TJUnited 2010 Team Description

Yaolong Huang, Chao Ma, and Jianrong Wang
Department of Computer Science, Tianjin University, Tianjin, China

Abstract. This paper describes the main features of the TJUnited 3D soccer simulation team, which will participate at RoboCup-ChinaOpen 2010 for the first time. After a few months design and implementation, the architecture and several key components, including World Model and Skills, have been completed. Here we introduce the key concepts in TJUnited, and we will give some conclusions and describe future research directions.

1 Introduction

RoboCup is an international research and education initiative whose purpose is to foster AI and intelligent robotics research by providing a standard problem, such as a soccer game. Among many leagues, the 3D soccer simulation league has been proved an excellent testbed for many novel approaches in AI and Robotics because of its high level of simulated reality and relative low-cost of experiments compared to other leagues by using simspark [1], the simulation server, as the standard platform.

As a new team in the 3D soccer simulation league, TJUnited is designed by 2 undergraduate students from Tianjin University. Before implementing TJUnited, most of our efforts went into collecting relevant materials and becoming familiar with RoboCup and 3D simulation. After a long time of preparation, we decided to adopt a layered architecture approach as our basis. However, we decided to implement it from scratch instead of using other team’s code as basis. Firstly, most teams’ code we could find so far was not suitable for our architecture. Making them fit into our design was harder than writing a new one. Secondly, we wanted to understand the concepts in our agent design more clearly from the implementation, which would undoubtedly improve our skills and help us identify more subtle problems in the RoboCup.

This article presents the TJUnited’s key concepts in the following sections. Section 2 introduces our overall agent architecture. Section 3 presents our World Model. Section 4 illustrates Skills design of our agent, followed by conclusion and future work in Section 5.

2 Agent Architecture

Because of the server/client architecture of simspark[2], which uses network connection to communicate with agents, each agent is able to perceive, reason and
act. Perception and action are accomplished via TCP connection. After receiving the perception messages, the agent should be able to use this information to update and maintain an internal representation of the environment it resides. We call the internal representation the World Model and will be detailed in Section 3. After updating, the agent should reason which action it should take to attain its goal. The capabilities of the agent, like walking and kicking, are the Skills component, which will be demonstrated in Section 4.

We adopted an Object-Oriented design similar to Vorst’s design[3], which has been proved to be good layered design, and implemented it in C++. The main component of our agent is presented in figure 1.

3 World Model

World Model is responsible for maintaining the internal representation of the environment and extracting useful high level abstraction from this information. We divided the environmental component into 4 sub-level[3]. Every level will use information provided by the lower levels to obtain a more abstract view of the game, which is key to success of the reasoning process of the agent.

The information provided is as follow:

- Level 1 stores the fixed information of the match, such as the field length.
- Level 2 stores the dynamic information, including the position and velocity of the objects on the field.
- Level 3 includes some abstract views on the state of the match, for instance the closest player to certain position or the ball.
- Level 4 provides the group and team level abstraction for the agent, which is the key to team cooperation.

### 3.1 Self-Localization

Without knowing the position of agent with respect to the global coordinate system, it is impossible to get to a target position on the field or to access the usefulness of passing the ball to the teammate. We used SEU’s approach to localize while there are 3 known fixed object, like flags, that are seen[4]. Furthermore, it is the usual case that less than 3 fixed objects are seen. In order to be able to localize the agent in such cases, we also used 2 fixed objects to complete the localization[5] without consideration of effects of z axis. Although this method requires some assumptions and approximations, it provides the agent with the ability to localize under a more restricted vision condition.

### 4 Skills

The *Skills* module represents the agent’s capability to influence the environment, such as walking. Without highly reliable and efficient skills, it is hard or even impossible to achieve the selected goal of the agent. After a long time of research, we decided to adapt a design similar to Vorst’s skill component[3] and add extra skills to it which are essential to the humanoid robot. We model various skills as classes instead of function out of the consideration for reusability. Additionally, all of the skills are managed by *Skill Manager*, which takes care of skills' parameters passing and invocation. By using this design, we can invoke the skills in a uniform format.

Compared with the 2D soccer simulation, the additional dimension and the humanoid robot model adds many new issues to skill design, like state-oriented skill. The skills of the agent should maintain some internal states to be continuously executed in certain timing sequence. For example, with walking, the agent should keep the current state of the walking phase, like lifting left foot, to enable it to determine the right skill primitive for the following cycle.

However, due to the limited time before the competition, we determined to adapt some of the skill’s implementations of HfutEngine3D[6], such as walking and getting up. Because of our modular design, we can later change our skill implementations without affecting other levels of skills.

### 5 Conclusion and Future work

In this paper, we addressed some key concepts in *TJUnited* and explained some of its detailed implementations. Currently, we have only finished the key parts of the agent, but have not yet implemented all of them. Furthermore, we found that the internal running mechanism of simspark[1, 7] will greatly influence efficiency of the agent design, which we have not paid much attention to.
For future directions, we are interested in adding sensor fusion technique[8] in our agent to refine the World Model, especially the localization. We are prepared to put more efforts into refining our skill implementation, especially walking, instead of using other teams’ implementation. We want to use the inverted pendulum method[9] to plan the walking pattern.

References

2. SimSpark User’s Manual
8. Alexander Ferrein, Lutz Hermanns, and Gerhard Lakemeyer, Comparing Sensor Fusion Techniques for Ball Position Estimation