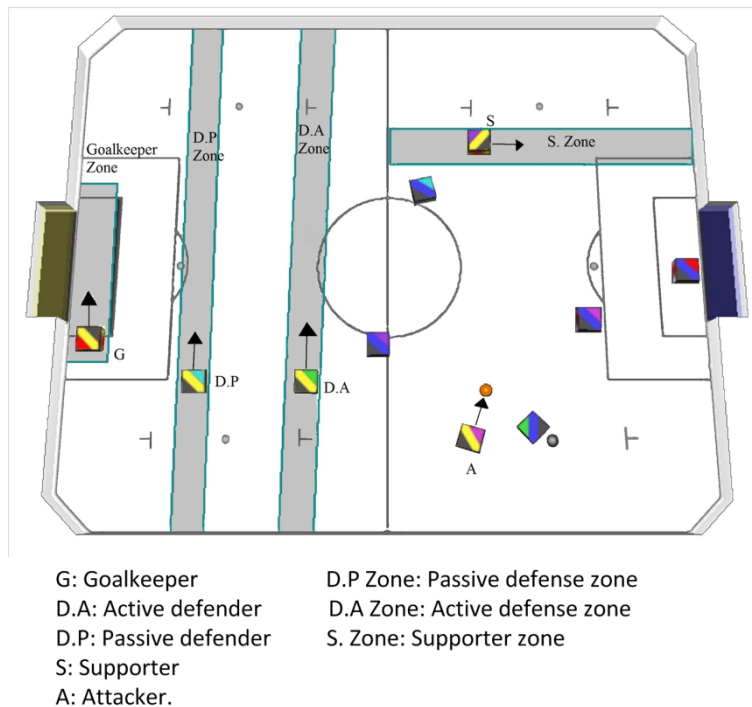


Figure 17. Example of roles and behaviors for the attack tactic.

#### 4. Experimental Results

The proposed strategy is implemented in Fira's SimuroSot using the C++ programming language. In order to validate the performance of the proposed strategy, 10 games each lasting 5 minutes were run against team strategy provided by default in the simulator, which consists of five constant roles. One is the goalkeeper, which blocks the ball in parallel to the goal line. Others are two defenders, which also block the ball in parallel to the goal line, distributed in the opponents' defensive zone. The last two roles are those of attackers, which basically go to the ball and shoot at their opponents' goal.

Games were played using the rules about fouls or stalemates of the SimuroSot league.<sup>1</sup>

The variables of interest to conduct the validation of a team's performance in each game included goals scored, successful passes, percentage of ball possession, and percentage of time spent in each tactic. Other variables taken into account were own goals by a team, goals disallowed due to fouls, and fouls and blocks per team.

Table 2 presents the score of games, successful passes, and percentage of ball possession in each game. The team with the proposed strategy is the home team, and the opponent is the team with the strategy provided by the simulator. The home team won all the games and showed a higher percentage of ball possession than the opposing players in the 10 games, displaying clear control throughout the games. It can also be seen that the number of successful passes of the team with the proposed strategy is higher than that of the opponent team, similar values being shown in all games.

Game		1	2	3	4	5	6	7	8	9	10	Average
Goals scored	Home	3	3	2	2	3	4	4	3	4	2	3
	Opponent	2	1	0	1	1	2	2	1	0	0	1
Successful passes	Home	5	3	6	5	4	5	6	3	6	5	4.8
	Opponent	1	2	2	0	1	1	4	0	1	0	1.2
Ball possession	Home	78%	67%	77%	66%	68%	76%	84%	61%	77%	69%	72.3%
	Opponent	22%	33%	23%	34%	32%	24%	16%	39%	23%	31%	27.7%

Table 2. Goals scored, successful passes, and ball possession per team.

Table 3 shows the fouls and stalemates, the own goals per team and the disallowed goals, which were goals scored after or whilst committing fouls. Many fouls and stalemates by the opponent team were observed in the games, due to the fact that in

---

1. [http://www.fira.net/contents/data/Middle\\_League\\_SimuroSot.pdf](http://www.fira.net/contents/data/Middle_League_SimuroSot.pdf)

many cases two or more opposing players go for the ball at the same time and block the ball, without coordination. One important reason why the number of fouls and stalemates were significantly lower in the home team lies in the fact that the strategy was designed taking into account the rules of the league, for example avoiding more than one player going for the ball at the same time, preventing them from defending with more than one player in the goal area, or avoiding defending with more than three robots inside the penalty area.

Game		1	2	3	4	5	6	7	8	9	10	Average
Fouls and stalemates	Home	2	1	2	1	4	2	2	0	3	1	1.8
	Opponent	11	7	8	5	12	8	7	13	8	8	8.7
Own goals	Home	1	1	0	1	0	2	1	0	0	0	0.6
	Opponent	1	0	2	0	0	2	0	0	0	0	0.5
Disallowed goals	Home	1	1	1	1	2	1	1	0	1	0	0.9
	Opponent	2	0	2	0	0	0	1	2	1	0	0.8

Table 3. Fouls and stalemates, own goals, and disallowed goals.

Table 4 shows the percentage of time for each tactic selected by the home team in each game. It can be seen that attack tactic (A.T) was not the most frequently selected in all the games. In fact, defense tactic (D.T) was the most frequently selected in two of the games. Furthermore, in most of the cases both percentages were quite similar.

Game	1	2	3	4	5	6	7	8	9	10	Average
D.T	34%	41%	40%	49%	46%	52%	47%	35%	43%	53%	44%
A.T	66%	59%	60%	51%	54%	48%	53%	65%	57%	47%	56%

Table 4. Percentage of tactic selected during the games.

In this respect the strategy designed using principle solution with dynamic assignation of roles showed an adequate performance by winning all the games and obtaining higher ball possession than the opponent players, although many defensive behaviors were performed throughout the games. This situation emerged as a response to the strategy of the opponent team. Although some goals were scored by the opponent, the defense tactic performed adequate activities and behaviors to respond to the opponent team, forcing it to select the attack tactic and scoring goals to win the games. It is noteworthy that the average selection of attack tactic was slightly higher than the selection of the defense tactic, which means that the ball spent more time in the offensive zone.

Many collaborative behaviors were observed during the games. In one situation, an active defender intercepted and sent the ball into the offensive zone, in some cases an active defender kept possession of the ball and became an attacker, and in other cases the attacker or supporter received the pass from an active defender. Another situation when the attack tactic was selected was that the attacker or supporter (which became the attacker) intercepted the ball and scored a goal. One example in the attack tactic is shown in Figure 18, where attacking behaviors are presented. In Figure 18(a) the attacker shoots the ball with the aim of scoring a goal, but the ball rebounds off an opponent. In Figure 18(b) the roles change and the supporter becomes the attacker and the attacker is now the supporter. The new attacker goes for the ball while the supporter is blocked by opponents, and in Figure 18(c) the attacker subsequently scored a goal.

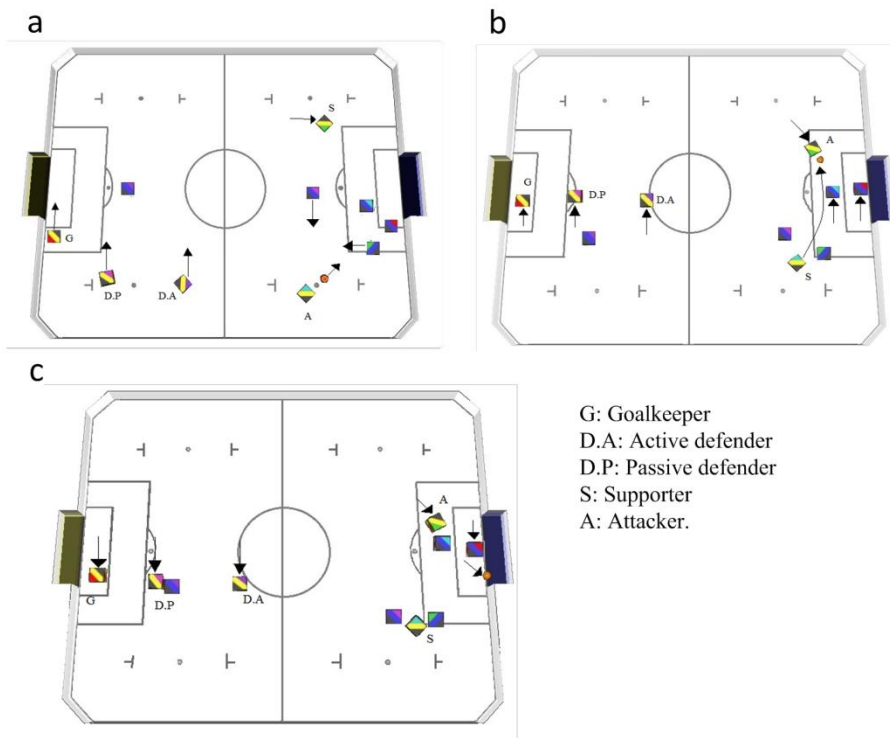


Figure 18. Example of behaviors in the attack tactic. The attacker shoots the ball into the opponents' goal. The ball rebounds and the new attacker goes to intercept the ball. The attacker then shoots and scores a goal.

## 5. Conclusions

Principle solution offers an alternative for modeling and designing robot soccer strategies in centralized robot soccer teams in a clear and intuitive way, adopting a hierarchical structure. This leads to a holistic model including the different layers of both software and hardware, which should be taken into account for the design of the architecture strategy. Principle solution allowed the identification of different functions, system elements with their influences, and structures for the design of strategy architecture for centralized robot soccer teams, where the decisions and data processing are carried out on a remote host computer, thereby presenting a different paradigm to that of fully distributed robot soccer teams. This facilitates the design of different rules for role selection and subsequent behaviors performed by robot soccer players,

depending on the dynamic conditions of the game. If changes of code were required, the specific modifications can be recognized by following the diagrams. This would allow an intuitive identification of code lines to be updated.

Another important advantage is the possibility of adapting this strategy to new roles or collaborative behaviors, such as behaviors that include coordination among the goalkeeper and defenders, as described in the introduction. It may also involve machine learning not only for the behaviors of the players, but also in the decision-making system. This strategy was implemented using the SimuroSot robot soccer simulator, playing 10 games against the strategy provided by the simulator, and showed a clear dominion in the games, winning all of the games and offering greater possession of the ball than the opponent. Although this model was simulated, the proposed strategy model could be implemented in the robot soccer MiroSot League, without any significant changes in the architecture presented.

One way to improve the performance of the robot soccer team, in order to score more goals or to avoid own goals, is to increase the collaborative behaviors. The reason why the home team did not score more goals, despite maintaining a greater percentage of ball possession, is similar to the reason why the home team scored own goals, and is related to the need to strengthen the performance of collaborative behaviors, between defenders and the goalkeeper in the defense tactic, and between the attacker and the supporter in the attack tactic. One way to improve this issue is to use learning algorithms in cooperative behaviors. Values of  $k_1$  and  $k_2$  are directly related with role assignment, and they did not affect the performance of the behaviors, although they could affect team performance by assigning incorrect roles to players.

In future research, machine learning will be implemented for the decision-making system used for role selection, in order to obtain better performance in role

assignment, and machine learning will also be incorporated into the behaviors in the search for a better execution of collaborative behaviors, for example by improving the coordination in a pass between supporter and attacker so as to be able to score a goal.

## References

- [1] A. Farinelli, L. Iocchi and D. Nardi, "Multirobot systems: a classification focused on coordination," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 5, October 2004.
- [2] A. Tews and G. Wyeth, "MAPS: a system for multi-agent coordination," *Advanced Robotics*, vol. 14, no. 1, pp. 37-50, 2000.
- [3] F. Stulp, H. Utz, M. Isik and G. Mayer, "Implicit Coordination with Shared Belief: A Heterogeneous Robot Soccer Team Case Study," *Advanced Robotics*, vol. 24, no. 7, p. 1017–1036, 2010.
- [4] X. Zhang, C. Liu, Q. Yu and Z. Li, "A Identification Method for Robotics Soccer," in *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, Wuhan, 2010.
- [5] J. G. Guarnizo, M. Mellado, Y. L. Cheng and A. Norheliena, "Strategy Model for Multi-Robot Coordination in Robotic Soccer," in *Applied Mechanics and Materials*, Malacca, Malaysia, 2013.
- [6] P. Riley and M. Veloso, "Recognizing Probabilistic Opponent Movement Models," in *RoboCup 2001: Robot Soccer World Cup V*, Springer Berlin Heidelberg, 2002, pp. 453-458.
- [7] R. Ros, J. L. Arcos, R. Lopez de Mantaras and M. Veloso, "A case-based approach for coordinated action selection in robot soccer," *Artificial Intelligence*, vol. 173, no. 9-10, pp. 1014-1039, June 2009.
- [8] J. Atkinson and D. Rojas, "On-the-fly generation of multi-robot team formation strategies based on game conditions," *Expert Systems with Applications*, vol. 36, no. 2, pp. 6082-6090, April 2009.
- [9] J. Cunha, R. Serra, N. Lau, L. S. Lopes and A. J. R. Neves, "Learning robotic soccer controllers with the Q-Batch," in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*, Espinho, 2014.
- [10] M. Riedmiller, T. Gabel, R. Hafner and S. Lange, "Reinforcement learning for robot soccer," *Autonomous Robots*, vol. 27, no. 1, pp. 55-73, July 2009.



- [11] J. Ibarra Zannatha, L. Figueroa Medina, R. Cisneros Limón and P. Mejía Álvarez, "Behavior control for a humanoid soccer player using Webots," in Electrical Communications and Computers (CONIELECOMP), 2011 21st International Conference on, San Andres Cholula, 2011.
- [12] G. Kontes and M. G. Lagoudakis, "Coordinated Team Play in the Four-Legged RoboCup League," in Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, Patras, 2007.
- [13] P. F. Palamar, V. A. Ziparo, L. Locchi, D. Nardi and P. Lima, "Teamwork Design Based on Petri Net Plans," in RoboCup 2008: Robot Soccer World Cup XII, 2009.
- [14] H. Costelha and P. Lima, "Robot task plan representation by Petri nets: modelling, identification, analysis and execution," *Autonomous Robots*, vol. 33, no. 4, pp. 337-360, November 2012.
- [15] P. H. Abreu, D. Castro Silva, F. Almeidae and J. Mendes-Moreira, "Improving a simulated soccer team's performance through a Memory-Based Collaborative Filtering approach," *Applied Soft Computing*, vol. 23, pp. 180-193, 2014.
- [16] Q. Guo and B.-Y. Wu, "Study and Application of Reinforcement Learning in Cooperative Strategy of the Robot Soccer Based on BDI Model," *International Journal of Advanced Robotic Systems*, vol. 6, no. 2, pp. 91-96, June 2009.
- [17] Y. Duan, Q. Liu and . X. Xu, "Application of reinforcement learning in robot soccer," *Engineering Applications of Artificial Intelligence*, pp. 936-950, October 2007.
- [18] M. Yang and Y. Jia, "Action Utility Prediction and Role Task Allocation in Robot Soccer System," in Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on, Guangzhou, 2012.
- [19] Y.-T. Wang and C.-H. Chen, "Design of an Action Select Mechanism for Soccer Robot Systems Using Artificial Immune Network," *Tamkang Journal of Science and Engineering*, vol. 11, no. 4, pp. 415-424, 2008.
- [20] Y. Lou, B. Chen and H. Shi, "Decision making model based on state assessment and hierarchical FSM in robot soccer," in Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on, Xiamen, 2012.
- [21] M.-Y. Shieh, . J.-S. Chiou and C.-I. Ko, "Adaptive Classification and Strategy Making System for Android Soccer Games," in Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on, Kuala Lumpur, 2012.
- [22] B. Fu, P.-x. Zhang and C.-f. Wang, "A Cooperation Strategy for Shooting in Robot Soccer Competition Based on the Multi-Suppose Tree," in 2012 International Workshop on Information and Electronics Engineering, Harbin, 2012.

- [23] C. E. Prieto, F. Niño and G. Quintana, "A goalkeeper strategy in robot soccer based on Danger Theory," in Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, Hong Kong, 2008.
- [24] P. Cardoso, L. Molina, E. O. Freire and E. A. N. Carvalho, "A Methodology to Designing Strategies for Robot Soccer Based on Discrete Event Systems Formalism," in Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian, Fortaleza, 2012.
- [25] K.-S. Hwang, W.-C. Jiang, H.-H. Yu and S.-Y. Lin, "Cooperative Reinforcement Learning Based on Zero-Sum Games," in Mobile Robots - Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training, J. Będkowski, Ed., Intech, 2011, pp. 289-308.
- [26] J. Gausemeier, R. Dumitrescu and D. Nordsiek, "Integrative development of product and production system for mechatronic products," Robotics and Computer-Integrated Manufacturing, pp. 772-778, August 2011.
- [27] C. Y. Low, N. Aziz, . M. Aldemir, M. Mellado, R. Dumitrescu and H. Anacker, "Principle Solution for Designing Collaborative Humanoid Soccer Robots," in International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Sarawak, 2012.
- [28] G. Klančar, B. Zupančič and R. Karba, "Modelling and simulation of a group of mobile robots," Simulation Modelling Practice and Theory, vol. 15, no. 6, pp. 647-658, July 2007.
- [29] J. Luckel, T. Hestemeyer and X. Liu-Henke, "Generalization of the cascade principle in view of a structured form of mechatronic systems," in Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on, Como, 2001.
- [30] J. Gausemeier, U. Frank and S. Kahl, "Specification technique for the description of self-optimizing mechatronic systems," Research in Engineering Design, vol. 20, no. 4, pp. 201-223, November 2009.