

## Robobahae 2005 Team Description

Saeed Danesh<sup>1</sup>, Reza Ahli Araghi<sup>1</sup>, Bardia Khalesi<sup>1</sup>, Mohamad Bostanian<sup>1</sup>, Reza Zafarani<sup>2</sup> and Mohsen Mossavi<sup>3</sup>  
<http://www.shbu.ac.ir/robocup>

<sup>1</sup> Computer Engineering Department, Sheikhbahae Institute of Higher Education,  
81797 Baharestan, Isfahan, Iran  
[saeed\\_danesh2020@yahoo.com](mailto:saeed_danesh2020@yahoo.com), [araghi\\_reza@yahoo.com](mailto:araghi_reza@yahoo.com),  
[semerdis2002@yahoo.com](mailto:semerdis2002@yahoo.com), [mbeng80@yahoo.com](mailto:mbeng80@yahoo.com)

<sup>2</sup> Faculty of Computer Engineering, University of Isfahan,  
81745, Isfahan, Iran  
[rzafarani@eng.ui.ac.ir](mailto:rzafarani@eng.ui.ac.ir)

<sup>3</sup> Faculty of Computer Science, Sheikhbahae Institute of Higher Education  
81797 Baharestan, Isfahan, Iran  
[mohsen@mossavi.com](mailto:mohsen@mossavi.com)

**Abstract:** This paper aims to describe the Robobahae team that is based on spades [1] and the 3D soccer simulation environment [2]. Despite the little time we had in developing the soccer server 3D binary, the first Robobahae team had some achievements in developing a new team.

## 1 Introduction

Since we had little time getting familiar with the rcserver3d and the rules are still not fixed definitely. We are trying to create simulation agents based on thinking and acting rationally. Our project was conceived as an effort to create intelligent players, capable of thinking like real soccer players and behave like a real soccer team. Therefore was quickened by designing a simple binary which had some simple but effective advancement.

## 2 SPADES

The simulation server is implemented above a platform called SPADES (System for Parallel Agent Discrete Agent Simulation). SPADES is a middleware system for agent-based distributed simulation (Riley 2003) [1]. It aims to provide a generic platform to run in multi-computer systems. It implements the basic structure to allow the interaction between agents and a simulated world in such a way that the users do not have to worry about sockets, addresses, etc. SPADES main features are: notify that returns an empty sensation (time notify) and after receiving it the agent is able to respond with actions. For example, if an agent received a sensation at cycle 100 and wants to produce an action at cycle 110 but the next sensation will only arrive at cycle 120, the agent can ask to receive a time notify message at cycle 110 and just reply after receiving it.

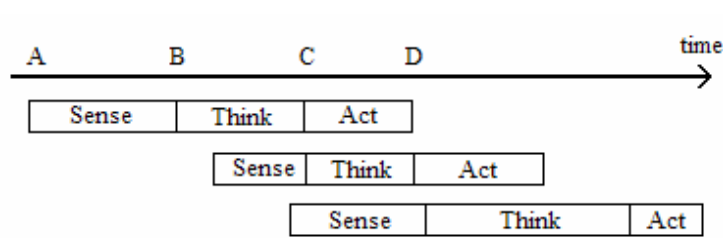


Fig.1– SPADES Sense-Think-Act Cycle

Fig.1 depicts the sense-think-act cycle and the time where each of its components runs. From A to B a sensation is sent to the agent. After receiving the sensation (from B to C) the agent decides which actions will be executed; then from (C to D) the actions are sent to the server.

In many agents, the sense, think and act components may be overlapped in time (like in Fig.1) there is just one restriction – the thinking cycles for one agent can not be overlapped. This constraint makes sense, since just a single processing unit is used per agent, and thus, just one sensation at time can be processed.

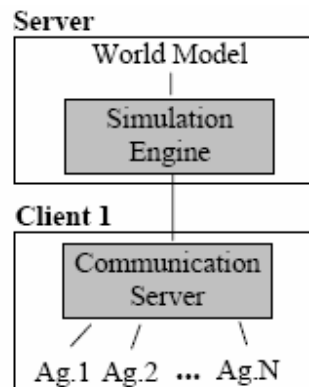


Fig.2 – SPADES Components diagram.

SPADES components are organized in client-server architecture (Fig.2). The Simulation Engine and the Communication Server are supplied as a part of SPADES; while the Agents and the World Model are built by the user and run upon the formers. The Simulation Engine is a generic piece of software that allows creating specific world models upon it. It runs on the server side and provides the interaction and communication between the agents and their world via the Communication Server. On the same machine it must run the specifications of the World Model that has the characteristics of the environment where the agent will act. Distributed along the clients are the agents and the Communication Server. The Communication Server must be present in all client machines end provides the communication between the agents and the Simulation Engine. It receives messages from the Agents and sends them to the server and vice-versa. The Agents also run on the client side.

### 3 Team Architecture

Robobahae 3D Soccer Simulation Team is an eleven autonomous software agent team in the purpose of consecutive research on Multi-Agent System. It operates in simulation system called rcssserver3D (3D soccer server) which enables two teams of agents to play a soccer game against each other. The 3D soccer server provides a fully distributed and real-time multi-agent environment where all teammates need to cooperate to achieve the common goal of winning the game. There are several requirements for the agent design:

- . What the world is like now.
- . What action agent should do?
- . What is will like if an agent had action on the world.

With attention to the agents sensing from the world and decision and effecting on world[3], Each agent's main loop is based on several simple actions which he can execute in each sensation and acting cycles. In each cycle the agent checks if the ball is kickable or not. The agent kicks the ball to the opponent's goal if he is standing in the right position for scoring toward the opponents goal, otherwise he will adjust himself in a way so that he can score in near future. Movement of the agent is defined in a way so that the agent accelerates for a while and breaks after a number of cycles so that he stops right behind the position he is moving to. The agents would move to their strategic positions when they should not move to the ball or kick the ball toward the opponent's goal.

### 4 Communication

As the basic simulation engine is based on SPADES, we use the SPADES agent library from P. Riley [4] to handle the communication with the server. The agent waits for a message from the server and updates its world model according to it. Then the player selects an action based on the new information, sends it to the server, and starts waiting for a new message from the simulator. This could be changed to a multi threaded model to decide upon arrival of a new message whether to finish the current decision process or to start a new one based on the new information.

### 5 World Model

The world model's function is to maintain an approximately precise mirror of the real world and provides the agent with all information he needs to know, e.g. positions or velocities of all objects on the field. It Parses the message received from the server, and updates the world model.

### 6 Movement

The small time we had in developing the soccer3D team was enough to show us how important movement was in a soccer3d simulation game. Therefore deep calculations were needed so that our player would accelerate and break in the right directions to move to a certain point on the field. If the movement is not considered correctly the agents would simply pass the point or stop behind it, thus it is a key factor in a soccer3D match. We simplified the problem by assuming that the starting speed is the same as the breaking speed, in fact the breaking speed is an opposite vector of the starting speed. We assumed that the player speeds up to the maximum speed possible with the starting drive vector before breaking, our experiments showed that around 35 sensations of acceleration with the starting speed would result in that. We found out that if the player speeds up for

35 cycles with a starting speed 10 cycles of breaking with the same speed (speeding in the opposite angle) would stop him. By testing different speeds and breaking we found out that a linear relation could be found between the distance player travels by a starting speed for 35 sensation and breaking for 10 cycles with different speeds. Therefore we calculated the distance player traveled for drive vector (1, 0,0) which was around 0.451 and each time the players used this constant to determine their first speed. E.g. if the player wants to travel 45 meters then he should accelerate for 35 sensations with the drive vector length of  $45/0.45$  ( Drive(100,0,0) ) and he should break after that for ten cycles with the drive vector in opposite angle ( Drive(-100,0,0) ).

## 7 Conclusion and Future Work

The Robobahae soccer simulation team was born to work on artificial intelligence (AI) and implementation of algorithms and AI methods . We want to approach the agents which can think like real soccer players. But several problems were ignored since there was not enough time to implement them in systematic way e.g. Neural Network, Fuzzy Logic, and Generic Algorithms. We want to upgrade our low-level and optimize are high-level code.

We hope to approach this goal as soon as possible. We are a new member of the Robocop family and also our team has the first experience in the competitions. Our future work is based on fixing the low levels of our code in a better way so that better high levels and decision layers could be based on that.

## References

1. Riley, P.: SPADES System for Parallel Agent Discrete Event Simulation, User's Guide and Reference Manual for Version 0.91
2. Koegler, M.: Simulation and Visualization of Agents in 3D Environments.
3. S.J.Russel –Peter Norvig , Artificial Intelligence: A Modern Approach (2nd Edition) , Prentice-Hall, 1995
4. Riley, P., Riley, G.: SPADES — A Distributed Agent Simulation Environment with Software-in-the-Loop Execution. In: Winter Simulation Conference Proceedings (2003) 817–825