Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents

Luís Paulo Reis[1], Nuno Lau[2] and Eugénio Costa Oliveira[1]

[1]LIACC – Artificial Intelligence and Computer Science Lab., University of Porto, Portugal
{lpreis|eco}@fe.up.pt, http://www.ncc.up.pt/liacc/
Tel.: 351-22-5081315, Fax: 351-22-5081315
[2]DET – Electronics and Telecommunications Department, University of Aveiro, Portugal
lau@det.ua.pt, http://www.ua.pt
Tel.: 351-234-370524, Fax: 351-234-370545

Abstract. In this paper we are proposing an approach for coordinating a team of homogeneous agents based on a flexible common Team Strategy as well as on the concepts of Situation Based Strategic Positioning and Dynamic Positioning and Role Exchange. We also introduce an Agent Architecture including a specific high-level decision module capable of implementing this strategy. Our proposal is based on the formalization of what is a team strategy for competing with an opponent team having opposite goals. A team strategy is composed of a set of agent types and a set of tactics, which are also composed of several formations. Formations are used for different situations and assign each agent a default spatial positioning and an agent type (defining its behaviour at several levels). Agent's reactivity is also introduced for appropriate response to the dynamics of the current situation. However, in our approach this is done in a way that preserves team coherence instead of permitting uncoordinated agent behaviour. We have applied, with success, this coordination approach to the RoboSoccer simulated domain. The FC Portugal team, developed using this approach won the RoboCup2000 (simulation league) European and World championships scoring a total of 180 goals and conceding none.

1 Introduction

In the last decades, there has been a growth on the popularity and amount of research in the field of Multi-Agent Systems (MAS). This growth came together with the increasing degree of complexity of the requirements induced by the analyzed application domains. In the last years, MAS research has been focused on more realistic environments that are becoming more and more real-time, continuous, non-deterministic, partially inaccessible, dynamic and noisy [20,21]. These kinds of scenarios include search and rescue-like scenarios as it is the case of RoboCup-Rescue domain [11], public transport coordination, mine clearance, land exploration and hospital/factory maintenance [4]. The complexity of these scenarios is even greater when they become multi-objective, simultaneously collaborative as well as adversarial environments as it is found in the simulated RoboSoccer [3,15,16] and robotics domain [27], as well as war scenario domains like battlefield combat [26]. In this kind war scenario domains like battlefield combat [26]. In this kind of domains, coordination must be achieved first, before the competition (game, rescue or battle) starts, through the definition of a flexible team strategy that all the agents know in advance and, second, during the competition, through communication and reactive reasoning based in the sensed information [20]. A good trade-off between social deliberation and reactivity is essential in most of these multi-agent systems applications. Reactivity is usually needed in order to enable the agents to react quickly to the events in the environment, while social deliberation is required for appropriate decision-making processes and to coordinate the agents enabling them to perform as a coherent team despite their own autonomy.

Several authors have proposed general models for flexible teamwork and ways to balance reactive behavior with the social deliberation. However, most of the approaches either are not sufficiently reactive to perform efficiently in real time and very dynamic domains or do not endow agents with sufficiently developed social behavior to perform intelligently as a member of a team in continuous, multi-objective and complex multi-agent environments. Some notable exceptions may be recognized, like Stone and Veloso work [21] that has been applied with success to RoboCup soccer and network routing, Tambe's STEAM [25] successfully applied in virtual battlefield simulations and Jennings' GRATE* [13] also applied in dynamic domains.

Peter Stone et al have proposed the use of "locker room agreements" [20,21] as a mechanism for defining pre-determined multi-agent protocols available for all the elements in the team [20]. They have used that mechanism to define a flexible teamwork structure including task decomposition and dynamic role assignment [20,21]. Their approach was implemented in the simulated robotic soccer team CMUnited [22] that won RoboCup world championships [8,9] RoboCup98 [1] and RoboCup99 [28]. Their team strategy is composed of formations using very simple protocols for switching between them (based on the result and time at each stage of the competition). Each formation assigns each agent a given role and protocols are suggested for enabling role exchange between the agents. The teamwork structure also includes set-plays, e.g. multi-step, multi-agent plans for execution in some situations. In this approach, roles may either be rigid or they may be somewhat flexible [21]. In their approach to robotic soccer roles correspond to a specific positioning in the field, like, for example a central defender.

Most implementations of multi-agent cooperation frameworks, rely on domain specific coordination. However, some relevant exceptions may be identified. ARCHON project [29], proposes a multi-agent cooperation system, in the domain of electricity transportation management, based on joint-intentions and on a general model of teamwork. Other example is Jennings' [13] joint responsibility framework, which is based on a joint commitment to the team's joint goal. His framework is implemented in the GRATE* system. GRATE* is a layered architecture in which the behaviour of an agent is guided by its mental attitudes, beliefs, desires, intentions and joint intentions. Agents are composed of two individual layers: a domain level system and a cooperation and control layer. In GRATE*, teamwork is executed when an organizer agent detects the need for joint action, becoming then the responsible for establishing the team and ensuring all member's commitments [13].

Other general model for teamwork is STEAM (simply, a Shell for Teamwork) [25]. STEAM is based in the joint intentions theory [14] but also on the SharedPlan theory [6,7]. STEAM uses joint intentions as the basis for teamwork but team members also build up a hierarchical structure of joint intentions, individual intentions and beliefs about the teammates intentions [25]. STEAM has been applied in several domains like the attack and transport domains that use an interactive commercial simulator developed for military training and RoboCup soccer server [3], in the context of the ISIS team. Although being far from a complete model of teamwork, STEAM attempt to bridge the gap from cooperation theory to its implementation is a remarkable one.

The trade-off between reactivity (essential to cope with the real-time, dynamic, noisy environment) and cooperation (needed to enable team joint behavior and to achieve overall team goals in the adversarial environment) is very difficult to achieve in the context of a general cooperation framework. In this paper we introduce a new approach to the coordination of a team of agents together with a method to balance reactivity and social behavior. The main innovations of our approach are:

- Balancing social behavior and reactivity through the distinction between active and strategic situations;
- Situation based strategic positioning – a policy used to position the agents in situations classified as strategic situations;
- Dynamic role and positioning exchange – enabling agents to switch roles (that define agent behaviors) and positionings (that define the places in a given formation);
- Formalization of what is a team strategy for a competition in partially cooperative and adversarial domains based on the concepts of tactics, formations and roles;
- A new agent architecture including a specific high level decision module capable of implementing the team strategy and supporting very flexible and efficient team performance.

The proposed approach is based on the definition of a team strategy using the concepts of tactics, formations and roles. Agents' decision making is based on a clear distinction between strategic and active situations. Based on this distinction, agents use, for strategic behaviour, Situation Based Strategic Positioning and, for active behaviour, domain specific high-level and low-level skills. To improve the flexibility of the team, agents are also able to switch their positions and specific behaviours (roles), at run-time, in the field. This mechanism called Dynamic Positioning and Role Exchange (DPRE) is based on previous work by Peter Stone et al [20,21] who suggested the use of flexible agent roles with protocols for switching among them. We have extended this concept and suggested that agents may exchange their roles (that correspond to agent types in our formulation) and their positionings in the current formation if the utility of that exchange is positive for the team. Moreover we propose a method to calculate that utility. Including DPRE in our robot soccer team implementation has significantly improved the overall team performance for teams of homogeneous agents and it may also be applied to heterogeneous agents.

This paper is organized as follows: Section 2 presents a formalization of a team strategy based on tactics, formations, positionings and agent types. The same section also describes Situation Based Strategic Positioning (SBSP) and Dynamic Positioning and Role Exchange (DPRE) mechanisms. Section 3 describes our agents' architecture and presents our high-level decision module control flow. Section 4 describes the application of our team strategy, SBSP and DPRE to the robotic soccer domain. The next section gives some experimental results. Finally, we present some conclusions as well as an outlook to future research we intend to do.

2 Team Strategy Formal Description

In our team coordination development we use the concepts of team strategy, role (agent behavior type), tactic, formation and positioning inside a formation.

Definition 1: *Team Strategy* is given by a set of *Tactics*, *Tactic Activation Rules*, a set of possible agent *Roles* and information concerning *Opponent Modeling Strategy*, *Teammate Modeling Strategy* and *Communication Protocols*.

$$TeamStrategy = (Tactics, TactActivationRules, Roles, OppModStrategies, TeammateModStrategies, CommunicationProtocols)$$

$$Tactics = \{Tactic_1, Tactic_2, ..., Tactic_{nTactics}\}$$

a set of tactics possibly applicable to the team

$$Roles = \{Role_1, Role_2, ..., Role_{nroles}\}$$

a set of roles (agent types) that define different agent behavior types

$$OppModStrategies = \{OppModStrat_1, OppModStrat_2, ..., OppModStrat_{nOppModStrategies}\}$$

a set of Opponent modeling strategies

$$TeammateModStrategies = \{TeamModStrat_1, TeamModStrat_2, ..., TeamModStrat_{nTeamModStrat}\}$$

a set of Teammate modeling strategies

$$CommunicationProtocols = \{ComProtocol_1, ComProtocol_2, ..., ComProtocol_{nComProtocols}\}$$

a set of communication Protocols

Tactics are used to supply the team with a set of *Formations* (see definition 3) that give the agents general strategic *Positioning* and *Role* information (agent individual characteristics). Tactics are selected according to *Tactic Activation Rules* that use G*lobal* information. For example if the competition is near the end, our team is losing it, but it is clear that it still has a chance of winning, then a more risky and aggressive tactic should be used. If it is clear by the competition statistics and opponent model information that the competition is surely lost then, a more defensive tactic, that tries to minimize the damage, may be the best one. For each *Tactic* a set of *Formations* and a set of *Preset Plans* are defined. These *Formations* and *Plans* are used in adequate situations.

Definition 2: A *Tactic* is defined by a set of *Formations*, *Formation Activation Rules*, and a set of *Predefined Plans*.

$$Tactic_i = (Formations_i, FormationActivRules_i, PresetPlans_i) \quad \forall i = 1..ntactics$$

Definition 3: A *Formation* is defined by the *Positioning* (which includes the *Reference Position*, *Role* and *Importance*) of the agents inside the *Formation*.

$$Formations_i = \{Formation_{i,1}, Formation_{i,2}, ..., Formation_{i,nformations_i}\}$$
$$\forall i = 1..ntactics$$
$$Formation_{i,j} = (AgentPositioning_{i,j,1}, AgentPositioning_{i,j,2}, ..., AgentPositioning_{i,j,nAgents})$$
$$\forall i = 1..ntactics, \quad \forall j = 1..nformations_i$$

Definition 4: A *Preset Plan* is defined by its *Plan Activation Information*, and the agents *Positioning evolution, Role evolution* and *Actions* along the time.

$$PresetPlans_i = \{PresetPlan_1, PresetPlan_2, ..., PresetPlan_{nplans_i}\}$$
$$\forall i = 1..ntactics$$
$$PresetPlan_{i,k} = (PlanActInfo_{i,k}, PlanAgentPositioningEvolution_{i,k},$$
$$PlanAgentRolesEvolution_{i,k}, PlanAgentActionsEvolution_{i,k})$$
$$\forall i = 1..ntactics, \quad \forall k = 1..nplans_i$$

Definition 5: The *Positioning* of an agent inside a formation is defined by its *ReferencePosition*, *PositioningRole* and *PositioningImportance*.

$$Positioning_{i,j,p} = (ReferencePosition_{i,j,p}, PositioningRole_{i,j,p}, PositioningImportance_{i,j,p})$$
$$\forall i = 1..ntactics, \quad \forall j = 1..nformations_i, \quad \forall p = 1..nplayers_{i,j}$$

For each positioning inside a given formation, a reference position is defined:

$$ReferencePosition_{i,j,p} = (ReferencePositionX_{i,j,p}, ReferencePositionY_{i,j,p})$$
$$ReferencePositionX_{i,j,p} \in \{-field\_length..field\_length\}$$
$$ReferencePositionY_{i,j,p} \in \{-field\_width..field\_width\}$$

This reference position is adjusted using the agent type and situation information to give that agent's strategic position at each time. Each positioning inside a formation has a *PositioningImportance*. For example, in the robosoccer domain, the central defenders are very important in defensive situations while the central forwards are very important in attacking situations. In a war scenario the home base last defenders are very important. This position importance is defined using a qualitative scale:

$$PositioningImportance_{i,j,p} \in \{VeryLow, Low, Medium, High, VeryHigh\}$$

Definition 6: *PositioningRole* defines the *Characteristics* of a given agent that occupies a positioning inside a given *Formation*. The *Roles* used as *PositioningRoles* in the *Formation* must be defined in the *TeamStrategy*.

$$PositioningRole_{i,j,p} \in \{1..nroles\}$$
$$\forall i = 1..ntactics, \quad \forall j = 1..nformations_i \quad \forall p = 1..nplayers_{i,j}$$

Roles may be used, for example in the RoboSoccer domain, to give agents specific characteristics like *Aggressive_Defender*, P*ositional_Defender*, P*ositional_Attacker*, etc.

Definition 7: A *Role* of the *Team Strategy* is defined by its *Active Characteristics, Strategic Characteristics* and by the *Critical Situation Rules*. The strategic state is abandoned to enter an active state if one of the *Critical Situation Rules* is true at a given time.

$$Role_i = (ActiveCharacteristics_i, StrategicCharacteristics_i, CriticalSituationRules_i)$$
$$\forall i = 1..nroles$$

Contrary to all the other concepts, *Role Strategic Characteristics* and *Active Characteristics* are domain dependent and must be defined accordingly. Figure 1 shows an example of this team strategy definition for an arbitrary domain.

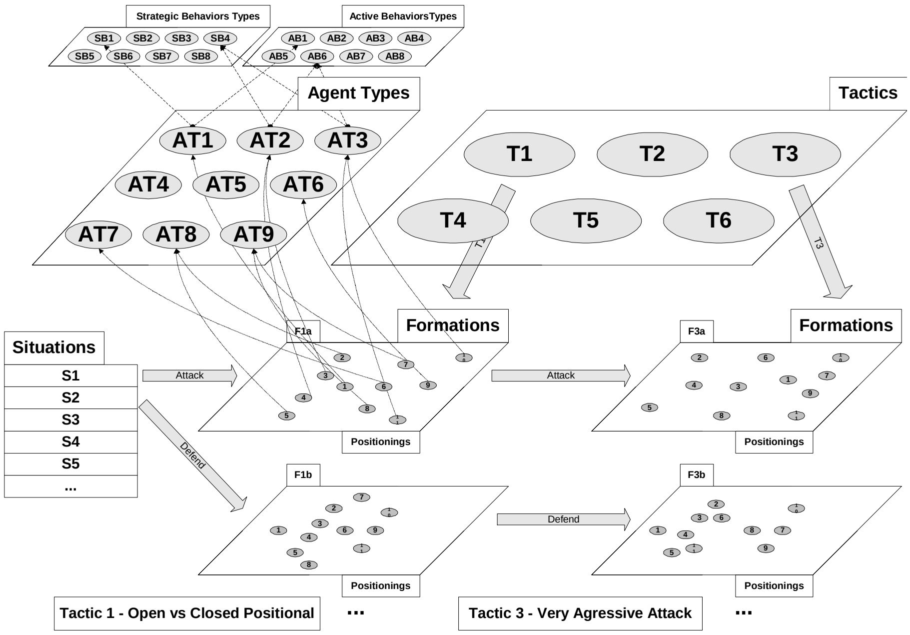

Fig. 1. Team strategy definition example

In Figure 1, the team strategy is very simple. It is composed by 6 tactics, each composed by several formations to be used in different situations (like attack, defend, etc.). Each formation uses different agent types for each of the different positionings. Each agent type is defined by its active and strategic characteristics.

3 Agent Architecture

We have chosen an agent architecture suitable for implementing our team strategy. This agent architecture is shown in figure 2. Agents include the traditional main control loop using perception interpretation and action prediction to update the world state, then deciding the appropriate action and finally executing the selected action. A high-level decision module is used to decide agent's current tactic, formation, role (agent behavior) and action at a given moment. Domain knowledge is structured in tactics, formations, agent roles, communication protocols, preset plans and game situations. These structures are predefined by the strategy designer according to the domain. Moreover, they are flexible enough to be fully instantiated before each competition by the team manager, according to each specific competition conditions, team capabilities and the opponent team foreseen characteristics. They may also be learned through both a training process and from other previous competitions.

Our agents build up a world model by interpreting the sensed information and predicting the results of the actions selected for execution. The main difficulties associated with this process are updating the world state effectively and choosing the appropriate action [20]. Communication also plays an important role here, and our agents use it for:

- Communicating their internal world states and situation information - keeping world representations more accurate;
- Communicate useful events (like a pass or a positioning exchange) in order to improve coordination.

3.1 Multi-Level World State Representation

World state representation includes information regarding several objects in the world and other high-level information like estimations about the final result of the competition, current time, opponent behavior, competition statistics, etc. We can separate this information in four levels of abstraction:

Global Information. High-level information needed to decide about the possible best team's tactic in a given moment. This information includes opponent team global behavior (quality, capabilities, aggressiveness, etc.) and high-level statistics from the competition (team's losses and successes while performing each specific collective action, etc);

Situation Information. Information that is relevant to the selection of the appropriate formation and for the situation based strategic positioning mechanism. This information is mostly concerned with the present and includes the formations of each team, field conditions, etc;

Action Selection Information. Information that is relevant to select an appropriate action like attacking and defending possibilities, moving options, interception possibilities, etc;

Physical State. Low level information, including the agent's state and the positions and velocities of the objects in the world.

Regarding the low-level information, there usually are several objects in the world which remain stationary, whose position is known and can be used for agents self-localization and there are several mobile objects whose localization must be continuously tracked. Each agent's internal world state stores an instantiation of all stationary as well as moving objects known. Sometimes, since world information is only partial, some objects although visible cannot be accurately identified. All moving objects have a representation of their locations and velocities stored with associated degrees of confidence values within interval [0,1]. The confidence values are needed because of the large portion of hidden world, which implies that the objects' positions stored are only estimations [20]. It is obviously a mistake to remember only objects that are currently in view, but it is also incorrect to assume that a mobile object will stay static (or continues moving with the same velocity) indefinitely. By decaying the confidence in unseen objects over time, agents can determine whether or not to rely on the current position and velocity values [2].

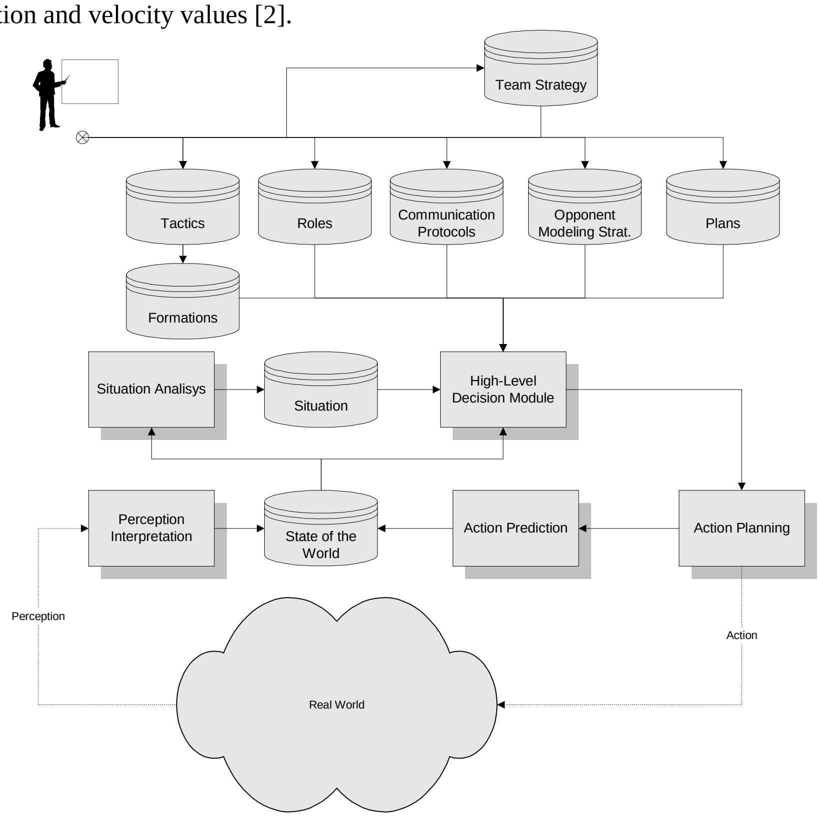

Fig. 2. Agent architecture

3.2 Situation Based Strategic Positioning with Dynamic Positioning and Role Exchange

We will now describe the two main agents positional coordination methods supplied by our strategy enabling the agents to cooperatively follow a given tactic and formation:

Situation Based Strategic Positioning (SBSP): This mechanism uses the situation information to select the best strategic position for each one of the players in the team. This strategic position depends on the current tactic and formation, the player role and positioning in the formation and the current situation.

Dynamic Positioning and Role Exchange (DPRE): This enables the agents to switch their positioning or their roles inside a given tactic and formation whenever that action leads to an improvement of the team global utility.

Each agent has an allocated positioning inside the current formation that changes dynamically with the competition specific situation. Due to the dynamic positioning and role exchange mechanism agents do not have fixed positioning inside the formation. For example, agent 2 can be at positioning 2 at a given time and at positioning 9 a few moments later. Also, positionings determine the agent's role (within the roles defined in the team strategy). Each role gives the agents different characteristics like tendency to be either positional or aggressive, tendency to be more offensive or defensive, etc. Changing positioning inside the formation also implies changing role. Of course in real applications this is not totally possible because a large part of the agent role characteristics depend on the physical and psychological characteristics of that agent. But if we have homogeneous agents in our team, it is rather straightforward and very useful to do this. For heterogeneous agents role adequacy must be also considered.

3.3 Situation Based Strategic Positioning

Our approach to building a cooperative team uses Situation Based Strategic Positioning enhanced with Dynamic Positioning and Role Exchange. As it was said before, this coordination policy for a team of agents is based on the distinction of Strategic and Active situations.

Definition 8: An *Active Situation* is a situation in which at least one *Critical Situation Rule* is fired for a given agent.

Critical Situation Rules are defined using Action Selection Information and take in account the present *Situation* (attack, defend, etc.).

Definition 9: A *Strategic Situation* is a situation in which an agent doesn't have any *Critical Situation Rule* fired.

If an agent is not involved in an active situation then it tries to occupy its strategic positioning that changes according to the situation of the game. Game information includes the competition time and result, opponent modeling information, competition mode, several statistics, attack and defensive information, positions, states and velocities of the objects in the world, etc. This information is complemented with other situation-based information like the plan currently in execution. Predefined strategic information includes several tactics, formations (for game situations), and agents' behaviors inside formations. Strategic behaviors, used in strategic situations, enable the team with a social behavior by allocating different tasks to different agents using a global perspective. Strategic behaviors provide the team with an efficient coverage of the whole field. The positions of agents in strategic mode maximize the options for cooperation with the agents in active mode. While agents in active mode use their specific domain knowledge to decide their action in a reactive decision mechanism, players in strategic mode fulfill the tasks allocated by the formation in use, by covering different sections of the field in the best possible way.

Definition 10: The *Situation Based Strategic Positioning of each agent* is a function of the current *Tactic*, *Situation* (that define the *Formation* in use) and *Agent Type* (that define the Agent Strategic Characteristics).

The situation information associated with the predefined strategic information is used to estimate the agent's own strategic position for each situation. It is also used to estimate his teammates strategic position for each situation. Knowing its own strategic position and the teammates strategic position, agents may decide if an exchange of positioning with a teammate is beneficial or not to the team and, therefore, if it is useful or not to perform dynamic positioning and role exchange.

3.4 Dynamic Positioning and Role Exchange

Our proposal includes the use of utility functions in order to evaluate usefulness of a positioning or role exchange.

Definition 11: A *Dynamic Positioning and Role Exchange* consists of the exchange (inside a given tactic and formation) of the *Positioning* and *Role* (including all agent behavior characteristics) between two agents.

As part of the world state representation, each agent has an estimate of the other agent's positions in the field:

$$AgentPositions = \{AgentPos_1, AgentPos_2, ..., AgentPos_{nplayers}\}$$

$$\forall p = 1..nplayers$$

$$AgentPos_p = (AgentPosX_p, AgentPosY_p)$$

Dynamic Positioning and Role Exchange aims to improve the performance of a homogeneous team of cooperating agents. At each time in the course of the competition, each agent has a positioning (place in the formation) and a role (behavior characteristics) inside the current active formation. Each agent keeps track of his positioning but also of the positioning allocated to each of the other agents in the team:

$$AlocAgentPositionings = \{AlocPositioning_1, AlocPositioning_2, ..., AlocPositioning_{nplayers}\}$$

$$AlocPositioning_p \in \{1..nplayers\}$$

If the team is composed of homogeneous agents each agent can carry out another role and occupy other positioning without any additional problem. Since the team is composed of cooperative agents, this positioning and role exchange takes place only when the agents can improve the global team utility by doing so. This utility is computed through the following formula:

$$\begin{aligned} DPREUtility(tact, form, p_a, p_b) = \\ &+ Utility(Position_{p_a}, Positioning_{tact,form,p_b}, PlayerRole_{tact,form,p_b}) \\ &+ Utility(Position_{p_b}, Positioning_{tact,form,p_a}, PlayerRole_{tact,form,p_a}) \\ &- Utility(Position_{p_a}, Positioning_{tact,form,p_a}, PlayerRole_{tact,form,p_a}) \\ &- Utility(Position_{p_b}, Positioning_{tact,form,p_b}, PlayerRole_{tact,form,p_b}) \\ &\forall p_a = 1..nplayers_{tact,form}, \forall p_b = 1..nplayers_{tact,form}, p_a < p_b \end{aligned}$$

If the DPRE Utility for a given pair (p_a,p_b) is positive, then the Dynamic Positioning and Role Exchange takes place, that is, each player assumes other one's positioning and role:

$$\begin{cases} AlocatedPlayerPositioning_{tact,\,form,\,p_a} = AlocatedPlayerPositioning_{tact,\,form,\,p_b} \\ AlocatedPlayerPositioning_{tact,\,form,\,p_b} = AlocatedPlayerPositioning_{tact,\,form,\,p_a} \end{cases}$$

The utility functions for each player positioning and role are concerned with the players' current position distance to his strategic position, the positioning importance, agent's states (physical conditions, objects carried, etc.) and the role characteristics. For heterogeneous agents the utility function must also take into account the agent's adequacy to perform each of the roles.

Because agents have local perspectives, their own utilities for a given DPRE exchange may be different and so, one of the agents may believe that the exchange is useful while the other believes the opposite. To deal with this problem, communication between the agents shall be used to synchronize the exchange. This situation is not very problematic because when one of the agents believes that a possible exchange has a positive utility, he will try to perform the exchange and communicate the teammate that fact. Meanwhile, the agent will go to the teammate position, which tends to increase the value of the estimated utility for this exchange for both agents.

3.5 High-Level Decision Module Control

We are not particularly concerned with the low-level skills and world state accuracy available to the individual agent but first and foremost we are concerned with its high-level intelligent decision capabilities and social abilities (coordination and communication). Therefore, the main features of our agents are included in the high-level decision module. Figure 3 depicts control flow inside that module which will be explained next.

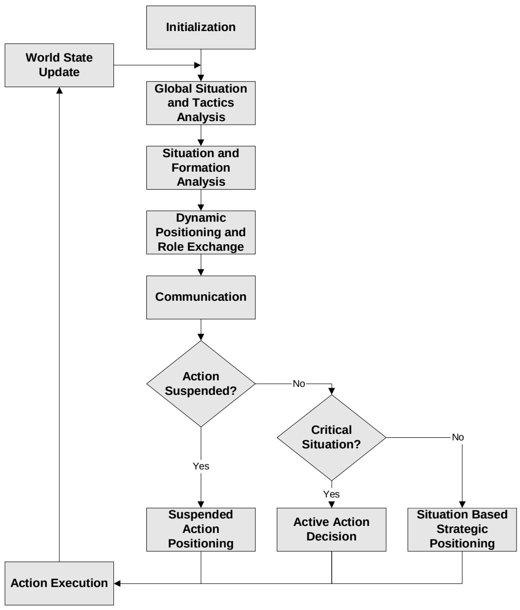

Fig. 3. Control flow of the high-level decision module

After the initialization of some internal structures that includes reading several configuration files, the agent enters the main control loop. This loop starts by running the tactic decision module in which the agent uses global situation information to decide the current tactic. This module is directly connected to the second module of the main control loop, which is concerned with situation analysis and formation selection. The control loop follows with dynamic positioning exchange and communication analysis.

If the competition is stopped (for example after a foul in a soccer game or a cease fire in a battlefield scenario), stopped action positioning is performed; if not, then if a critical situation is not identified, meaning that the agent does not need to get immediately into active behavior, global situation based strategic positioning is performed. Whenever the situation is critical, then an appropriate action is selected. The loop closes with internal activities not directly connected with decision-making, that is, action execution and world state updating.

4 Application of the Team Strategy to RoboCup

4.1 RoboCup Simulation League and the Soccer Server

Our formalization has been applied to the RoboSoccer simulated domain in the context of the FC Portugal simulated soccer team [17]. The application domain is based on the RoboSoccer server [3,15,16] that has been used as the basis for the RoboCup international competitions and for several associated research challenges [10]. The soccer server is a very complex and realistic domain. Unlike many discrete and accessible traditional Artificial Intelligent domains (like Chess or Checkers), soccer server comprises several real world complexity factors like asynchronous sensing and acting. The simulation comprises characteristics from robotic systems combined with characteristics from real human soccer players and soccer matches. The server's sensor and actuator noise models are inspired by typical robotic systems [20], while many other characteristics, such as limited stamina, movement and vision, are motivated by human limitations. This is indeed a very rich domain for the study of multi-agent real-time coordination and communication. Each team is composed of eleven software agents that must act autonomously, with limited perception, action and communication abilities and that have to collaborate in a real-time, noisy, adversarial environment to achieve common goals. Teams have two contradictory goals:

To score on the opponents goal. To accomplish this, players shall advance in the field towards that goal line;

To prevent the opponent’s from scoring in their own goal. To achieve this, players must cover the way to their own goal and shall not advance too much in the field.

Reactive behavior is essential in RoboCup games because of the real-time dynamics of the environment. However, the agent’s contradictory goals and the complexity of the environment, also demand an intelligent social behavior. Balancing both these approaches however it is not an easy task but is probably one of the major challenges in the RoboCup simulation league. To be successful teams must react very fast to changes in the environment (like ball velocity changes) but on the other hand, must also perform very complex collective moves (like organized joint attacks, keep a defensive line, etc.).

4.2 Team Strategy for the RoboSoccer Domain

To apply our team strategy formalization and SBSP to the RoboSoccer domain, we need to define specific *Critical Situation Rules*, *Agent Strategic Characteristics* and *Agent Active Characteristics*. *Critical Situation Rules* (CSRs) are used to identify situations in which strategic positioning must be abandoned and an active mode must be entered. Each *Role* that an agent can play has associated a set of *Critical Situation Rules.*

$$CriticalSituationInfo_i = \{CriticalSituationRule_{i,1}, CriticalSituationRule_{i,2}, ..., CriticalSituationRule_{i,ncriticalrules}\} \quad \forall i = 1..nroles$$

For each agent role, its Strategic Characteristics include Ball Positional Attraction that is used to adjust player's strategic positions towards the region where the ball is, and a positional rectangle that strategically the players should not leave (although they may leave it in active situations). For some roles (namely defenders and goal keeper) it is needed to stay always behind the ball x-coordinate. In some specific field regions, the players' strategic positions should be more attracted towards the position of the ball. This is the case of the forward players when the ball is near the opponent's goal line, or the defenders when the ball is near their own penalty area.

$$StrategicCharacteristics_i = (BallPositionalAttraction_i, PositionalRectangle_i, BehindBall_i, BallAttraction_i, BallAtractionRegion_i) \quad \forall i = 1..nroles$$

Active Characteristics are concerned with ball possession and ball recovery functionalities. Ball possession functionalities rely on characteristics that are divided into four main groups dealing respectively with passing, forwarding, shooting and dribbling. Ball recovering functionalities depend on the abilities for intercepting the ball, marking an opponent that has the ball, marking an opponent without ball, covering the goal from an opponent, obstructing an opponent, covering a pass line, get free to receive a pass and prepare the reception of a pass. Each ball possession and ball recovery ability is composed of an evaluation rule and a behavior. Evaluation rules are used in order to choose the best possible ability to use at a given moment.

As it was said before, information is used at different levels of detail. *Global Information* refers to high-level information of the game and is used to decide the appropriate *Tactic* to use. In the RoboSoccer specific application, this information may include: information about time, current result, game statistics (number of shoots, corners, successful passes, ball possession time, etc.) and opponent modeling information (which includes the opponent team tactic, formation and players' roles, positions and characteristics).

Situation Information is used (along with *Global Information*) for a dynamic selection of the *Formation* to use. This information includes *Ball Possession Information* (ball possessor, confidence, team in possession, time of possession, etc.), bll information (position, velocity) and player's information (teammates and opponents).

$$GlobalSituation = (Time, Result, StatisticsInfo, OppModInfo)$$
$$Situation = (BallPossesionInfo, BallInfo, TeammateInfo, OpponnentsInfo)$$
$$StatisticsInfo = (BallPossessionInfo, ShootInfo, PassInfo, DribbleInfo, CornersInfo, OffSideInfo, FreeKickInfo, KickInInfo, GoalieKickInfo)$$
$$PositionalInformation = (BallLowLevelInfo, OffSideLines, TeammateLowLevelInfo, OpponentLowLevelInfo, InterceptionInfo, CongestionInfo, DangerInformation)$$
$$ActionInformation = (PassInformation, ForwardInformation, ShootInformation, DribbleInformation, ...)$$

Statistics Info is concerned with game statistical information. This includes, for each one of the teams, statistics concerning ball possession time by field region (for each team), shooting statistics (number of shoots, their characteristics and results), statistics about passing (number of passes, their characteristics and results) and the same type of statistics for forwards and dribbles. Statistics for stopped game situations are also included for corner, offside, free kick, kick in and goalie kick.

4.3 Situation Based Strategic Positioning for RoboSoccer

Based on the player allocated positioning and on the role allocated by the formation to that specific positioning, the player's strategic position in each situation is calculated using the following process:

$$AlocRole_p = PlayerRole(AlocPositioning_p)$$
$$PlayerStrategicPosition_p = AdjustedPosition(ReferencePosition(AlocPositioning_p) + BallPosAttraction(AlocRole_p, BallPos, BallPosAttrac, PosRectangle) + BallAttraction(AlocRole_p, BallPos, BallAttraction, BallAttractionlRegion)), BehindBall(AlocRole_p), OffsideConversion(AlocRole_p))$$

The player strategic position is calculated as an adjusted position (using the *behind ball* and *offside* conditions) of the sum of the reference position (that depends on the tactic and situation) with the ball positional attraction and ball attraction in specific regions of the field. The factors used to adjust this reference position depend on the agent type and their definition consists basically in defining the agent type strategic characteristics.

Contrarily to other positional mechanisms that do not use tactics or agent types, like Stone's SPAR [20], SBSP enables a very flexible positioning of soccer players. SBSP enables the team to have completely different shapes for different situations. For example, the team may have a very compact shape in a defend situation and spread in the field on an attack situation (just like a real soccer teams). It also enables the team to be correctly positioned in situations like goal scoring opportunities (in which players must assume possible shooting positions inside the opponent's area). Finally it enables different players (that have different agent types) to have completely different positional behaviors.

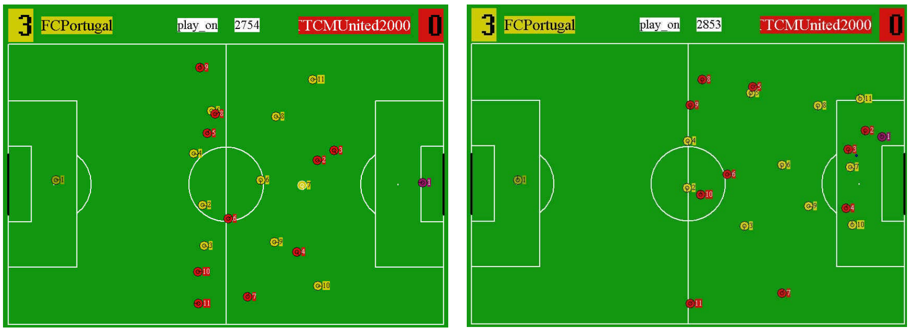

Fig. 4. Situation Based Strategic Positioning in an attack situation and in a goal scoring opportunity situation in RoboCup 2000 game against ATTCMU2000.

SPAR [20] is based on attractions (to active teammates, ball and opponent's goal) and repulsions (from opponents and passive teammates) and does not consider situations or different positional behaviors for different player types. Thus players are unable to have this flexibility. This difference is quite visible in games between FC Portugal and CMUnited99 and was also quite visible in the game between FC Portugal and ATTCMU2000 in RoboCup2000 (Fig. 4).

Strategically players should keep the position determined by situation based strategic position. But because agents are also involved in active behaviors, their real positions in the field may differ considerably from the strategic positions. For these situations the DPRE mechanism is very useful to keep the formation shape.

4.4 Team Strategy Practical Application

A team strategy includes several different tactics and conditions for the activation of those tactics. An example is a very simple team strategy that uses only three tactics: Normal 433, Aggressive 442 and Very Aggressive 235. The rules for selecting the appropriate tactic could be that in the beginning of the game or if winning, the 433 tactic would be selected. If the game is a draw after the game interval or if losing by 1 goal, the Aggressive 442 would be selected. If the game is a draw near the end or if losing by more than one goal, the Very Aggressive 235 tactic would be selected.

Each one of the tactics uses several formations composed of different players with different roles. Typical formation types are used for defending (when the opponents have control of the ball) and for attacking (when the team has ball possession). Other types of formations could be used for transporting the ball from defense to attack, for goalie free kick, corners, etc. A simple example for the first tactic is the use of only two formations: a simple 433 closed formation for defending and a simple 433 open formation for attacking (as illustrated in Figure 5).

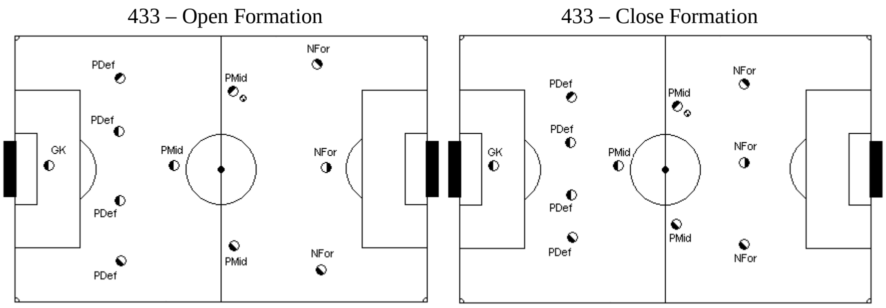

Fig. 5. Typical 433 Formations: In the open formation, the players are more spread over the field. In the close formation (used for defending) players are more concentrated in the middle of the field. Player roles are also represented (GK-Goal Keeper, PDef-Positional Defender, PMid – Positional Midfielder and NFor –Normal Forward).

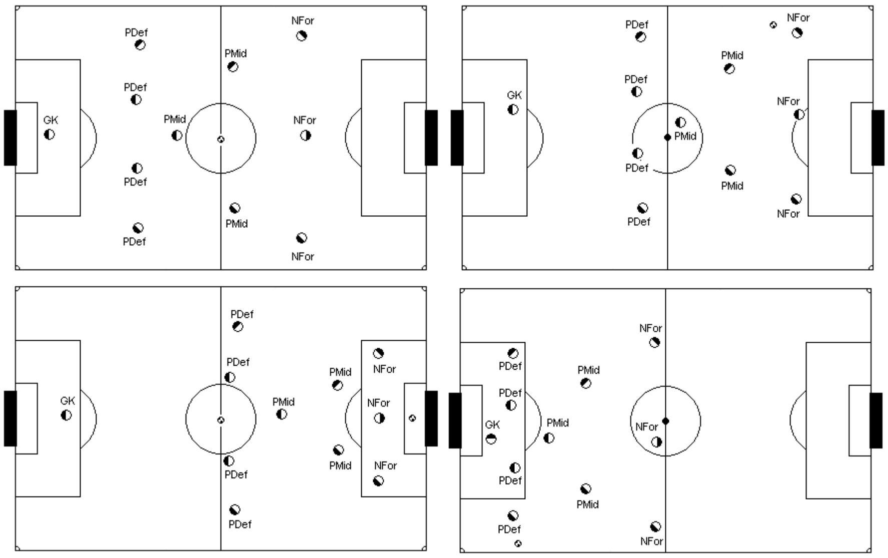

Fig. 6. Illustration of the Situation Based Strategic Positioning concept, considering only the ball positional attraction and ball attraction parameters. In the top left figure, the ball is in the center of the field, on the top right figure the ball formation positional attraction is shown. In the bottom left figure, the formation is attracted by the ball but also, since the ball is in a ball attraction region, forwards and two of the midfielders are directly attracted by the ball. The bottom right figure shows a similar situation with the ball on the defensive side of the field.

Positions of the players inside the formations and their respective roles are different depending on the tactic and on the game current situation. The formations shown in Figure 5 assume that the ball is on the middle of the field and that all players are in a strategic behavior. But, like it was said before, the strategic positioning of the players is “situation based” and depends on the tactic and formation in use (and the player role in that formation as well) plus several situation parameters (Figure 6).

As one can see in figure 6, the ball represents a crucial part of the situation in the SBSP mechanism. The formation is attracted by the ball, adjusting itself to the game situation. If the ball is in a Ball Attraction Region (like the opponents area), several players are attracted directly towards the ball as it is illustrated in the bottom left image in Figure 6. Besides the ball attraction, players, depending on its role definition, may stay always behind the ball and may avoid staying in an offside position. Using this formation definition mechanism, we can define players with totally different behaviors. For example, giving a player a high positional attraction in the x-axis makes the player perform long runs along the field accompanying the ball movement along the field. Giving the player a role with a very high ball attraction and setting the ball attraction region for that role to be the whole field, configures a player that always goes towards the ball position. This kind of players can be very annoying to some opponents.

5 Experimental Results and Discussion

Our agents' coordination proposal has been implemented in the FC Portugal RoboCup simulation league team. The team was implemented using a world state update model and low level skills based on CMUnited99 publicly available source code [22]. Modifications were performed on the interception, kicking and dribbling abilities of the players [17]. FC Portugal also uses intelligent perception and communication [17] and integrates soccer knowledge in the high-level decision modules for ball possession and ball recovery [17,18].

FC Portugal defeats very easily all the teams that have competed in Stockholm for RoboCup99. Table 1 summarizes the results achieved[1] in a series of 10 games against eight of the most well known teams that competed in Robocup99. The results were achieved using always a simple team strategy based on a 433 positional tactic (used when the team is wining or drawing the game) and a 442 aggressive tactic (used when the team is losing). Both tactics included only two situations (defending and attacking) and used open formations for attacking and closed formations for defending. The team used "positional agent types" very similar to the ones used by FC Portugal in Melbourne RoboCup 2000.

Teams	Wins	Draws	Losses	Score	Mean
CMUnited99 (U.S.A)	10	0	0	113-0	11.3-0.0
Magma Freiburg 99 (Germany)	10	0	0	144-0	14.4-0.0
Essex Wizards99 (England)	10	0	0	161-0	16.1-0.0
11 Monkeys 99 (Japan)	10	0	0	238-0	23.8-0.0
Mainz Rolling Brains99 (Germany)	10	0	0	209-0	20.9-0.0
Brainstormers99 (Germany)	10	0	0	146-0	14.6-0.0
Cyberoos99 (Australia)	10	0	0	254-0	25.4-0.0
Zeng99 Acamp (Japan)	10	0	0	244-0	24.4-0.0
Total	100	0	0	1509-0	18.9-0.0

Table 1. FC Portugal scores against RoboCup99 teams.

[1] The results were achieved using three separate machines running Linux RedHat 6.1 (one for each team and one for the server) connected through a 10Mbit network. Pentiums 550MHz, 128 MB were used for the server and for FC Portugal team and a Pentium 550MHz with 256MB of memory was used for the opponent team.

The results clearly show that FC Portugal could easily beat all those opponents. The team was capable of scoring an average of more than 11 goals to the previous undefeated champions CMUnited 99 [23]. Also against other very good teams, like the second (Magma Freiburg [5]) and third (Essex Wizards [12]) places of RoboCup 99, FC Portugal team was able to score many goals.

In the RoboCup 2000 European and World championships, these results were confirmed. Although competing with well established teams, and being a very recent team, FC Portugal won the European RoboCup 2000, held in Amsterdam (May 29-June 2), and the World RoboCup 2000, held in Melbourne (August 28 - September 3). In these two competitions, FC Portugal scored a total of 180 goals, without conceding a single goal. Table 2, summarizes the results achieved by FC Portugal in both competitions.

Euro RoboCup – Amsterdam	Score
Essex Wizards (England)	3 – 0
Lucky Luebeck (Germany)	13 – 0
Cyberoos (Australia)	4 – 0
Pizza Tower (Italy)	22 – 0
Polytech (Russia)	19 – 0
PSI (Russia)	6 – 0
Wroclaw (Poland)	13 – 0
Essex Wizards (England)	5 – 0
Karlsruhe Brain. (Germany)	2 – 0
Total Score	86 – 0

RoboCup 2000 - Melbourne	Score
Oulu2000 (Finland)	33 - 0
Zeng2000 (Japan)	18 - 0
Robolog (Germany)	20 - 0
Essex Wizards (England)	7 - 0
Karlsruhe Brain. (Germany)	3 - 0
YowAI (Japan)	6 - 0
ATTCMU2000 (U.S.A)	6 - 0
Karlsruhe Brain. (Germany)	1 – 0
Total Score	94 - 0

Table 2. Scores of FC Portugal in EuroRoboCup2000 (Amsterdam) and RoboCup2000 (Melbourne).

In Amsterdam FC Portugal used almost unchanged CMUnited99 basic skills [22] (kicking, dribbling, interception, etc.) but the team was able to beat other teams with much better low-level skills (e.g. Essex Wizards [11] with a better dribbling ability, Karlsruhe Brainstormers [19] with a more powerful kick). The main reasons for this (apparently strange) success were, at the social behavior level:

Team strategy based on very flexible tactics with well-conceived formations for different game situations and flexible player types;

Situation based strategic positioning that enabled the team to move in the field as a real soccer team;

Dynamic positioning and role exchange mechanism that enabled the team to keep higher levels of stamina and a reduced number of useful positions uncovered;

Associated with this, the individual decision capabilities of the agents (ball possession and ball recovery decision modules) were also very important.

In Amsterdam FC Portugal used the SBSP mechanism combined with the DPRE and individual decision modules to explore the free space on the field to attack and to cover that same free space while defending. Against teams that had good positional systems this playing type revealed to be less effective and FC Portugal could not score many goals.

In Melbourne, several teams used positioning mechanisms similar to SBSP, (mostly the ones that competed in Amsterdam, with whom FC Portugal played half of its games). This way, the positional advantage, FC Portugal had in Amsterdam, was only totally decisive in games against not very strong teams. In the games against very good teams, FC Portugal superiority was also related, not only with the team strategy, SBSP and DPRE but also with the intelligent perception and communication mechanisms [17], strong kick based on optimization techniques [17], marking techniques and debugging tools used [17].

Teams	Place	W	D	L	Score
Brainstormers2K (Germany)	2nd	10	0	0	24-0
ATTCMU2000 (U.S.A)	3rd	10	0	0	71-0
CMUnited99 (U.S.A)	4th	10	0	0	113-0
Essex Wizards2000 (England)	7th	10	0	0	68-0
Cyberoos2000 (Australia)	9th	10	0	0	234-0
Robolog2000 (Germany)	13th	10	0	0	168-0
Total		60	0	0	678-0

Table 3. Scores of FC Portugal in series of 10 games against some of the best RoboCup 2000 teams. Place indicates the ranking of these teams in RoboCup 2000, while W, D and L, stand respectively for wins, draws and lost games. Score represents the combined score of 10 games.

Table 3, shows FC Portugal scores against six of the best teams in RoboCup 2000[2] [24] and Table 4 shows FC Portugal results turning off the SBSP or/and the DPRE mechanisms. Turning off the SBSP mechanism makes the agents assume always an active behavior. To implement this, a new critical rule that selects the best ball recovery behavior to perform in a given situation was added. This way, if no other critical situation rule is activated, the agent selects between the available ball recovery behaviors and executes the best one. Ball recovery behaviors considered include interception, passive interception, prudent interception, mark opponent, mark pass line, go to ball position, and get free from opponents.

The results show that turning off the SBSP mechanism makes the team perform much worse. The team is unable to win all the games and the games against Karlsruhe Brainstormers are now very tied (although Karlsruhe has still many problems in scoring). Against CMUnited99, results are also rather worse. Against this team, losing the positional advantage makes the game much more tied and FC Portugal loses the pressing capability over CMUnited. The result is that the team is unable to score many goals and CMUnited99 has some scoring chances.

---

[2] Unfortunately it is not possible to present results against YowAI2000 and Magma Freiburg that were fifth in RoboCup 2000, because their binaries are still not available.

	Without SBSP				Without DPRE				Without SBSP and DPRE			
Teams	W	D	L	Score	W	D	L	Score	W	D	L	Score
Brainstormers2K (Germany)	5	5	0	8-2	8	2	0	12-1	3	6	1	6-2
ATTCMU2000 (U.S.A)	9	1	0	34-0	10	0	0	56-0	10	0	0	28-0
CMUnited99 (U.S.A)	10	0	0	49-0	10	0	0	85-0	10	0	0	38-0
Essex Wizards2000 (England)	10	0	0	39-0	10	0	0	62-0	9	1	0	44-0
Cyberoos2000 (Australia)	10	0	0	108-0	10	0	0	184-0	10	0	0	96-0
Robolog2000 (Germany)	10	0	0	111-0	10	0	0	142-0	10	0	0	98-0
Total	54	6	0	349-2	58	2	0	541-1	52	7	1	310-2

Table 4. Scores of FC Portugal with and without SBSP and DPRE against several RoboCup 2000 teams.

Turning off the DPRE mechanism does not affect much the results against Essex Wizards. Analyzing the games, the explanation seems to be that Essex Wizards have a very slow type of game and so dynamic role and positioning exchange is not essential.

Tests were also performed using FC Portugal without both mechanisms against the best RoboCup2000 teams. The team is still superior to its opponents but has now lots of difficulties to overcome Brainstormers (conceding two goals and losing a game 1-0). Brainstormers are able to dominate the game and FC Portugal has lots of difficulties to enter Brainstormers middle field. However, FC Portugal communication strategy, intelligent perception, very good decision mechanisms (based on real soccer knowledge) and excellent goalkeeper, are still sufficient to win most of the games.

6 Conclusions and Future Work

In this paper we have presented a new approach to build cooperative teams of agents performing in dynamic real-time adversarial environments, based on a formalization of what should be a team strategy, a situation based strategic positioning mechanism, a dynamic positioning and role exchange mechanism and a new agent architecture. Following these guidelines we have selected the RoboSoccer simulated domain as our test bed and implemented a new RoboSoccer team – FC Portugal – that uses this team strategy. FC Portugal has shown the usefulness of the proposed approach through the results achieved in RoboCup2000, becoming the undefeated European and World champion, scoring 180 goals without conceding a single goal.

One of our main contributions in this paper is the formalization of the team strategy concept. As shown in section 2, a large part of this formalization can be applied to any domain with spatially distributed agents and any kind of team composed of homogeneous agents. In our formalization a team strategy is composed of a set of tactics and a set of possible agent types (defining agent behaviours at several levels). Each tactic is composed of several formations that are applied in different situations. Each formation is a flexible distribution of the agents in the field and has an assignment of agent types to each agent. The strategic position of the agents in the field is also dependent on the situation and of the positioning assigned in that formation to each agent. This nested strategic and tactic knowledge representation enables the development of very flexible teams, capable of assuming social behaviour (in strategic situations) and a more reactive behaviour (in active situations). Moreover the distinction between strategic and active situations gives the agents more sophisticated means of achieving global coordination of the team. When the agent is not on a critical situation (one in which he will be engaged in active behaviour soon), it assumes that it is in a strategic situation. In this situation it tries to position itself in the best possible strategic position.

It is also significant that, in our teamwork coordination approach, there is no conflict between simple agent reaction and deliberation about team coordination. In fact, situation-based analysis other than just trigger individual agent's actions, also induces appropriate tactical changes, leading all the team to adopt a new formation and therefore maintaining some sort of team coherence and coordination adjusted to the new situation. This reactively oriented change of global coordination framework (formations used for specific situations) prevents agents from performing isolated from the rest of the team while paying attention and reacting to the situation dynamics.

Our Dynamic Positioning and Role Exchange mechanism is another contribution of this work. This policy enables a better use of the team resources by exchanging their positioning and role. If, in a given situation, it is better for the team global utility that two agents switch places, the exchange is then performed. However, since the agents are homogeneous (except for the goal keeper), a role (agent type) exchange can also be performed (each agent is equally good in each one of the possible roles). This is useful to keep the formation characteristics. For heterogeneous agents a similar mechanism can be adopted extending this framework with an agent vs role capability matrix and changing the DPRE utility functions in accordance.

The results achieved by FC Portugal show the usefulness of the proposed team strategy, agent architecture and the adequacy of SBSP and DRPE in the context of the simulated RoboSoccer domain. Future work will be concerned with the application of this framework to other domains and its extension to heterogeneous agents.

Acknowledgments

The authors would like to thank to Peter Stone, Patrick Riley and Manuela Veloso for making available the low-level source code of their CMUnited99 simulation league RoboCup team. This saved us a huge amount in testing our approach in the framework of the FC Portugal project. We would also like to thank the financial support for this work of the FCT - Portuguese Foundation for Science and Technology, Compaq Portugal, PT Innovation, University of Aveiro, LIACC – Artificial Intelligence and Computer Science Laboratory and APPIA – Portuguese Association for Artificial Intelligence. Our thanks goes also to the anonymous reviewers that through their very detailed reviews contributed to increase the quality of this paper.

References

[1] M. Asada and H. Kitano, editors. *RoboCup-98: Robot Soccer World Cup II*. Springer, Lecture Notes in Artificial Intelligence, 1999
[2] Mike Bowling, Peter Stone, and Manuela Veloso. Predictive memory for an inaccessible environment. *In Proceedings of the IROS96 Workshop on RoboCup*, pages 28--34, Osaka, Japan, November 1996. (p. 68)
[3] Emiel Corten et al. Soccerserver Manual, Version 5 rev 00 beta, at URL http://www.dsv.su.se/~johank/ Robocup/ manual, July, 1999
[4] Keith Decker. Task Environment Centered Simulation. In M. Prietula, K. Carley, and L. Gasser, editors, *Simulating Organizations: Computational Models of Institutions and Groups.* AAAI Press/MIT Press, 1996
[5] Klaus Dorer, The Magma Freiburg Soccer Team, in Manuela Veloso, Enrico Pagello and Hiroaki Kitano, editors. *RoboCup-99: Robot Soccer World Cup III*. Springer, Lecture Notes in Artificial Intelligence, 2000
[6] B. Grosz,. Collaborating systems. *AI magazine*, 17 (2), 1996
[7] B. Grosz, and S. Kraus. Collaborative plans for complex group actions. *Artificial Intelligence*, 86, 269—358, 1996
[8] Hiroaki Kitano, M. Asada, Y. Kuniyoshi, I. Noda and E. Osawa. Robocup: The Robot World Cup Initiative. In *Proceedings of IJCAI95 Workshop on Entertainment and AI/Alife*, 1995
[9] H. Kitano. RoboCup: The Robot World Cup Initiative, *Proceedings of the 1st International Conference on Autonomous Agent (Agents97)*, Marina del Ray, The ACM Press, 1997.
[10] Hiroaki Kitano, M. Tambe, P. Stone, M. Veloso, I. Noda, E. Osawa and M. Asada. The Robocup Synthetic Agents' Challenge. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997
[11] H. Kitano, et al., RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research, *Proceedings of IEEE Conference on Man, Systems, and Cybernetics (SMC-99)*, 1999
[12] Kostas Kostiadis and H. Hu. A Multi Threaded Approach to Simulated Soccer Agents for the RoboCup Competition, *IJCAI'99 workshop on RoboCup*, 1999.
[13] N. Jennings. Controlling cooperative problem solving in industrial multiagent systems using joint intentions. *Artificial Intelligence*, 75, 1995
[14] H. J. Levesque, P. R. Cohen and J. Nunes. On acting together. *In Proceedings of the National Conference on Artificial Intelligence.* Menlo Park, California, AAAI press, 1990
[15] Itsuki Noda. Soccer Server: A Simulator of Robocup. In *Proceedings of AI symposium '95 Japanese Society for Artificial Intelligence*, pp. 2934, 1995
[16] Itsuki Noda, Hitoshi Matsubara, Kazuo Hiraki, and Ian Frank. Soccer server: A Tool for Research on Multiagent Systems. *Applied Artificial Intelligence*, Vol. 12, pp.233-250, 1998
[17] Luis Paulo Reis and Nuno Lau, FC Portugal Team Description: RoboCup 2000 Simulation League Champion, *In Peter Stone, Tucker Balch and Gerhard Kraetzschmar, editors, RoboCup-2000: Robot Soccer World Cup IV*, Springer, Berlin, 2001, to appear
[18] Luis Paulo Reis and Nuno Lau, FC Portugal Approach Overview, Accessible from http:/www.ieeta.pt/robocup/overview.htm, November of 2000

[19] Martin Riedmiller et al. Karlsruhe Brainstormers 2000 - A Reinforcement Learning approach to robotic soccer. *Proceedings of the Fourth International Workshop on RoboCup*. Melbourne, August 2000

[20] Peter Stone. *Layered Learning in Multi-Agent Systems*. PhD Thesis, School of Computer Science, Carnegie Mellon University, 1998

[21] Peter Stone and Manuela Veloso. Task Decomposition, Dynamic Role Assignment, and LowBandwidth Communication for RealTime Strategic Teamwork. *Artificial Intelligence*, 110 (2), pp.241-273, June, 1999.

[22] Peter Stone, Patrick Riley and Manuela Veloso. CMUnited-99 source code, 1999. Accessible from http://www.cs.cmu.edu/ ~pstone/RoboCup/CMUnited99-sim.html.

[23] Peter Stone, Patrick Riley and Manuela Veloso. The CMUnited-99 Champion Simulator Team, in Manuela Veloso, Enrico Pagello and Hiroaki Kitano, editors. *RoboCup-99: Robot Soccer World Cup III*. Springer, Lecture Notes in Artificial Intelligence, 2000

[24] Peter Stone, Tucker Balch and Gerhard Kraetzschmar, editors, RoboCup-2000: Robot Soccer World Cup IV, Springer, Berlin, 2001, to appear

[25] Milind Tambe, Towards Flexible Teamwork, *Journal of Artificial Intelligence Research* 7, pp. 83-124, 1997

[26] Milind Tambe. Implementing Agent Teams in Dynamic Multi-Agent Environments. *Applied Artificial Intelligence*, vol. 12, 1998

[27] Manuela Veloso, Michael Bowling, Sorin Achim, Kwun Han, and Peter Stone. The CMUnited98 Champion Small Robot Team. In Minoru Asada and Hiroaki Kitano, editors, *RoboCup98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999

[28] Manuela Veloso, Enrico Pagello and Hiroaki Kitano, editors. *RoboCup-99: Robot Soccer World Cup III*. Springer, Lecture Notes in Artificial Intelligence, 2000

[29] T. Wittig. ARCHON – An Architecture for Multi-Agent Systems, *Ellis Horwood Limited*, 1992