SharifII Soccer Simulation Team

Dr. Jafar Habibi, Ehsan Foroughi, Mehran Motamed
Pooya Karimian, Hamed Hatami, Hossein Fardad

Sharif University of Technology

1 Introduction

This paper describes ParsAI, a team developed and working at Sharif University of Technology. It is mainly concentrated on two distinct works, an idea for decision making components and a simple but powerful easy-to-debug implementation of clients.

2 Special Team Features

Our team's power comes from its simplicity of implementation and powerful design based on layered architecture as described in the next section.

Another feature of the team is the usage of the simple but powerful method of weighted comparison in all components. This method is to some extent similar to a neural network except for the method of weights calculation.

3 The layered architecture of our agent

Handling large programming projects requires a well-designed architecture. A common method is to divide the problem into distinct layers, each of which uses the previous layers' services to provide some predefined services for next layers; therefore, precise definition of each layer is necessary.

Beside all its known advantages, this model allowed us to divide both the problem and the program into many parts and helped us to work perfectly in a group, because it fixes and abstracts all the intermediate interfaces of components.

Figure 1 shows an overview of the architecture. The arrows indicate the way information is passed to other layers.

Our client model has a four-layered architecture as follows:

3.1 The Connection layer

This layer consists of a network socket and some low-level procedures for receiving sensor information from the server and sending commands to it.The main loop of the program simply gets messages one by one from server (using a blocking mode socket), processes them and passes the control to worldmodel layer to update data.

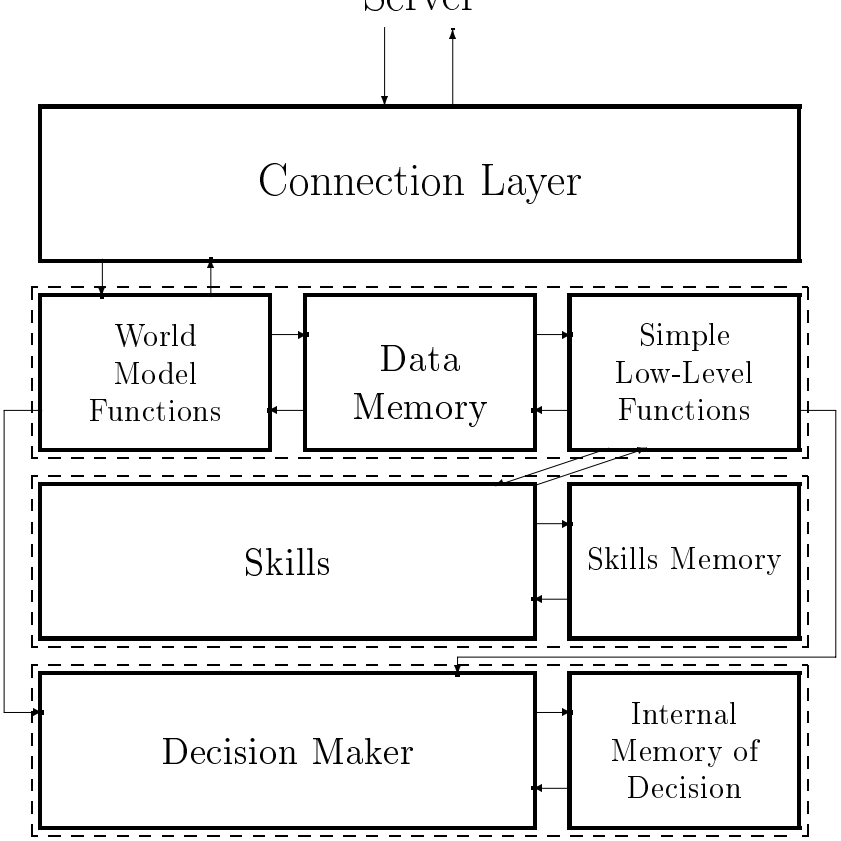

Fig. 1. The agent four layered architecture. From top : (1) Connection layer (2) World model layer (3) Skills layer (4) Decision making layer

3.2 The World Model layer

This layer is very important due to the dynamic nature of the problem. It provides and maintains the information which the agent needs for making decisions. This layer gets raw data from sensors, and parses them. Then it analyzes them to extract relatively useful data. In this part previous data is used for fields that we don't get any information. To store the data into memory we use a structure that stores the time of last updatation and also the reliability (certainty or the range of error) of the data. This information prevents the client from confusion that usually occurs due to old or incorrect data, allowing us to compare values gained from different sensors. Note that all world model updators have to consider and update these values.

This layer also provides some intermediate functions for sending commands to the server. These functions update the worldmodel assuming the server will receive and change the world model at the end of that cycle. We used this method because the probability of "message reaching server correctly in synchronous time stepping model" is very high and all messages usually reach server within a cycle time.

For the parts of world which can't be seen, our agents use a combination of previous data and the data they receive from the hear sensor. In this combination the certainty values are of great importance that allow the result tending to be based on previous data in first few cycles and tending to hear sensor information

as the time passes on (from the last visual information about that object). In practice the accuracy reduces noticeably when this communication is turned off.

This layer also includes the Positioning component that finds the Player's absolute position in the field using the vision based sensors. In this part we have used the Hill Climbing algorithm to find the player's position that helped us reach the accuracy of less than 0.1-0.2 meter error in average compared to average error of 0.5-0.7 meters in the normal methods (a superb improvement).

3.3 The Skills layer

This layer consists of individual skills components that form the base of soccer playing behaviors of agents. We have implemented and used these skills :

- Passing: Includes Normal Pass, Forward Pass and Goaler Pass. Finds the best teammate considering all opponents by calculating a value for each possibility (the safeness comes here) and finding maximum value.
- Dribbling: Using many methods of dribbling like spinning the ball, running with ball and going through with a sudden movement; the main goal of this skill is to dribble forward toward a destination.
- Complex Kick: This skill, normally used in combination with Goal Scoring and Passing skills, is used to increase the speed of the ball with subsequent kicks toward some predetermined destination.
- Shoot to Goal: This skill considers the possibility of Shooting to opponent goal using both Normal and Complex Kicks and calculates the best direction.
- Getting the Ball: This skill calculates and finds the best way to reach the ball before opponents. When the Agent decides to have control over the ball, and the ball is not in his catchable area, he probably uses this skill.
- Positioning Without the Ball: This skill controls the movements of players without the ball in the field to achieve a good arrangement and positioning.

3.4 The Decision Making layer — Team's strategy

This layer makes decisions based on information received from other layers specially the world model layer. It also includes team strategies and agent reactions to normal and abnormal situations. Implementation is of great importance because of the complexity of this layer, and it needs a powerful abstraction to include all situations so we can easily add new ideas and reactions to it.

The real time, dynamic environment makes the decision making problem more challenging. So while keeping the simplicity we tried a new approach to the problem of choosing a good strategy : Weighted Comparison.

In this method, used both in skills and strategy layer, layer by layer, each component calculates a value for each action. This weight calculated (usually linearly) from values of underlying parts, represents the usefulness of the action so that the strategy layer can choose the best action by comparing this values.

4 Team Development

For the beginning of our team development we used "libsclient-4.00", a common library for connection layer, first developed by Noda Itsuki, but little by little we customized the library and optimized it for server version 6.00. Also at the last step we added our powerful positioning system to it. Its source code is openly released and can be found in our web site.

We also wrote a debugger and a real-time world model visualizer for our agents during the team development that helped us a lot in the implementation.

Team Leader: Dr Jafar Habibi — email: habibi@sharif.ac.ir
Team Members:

Ehsan Foroughi (attend the competition)
Mehran Motamed (attend the competition)
Pooya Karimian
Hamed Hatami
Hossein Fardad

- {foroughi,motamed,karimian,hatami,fardad}@linux.ce.sharif.ac.ir
- Country: Islamic Republic of Iran
- All are undergraduate students of computer engineering in Sharif University of Technology

Web page http://www.parsai.com

5 Conclusion

We have decided to reduce the complexity of implementation while increasing the power of the team by avoiding case consideration and using formal, universal methods and designs. This layered design will remain as the base of our client.

In the mean time we are working on the improvement of the algorithms and completion of the parts that are not implemented yet. In the Connection Layer and World Model Layer we don't need any noticeable changes except for a better accuracy of the objects in the World Model. There remains some work in the Skills Layer and the main concentration is on Strategy Layer. Some learning based strategies are also planned to be tested in future.

Some new topics are introduced in relation to the decision making algorithm discussed in section 3. The most important of them is usage of the Genetic Programming method to determine the weights. As you may notice, the core of the decision making components is to determine the weights correctly and efficiently. This converts the strategy and decision component of our program into a table of quotients. So we can use Genetic Programming methods to find the efficient table. Note that the problem of writing a successful team is now converted to maximizing the value of a function from a table of inputs to a number representing the success of the team (teams performance).