RoboLog Koblenz

Jan Murray, Oliver Obst, Frieder Stolzenburg

Universität Koblenz-Landau, Fachbereich Informatik
Rheinau 1, D–56075 Koblenz, GERMANY
{murray,frvit,stolzen}@uni-koblenz.de

1 Introduction

The RoboCup scenario yields a variety of fields of research. The main goal of the RoboLog project, undertaken at the University of Koblenz in Germany, is the specification and implementation of flexible agents in a declarative manner. The agents should be able to deal with the real-time requirements but also be capable of more complex behavior, including explicit teamwork. To this end, we develop a *declarative multi-agent script language* for the specification of collective actions or intended plans that are applicable in certain situations. The agents should be able to recognize such situations by means of *qualitative spatial reasoning*, possibly supported by communication.

The RoboLog team is based on an architecture with four layers, where layer 1 deals with the synchronization with the SoccerServer and realizes the low-level skills. Layer 2 handles qualitative spatial reasoning. More complex actions and teamwork are realized in layers 3 and 4. The focus of this paper is laid on the first layer, especially on position determination (see Sect. 3). For a more detailed description of the higher layers see the paper *Spatial Agents Implemented in a Logical Expressible Language* in this volume.

At the RoboCup-99 competition RoboLog Koblenz played in Group C in the Simulator League. The team lost only one match and managed to achieve a draw in the other three. Unfortunately, this did not suffice to enter the elimination round. The match RoboLog Koblenz vs. IALP (1–1) was the most interesting match we played, in so far as it was the *only* drawn match in the whole competition, that did not end with a score of 0–0.

Team Leader: Frieder Stolzenburg
Team Members:

Dr. Frieder Stolzenburg
- Universität Koblenz-Landau
- Germany
- Researcher
- attended the competition

Oliver Obst, Jan Murray
- Universität Koblenz-Landau
- Germany
- Master Students
- attended the competition

Björn Bremer, Michael Bruhn, and Bodo van Laak
- Universität Koblenz-Landau
- Germany
- Master Students
- did not attend the competition

Web page: http://www.uni-koblenz.de/ag-ki/ROBOCUP/

Fig. 1. The RoboLog Koblenz team.

M. Veloso, E. Pagello, and H. Kitano (Eds.): RoboCup-99, LNAI 1856, pp. 628–631, 2000.

2 Team Development

The RoboLog Koblenz players were implemented by a team of 3 to 5 people. Most of them were students preparing their course work or diploma theses. We were able to conduct several test games with different scores on our local network—a 100 MBit Ethernet, sometimes using a Sun Ultra-Enterprise with 14 processors *à* 336 MHz and 3 GB main memory. Fig. 1 lists the team members of RoboLog Koblenz.

3 World Model

For each agent, the RoboLog interface—written in *C++*—requests the sensor data from the SoccerServer. By this, the agents' knowledge bases are updated periodically. If some requested information about a certain object is currently not available (because it is not visible at the moment), the most recent information can be used instead. Each agent stores information about objects it has seen within the last 100 simulation steps. So we can think of it as the agent's memory or recollection. The passing of time can be modeled in several ways with RoboLog. It provides various means for creating snapshots of the world and defining an event-driven calculus upon them, translating the agent's view of the world into a propositional, qualitative representation in *Prolog*.

The RoboLog system provides an extensive library that makes precise *position determination* possible. The whole procedure is able to work even when only few or inconsistent information is given. First of all, an agent has an egocentric view of the world. The actual sensor data provide more or less precise information about the positions of other agents, landmarks and border lines relative to the agent's position and orientation. The RoboCup scenario yields a frame of reference with absolute coordinates, given by the geometry of the playing field. Knowing the absolute position allows the agent to identify its own location wrt. other objects on the map, even if they are not visible at the moment. Secondly, an absolute frame of reference is helpful in order to communicate with other agents in situations where cooperative actions are appropriate.

The first localization method in RoboLog—introduced in [1]—requires (only) three or more directions to visible landmarks relative to the orientation of the agent to be known. Provided that at least three of them and the position of the agent neither form a circle nor lie on a straight line, the absolute position and orientation of the agent can be computed with a time complexity that is only linear in the number of landmarks. If the corresponding equation system in complex numbers is over-determined, and the data is noisy, the procedure estimates the position applying the least squares method. Without reference to a third landmark, the actual position in question lies on a segment of a circle. This is shown in Fig. 2.

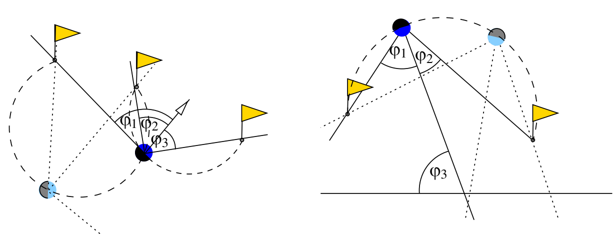

Fig. 2. Self-localization by angle information.

However, we also need procedures that are able to work, if only limited sensor data is available. If less than three landmarks or border lines are visible, there are two cases where the position of the agent is uniquely determined even then. Firstly, knowing the distances and angles to two landmarks is sufficient. Secondly, the position can be determined if the relative position of one landmark and of a point on a border line where the center line of the view axis of the agent crosses this line is given, provided that both points are not identical. In addition, by keeping track of the movements of the agent, it is also possible to estimate the current position. All these methods are implemented in the RoboLog module, in order to lay a solid quantitative basis for the qualitative reasoning in the higher layers.

4 Communication

The RoboLog agents use communication in order to clarify situations. Very frequently an agent is not able to recognize a situation or its own role therein, because of insufficient information. These drawbacks can be overcome by the use of communication. If an agent recognizes a situation in which a multi-agent script can be executed, it sends this fact to the others by communicating a Prolog predicate, that is executed by *all* RoboLog agents receiving it. If this command turns out to be relevant for a player, i.e. the agent has to take part in a collaborative action, it will be executed. Otherwise the agent just ignores the message. For this communication to work the agents rely on the fact, that their internal structure is the same, and more that they are all implemented in Prolog. We use communication especially for initiating teamwork such as double passing.

5 Skills

The RoboLog agents are equipped with several basic skills like *dribbling* or *kicking* to a certain position (both programmed in Prolog). These skills are not very sophisticated, which proved to be a disadvantage during the RoboCup-99 competition. All agents are clones of each other except for the goalie, which we will now describe in greater detail.

The goalie's main objects are to keep between the ball and the goal and not to lose sight of the ball. Its behavior depends mainly on the movement and (qualitative) position of the ball. The agent partitions the playing field into several regions, e.g. *opponent half* or *penalty area*. Based on the region the ball is currently in and the extrapolation of its trajectory, the goalie chooses a position to move to. It also takes into account, if the ball is owned by the opponent team or in the middle of a shot. If the ball is in a shot, the agent tries to intercept it, but otherwise it just blocks the way from the ball to the center of the goal line. After any movement it turns its neck towards the expected position of the ball, thus keeping it in sight.

6 Strategy

The underlying basic strategy of the RoboLog Koblenz team simply is moving towards the opponent goal (possibly avoiding obstacles) and trying to score. If a RoboLog agent has the ball, it tries to recognize the current situation as one in which a (multi-)agent script is applicable. In this case the script will be executed, yielding behaviors like (double) passing. Otherwise the agent sticks to the default strategy until a script can be applied.

When the other team has the ball, the players try getting it back in the following manner. Each agent checks, if at least two team mates are nearer to the ball. If that is not the case, it tries to intercept the ball. This "double attack" proved useful as it prevents situations in which no agent goes to the ball due to sensor uncertainty. Players, that are too far away to be involved, return to their special home positions.

7 Special Team Features

The RoboLog Koblenz agents make use of explicit teamwork during a game. In many other teams multi-agent cooperation emerges just as a consequence of the behaviors of the single agent. RoboLog agents, however, actively try participating in collaborative behavior specified in a multi-agent script language. The agents explicitly make use of communication to tell other players to take part in a collective action. A more detailed description of the multi-agent language can be found in the paper *Spatial Agents Implemented in a Logical Expressible Language* in this volume.

8 Conclusion

The RoboLog system provides a clean means for programming soccer agents declaratively. Cooperative Behavior can be modelled explicitly by means of multi-agent scripts. A qualitative spatial representation allows agents to classify and abstract over situations. However, the imperfection of the low-level skills turned out to be a major disadvantage of RoboLog Koblenz. The current approach only allows to react *ad hoc* to external events or interrupts. Future work therefore includes reimplementing a part of the RoboLog interface in order to enhance the basic skills as well as extending them by a number of additional abilities.

Promising areas of further research are the specification of a more sophisticated communication paradigm and the use of logical mechanisms within the lower levels of our approach. Deduction could be used to build a more complete view of the agent's world. In addition, the robustness of the decision process can be improved by means of defeasible reasoning and the use of any-time reasoning formalisms. The application of these techniques to real robots is one of the next steps of our research activities.

References

[1] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, Apr. 1997.

[2] A. S. Rao. AgentsSpeak(L): BDI agents speak out in a logical computable language. In W. van de Velde and J. W. Perrame, editors, *Agents Breaking Away – 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, LNAI 1038, pages 42–55, Berlin, Heidelberg, New York, 1996. Springer.