

Team Description RoboLog Koblenz 3D ^{*}

Heni Ben Amor, Joschka Bödecker, Anita Maas, Irene Markelic, Jan Murray,
Oliver Obst, Christoph Ringelstein, and Markus Rollmann

Universität Koblenz-Landau, AI Research Group, D-56070 Koblenz
{amor, jboedeck, amaas, markelic, murray, fruit, cringel, rollmark}@uni-koblenz.de

Abstract. This paper describes our first efforts of developing a new team for RoboCup Simulation League 3D. We are aiming at creating methods to implement physical multiagent systems that on the one hand mid and long term goals while staying reactive on the other hand. Ideally, a team of agents can be created in a modular way so that it is effortless to replace single agents with other ones. Similarly, it should be easy for the user to specify the goal of the system. First steps towards such a flexible system is to create an architecture that facilitates this.

1 Introduction

This is a description of RoboLog3D, our team for 3D RoboCup simulation league. One of the first steps towards a team within this new league is to create an architecture for an interface to the simulator and a way to organize the different agent behaviors.

Because the architecture of our 2D team is not very modular, we decided not to directly port the RoboLog 2D kernel for the new team. But instead of creating the architecture from scratch, we are using the class object framework (*zeitgeist*) coming with the simulator. First steps towards this approach can already be seen in the sample client coming with the simulator, which was also developed by us.

2 Reusing the Simulator Architecture for the Team

Zeitgeist is the class object framework that was developed by us for the 3D simulator [3]. Here, class objects are basically class factories that allow to create instances of classes during run time based on a string interface. Additionally, with *zeitgeist* we can request type and interface information from object instances, so that scripting languages can be attached easily.

Another part of *zeitgeist* is a kind of registry for class objects. *Zeitgeist* provides a tree-like structure to manage class objects; these can be accessed via the path name through the tree. This kind of design can also be found in the resource manager architecture of the QNX operating system [4].

By implementing all classes of the RoboLog3D kernel in the *zeitgeist* framework, it is possible to define interfaces for basic player modules in a generic way. Later on, it

^{*} This research has been partially supported the special research program DFG-SPP 1125 funded by the German Research Council (DFG).

is possible to exchange single modules without breaking any existing code, as long as the defined entry points and interfaces are maintained. In the context of a soccer agent, modules are for example the world model holding the data the agents perceive or single behaviors.

3 Important Agent Behaviors

Work at our new team has just started; at the time of writing this description we are about to implement some of the basic behaviors of our team. To our opinion, a key functionality for a 3D team will be the behavior to approach a certain point on the soccer field as fast as possible, and, similar to that, the behavior to approach the ball. Other than in the 2D simulator, the 3D simulator uses a physics engine to calculate velocities. Because of this, formulas for updating positions and velocities are more complicated than in the 2D case, and they are more or less hidden to the user. This means that it is more difficult to exactly calculate future ball positions; other means for doing that have to be employed. We are currently experimenting with combining two approaches which already have been tested by us separately in the 2D case: The first approach is an interception method based solely on qualitative data, see [5] for details. The other approach is a way to describe agent actions as kind of steering behaviors, which we have used in 2D simulation league for example to dribble with the ball while avoiding obstacles [1].

4 Agent World Model

Because agents in 3D soccer simulator have a 360 degrees view field, data agents get about other objects is more complete than in the 2D case. On the other hand, agents do not get velocity information from 3D soccer simulator, which makes it necessary to keep track of object locations to calculate the velocities. Velocities of objects are especially important for steering, because acceleration takes a few times longer than in the two dimensional simulation. To estimate the agent's position, we intend using the approach described in [2], which we also implemented for our 2D team. Currently however no effort has been taken into developing a world model yet; to test our behaviors we are exploiting the 3D simulator features to get full information about the environment. Later this approach will still be useful to test the quality of the estimated data.

References

1. Heni Ben Amor, Oliver Obst, and Jan Murray. Fast, neat and under control: Inverse steering behaviors for physical autonomous agents. *Fachberichte Informatik 12–2003*, Universität Koblenz-Landau, Universität Koblenz-Landau, Institut für Informatik, Rheinau 1, D-56075 Koblenz, 2003.
2. Margrit Betke and Leonid Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, April 1997.

3. Marco Kögler and Oliver Obst. Simulation league: The next generation. In Daniel Polani, Andrea Bonarini, Brett Browning, and Kazuo Yoshida, editors, *Proceedings of the RoboCup Symposium 2003*, July 2003. An LNAI version of the proceedings will appear later.
4. QNX Software Systems Ltd. QNX Neutrino Realtime Operating System: System architecture. http://www.qnx.org/developer/docs/momentics_nc_docs/neutrino/sys_arch/about.html, 2003.
5. Frieder Stolzenburg, Oliver Obst, and Jan Murray. Qualitative velocity and ball interception. In Matthias Jarke, Jana Köhler, and Gerhard Lakemeyer, editors, *KI-2002: Advances in Artificial Intelligence – Proceedings of the 25th Annual German Conference on Artificial Intelligence*, number 2479, pages 283–298, Berlin, Heidelberg, New York, 2002. Springer.