# RMAS_ArtSapience RoboCup Soccer Simulation 2D Team Description

Slim Abdennadher, Amir Rashed, Amr Saleh, Caroline Sabty, Catherine Antoniou, May Abbas, Mustapha Abdallah, Rania Said, and Yahia ElGamal

The German University in Cairo - GUC
slim.abdennadher@guc.edu.eg

**Abstract.** The main focus of the paper is a multi-level architecture for a dynamic multi-agent planner. The presented planner provides generic interfaces for the ease of integration in different multi-agent environments. The paper also presents a generic way for real time positioning in a multi-agent environment where there are competing groups of agents. In order to deal with a multi-agent environment where different positions have variable importance, an architecture of a generic fuzzy controller is presented to deal with that.

## 1 Introduction

RMAS_ArtSapience was founded by the Robotics and Multi-Agent Systems research group (RMAS), the faculty of Media Engineering and Technology (MET), the German University in Cairo (GUC), Egypt. The German University in Cairo participated in the RoboCup in 2009 and 2010. In the Festo Hockey League in 2009, the German University in Cairo achieved the $2^{\text{nd}}$ place where the best result of the team was a 6-0 against Polytech'Lille from France. In the Festo Logistics League in 2010, the German University in Cairo achieved the $1^{\text{st}}$ place where the best result of the team was 9-0 against both Robo-Erectus from Singapore and ROBOLOG from Switzerland. In 2011 the activity is expanded by adding the Soccer Simulation 2D and the Rescue simulation competitions to the competitions in which the German University in Cairo participates. All the members of the team are $3^{\text{rd}}$ to $9^{\text{th}}$ semester undergraduate students.

RMAS_ArtSapience team is based on the UVA base of 2003 where some modifications and additions on the world model and behaviors modules were applied. On top of the base code a multi-level multi-agent planner was implemented.

The remainder of the paper is organized as follows. Section 2 discusses the modifications and additions applied by RMAS_ArtSapience on the UVA base. Section 3 explains the architecture of the multi-level multi-agent planner. Finally, section 4 concludes the paper with a summary and a short discussion for future work.

## 2 Additions and Modifications to UVA Base

The UVA base code was enhanced by applying some additions and modifications to the behaviors, utilities, world model and main modules. This section focuses on some of the additions applied to the behaviors and world model modules.

### 2.1 Behaviors Module

When a team is defending, the main aim is to retrieve the ball. One of the ways of ball retrieval is interception of opponent's pass. For an agent to successfully intercept a pass, the agent has to be faster than the opponent's agent to which the pass is made by having a good marking position for the opponent. UVA base marking behavior offers three possibilities only for the marking positions of an opponent by getting a position between the opponent and the goal or the ball or the mid point of the two positions.

In order to provide a more precise marking behavior with infinite number of marking positions for an opponent, a dynamic fuzzy controller was implemented. The input 'I' to the fuzzy controller represents the difference in x-axis between the position of the opponent and the position of the ball. The output 'O' from the fuzzy controller is dynamic in a way that its extremes are set in real time to the goal and ball marking positions. If 'I' is -ve, it's inferred that the agent is close to the goal where 'O' tends to goal marking and vice versa if 'I' is +ve.

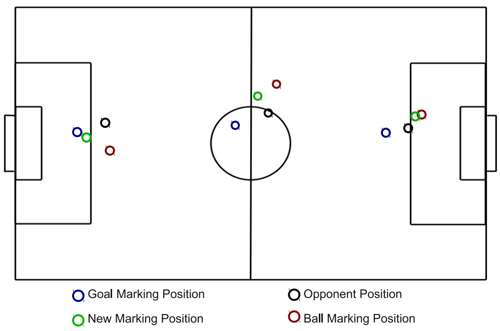


**Fig. 1.** Various Marking Scenarios

### 2.2 World Model Module

Work in the world model targeted two problems. The first problem was that the UVA's strategic position getter considered the ball position as the only dynamic factor. It was seen that it would be better to take the positions of surrounding teammates and opponents into consideration to attract/repel to/from them. The new strategic position getter would help the agents act more effectively in defense and attack, and become more flexible by adapting to different formations of the opponents using the same formation. The second problem was about deciding when to mark an opponent and the maximum distance to be kept from the opponent while marking him. Although close marking is always the best choice from the defensive point of view, it has drawbacks on the stamina of the agents. To solve the marking problem, two controllers were implemented: mark activation controller and mark distance controller. The mark activation controller is a fuzzy-based controller that controls whether it's necessary to mark a certain opponent or not. The mark distance controller defines the maximum distance to be kept from the opponent by considering the distance of the opponent from the goal and the ball.

**Strategic Positioning** The problem is modeled as a physics problem of the center of mass [5]. The pitch is viewed as a plane of mass where strategic positions are chosen in a way that achieves balancing the distribution of mass on the pitch. Each of the agents in addition to the ball is given a +ve/-ve mass, where the sign of the mass depends on which team have the ball. Weights vary according to the objects' relative current position and velocity. In the scenario, agents tend to move to more logical positions than only moving towards the position of the ball.

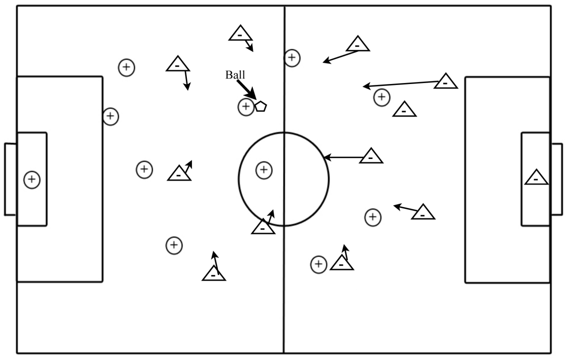


**Fig. 2.** Scenario for the strategic positions getter

**Marking Activation Controller** The mark activation controller depends on the position of the opponent on the pitch. The logic followed for the controller is that in the defensive situation defenders try to achieve close marking while not getting far from their positions to keep the defensive structure taking into consideration that a far away opponent will be marked by another defender. The input 'I' to the controller is the position of the opponent which represents how sensitive is the situation (the closer to the goal the more sensitive it becomes). Output 'O' defines distance 'D' which is directly proportional to the sensitivity analyzed from 'I'. 'D' represents the maximum distance for the opponent to be far away from the strategic position of the agent for the agent to decide whether to mark the opponent or not.

**Marking Distance Controller** The marking distance controller defines how close should the marking be depending on the opponent's distances to the ball and goal. In order to make the controller generic and dynamic, a weight is added to each distance and the equation is normalized by dividing by the output coming from the marking activation controller. Below is the equation used inside the controller.

$$\psi = \frac{(\omega_\beta \times \beta) + (\omega_\zeta \times \zeta)}{\gamma} \quad (1)$$

- $\psi$: is the marking distance to be passed to the marking behavior,
- $\omega_\beta$: is the weight of the opponent's distance to the ball,
- $\beta$: is the distance of the opponent to the ball,
- $\omega_\zeta$: is the weight of the opponent's distance to the goal,
- $\zeta$: is the distance of the opponent to the goal,
- $\gamma$: is the output of the marking activation controller.

The following figure show some marking scenarios. The scenarios show how much distance the agent had to move from the strategic position to do the marking.

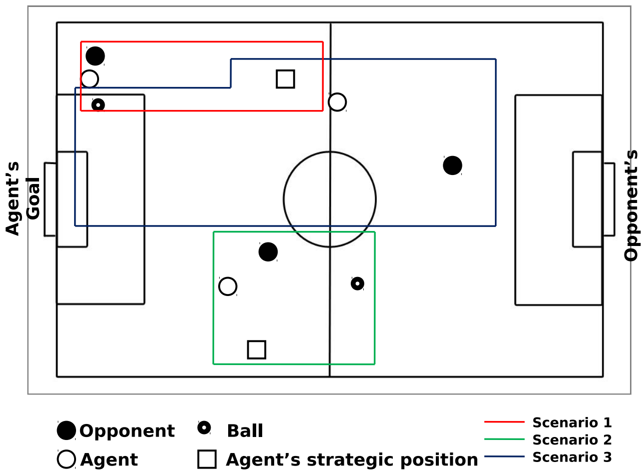


**Fig. 3.** Marking distance scenarios

## 3 The Multi-Level Multi-Agent Planner

The architecture of the planner has three distinct levels where each level is responsible for a certain part of the planning and execution process [1]. This section discusses the details of each of these levels. The following figure shows the architecture of the planner and how it communicates with the world model and the behaviors.

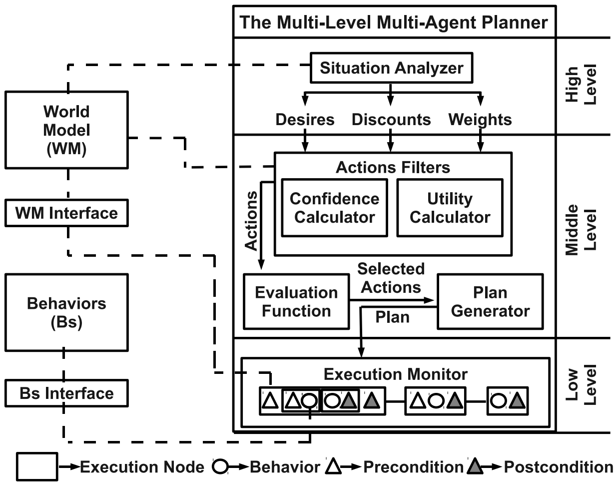


**Fig. 4.** The Multi-Level Multi-Agent Planner

### 3.1 High Level Planner

The high level contains the situation analyzer which considers general attributes like time, stamina, current score, agent's role, agent's current position and ball current position. The situation analyzer calculates for each of the possible actions the current desire and the current discount value. The situation analyzer also calculates the weight by which each of the desire, discount value, confidence and utility contributes in each action's evaluation [4].

The main controller in the situation analyzer is a fuzzy-based controller for pitch evaluation. As shown in the figure below, the controller can divide the pitch into any number of rectangles giving each rectangle a membership function representing the evaluation of the rectangle. According to the required functionality, the point given to the controller can be the agent's position, the target position or the position of any object on the pitch.

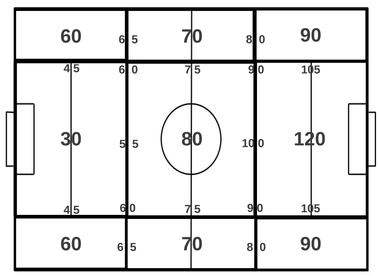


**Fig. 5.** Fuzzy-Based Pitch Evaluation Controller

### 3.2 Middle Level Planner

The middle level contains three components. The actions filters generates a finite set of actions where each action has a behavior, desire, discount, confidence and utility [2]. The evaluation function selects actions to be done for the next N cycles. The plan generator generates a short to middle term plan [3].

### 3.3 Low Level Planner

The low level has the execution monitor which provides flexibility by making conditions presence in nodes optional and by being able to deal with nested nodes. A node is executed if the precondition is satisfied and is removed if the postcondition is satisfied [1].

## 4 Conclusion and Future Work

The presented planner is aimed to be a generic planner suitable for different multi-agent systems scenarios by dealing with the abstract meaning of behaviors, conditions, desires, etc rather than dealing with the specific scenario of the RoboCup Soccer Simulation 2D. To validate that aim, the planner is used by RMAS's Festo Logistics League team. Dividing the planner into levels helped maximizing the performance and minimizing resources consumption as not all levels are executed every cycle.

To improve our research results, the following topics are included for future work:

– The coach should be able to analyze the formation of the opponent, and each opponent's moving area where information should propagate to the agents' high level planner for more precise calculations. This can have an impact on the formation/strategy to follow.
– The coach should take over control by giving orders during play off states for better utilization and precision in play off states.
– Communication should be integrated with the high level and set plays should be defined and integrated with the low level for better coordination and cooperation during attacking.

## References


1. Stone, P., Veloso, M.: Task decomposition and dynamic role assignment for real-time strategic teamwork. In Müller, J., Singh, M.P., Rao, A.S., eds.: In Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98), Heidelberg, Germany, Springer-Verlag (1999) 293–308
2. Bellman, R.: A markovian decision process. Journal of Mathematics and Mechanics **6** (1957)
3. A., R.: Howard Dynamic Programming and Markov Processes. The M.I.T. Press (1960)
4. van Nunen, J.E.E.: A set of successive approximation methods for discounted markovian decision problems. Mathematical Methods of Operations Research **20** (1976) 203–208
5. Reis, L., Lau, N., Oliveira, E.: Situation based strategic positioning for coordinating a team of homogeneous agents. Balancing Reactivity and Social Deliberation in Multi-Agent Systems (2001) 175–197