Middle Robots League, Team *Stuttgart-CoPS*, pages 195–199
http://www.ep.liu.se/ea/cis/1999/006/31/

CoPS-Team Description

Stuttgart-CoPS

Norbert Oswald, Michael Becht, Thorsten Buchheim, Günter Hetzel, Georg Kindermann, Reinhard Lafrenz, Matthias Muscholl, Michael Schanz, Moritz Schulé, Paul Levi

NO, MB, TB, GH, GK, RL, MM, MS, MS, PL: University of Stuttgart, Germany

Abstract. *This paper presents the hardware and software design principles of the medium size RoboCup Team* CoPS *which are developed by the image understanding group at the Institute for Parallel and Distributed High Performance Systems (IPVR) of the University of Stuttgart. By adapting already successfully tested multiagent software concepts by our group to the domain of robotic soccer we intend to improve those concepts at the field of realtime applications with uncertain sensory data.*

1 Introduction

Multiagent theory has become a popular research area in the context of artificial intelligence. Although applied to many domains, the full potential of this paradigm developes especially in situations where decisions have to to be made upon uncertain data or partial information as e.g. in robotics. Deficiencies of a single agent in perceiving its environment and having only partial knowledge about its surroundings shall be compensated by its ability to exchange information with other agents.

Furthermore multiagent systems support the idea of the whole being more than the sum of its parts, meaning that the problem solving potential of a multiagent system exceeds by far the capabilities of the agents within the system. This can be achieved by providing the agents with proper communicational abilities to negotiate and coordinate their actions.

The research group **Image Understanding** of the **Institute for Parallel and Distributed High Performance Systems** has been working for several years in the field of multiagent-systems developing a multiagent architecture [1], a theoretical framework for cooperating agents [2] and applying cooperative concepts in computer vision [3]. With our medium size robotic soccer team *CoPS* (Cooperative Soccer Playing Robots) those developed concepts are tested and adapted to the real world application of a soccer match.

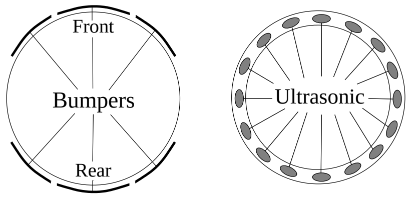

Figure 1: Pre-installed Sensorics of the Super Scout

2 Hardware Equipment

Our RoboCup Team consists of 5 Nomad robots of type *Super Scout* equipped with additional sensorics and a kicking device. The onboard computer system of the Super Scouts consists of a Pentium 233 MMX with 64 MB and a hard disc of 4 GB capacity. The two wheel differential drive allows robot motion with a maximum speed of $1m/s$ and an acceleration of $2m/s^2$.

Each of these robots is already equipped with odometric sensors, a tactile bumper ring of 6 independent sensors and 16 ultrasonic sensors arranged around the outside perimeter of the robot. The 6 tactile sensors allow a coarse localization of physical contacts with the environment in the front-left/center/right or the rear-left/center/right of the robot.

The ultrasonic sensors are merely used for obstacle avoidance purposes. Nevertheless, there are situations where physical contacts with the environment are inevitable. In those cases the bumper sensor information is needed to react appropriately (e.g. stopping a backward movement, if the bumper sensor in the rear is activated).

2.1 Modifications and additions

We installed a 1/3″-Chip CCD camera with a 752x582 resolution delivering a video signal to a MATROX frame grabber on the Pentium board. By image processing algorithms the ball is extracted from the grabbed pictures and its current position relative to the robot is estimated.

As the precision of the odometric sensors degrades constantly with every movement of the robot, we equipped each robot with a SICK laser range finder [1] for a more robust and exact self localization. Within an angle of 180° the range finder provides depth information with a resolution of 0.5° and an accuracy of 5cm. To derive the actual position from the laser data the measured laser image is iteratively rotated and shifted until it fits best a predefined model of the football field. For a faster computation of the robot's position by laser we make use of the odometric data which is taken as starting point for the iterative search. After accomplishing the self localization the odometric values are updated with the new values. However, because of this rather time consuming procedure, a localization by laser should be applied as rarely as possible.

For further computational power a Toshiba Portégé 7010 CT Laptop was mounted on top each robot. The Pentium board as well as the notebook

---

[1] We'd like to thank the SICK AG,Germany for lending us three Laser Range Finders

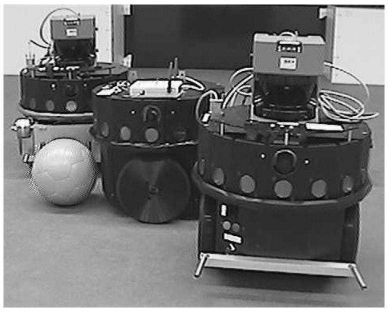

Figure 2: Equipment of a soccer robot (1)

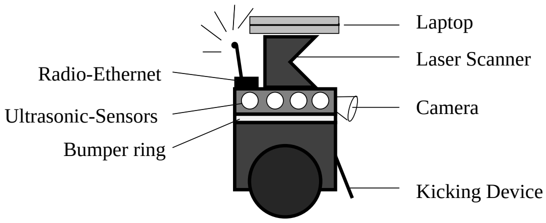

Figure 3: Equipment of a soccer robot (2)

are connected to a radio-ethernet HUB enabling communication between the board and the laptop as well as communication with other robot agents.

For passing and kicking the ball we constructed a special kicking device which is driven by two solenoids. The camera, laser range finder and kicking device are mounted towards the front of the robot players. Only the goalkeeper's sensors are shifted by 90° as it only moves unidirectionally forwards or backwards to defend the goal.

3 Software Concepts

3.1 Agent Architecture

In our software model each robot consists of a set of concurrent software modules, so called elementary agents (EA). Each EA has special plans to perform tasks which can be requested by other elementary agents. We classify EAs into three categories according to the level of abstraction of the tasks they perform.

EAs concerned with simple control tasks of the sensors and actuators are situated at the lowest layer within the architecture, **the reflexive layer**. Reflexive tasks generally involve only simple planning or no planning at all, guaranteeing an immediate response on a request. The Pilot-EA e.g. performs tasks like driving a given route straight ahead and rotation by a given angle, while the Localizer-EA determines the current position by odometric and laser range finder data.

More complex tasks are performed by the EAs at **the tactical layer** of our software architecture. An EA concerned with scene-detection aggregates

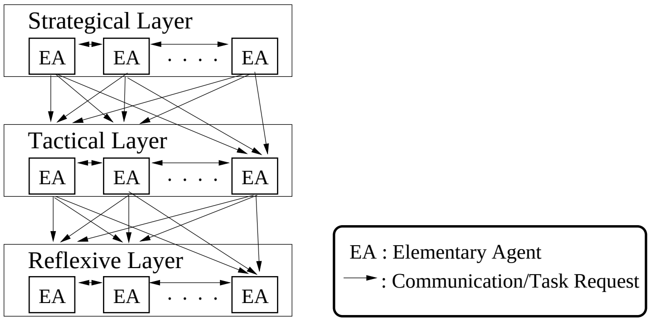

Figure 4: Cooperative Agent Architecture

and evaluates information from the sensors of a robot and builds a model of the environment which can be requested by other agents. A navigator-EA, also located at this level and responsible for vehicle control, is concerned with tasks like *getting the ball* or *moving towards the goal*. Planning such tasks requires information about the environment which is provided by the scene-detection.

Finally, at the strategical layer, the Strategist-EA is concerned with long term goals and team coordination. It generally has to react only on external signals like e.g. *start of the game*, *goal* and *end of the game*. All EAs can send task-requests to each other within the same robot as well as to elementary agents of other robots. Thus, data-uncertainties concerning the environment can be compensated and a means for coordinating joint actions is provided. For a more detailed description of our software architecture we refer to our technical paper [4].

3.2 Implementation Issues

The elementary agents were all designed as separate multi threaded processes on a LINUX system, a communication thread waiting for task requests from other agents and further executional threads processing requested tasks. For interagent communication we used the freely available CORBA implementation MICO which facilitated the distributed agent modelling a lot.

References

[1] P. Levi, M. Becht, R. Lafrenz, and M. Muscholl. COMROS - A Multi-Agent Robot Architecture. In *DARS 3*. Springer-Verlag, 1998.

[2] 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.

[3] N. Oswald and P. Levi. Cooperative vision in a multi-agent architecture. In *LNCS*, volume 1310, pages 709–716. Springer-Verlag, 1997.

[4] N.Oswald, M.Schulé, and R.Lafrenz. A cooperative architecture to control multi-agent based robots. Submitted to : Proceedings of the IJCAI RoboCup Workshop, 1999.