

# The Ulm Sparrows: Research into Sensorimotor Integration, Agency, Learning, and Multiagent Cooperation

Gerhard K. Kraetzschmar<sup>1</sup>, Stefan Enderle<sup>1</sup>, Stefan Sablatnög<sup>1</sup>, Thomas Boß<sup>1</sup>,  
Mark Dettinger<sup>2</sup>, Hans Braxmayer<sup>1</sup>, Heiko Folkerts<sup>1</sup>, Markus Klingler<sup>3</sup>,  
Dominik Maschke<sup>1</sup>, Gerd Mayer<sup>1</sup>, Markus Müller<sup>1</sup>, Alexander Neubeck<sup>1</sup>,  
Marcus Ritter<sup>1</sup>, Heiner Seidl<sup>3</sup>, Robert Wörz<sup>1</sup>, and Günther Palm<sup>1</sup>

<sup>1</sup> Dept. Neural Information Processing,

<sup>2</sup> Dept. Software Engineering and Compiler Construction,

<sup>3</sup> Dept. Mathematics and Economics,

University of Ulm, Oberer Eselsberg, 89069 Ulm, Germany

[sparrows@neuro.informatik.uni-ulm.de](mailto:sparrows@neuro.informatik.uni-ulm.de),

WWW: <http://smart.informatik.uni-ulm.de/SPARROWS/>

**Abstract.** We describe the motivations, research issues, current results, and future directions of THE ULM SPARROWS, a project that aims at the design and implementation of a team of robotic soccer players.

## 1 Motivation and Research Goals

Engineering a team of robotic soccer players that is able to successfully compete in championship games requires truly interdisciplinary research and the effective cooperation of many disciplines. This is our main scientific reason for setting up a local research effort related to ROBOCUP and to implement THE ULM SPARROWS, a team of soccer agents that is currently projected to participate both in the mid-size real robot league as well as the simulation league. A secondary reason is the coincidence of ROBOCUP research issues[1] with mid-term research goals of SMART[2]<sup>1</sup>, a local, large, interdisciplinary joint research project that aims at the development of a complete cognitive architecture for an adaptive robotic agent performing complex tasks. Not the least reasons for starting ROBOCUP activities were their attractiveness to students and its potential for appealing educational activities.

Considering the wide variety of fields involved with ROBOCUP, no single group or team can equally well address all research issues of interest. In order to ensure high-quality research results, both a long-term commitment to ROBOCUP activities and a well-focused research agenda are essential. For our team, the following open research issues are of particular interest: *Learning* elementary football skills from demonstration, by reinforcement or teaching, *Spatial Modeling* of highly dynamic environments including their rapid update by sensor

---

<sup>1</sup> See [www.uni-ulm.de/SMART/](http://www.uni-ulm.de/SMART/) for more information.

interpretation and fusion, *Emergent multiagent cooperation* for achieving team play without explicit communication, *Robot control architectures*, *Neurosymbolic integration* in robot control architectures. Furthermore, we also favor the simultaneous use of *simulated and real-world* testbeds for robotics research, in particular the development of good simulation models of sensors and actuators of (increasingly complex) robotic agents.

Solving these problems, even making good progress, will undoubtedly take many years, and our initial focus is to lay the groundwork on which these problems can be studied.

## 2 Architectural Overview

Developing both a simulation *and* a real robot team at the same time is a serious challenge. In order to permit as much coherence as possible between both development efforts, we try to develop a *common* control architecture to be used by agents in both teams.

The architecture of our agents is structured into a reactive and a deliberative layer (see Figure 1). In the reactive layer, a sensor interpretation module, a dynamically determined set of behavior modules, and an arbiter for prioritizing conflicting primitive actions are concurrently executed. Together, they implement a fast sensorimotor feedback loop and ensure timely response to changes in a highly dynamic environment. The deliberative layer itself consists of three levels: *i*) elementary tasks (behaviors), *ii*) coherent action patterns (combinations of behaviors to form simple playing patterns like passing the ball, shooting a goal, etc.), and *iii*) cooperative action patterns (patterns, where several team players take different roles and interact). On each level, a situation/play pattern associator evaluates the currently perceived situation and matches it against a database of situation/behavior patterns (aka plan library). If a matching pattern is found, it dynamically modifies the set of concurrently executed behaviors in a controlled way. The *elementary task behaviors* implement a set of soccer-oriented reactive action patterns. These patterns are abstractions of short sequences of low-level commands which the agent can execute directly. At the same time, the behaviors provide a certain degree of isolation from low-level details, thereby reducing dependency on particular details of the soccer server protocol or the hardware.

Building upon the behavior level, *coherent action patterns* focus the selection of behaviors to locally coordinated action sequences, which exhibit coherence over time. Thus, action patterns provide situation-oriented partitioning of the space of possible behavior sequences and focus of attention for a single agent.

Coordinated team play is dealt with on the group level. Albeit we assume explicit mechanisms to achieve cooperative behavior of soccer agents, we do neither assume nor rely on explicit communication. Cooperative behavior among multiple agents in games like soccer is an emergent phenomenon. Agents have a database of stored *cooperative play patterns*. Each agent continuously tries to match the currently perceived situation against this database. If it can match

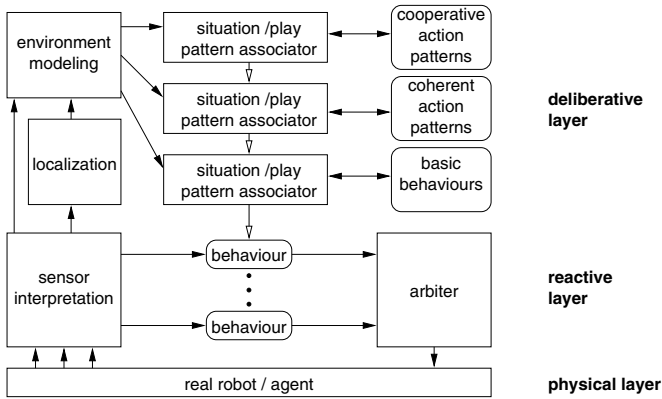


Fig. 1. Architectural Overview

itself *and* at least another teammate with roles in a tactical move, this move is executable. Currently, such a move is executed directly, but situated valuation and selection of several executable moves will be necessary in the future. Whenever no cooperative play patterns are executable or in the case of failures, the agent falls back to using single agent coherent play actions. Currently, all players except for the goalkeeper use the same database of cooperative play patterns.

### 3 The Robot Sparrows: The Middle Size Robot Team

The real robot team of THE ULM SPARROWS consists of four robots: three Pioneer-1 and a tracked vehicle as goalie (see Figure 2). Our robots are com-

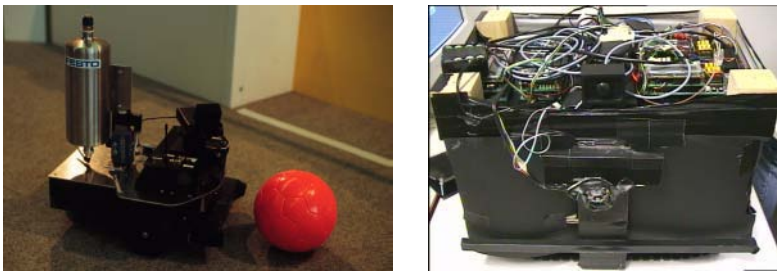


Fig. 2. THE ULM SPARROWS: a field player (left) and the goalkeeper (right).

pletely autonomous, i.e. all computation is performed onboard; no radio links to offline workstations are not needed for play action, but may be added for

debugging purposes. The robots are equipped with a Pentium CPU board in PC 104 technology, including various I/O interfaces, harddisk, and a framegrabber. A wide angle color camera is mounted on a servo providing 180 degrees of pan angle. Thus, our field players do have a neck, which permits (largely) independent control of view direction and movement heading. Together with the wide angle camera, the robot has almost complete view of its surroundings without having to turn its body. This proved to be very useful as our robots proved to be able to detect the ball without long exploratory movements on the field.

The vision system implements a real-time color blob tracking algorithm. Research to investigate the combined use of color and other features like shape or texture is underway.

A real problem for Pioneer robots is kicking the ball. Their acceleration and top speed is too slow to allow for competitive kicks by just pushing the ball. Given the size limitations for the mid-size league robots, designing kickers for the Pioneers is not easy. We opted for a solution using pneumatics. A high pressure air tank was added. An electrically triggerable valve gives air into a pneumatic cylinder, which carries a plastic paddle at its end that actually exerts force onto the ball.

*Movement Control:* The movement of the robots is controlled by a behavior-based reactive control system as sketched in Figure 1. All lower-level action is determined by *behaviors*. The actions a behavior can control are setting the speed or direction of the robot, setting the camera angle, or changing state variables, like *ball-is-visible*.

A pool of mutually exclusive basic behaviors (see bottom of the deliberative layer) are waiting to be activated. Each one of these behaviors implements a small, simple task, like searching for a goal, moving to the ball, or kicking. They all have in common that they cannot be performed simultaneously and must therefore be activated sequentially. An example sequence for shooting a goal could be: *search for goal*, *search for ball*, *move to ball*, *kick ball into goal*. Behavior sequences are invoked by deliberative level, which corresponds to the levels coherent action patterns and group patterns in the common architecture and at the moment holds a number of hand-coded plans.

Aside of the mutually exclusive sequential behaviors, the behaviors in the reactive layer can be activated in parallel. For example, an *avoid* behavior for avoiding collisions with walls or other robots or a *stall* behavior for detecting stalled wheels may be executed simultaneously. However, when the robot approaches the ball, the *avoid* behavior must be temporarily suppressed. The concurrent execution of parallel behaviors is also represented in higher-level plans. We use a very simple fixed priority scheme to arbitrate the proposed actions given by the active behaviors.

*Orientation Estimation:* The absolute position of the robot is not tracked. Only information about the orientation is collected. As soon as the vision system identifies a goal, the orientation is updated accordingly. While none of the goals is visible the odometry gives us a good enough estimate of the orientation. Competition experience proved that this concept suffices for achieving coherent

action patterns. But as CS-Freiburg demonstrated, exact information about the own (and the team members) position is essential for cooperative group play. Our future research will therefore try to extend our capabilities in position estimation.

## 4 Simulation Team

THE ULM SPARROWS simulation team consists of two different variants of agents: a generic field player agent and a goalie agent. The software architecture of both types of agents consists of two modules: *i*) the world model, which collects and integrates the information sent by the soccer server, and *ii*) the action selection module, which itself is divided in three levels as described in Section 2. World modeling and action selection are run in separate threads. This proved problematic as our development platform (Linux) was different from the platform used in Paris (Solaris).

*World Model:* The world model parses the messages from the server and tries to keep a consistent model of the situation of the game by accumulating the information provided by the soccer server into an allocentric representation. By fusion of extrapolated information with the actual snapshots a more complete and stable representation of the current situation is achieved. Future versions of the world model will try to integrate more sophisticated approaches for modeling the uncertainty which is involved in the perception. The competition gave us the important possibility to collect data sets with a variety of opponents, which exhibited a lot of different strategies.

*Action Selection:* The action selection module largely follows the common agent architecture described in Section 2.

*Behaviors:* At the elementary task level a set of robust behaviors is provided which serve as basic building blocks for higher-level coherent single agent action and coordinated group behavior. Behaviors implement a soccer-oriented abstraction of primitive actions by sequencing multiple atomic actions over relatively short periods of time. In order to build a basic soccer player, an agent should at least be able to perform the following list of behaviors: moving with and without the ball while avoiding collisions, dribbling, shooting (to score a goal), intercept, pass to stationary and moving team members. At any given time only one of these behaviors is active. It remains active until the task is believed to be completed. Deciding the termination condition is usually performed by the upper levels of our architecture, but behaviors can themselves provide hints whether they consider their task finished or not, or whether they cannot fulfill the desired action. For example the passing behavior would signal an error if the player does not have the ball.

*Coherent Action Patterns:* The space of all possible situations is partitioned into six prototypical situations, each of which induces a different action pattern. Partitioning the situation space is performed according to the following two criteria: *i*) the distance of the individual player to the ball (near/far), and *ii*) the team currently in possession of the ball (own/opponent/unknown). According to

the result of the match of the actual situation with the six prototypical situations, the player follows a predefined strategy invoking a reasonable behavior of the agent.

*Group Play Patterns:* For emergent multiagent cooperation in soccer, an agent has to recognize situations where cooperative play is possible. For that, the agent must be able to match itself and other team member with different roles required by group play patterns. The group play pattern module examines the overall situation as given by the world model. As soon as it is able to match one of the prestored situation patterns to current perception, the agent identifies its role in this pattern and starts executing the predefined plan, which is bound to this role. Cooperative group play will only emerge, however, if several agents dynamically activate the same group play patterns with the correct role instantiations. While executing a group play pattern, an agent monitors its progress.

## 5 Conclusions

In this paper, we presented an overview on the research done in the THE ULM SPARROWS team as well as the current state of our soccer teams. Most work on teams has been performed within the last 2–3 months, after a sufficient number of students could be interested and engaged itself into team implementation. As all people involved were new to ROBOCUP, getting working teams of robot soccer players (and a collection of software agents) was our main focus for ROBOCUP-98. With all the basic groundwork done, the foundation for doing serious research on the issues discussed earlier is laid. Modelling the environment, the overall agent architecture, and learning schemes for basic skills will be in the center of research within the next year.

The results of Paris showed that our architecture is sufficient to be used for our further research. Minor hardware and software problems that harmed performance in Paris could be identified and resolved with the experiences gained in a real competition.

## References

1. Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. RoboCup — a challenge problem for ai. *AI magazine*, pages 73–85, Spring 1997.
2. Günther Palm and Gerhard Kraetzschmar. Integration symbolischer und subsymbolischer Informationsverarbeitung in adaptiven sensomotorischen Systemen. In Matthias Jarke, Klaus Pasdach, and Klaus Pohl, editors, *Informatik 97: Informatik als Innovationsmotor*, Informatik aktuell, pages 111–120, Heidelberg, Germany, 1997. GI, Springer.