Avan Simulation Team

Mahmood Rahmani, Saeed Chitgar, Alireza Fasih,
Seyyed Mehdi Hatami, Alireza Mohammad Shahri

Introduction

This paper presents an overview of Avan's approach to build multi-agent systems and activities in RoboCup. Our scientific approach in working on multi-agent systems and distributed intelligence is based on *Planning*. We have paid attention to **3T** as a known autonomous agent control architecture. In this way some strategies have been selected to provide significant assistance in the design of autonomous intelligent control. The most important items are: *Modularity* for simplifying both design and control, *Hierarchically Organized Action Selection* for focusing attention and providing prioritization when different modules conflict, and *Environment Monitoring System* for allowing attention to shift and priorities to be reevaluated [1].

3T architecture have been selected to improve some features of Avan's autonomous agents, such as communication, planning, action scheduling, execution monitoring, coordination and learning. Reactive Action Packages (**RAP**) is an architecture that has been used for the middle layer of 3T. Our main goal in this project was to create reactive, flexible, situation-driven and coachable plans using RAP architecture [2],[3] .

Debugging and verifying for such a distributed control program is so difficult and directly related to their distributed nature. We have developed a tool called *Task Monitor* for sequencing (RAPS) and control skills tier. It shows hierarchical actions, although it would be possible to display flat action structure (where all actions are leaf actions) [4].

The control system and programming of Avan agents are provided via a GUI called *Plan-Editor,* in which executable actions are treated as basic building blocks that can be chained together to achieve a larger more complex goal in the style of an *Skill Network*. To achieve this and in order to have a coachable team, a language based on Clang has been used, with some conventional loops and conditional selection. The *plan file* is the output of the Plan-Editor in Clang and represents the behavior of the agent. For an agent in the plan file there is a set of capabilities (potential objective or goal) and a collection of reuseable generic behaviors that may be planned and executed to achieve the objectives [5].

Members

Stuff:

Dr. Alireza Mohammad Shahri, <shahri@iyst.ac.ir>

Team leader:

Mahmood Rahmani, <rahmani_mahmood@yahoo.com>

Team Members:

Saeed Chitgar, <saeed_chitgar@yahoo.com>
Alireza Fasih, <ar_fasih@yahoo.com>
Seyyed Mehdi Hatami, <hatami_mahdi@yahoo.com>

Team Web page

http://www.qazviniau.ac.ir/robocup

References

[1] J. Bryson, *Cross-Paradigm Analysis of Autonomous Agent Architecture*, Journal of Experimental and Theoretical Artificial Intelligence, Volume 12, Number 2, April 2000

[2] P. Bonasso, D. Kortenkamp, D. P. Miller and M. Slack, *Experiences with an Architecture for Intelligent, Reactive Agents*. Intelligent Agents II, Springer: 187-202,1995.

[3] R. J. Firby, *Adaptive Execution in Complex Dynamic Worlds,* Ph.D. Thesis, *Yale University Technical Report*, January 1989.

[4] D. Kortenkamp, *Designing visualization tools for a distributed control architecture*

[5] J. R. Graham, K. S. Decker, M. Mersic, *DECAF - A Flexible Multi Agent System Architecture*, 2000.