The Paderkicker Team
RoboCup 2009

Bernd Kleinjohann, Philipp Adelt, Willi Richert, and Claudius Stern

Faculty of Electrical Engineering, Computer Science and Mathematics,
University of Paderborn, Germany
`teamleader@paderkicker.uni-paderborn.de`

Abstract. The Paderkickers are a second generation robot soccer team with roots in automotive technology like automotive-capable microcontrollers or communication over CAN bus. Each robot integrates a self-sustained network of decentralized sensor-/actor-coupling nodes, internally interconnected and supervised for fail-safe operation. In this paper we present how we integrate and process the extracted real-time data in our behavior-based system.

1 Introduction

The Paderkicker RoboCup group [1] consists of seven players (Fig. 1) that already participated successfully in the RoboCup world championships 2006 in Bremen as well as in the GermanOpen competitions in 2004–2008. This second generation improves upon the first RoboCup player generation with omniwheels, better vision capabilities and a more distributed internal design. Started as a student project team for graduate students in 2001, the project has a strong emphasis on the education of students of the Unversity of Paderborn. Most students

Fig. 1. Field players

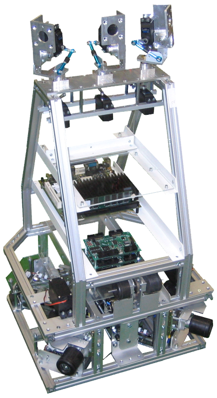

Fig. 2. Aluminium frame construction

come from computer science centric courses with some coming from mechanical or electrical engineering. They work in specialized teams on hardware, software and HW/SW codesign issues. In an initial seminar the students delve into selected topics of the RoboCup domain. Afterwards the smaller teams have to carry out self-chosen topics in the Paderkicker area. Team leaders are Bernd Kleinjohann, Philipp Adelt, Willi Richert and Claudius Stern.

Altogether about 125 students have participated in this project team over the years and thus collected experience how to combine real-time embedded systems with intelligent behavior. Another offering are two courses "Embedded systems" and "Intelligence in embedded systems", held for master students in computer science. In addition to education, the research interests comprise embedded real-time architectures [3, 13–16, 18], real-time vision [14–16, 18] and visualization [6, 5], learning and adaptation from limited sensor data, skill learning and methods to propagate learned skills and behaviors in the robot team [12, 11, 10, 9, 8, 4]. However, our goal is not to carry out research for specific solutions in the robotic soccer domain, but to use and test advanced techniques from different research projects. The Paderkicker platform serves as a testbench for the collaborative research center 614 (SFB 614) and priority program organic computing (SPP OC 1183, both funded by the Deutsche Forschungsgesellschaft). Furthermore, the knowledge in vision, motion and object tracking has been used in the AR PDA (Bundesministerium für Bildung und Forschung) project [7]. We will now introduce the architecture of the field players.

2 Robot outline

The current generation of the Paderkicker robots is equipped with an omnidirectional drive which enables the robot to do translational and rotational movements simultaneously. This is a great advantage over the prior generation that featured a differential drive with two driven wheels. Here a four wheel omnidirectional drive is used instead of a three wheel one. The construction of the wheel suspension ensures that all four wheels are pressed onto the ground which leads to enhanced stability.

Besides the driving system, the ball handling system has been redesigned from scratch. The ball handling system consists of two main components: the ball kicking system and the dribbling system. The previously used mechanical kicking system has been replaced by an electromagnetic solenoid which provides more control over the kicking power and reduces the actuation latency. The ball dribbling system has been redesigned to be more robust concerning collisions. All servo motors have been mechanically decoupled with rubber blocks so that even hard collisions will not harm the servos with excessive mechanical shocks.

The same mechanical decoupling has been applied to the servos of the active vision system to tolerate collisions with high kicked balls. In contrast to omnivision systems that are currently used by many other teams, three individual pan-tilt cameras are used in the vision system. Each camera may independently focus and track a different object of interest like ball, goal or other robots.

3 System design

In this section the structure of the Paderkicker robot will be shown. First, the functional architecture will be described. Then we will show how this logical structure maps onto a hardware structure. After the description of the underlying structures, the behavior system as well as the vision system will be introduced.

3.1 Functional architecture

During the system design process, four main functional units were identified (vision, driving, ball handling and behavior) and designed in a modular way. A robot of the Paderkicker team consists of a behavior module, the vision module, the driving module, and the ball handling module. The function of the last three is self-explaining by their respective names. The behavior module is the topmost module in a robot's hierarchy. It controls the robot's overall behavior.

The different modules are realized in a distributed way as described below. All components communicate with a message format which is used in the entire system independent of the respective medium for communication.

The functional units were further divided into sub-modules as depicted in Figure 3. This structure allows the independent development of the different functional units. Furthermore, the functional units were designed to work autonomously on their own presenting an already abstracted interface to the rest of the system. A dedicated interface sub-module manages the communication and merges data. This hierarchical structure enables the functional unit "Behavior module" to act on a very high level of abstraction.

As an example, the driving module is designed to work autonomously and part of the robot's low-level behavior has been mapped to it. Distributing the drive-control task to a group of sub-modules instead of using only one monolithic module leads to more flexibility and robustness. The sub-modules within are realized on individual microcontroller boards working as a distributed system. Each microcontroller board realizes an individual motor controller and odometry data logger with a short measurement-control latency and therefore can react very fast. Each board also incorporates an emergency handling unit which leads to a more robust behavior of the whole driving module.

3.2 Hardware architecture

The functional structure described above is mapped onto a hardware architecture as depicted in Figure 4. The central processing unit is a Intel Core Duo PC board running Linux. The vision algorithms and the behavior system are realized here. The Mini-ITX board is equipped with a Mini PCI wireless LAN card and handles team communication.

As described above, the modules for ball handling and driving are divided into sub-modules. These sub-modules are realized on microcontroller boards equipped with an Atmel microcontroller which comes with an on-chip CAN bus interface. Groups of microcontroller boards communicate over CAN with the members of

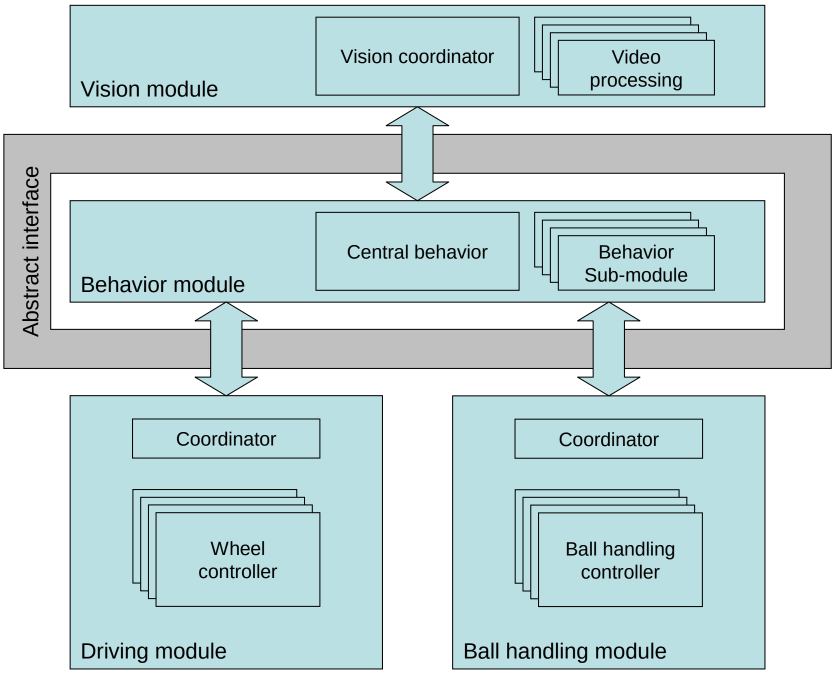

Fig. 3. Paderkicker hierarchical module structure

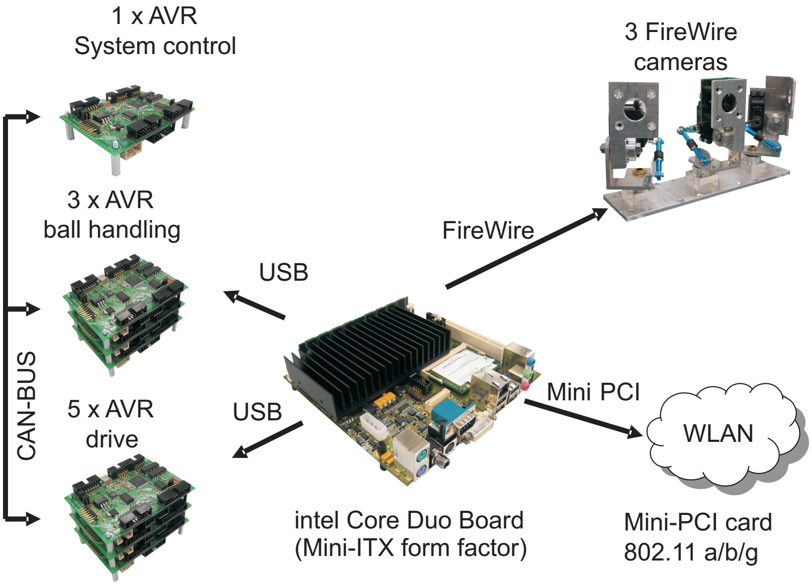

Fig. 4. Paderkicker hardware architecture

the according group. One dedicated microcontroller board in each group manages the communication with the central Mini-ITX board over a USB connection.

4 Behavior based system

The actual version of the behavior system is realized as a parallel distributed software system (Figure 5), where parallel running processes are responsible for the dedicated functional hardware units vision, driving, ball handling (Figure 3). In addition, a new timing concept now allows the different subsystems like the above mentioned to run at different cycle duration. Using a double buffered shared memory approach it is no problem if e.g. the cycle time of the vision system increases because the analyzed image contains more detectable objects than usual or if the ball handling component has to run at a higher frequency than the driving component.

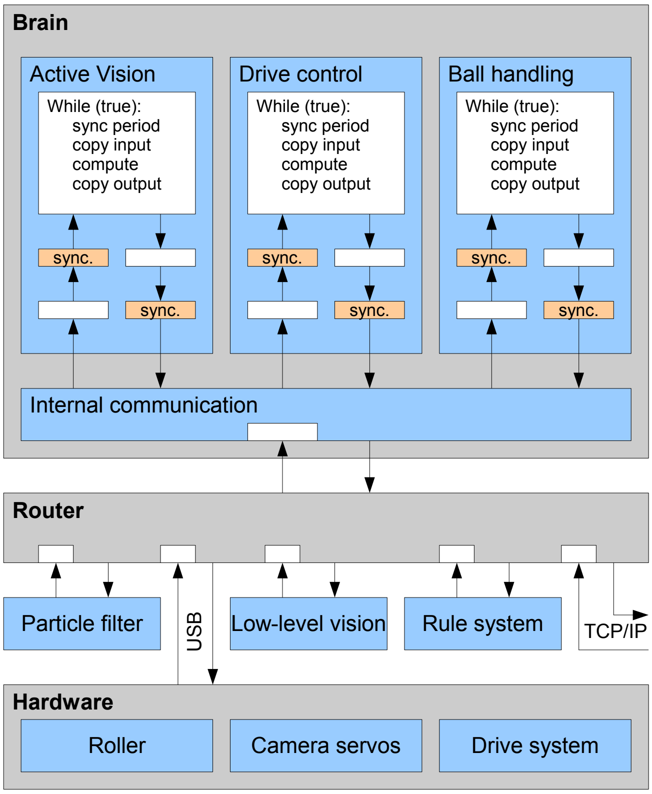

Fig. 5. Asynchronous architecture for the behavior module

The architecture's design is driven by the need of the sub-modules *Active Vision*, *Driving*, and *Ball Handling* to run at different sample rates. In the former architecture all the functionality was done in the same module at the same speed. The problem was that functionality that needs to run at a high speed got at some point corrupted data from modules running at slower speed, which lead to unpredictable behavior in some cases. To avoid this, at first the different functionality was identified and regrouped in separate sub-modules. Then we introduced a double-buffered communication mechanism that separates the actual data on which the individual modules are working on from the communication process.

Each sub-module has its own cycle. The sub-module's output is is either new data for the other modules or part of the final action which first has to be sent to the hardware via the *Router* (Figure 3). All sub-modules are running concurrently.

5 Vision system

The vision system also has been designed using the paradigm of hierarchical distribution. The vision systems span four levels of abstraction, beginning with the low-level vision based on an optimized algorithm for low latency real-time color segmentation [17]. The original algorithm has been adapted to run on a PC under an ordinary Linux system. Three digital FireWire cameras are mounted on pan-tilt units to cover the whole 360° view. Each camera is handled by an independent task doing the low-level image processing. On the next level of hierarchy the outputs of these tasks are merged into a robot-centric view of the surrounding objects and landmarks. Figure 6 shows a visualization of the particle filter. Each triangle indicates a hypothesis of the robot's position with the hollow triangle being the resulting position estimation of the robot in the world coordinate system. An abstract interface is presented to the next level of hierarchy enabling the user of the interface to specify e.g. scan modes of the cameras.

The next level of abstraction includes two particle filters [2] and a specialized control module. One particle filter estimates the robot's position relative to known landmarks. The second particle filter estimates the position of the ball relative to the robot. The control module again presents an abstract interface to the next level of abstraction. Using this interface two views are accessible. One "global view" with global world coordinates including all absolute coordinates of objects and landmarks. However, the second view is robot-centric using relative coordinates.

The behavior based system descibed in Section 4 is located on the highest level of abstraction. A dedicated module within this system takes care about the behavior of the underlying vision system, e.g. which part of the field is to be examined or whether the ball has to be tracked.

Compared to systems using an omnivision camera [19], on our system the resolution is higher for a given viewing direction. Furthermore the system allows

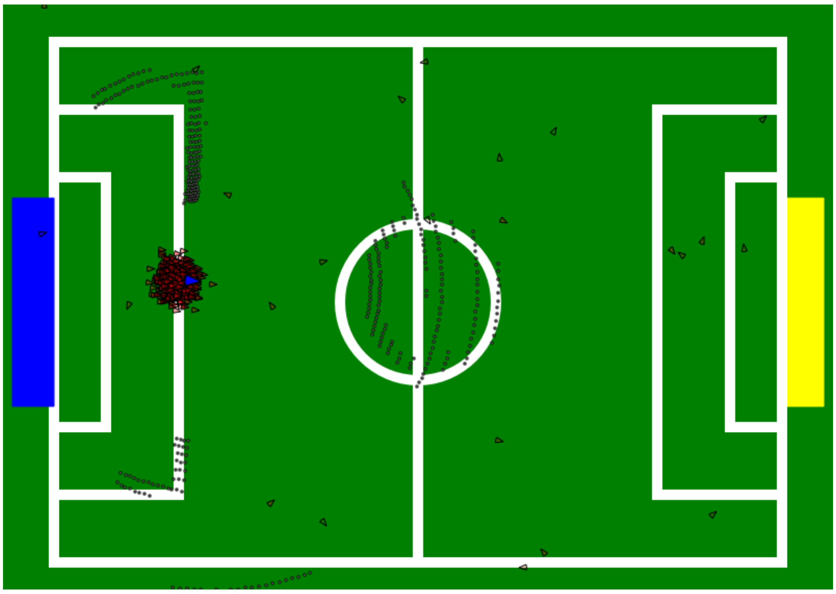

Fig. 6. Visualization of the particle filter and the robot's perceived artefacts (dots).

the over-sampling of a specified region of interest. Due to the constant usage of abstraction throughout the system this is done autonomously, e.g. for the position of the ball. This enables the system to recognize even distant objects that would be indistinguishable in a typical omnivision setup with only one fixed camera.

6 Coordination of functional units

The architecture does not impose limits upon the way data is exchanged between functional units. Most units will work asynchronously regarding each other and can work in a time-triggered or event-triggered manner. An example for an asynchronous time-triggered operation are the cameras attached to the vision system that will deliver data in periodic intervals that cannot practically be synchronized with the rest of the system. The high-level behavior system is running at a different rate unsynchronized to the cameras. In contrast, sensors like a ball detection sensor can trigger event processing and event messages that are non-deterministic in their timing.

To bridge the gap between such different execution semantics, an abstraction layer is introduced. It decouples the communication of the unit. Double buffering with atomic copying is used to ensure integrity for data transfers. Depending on the type of data, new data either is queued or overwrites an old value for a last-recently-received type of information.

7 Learning in Teams

One purpose of our Paderkicker soccer team is the investigation of appropriate means to propagate learned knowledge in teams of robots [9, 8, 4]. The learned knowledge can be on the one hand low-level skills, and on the other hands high-level strategies making use of those skills. We have developed an approach by which a robot is able to learn skills most natural to its own morphology by developmental self-exploration. This is a learning behavior also found in early childhood. Together with the learned skills the robot also learns the means to classify observed behavior from other robots. This data is then processed to improve the strategy [11]. As every robot has a different perception stream the skills will be different within the robot team. Nevertheless, they are able to imitate each others' strategies thus solving easily the correspondence problem often found in imitation literature and speeding up the overall learning effort [12].

8 Conclusion

With the architecture described in this paper we have gradually improved our robot team. With our new design for the goal keeper, team support, and our enhanced real-time capabilities for the vision module we believe that our robot team is a robust platform that is able to keep up with the competing RoboCup teams this year.

Furthermore, the platform reflects our educational concept. Master/diploma students in computer science (with emphasis on embedded systems) and master/diploma students in "Ingenieurinformatik" (combination of computer science and electrical or mechanical engineering) are educated in computer science solutions suitable for engineering problems especially in the automotive domain.

Acknowledgement We thank Matthias Schmitz, Alexander Jungmann, Jan Schulte-Landwehr, Bastian Nordmeyer, and the numerous students whose work has taken the project forward. We also would like to express our gratitude to our C-LAB colleagues for bearing our experiments.

References

1. The Paderkicker Team, 2009.
2. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 50(2):174–188, Feb 2002.
3. Bernd Kleinjohann, Lisa Kleinjohann, Willi Richert, and Claudius Stern. Integrating autonomous behavior and team coordination into an embedded architecture. In Pedro U. Lima, editor, *Robot Soccer*, chapter Integrating autonomous behaviour and team coordination into an embedded architecture, pages 253–280. Pro Literatur Verlag / ARS, December 2007.

4. Markus Koch, Willi Richert, and Alexander Saskevic. A self-optimization approach for hybrid planning and socially inspired agents. In *Second NASA GSFC/IEEE Workshop on Radical Agent Concepts*, 2005.
5. Rafael Radkowski, Jürgen Gausemeier, Bernd Kleinjohann, Willi Richert, Philipp Adelt, and Henning Zabel. Augmented reality to support the testing of autonomous systems by the example of soccer robots. In *53. Internationales wissenschaftliches Kolloquium*, volume -. Technische Universität Ilmenau, Technische Universität Ilmenau, 8 - 12 September 2008.
6. Rafael Radkowski, Willi Richert, Henning Zabel, and Philipp Adelt. Augmented reality-based behavior-analysis of autonomous robotic soccers. In *IADIS International Conference of Applied Computation*, Algarve (Portugal), 2008.
7. Christian Reimann. Kick-Real - a mobile mixed reality game. In *ACE2005, ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 2005.
8. Willi Richert, Bernd Kleinjohann, and Lisa Kleinjohann. Evolving agent societies through imitation controlled by artificial emotions. In M. Huang, X.-P. Zhang, and M. Huang, editors, *ICIC 2005*, number 3644 in LNCS, pages 1004–1013. Springer-Verlag Berlin, 2005.
9. Willi Richert, Bernd Kleinjohann, and Lisa Kleinjohann. Learning action sequences through imitation in behavior based architectures. In *Systems Aspects in Organic and Pervasive Computing - ARCS 2005*, number 3432 in LNCS, pages 93–107. Springer-Verlag Berlin, 14 - 17 March 2005.
10. Willi Richert, Olaf Lüke, Bastian Nordmeyer, and Bernd Kleinjohann. Increasing the autonomy of mobile robots by on-line learning simultaneously at different levels of abstraction. In *International Conference on Autonomic and Autonomous Systems (ICAS'08)*. IEEE Computer Society, March 2008.
11. Willi Richert, Oliver Niehörster, and Markus Koch. Layered understanding for sporadic imitation in a multi-robot scenario. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*, Nice, France, 2008.
12. Willi Richert, Ulrich Scheller, Markus Koch, Bernd Kleinjohann, and Claudius Stern. Integrating sporadic imitation in reinforcement learning robots. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL'09)*, 2009.
13. Claudius Stern, Philipp Adelt, Willi Richert, and Bernd Kleinjohann. Hierarchically distributing embedded systems for improved autonomy. In *Proceedings of the Distributed Parallel Embedded Systems Workshop (DIPES)*, Distributed and Parallel Embedded Systems, 7 - 10 September 2008.
14. Dirk Stichling. *VisiTrack - Inkrementelles Kameratracking für mobile Echtzeitsysteme*. PhD thesis, Universität Paderborn, Fakultät für Elektrotechnik, Informatik und Mathematik, 2004.
15. Dirk Stichling and Bernd Kleinjohann. CV-SDF - a model for real-time computer vision applications. In *IEEE Workshop on Application of Computer Vision*. IEEE, December 2002.
16. Dirk Stichling and Bernd Kleinjohann. Low latency color segmentation on embedded real-time systems. In Bernd Kleinjohann, K.H. Kim, Lisa Kleinjohann, and Achim Rettberg, editors, *Design and Analysis of Distributed Embedded Systems*. Kluwer Academic Publishers, 2002.
17. Dirk Stichling and Bernd Kleinjohann. Low latency color segmentation on embedded real-time systems. In *Design and Analysis of Distributed Embedded Systems*. Kluwer Academic Publishers, November 2002.

18. Dirk Stichling and Bernd Kleinjohann. Edge vectorization for embedded real-time systems using the CV-SDF model. In *Proceedings of the 16th International Conference on Vision Interfaces (VI 2003)*, June 2003.
19. Felix v. Hundelshausen, Sven Behnke, and Raúl Rojas. An omnidirectional vision system that finds and tracks color edges and blobs. *Lecture Notes In Computer Science*, 2377:374–379, 2002.