ROBOTOOS 2007 – 2D SOCCER SIMULATION TEAM DESCRIPTION

ARMIN ILDERMI, S. JAVAD HABIBI, MAJID ABEDINZADEH, SINA MOLAVIPOUR, PARHAM MIRZAYIAN

Mashhad Student Robotics Research Center, Mashhad, Iran
E-mail: `mashhadrobocup@yahoo.com`

Abstract— This paper intends to give an overview of our Soccer Simulation Team work done on 2D version. The main development we made was using a two-phase selection mechanism to determine the best action among all possible ones carried out by the ball controller agent for a given situation. Our aim is to construct stable and flexible agent architecture for our further development and research.

Keywords— Soccer simulation, multi-agent systems, decision making, interception, positioning

1 Introduction

The main goal of robotic soccer is to have a perfect domain for researchers and a standard problem for investigating and examining new artificial intelligence as well as multi-agent approaches and techniques. In RoboCup simulation league, teams of 11 autonomous software agents compete against each other by using RoboCup soccer server simulator software which is available from the official simulator website [1]. The ball controller agent is capable of performing shoot, passes or dribble actions and informs the server of his decision so that the server updates the playing environment.

2 Decision Making Module

For the basic decision making module we take advantage of Nexus team work on 3D framework [2]. In case the player number is 1, it would be considered as the goalie so the defined method is taken for the agent. If the player is not goalie and handles the ball control, i.e. located in the minimum required distance to kick the ball, *TryHandle* module selects one of the high-level actions pass, shoot, or dribble. If ball is not kickable then if the agent is the closest of all its teammates to the ball, *TryIntercept* module will select a move toward the ball action for the agent. If none of the above two situation, the player must go to an appropriate position according to the *TryPositioning* module.

```
if(!handleball.TryHandle())
{
        if(!intercept.TryIntercept())
        {
                Positioning.Position();

        }
}
```

In implementation of the *TryHandle* module, we used UVA team [3] structure in which divides the filed into some regions shown in figure 1, to consider best agent action considering position and game status.

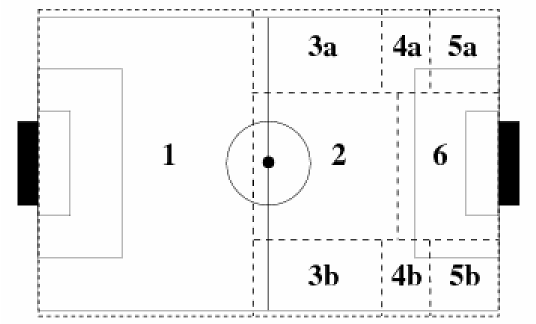

Figure 1. Strategic filed partitioning

3 Two-phase Action Selection Mechanism

To determine the best action among all possible ones carried out by the ball controller agent for a given situation, we first recognize the best of each action, i.e., the best shoot, the best dribble, and the best pass, independently. It is clear that, when the best possible shoot is sought the parameters that affect the shooting action are considered, only. For dribble and pass actions the same kind of process is followed.

In the next phase, we select the best of bests, i.e., the system chooses the best action among three best actions shoot, dribble, and pass. In this phase, common measures are used in order to evaluate the actions. Figure 2 shows the two-step evaluation method in which in the first phase it finds the best possible shoot, pass and dribble using specific measures. In the second phase, it selects the actual action to take, using common measures. To determine the priority in the second step, the calculated priorities in the first step is not considered [4].

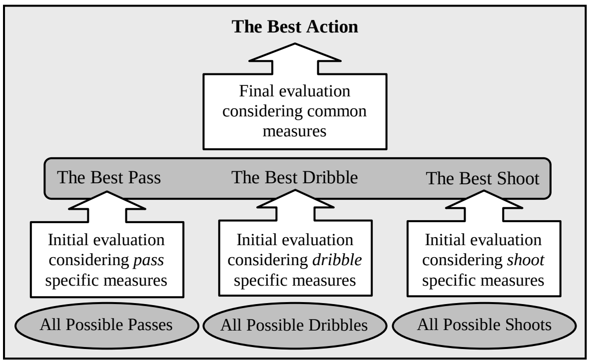

Figure 2. The two-phase selection diagram

4 Planning Approach

In Artificial Intelligence, problem solving is to transit state from initial state to goal state. There are two types of planning, namely deliberative planning and reactive planning. The former is to end all series of action before it really acts. So deliberative planning requires much computation resources. And it has also a problem of poor adaptability to dynamic environment.

On the other hand, reactive planning has a good adaptability, but in many cases, it does not select the best choice, and it needs more optimization. So there is a potent trade problem in these two types of planning. To solve this problem, we propose three layers planning, Strategy, Group, and Individual.

4.1 Strategy Layer

Strategy Layer covers all teammates. In this layer agents select their formation, tactics, and decide the policy of resource management. These must be decided depending upon opponent model. Static role assignment is done in this layer. Static Role is a role set, like Goalie, Defensive Half, etc. There are many types of team styles, indeed. So we need to adapt them effectively.

4.2 Group Layer

Group Layer planning include about three or four teammates in local state near ball. In group layer agents are assigned a Dynamic Role like, ball handler, and supporter. An agent, who ends the chance, can be a reactive cooperative planner. If there are no fatal condition to execute the plan, agreement will be done, and plan in group level can be executed.

4.3 Individual Layer

Individual Layer planning covers only 1 vs. 1 state. Agent selects most suitable pre-planned module. There are fatal conditions, which agents withdraw his plan in every simulation step. For example agent cannot end pass course in defense area, he makes a decision of clearing ball.

5 Strategy Architecture

Real-time decision-making in a match is a complex task. As so many factors have to be considered, we structured the whole task into several Modules including communication ,visual control , handle ball offense positioning Defense positioning etc more attention was paid to the later four the whole Strategy architecture is shown in figure 3 the following Explains the typical components in the architecture.

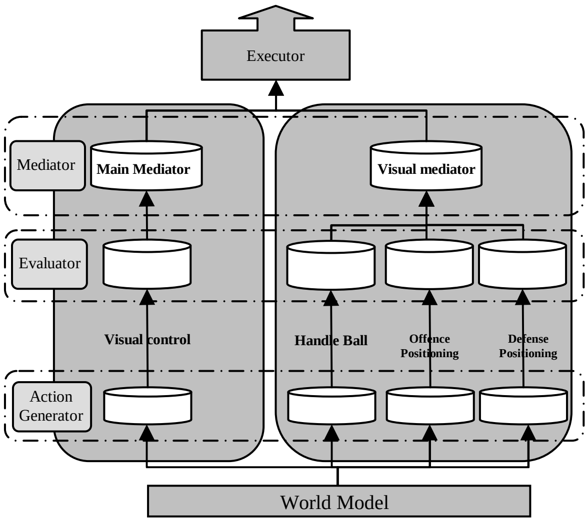

Figure 3. Strategy Architecture

6 Formations and Situation Analysis

In the formation decision module, agents use a set of predetermined rules, game high-level information (such as the result and time) and (if available) coach information to decide which is the current information. Formation information includes a set of flexible positions, roles and rules to change these positions in a soft way according to game low-level information (such as the ball and players current positions).

This module is directly connected to the second module of the main control loop, which is concerned with game situation analysis. This analysis includes who is attacking and the identification of game specific situations (like free kicks). To each situation a different sub formation (instantiated using the formation information) is assigned.

7 Situation Based Strategic Positioning

In non-critical situations, players try to position themselves in strategic positions that are dynamically determined using game information and rules available to all team agents. This way, even without visual or audio information, a given player, through a detailed situation analysis, may know with good accuracy all teammates positions in the field!

This is very similar to what happens in a real football team in which all players, analyzing the game situation, are able to identify known situations and guess teammate positions according to the identified situations. This technique assures at all steps that players are well distributed along the field, i.e. defensive regions are covered and a lot of attack options are available.

8 Future Work

Our future work would mainly based on developing a fuzzy decision making system which tells an agent which direction to look. Considering the noise and uncertainties in the agent actions, fuzzy decision making seemes suitable to be used in the soccer simulation environment. This approach is much less time consuming and very simple to follow, in contrast with the other analytical implementations with complex conditions in the codes.

As the number of input variables may increase, filling in all the possible rules becomes a time consuming task. Therefore rules that only include a few of the input variables and cover greater subspaces in the control space are used. As the second reason to use fuzzy systems, they are not sensitive to the completeness of the rule base as long as the boundary rules are preserved in the fuzzy associative memory. Having completed the bank of fuzzy if-then rules, c*enter of maximum* defuzzification method would be used which is continuous and computationally efficient.

References

[1] Chen y M., Dorer K., Foroughi E., et al., "*RoboCup Soccer Server*", User manual for Soccer Server V 7.07 and later, February 11, 2003.

[2] Salmani V., Seifi F., et al. "Nexus 2005 - 3D Team Description", RoboCup Soccer simulation team escription, International Symposium of RoboCup, July 18-19, 2005, Osaka, Japan

[3] de Boer R., Kok J. R., "The incremental development of a synthetic multi-agent system: the UvA Trilearn 2001 robotic soccer simulation team". Master's Thesis, University of Amsterdam, The Netherlands, 2002.

[4] Salmani V., Naghibzadeh M., Seifi F., Taherinia A., "*A two-phase mechanism for agent's action selection in soccer simulation*", accepted to be published on International Conference on Artificial Intelligence and Data Mining, ALDM - 2005: February 25-27 , 2005 Istanbul Turkey