CoPS-Team Description

R. Lafrenz, M. Becht, T. Buchheim, P. Burger, G. Hetzel, G. Kindermann, M. Schanz, M. Schulé, and P. Levi

Institute for Parallel and Distributed High Performance Systems (IPVR)
Applied Computer Science - Image Understanding
University of Stuttgart, Breitwiesenstr. 20-22, 70565 Stuttgart, Germany
robocup@informatik.uni-stuttgart.de

Abstract. The control software of the robot soccer team CoPS is designed as a multi-agent-system. The basis for a cooperation between the robots is a suitable environment model based on uncertain sensory data and communication.

1 Introduction

In the last two years, the CoPS-team (Cooperative Soccer Playing Robots) competed successfully in the RoboCup tournaments. The robots of the team are shown in fig. 1.

Fig. 1. The CoPS team of soccer robots (March 2001)

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 635–638, 2002.

The control software of our robot team is designed as a Multi-Agent-System ([1], [2]). Although applied to many domains, the full potential of this paradigm developes especially in situations where decisions have to be made upon uncertain data or partial information.

2 System Architecture

In our architecture, not only a whole robot, but also major software components of a robot are modelled as agents. Each single robot has the internal structure shown in fig. 2.

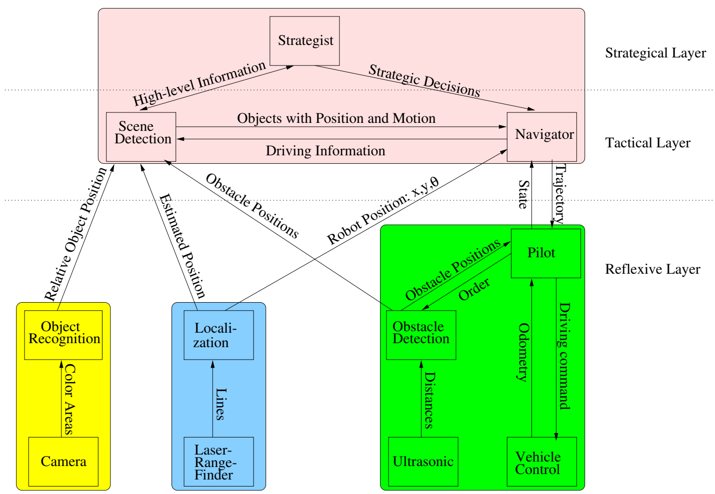

Fig. 2. System architecture

Logically, the software is structured in three layers: reflexive, tactical, and strategical. These layers consist of elementary agents which are described in section 3. The assignment of an agent to a layer helps to design and understand the system.

The Reflexive Layer The elementary agents in the reflexive layer control the hardware of the robot and perform sensing and acting, e.g. *Camera* or *Vehicle Control*. Other kinds of *elementary agents* in the reflexive layer fuse the data of different type or transform sensory data into information, e.g. the *Object Recognition*.

The *Pilot* has to carry out the orders given by the Navigator. As an agent, the Pilot is responsible for leading the robot to a given destination. It has to avoid collisions, and needs access to the results of the Object Recognition, which are provided by the later described Scene Detection. It is able to perform minor corrections to the path planned by the Navigator.

The Tactical Layer In the tactical layer we find two agents, the *Scene Detection* and the *Navigator*. The Scene Detection is responsible for storing and updating the information about the current situation. As an agent, the Scene Detection collects all relevant information form the sensory agents. This local environment model represents the scene, as seen from a single robot. In order to obtain robustness and a better understanding of the current scene, communication within the team is necessary. The communicated information is fused with the local information and also stored in the Scene Detection.

One of the basic requirements for that is the ability to distinguish between team-mates and oppenents. By assuming same shape and colors for robots of both teams, this distiction is impossible without cooperation, in other words, indirect information. But, by exchanging position information for all robots within the team, the recoginized objects can be identified as team-mates or opponents. In addition, the position estimation of a team-mate can be verified.

The Scene Detection answers requests of the Navigator. The navigator generates abstract driving commands (like search ball, dribble ball, etc.) which are executed by the pilot.

The Strategical Layer In our concept, there is only one agent placed in the strategical layer: the *Strategist*. It reacts to infrequent external events (e.g. start of game, goals) and decides on the way of further playing. Thus it decides on the long-term strategies of the game, for example offensive or defensive playing mode. Therefore, it communicates with two other agents, which are placed in the tactical layer: the Scene Detection and the Navigator. In addition, the Strategist is able to communicate with the strategists of other players in the team. Sometimes a single player is not able to recognize the current situation. Therefore we provide communication beyond the boundaries of a single robot. In principle, any elementary agent is able to communicate with any other agent. Hence, very complex and powerful negotiations are possible.

3 Structure of Elementary Agents

The elementary agents are the autonomous function moduls of the system. They get their functionality from their plans. These plans control the interplay of the sensing and acting. The *body* of the elementry agent is supervised by the *decision unit*.

Fig. 3 shows the structure of an elementary agent. The *decision unit* is responsible for negotiation between the agents and for internal configuration. Data is transposed through seperate data channels. The configuration of this channels is done by the *decision unit*. *State* contains the current state of the elementary agent. *Model* administrates the statical or lernable knowledge. The internal cycles range from simple and fast react behavior (controller) to complex planning of each step using the knowledge of the *model*.

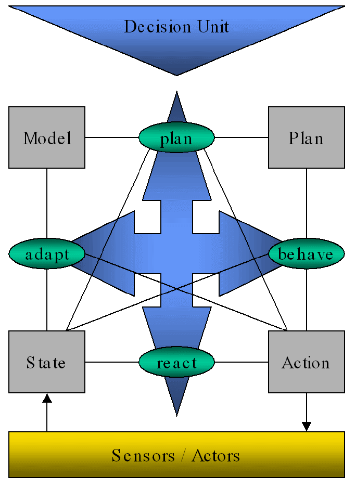

Fig. 3. Structure of an *elementary agent*

4 Conclusion and Outlook

In this paper, we have described the system architecture of our soccer robots. Currently, our research interests concentrate on cooperative environment modelling, cooperative behavior, and we plan to include reinforcement learning in our system.

References

[1] M. Becht, M. Muscholl, and P. Levi. Transformable multi-agent systems: A specification language for cooperation processes. In *Proceedings of the World Automation Congress (WAC), Sixth International Symposium on Manufacturing with Applications (ISOMA)*, 1998.

[2] R. Lafrenz, N.Oswald, M.Schulé, and P. Levi. A cooperative architecture to control multi-agent based robots. In *Proceedings of PRICAI*, 2000.