3T Architecture for the SBCe Simulator Team

Eslam Nazemi, Mahmood Rahmani, and Bahman Radjabalipour

Electrical and Computer Engineering Faculty
Shahid Beheshti University
nazemi@cc.sbu.ac.ir
{m-rahmani, b-radjabalipour}@ce.sbu.ac.ir

Abstract. In this paper we briefly describe the process of action control used by the agents of the SBCe Simulator team. It is based on 3T Architecture, which helps to coordinate planful activities with real-time behavior for dealing with dynamic environments.
Our main goal in this project was to achieve some features of teamwork in the multi-agent system of simulated RoboSoccer such as Combination Play and Dynamic Positioning. To achieve this, SBCe used some real soccer rules and knowledge. SBCe's low level structure is based on the CMUnited-99.

1 Introduction

The design of intelligent agents is an important research direction within Multi-Agent System (MAS), where the behavior of a society of agents is described by modeling the individuals and their interactions from a local, agent-based perspective. Thus, finding appropriate architectures for these individuals is one of the fundamental research issues within agent design. Our main goal in this project is to achieve an architecture that supports some teamwork features of the multi-agent environment, such as Dynamic Positioning [2] and Combination Behavior. To achieve this, the SBCe uses the 3T Intelligent Control Architecture [5]. Also it uses some real soccer based on PCB (Pressure-Cover-Balance) System and MACP (Multi-Agent Combination Pattern) [6]. We argue that these techniques are very efficient in the coordination of agents in a team.
Section 2 discusses about the team development. Section 3 describes the agent architecture. Section 4 deals with the main concepts of our team, namely PCB and MACP. Finally we present future work and conclusions.

2 Team Development

Team Leader:

Eslam Nazemi
- Iran
- Faculty member

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 487–490, 2002.

Team Members:

Mahmood Rahmani
- Iran
- Undergraduate student
- In charge of the design and implementation of the Virtual Soccer Team "SBCe"; the body of his B.Sc. project at the university.

Bahman Radjabalipour
- Iran
- Undergraduate student

3 Agent Architecture

The 3T architecture separates the general agent intelligence problem into three interacting layers or tiers (and is thus known as 3T). Skill layer that is a dynamically reprogrammable set of reactive skills coordinated by a skill manager. Sequencing layer that is a sequencing capability that can activate and deactivate sets of skills to create networks that change the state of the world and accomplish specific tasks. Planning layer that is a deliberative planning capability that reasons in depth about goals, resources and timing constraints.
We have modified the 3T that acts as a simulated robot. This is being accomplished by augmenting the skill layer of the architecture with a set of primitives or retrieving commands that are accepted for SoccerServer [1].
We have put our Tactic layer in the Sequencing layer. The Tactic layer activates a specific set of skills in the skill level (reactive layer). The Tactic layer will terminate actions, or replace them with new actions when the monitoring events are triggered, when a time out occurs, or when a new message is received from the deliberative layer indicating a change of plan.
We have developed our Strategy layer in the Planning layer that operates at the highest possible level of abstraction so as to make its problem space as small as possible. Figure 1 shows the relationship of the three layers and their interactions.
We have developed Skill Manager which acts as an interface between a set of situated skills and the rest of the architecture. Situated skills represent the architecture connection with the SoccerServer. For example, one might develop a situated skill for kicking the ball. Such a skill will only be useful if the ball is currently kickable. In other situations the skill will fail. The Skill Manager also handles the interface with the sequencing system by providing all of the communications and asynchronous events that the Sequencer (Tactic layer) needs in order to stay coordinated with the skills.
The SBCe's low-level skills are almost entirely based on the source code of CMUnited-99 team [3] with a substantial modification in order to be able to include the Skill Manager.
To accomplish tasks that the agent routinely performs, the architecture has a sequencing system. In our case, the Sequencer is a tactic interpreter. In it's simplest form a tactic is a description of how to accomplish tasks in the game.

The Sequencer Interpreter runs a continuous loop installing new goals on its agenda, deinstalling old goals that have been achieved or that have failed, activating or deactivating skills and watching for responses from event skills. On each cycle of interpreter, for each skill that needs activation or deactivation, an activate or deactivate message is sent to the Skill Manager which acknowledges the messages and activates or deactivates the skills. Usually the Sequencer will have requested activation of event skills; and in those cases on each cycle it listens for a response from the Skill Manager whether an event has occured or not.

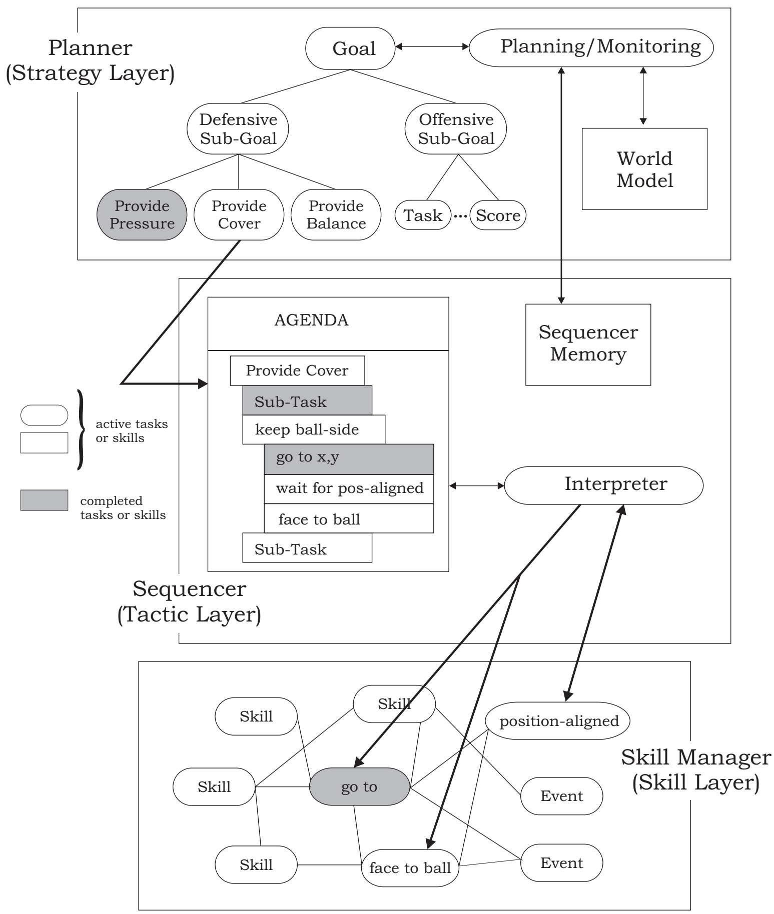

Figure 1. 3T Intelligent control Architecture

4 Defensive and Offensive Systems

SBCe uses the defensive positioning system called "PCB" and the offensive system called "MACP".

In the PCB system the fastest agent to the ball possessor provides **Pressure** on the ball. The second closest agent must provide **Cover**. Other agents will provide **Balance** to the defense. In this system, in general, a defender stands **goal-side and ball-side** of its mark and a midfielder stands ball-side.
There are a number of different ways in which an off-ball player can provide support for an on-ball attacker. These are called the Multi-Agent Combination Pattern or Combination Play. For example, in the attacking positions, the basic two-agent combination pattern can be divided into four basic categories, including passes to the side of the defender:

- Passes behind the defense (slotted or through pass — diagonal run).
- Passes to the side of the defense (give-and-go — overlap — wall pass).
- Passes in front of the defense (square balls — drop passes).
- Faked passes and other tricks (take over — dummy run).

5 Future Work and Conclusions

The modularity of the architecture provides several benefits. The bottom-up development allows one to add functionality incrementally to the agent.
Having achieved this framework, we will in future investigate the integration of other AI disciplines. For example machine learning techniques can be investigated from case-based reasoning in the planning layer to reinforcement learning in the skill layer [4].

Acknowledgment. We like to thank CMU for their low level source code which was used as a base of our team.

References

1. Ituki Noda, Hitoshi Matsobara, Kazuo Hinaki and Ian Frank, SoccerServer: A Tool for research on multiagent system- Applied Artificial Intelligence, 12: 233-250, 1998.
2. Luis Paulo Reis, Nuno Lau, FCPortugal Team Description, RoboCup 2000 Simulation League Champion, 2000.
3. Peter Stone, Patrick Riley and Mannuela Veloso, Cmunited-99 Source Code, Accessible from http://www.cs.cmu.edu/ pstone/RoboCup/CMUnited99.sim, 1999.
4. R. Peter Bonasso and David Kortenkamp, An intelligent agent architecture in which to pursue robot learning. In proceedings of the MLC_COLT'94 Robot Learning Workshop, 1994.
5. R. Peter Bonnaso, David Kortenkamp, David P.Miller and Marc Slack, Experiences with an Architecture for Intelligent, Reactive Agents. Intelligent Agents II, Springer: 187-202,1995.
6. Shel Fung, Basic Coaching Manual, Accessible from http://www.usc.mun.ca dgraham/manual, 1999.