Persepolis 2002 Team Description

Shahab Javanmardy, Mohammad Nejad Sedaghat, Nader Talebi
Nina Gholami, Anis Yousefi

{shahab, msedaghat, nader, ngholami, ayousefi}@Javanroboclub.com

Javan RoboClub, Http://www.javanroboclub.com
Javan Farhangsara , Khavaran Ave., Tehran, Iran

http://www.javanroboclub.com/~persepolis/index.html

Abstract. In this paper, Persepolis team agent's implementation is described. A four-layer architecture is proposed and each layer is briefly presented. The emphasis of the article is on a novel approach in reasoning layer, which presents *characterized agents, single player opponent modeling* and *collaboration* among characterized agents. The primary results show a good performance with respect to available literature.

1 Introduction

RoboCup competition is one of the most important events in AI field and a very rich domain to work on Multi-Agent planning and collaboration. In the teams with homogeneous agents, it is hard to implement various strategies and stay adapted to the environment. To address this problem, we investigate the concept of *Characterized Agents*. In this paper the multi-threaded [1][2] and multi-layered architecture [3] of the agents have been described at the first stage. The synchronization with the server [4] and making the world model accurate [5] are also discussed. After this, the usage of Optimal Scoring Policy [6] to improve the efficiency of shoot skill is described. In the next section the concept of *Characterized Agents* is introduced following by the *Single Player Opponent Modeling* approach. The last part is devoted to *collaboration* among *Characterized Agents.*

2 Architecture

Persepolis uses a multithreaded architecture, which allows each agent to have a separate thread for perception, decision, and action [1][2]. Also a layered architecture is used including Interface Layer, Action Generator Layer, Evaluator Layer, and Reasoning Layer [3]. The threads used for perception and action form the bottom layer of our architecture named Interface Layer. The Interface Layer hides the detail of networking from higher layers and builds the world model of each agent. To be synchronized with the server the External Basic method is applied [4]. Above this layer, we have the Action Generator Layer, which provides the possible actions. The third layer is Evaluator, which evaluates the actions generated by the lower layer and assigns a priority to each action that mean how valuable it is. The next layer is called Reasoning Layer, which receives many actions with different priority. Its task is to find the optimal action.

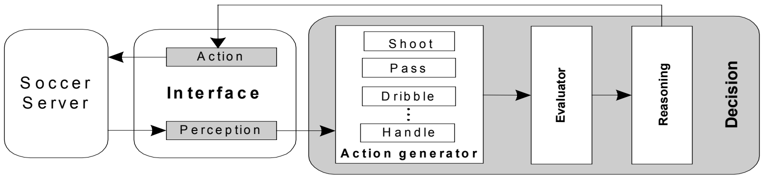

Fig.1. Multi-threaded and Multi-layered Architecture

3 World Model

The world model of each agent is a probabilistic representation of world state based on past perceptions. For each object, an estimation of its position and velocity is stored together with a confidence value that is determined based on the distance to the seen object [5]. To unify the world model of agents, the player with the best view of the field, reports his sight information to teammates.

4 Improving the Shoot Skill

To improve the shoot skill we use the Optimal Scoring Policy [6], which determines the optimal shooting point in the goal together with an associated probability of scoring when the ball is shot to this point. The weakness of this approach is, the probability of passing the ball from the goalkeeper is calculated based on a typical goalkeeper, but different goalkeepers will result different probabilistic models. We want to improve this approach by adapting the model during the game.

5 Characterized Agents

In many soccer simulation teams, despite of the clearly *different responsibilities* of the players, their decision-making mechanism is the same. this will result the agents unexpectedly act the same in similar cases. For example we expect a midfielder to be a good passer but in the case of forward we need a player who can score easily.

The other problem is that the agents cannot recognize weak and strong points of the opponent. This will cause an action to be lost several times. For example, we have noticed that the FC Portugal agents lost many goal chances because they didn't consider the capabilities of current goalkeeper. Therefore the opponent players' capabilities should be considered in decision-making mechanism. The emphasis of our approach is on having heterogeneous players and stay adapted to opponent team the same as the one in real soccer match. The agents can be different in two ways:

1. Weighted action selection structure.
2. Action performing mechanism.

5.1. Weighted Action Selection Structure

Since the decision-making mechanism is based on the evaluation results of the Evaluator Layer, we can increase the chance of selecting an action by assigning a suitable weight to each result.

5.2. Action Performing Mechanism

The actions' result is evaluated considering the type of player, so it is required to have different algorithms for an action in different players. For instance, we expect a defender to pass in a way that jams the opponent attack, but a midfielder should pass in the way that would be a start of an aggressive movement.

In *Characterized Agents* it is possible to have an efficient and time-consuming algorithm, just for the actions that need to be done more accurately by this player. In this approach, a defender does not need to have an efficient algorithm for shoot skill. Each player has a sequence of actions, which is sorted according to their *precedence.* The precedence of an action is determined with respect to our expectancy of this player. For example, the action sequence is as follows: shoot, pass, and dribble, etc. for a forward. The *precedence of the action* should be considered in designing of the algorithm for a particular player.

5.3. Player Type

The mentioned approaches will result that each agent acts differently in either selecting or performing an action. The set of weights and algorithms will form a particular player type. A player type presents the tendency of the player to perform an action in a special way. The agent with a particular player type presents a special behavior, which makes it different from the other players. Therefore they are called *Characterized Agents*.

6 Single Player Opponent Modeling

In real soccer, the team's strategy is selected considering the opponent's capabilities. In addition, each player considers the opponent's capabilities when trying to select an action. Up to now, in soccer simulation teams, the different strategies have been implemented by using different formations. Actually opponent players' skills were not considered in choosing a strategy. To increase the success chance of an action, the weak and strong points of the opponents should be respected. This means, from our point of view, opponent agents are characterized and we judge about their characteristics, considering the result of his actions.

The players' skills are recognized in the following way, if one of our actions has been failed several times, having a particular opponent player in the opposite site, we guess that the opponent player has a perfect algorithm for this special skill. To recognize weak points of each opponent player we act in the same way.

After identifying the opponent, the weight of each action is updated dynamically regarding to the weak and strong points of the opposite player. For example, a player evaluates pass action, considering interception skill efficiency of the opposite player.

7 Collaboration

Since soccer is a teamwork play, collaboration is unavoidable, and also tactical actions require collaboration among the players. To achieve collaboration, one of the methods would be reporting the selected strategy through the messages. But this is not a suitable method, because communication among the agents is limited in the simulation environment. The other method is "Distributed Planning" introduced by TsinghuAeolus [7]. In this method every player assumes that his world model is noiseless and with this assumption, the agents finds the current strategy and his role. To avoid confliction, each agent assumes that other teammates have the same world model. Therefore they should know his role in current strategy. Although this is a useful method in homogeneous multi-agent systems, it does not work properly in

*Characterized Multi-agent Systems (CMAS).*So it is required to use a new method for collaboration among the *Characterized Agents*.

The chief point is, the team members coordinate with each other to achieve a unique goal. In this approach, the player who has the ball, is the designer of the game. It means that he chooses the strategy and performs his task, considering his characteristics and the current situation. The other players, knowing the designer's characteristics and his decision-making mechanism, will guess the selected strategy and perform their tasks.

A strategic movement is a set of conscious individual motions, which is done to achieve a particular goal. In a system including *Characterized Agents*, each of the individual motions should be done by the player who has an optimal algorithm for it. Therefore the player has to consider his teammates' skills in each step and acts in a way that causes the following action to be carried out by a player who has an optimal algorithm for it. For instance, the defender with the ball should pass to a midfielder with a proper position and also has an optimal pass algorithm.

Considering the existence of *Characterized Agents* in the team, collaboration among the agents becomes very important. Hence we have considered collaboration in *Characterized Multi Agent Systems (CMAS)* as one of our most important goals.

8 Conclusions and Future Work

The characterized agent idea, causes each player knows his role and performs his responsibilities as it should be. In addition, by identifying the opponent players, the player adapts to the environment more efficiently. In the next stage we will enhance the idea of characterized agents by making them more specialized. The main focus will be on collaboration in *Characterized Multi-agent systems (CMAS).*

References

1. Kostas Kostiadis, Huosheng Hu: A Multi-Threaded Approach to Simulated Soccer Agents for the RoboCup Competition. In Proceedings of the 16th IJCAI RoboCup Workshop (1999), 137-142.
2. Remco de Boer, Jelle Kok, Frans Groen: UvA Trilearn 2001 Team Description Paper. At http://carol.wins.uva.nl/~jellekok/robocup/ (2001).
3. Yao Jinyi, Chen Jiang, Cai Yunpeng, Li Shi: Architecture of TsinghuAeolus. At http://166.111.68.50/RoboCup (2001).
4. Marc Butler, Mikhail Prokopenko, Thomas Howard: Flexible Synchronisation within RoboCup Environment: A Comparative Analysis. In Proceedings of the RoboCup 2000 Workshop (2000), 119-128.
5. Peter Stone,Patrick Riley,Manuela Veloso:The CMUnited-99 Champion Simulator Team. In Proceedings of the RoboCup 1999 Workshop (1999), 35-48.
6. Jelle Kok, Remco de Boer, and Nikos Vlassis. Towards an optimal scoring policy for simulated soccer agents. At http://carol.wins.uva.nl/~jellekok/robocup/ (2001)
7. Yao Jinyi, Chen Jiang, Cai Yunpeng, Li Shi: Global Planning from Local Eyeshot: An Implementation of Observation-based Plan Coordination in RoboCup Simulation Games. At http://166.111.68.50/RoboCup (2001).