Iranians Team Description:
Robocup 2002 Simulation League Champion

Team Leader: A. Rahmani
Members: H.Mousavian, R.Zakeri, H.Hemmati,
A.Abdorazaghi, A.Aavani, V.Rafe

{rahmani, mousavian, rafe}@iust.ac.ir
{r_zakeri, h_hemmati, abdorazaghi, a_aavani}@mail.iust.ac.ir

Computer Engineering Department - Iran University of Science and Technology

February 2002

Abstract. Soccer is one of the most attractive sports on the world. There are millions of people spending several hours every week watching their favorite team matches. Regard to this high level of fascination, Robocup Simulation League was established to use this field as a rich domain for the study of Multi-Agent Systems and other related topics. In a real soccer team, every player knows his/her duties and performs them using some inferences about his around world. These inferences are based on some specifiable rules. We are using this idea to write our rule based virtual soccer players.

1 - Introduction

Robocup Simulation League, regularly held from 1997, is one the contests that potentially attracts many students because of its scientific aspects. Robotics in conjunction with Soccer, each of which individually fascinating, has opened new ways for research in various fields such as Multi Agent Systems, agents collaboration and cooperation, strategy acquisition, real-time reasoning and planning and so forth [1]. In this paper we will first study *Iranians* team architecture in section 2; then in section 3 team s world model is introduced and in section 4 we will see how to use Rule-Based Expert Systems model in rule abstraction. Finally a brief conclusion is presented in section 5.

2 - Team Architecture

Utilizing rule-based expert systems is one of the well-known methods in AI. There is a type of rule abstraction in these systems, which we have used in implementation of our simulated soccer agent. *Iranians* simulation team architecture is shown in figure 1. As seen in this figure, coming information from the *server* in each cycle are parsed and transformed to sensational, auditive and visual data that update the *world model*. Each of *Skills & Tactics* and *Strategy* units recommend a suitable strategy and new skill or tactic with regard to the world model. At last

Decision Making unit tells each player to play on with his current strategy, skill and tactic or take a new one. *Coach* can also change team strategy and in this case coach s command has higher priority than that of the agent itself. Each skill or tactic is composed of minor and in-parallel executable *tasks*. These tasks in turn are composed of one or more basic commands. *Basic commands* unit is responsible for receiving different commands from various tasks and sending them to the server in the form of suitable messages. Skills & Tactics unit includes some individual and group skills that are already designed and in each instant there is only one active skill or tactic. Strategy unit includes team formation, team offensive or defensiveness and role assignment regarding heterogeneous players features.

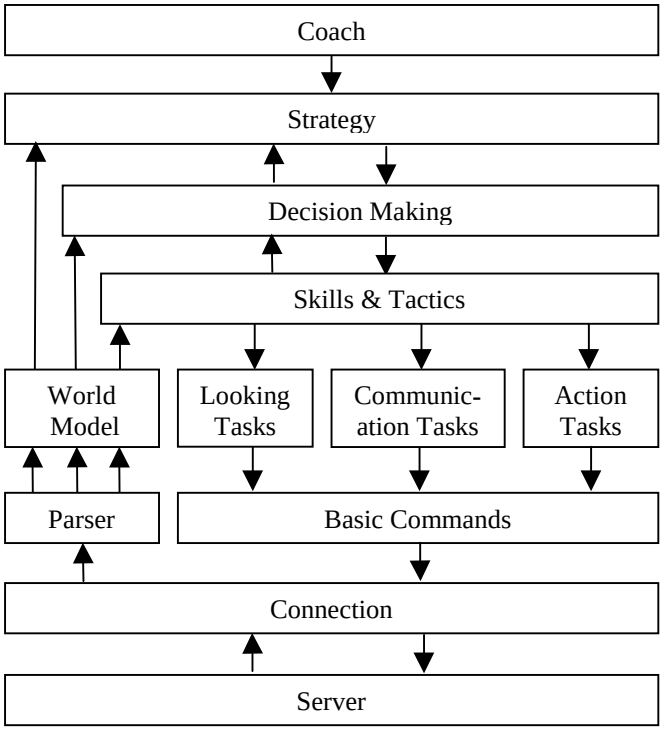

Figure 1 : Architecture

3 - World Model

The world model of each agent is updated on each data reception from the server. In the case of no visual data reception about an object or when it is not in the view-cone of the agent the information about this object such as position and speed is estimated regarding performed action in the last cycle. The world model of an agent is its memory plus some necessary methods for working with memory. So we have designed our agents world model structure with regard to the memory structure for storing information about objects. As seen in figure 2 objects are divided into two main groups of *Mobile* and *Stationary* objects. World model in this structure is similar to that of CMU98 team[2]. Stationary Objects do not need memory and are used merely for players current position estimation; but Mobile Objects do need memory. The precision and accuracy of the information stored in the world model is very important since all decisions in an agent are made upon them.

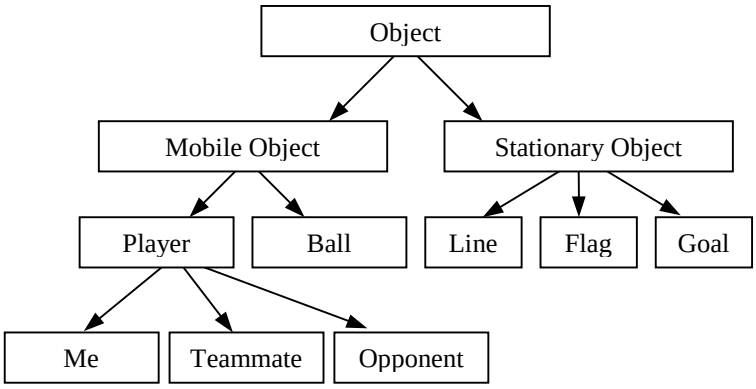

Figure 2 : World Model Structure

4 - Rule Abstraction

The important point in the design of *Iranians* team is that we have tried to reach a new type of rule abstraction in which the programmer can write the rules for an agent in a text file using a specified grammar instead of manually coding these rules. Each agent translates this file to a useful structure using a built-in compiler (figure 3). This structure in companion with the information stored in the world model form the input of an inference mechanism. According to the architecture cited in section 2, Skills & Tactics, Strategy and Decision Making units utilize this inference mechanism for their selections and decisions. This property allows us to program an agent without manipulating its internal code. So we have a programmable agent. The best advantage of this rule abstraction is that making changes in the behaviors of an agent and testing different rules and combining them together is easily performed. Also using this technique, it is very easy to teach a simulated soccer agent the experiences of a real footballer. In fact the rule abstraction idea of our team is a conceptual difference between our viewpoint and previously implemented teams. We think using rule abstraction, our team can be a good beginning point for new teams. We discussed this topic more on our paper named Using Rule-Based Expert Systems Model in Robocup Simulated League . Another paper of ours, Handling Decision Problem for Simulated Soccer Agent Using Decistion Tree Method , more specifically discusses decision making problem. These papers are to be submited in Robocup 2002 International Symposium.

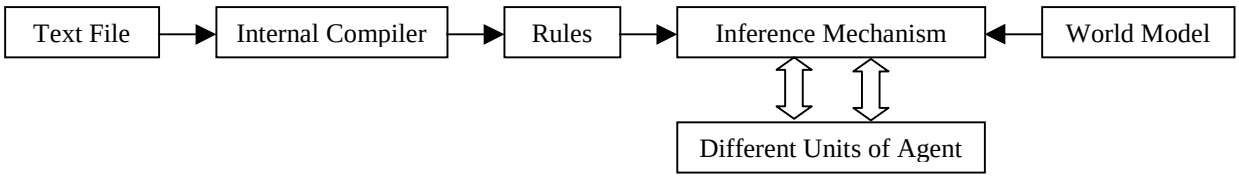

Figure 3 : Using Rule Abstraction

5 - Conclusion and Future works

Using rule abstraction idea seems to results in a new generation of simulated soccer agents. This generation is more flexible and easier to improve than present agents. To improve this model we want to examine possibility of:

- Generalization of abstraction to skills, tactics and strategies.
- Logical inference on current rules and making new rules.
- Setting utilized coefficients in rules using Agent Offline Learning.

References :

[1] K. Kostiadis and H. Hu: "Reinforcement Learning and Co-operation in a Simulated Multi-agent System", Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 990-995, Kyungiu, Korea, 17-21 October 1999

[2] Peter Stone, Manuela Veloso, Patrick Riley: "The CMUnited-98 Champion Simulator Team", Robocup-98: Robot Soccer World Cup II, Springer Verlag, Berlin, 1999.