# WF Wolves KidSize Team Description RoboCup 2009

Reinhard Gerndt, Stefan Krupop, Carsten Schridde, Jan Carstensen, Sönke Michalik, Sören Michalik, Timo Büchner, Eduard Siewert, and Richard Benedix

University of Applied Sciences, Department of Computer Sciences,
Am Exer 2, 38302 Wolfenbüttel, Germany
{r.gerndt,s.krupop,c.schridde,j.carstensen
soe.michalik,so.michalik,t.buechner,e.siewert,r.benedix
}@fh-wolfenbuettel.de
http://robocup.fh-wolfenbuettel.de

**Abstract.** This is the Team Description of the WF Wolves, the RoboCup Team of the University of Applied Sciences Wolfenbüttel, for the RoboCup 2009. The University was involved in autonomous robot projects, mostly Unmanned Aerial Vehicles (UAVs) for some time. In the year 2006 a RoboCup team has been founded to participate in the RoboCups Mixed Reality (MR) league. In the 2007 RoboCup the team reached the fourth place, in 2008 first place of the MR competition. After bringing many improvements to the MR community, the team is now looking forward to apply some of the developments to humanoid robots and competing in the Humanoid KidSize League as well. The team will use a modified RoboBuilder-Huno for the RoboCup competition.

## 1 Introduction

Our team has been founded in the year 2006 at the University of Applied Sciences Braunschweig/Wolfenbuettel. The interdisciplinary team is supported mostly by the computer science, electrical and mechanical engineering departments. It is mostly run by a group of 10 to 20 students, supported by faculty members. The team participated in the RoboCup 2007 in Atlanta, where it reached the fourth place in the Mixed Reality Competition. At the RoboCup 2008, the team won the Championship in the Mixed Reality. The team also successfully participated in local RoboCup events like the Japan and German Open. The MR soccer is played by real micro robots on a horizontally mounted display with virtual field and ball. The team designed several versions of agent software to control the behavior of the micro robots. For the Mixed Reality community the team contributed to many hard- and software developments, such as the micro robot hard- and firmware and other components like a specialized battery charger. In 2008, the team started its activities with humanoid robot soccer. A universal software architecture allows to use the agents from the MR robots for the humanoids as well.

## 2 About the Team

The humanoid group within the team currently has nine graduate and undergraduate student team members from different faculties such as computer science, electrical engineering and mechanical engineering. It is organized independently but supervised by the faculties.

## 3 Mechanical and Electrical Design

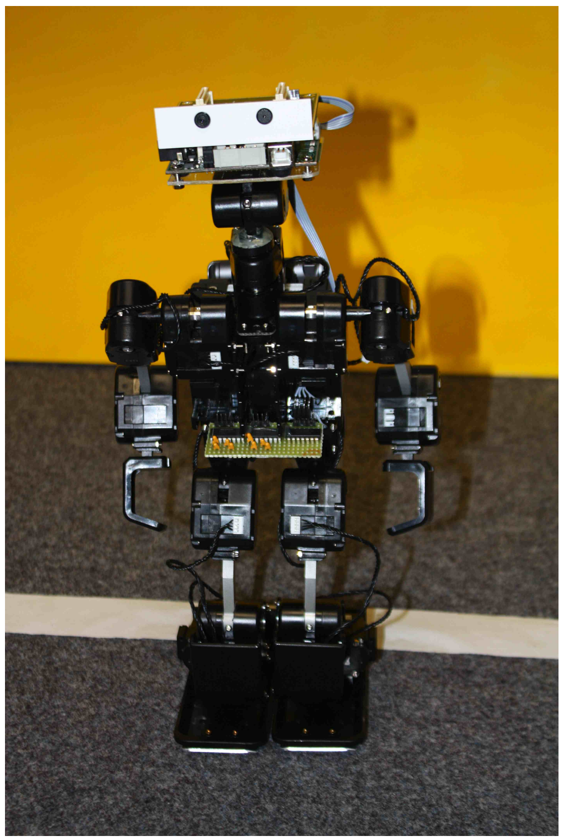

**Fig. 1.** WF Wolves 2009

The basic set of the "RoboBuilder 5710K CreatorsKit" was enhanced by stronger servos and stronger joints. Legs and arms have been enlarged and the size of the feet was reduced to assure compliance with the RoboCup rules. A pan/tilt unit was added to allow 2 DOF movement of the head. The Robot has 18 Degrees of Freedom total. Each Leg has five Degrees of Freedom, each arm has three and two are used to move the head (pan/tilt). The robot is equipped with 3-axis acceleration sensors and two 2 Megapixel cameras. The robot is controlled by three processors. High-level control is done by a PDA Pocket Loox N560. The PDA has a 624 MHz Processor and uses 64 MB RAM. It is equipped with 802.11g WLAN. The Loox is powered by its internal LiIon accumulator with 1200mAh. The image processing unit works with an Analog Devices Blackfin BF-561 Dual-Core DSP Processor with each core running at 500 MHz . The module has additional 64MB SDRam and 8MB Flash for the image processing algorithm. The DSP processor architecture features two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs and a 40-bit shifter for processing the images. The Dual 12-Channel DMA is used to transfer the camera-data into the SD-Ram and internal L1 Cache with multiple Transfer-Buffers. with the multiple DMA-Channels the image data can be transfered simultaniously from both camera ports. Low-level control is implemented on an AVR microcontroller. Image processing, low-level control and servos are powered by an additional battery.

## 4 Perception

The objective of the perception module is to locate objects as well as a number of marks on the field. By using stereo vision, we are able to determine the distance to the objects and marks considerably accurate. The left picture shows the sampling points which are possibly an object. Afterwords the area around these points is scanned more detailed. The scan-line algorithm checks the specific area and supports us with information about the size of the object. The areas which is detected to belong to the object are highlighted in the right picture. By only using the U and V Channels for color detection, the YUV format improves the detection of colors with different light intensities. An additional run-detection algorithm is used to identify the lines on the field. By knowing the lines we are able to detect specific marks on the filed "crossings". Based on the found object and an internal model, the robot will try to localize itself on the field.

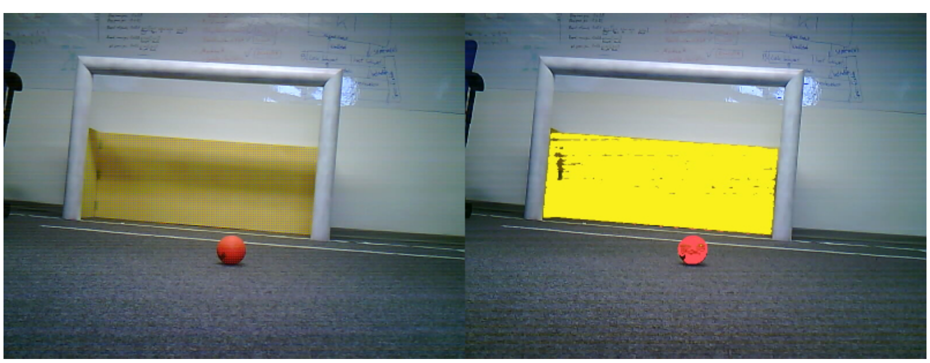

**Fig. 2.** Perception

## 5 Robot Control

We are using a software framework that allows to use the same agent we are using in the Mixed Reality League for the humanoid League. This is possible due to two abstraction layers. The Hardware Abstraction Layer (HAL) allows to run the same software on different kinds of humanoid robots. The Software Abstraction Layer (SAL) is used as an interface for different platforms of the RoboCup. In the Localization module the data from the camera unit is assembled into a bot specific Perception. The Localization Interpreter interprets the Perception to update the WorldState (sensor fusion). The Agent gives High-Level soccer statements to the Instruction Interpreter, where commands such as: "GoToBall", "KickToGoal", are transformed into HAL commands. The Instructor forewards the assembled movement commands to the AVR microcontroller where they are executed. The instructions are physically executed by running Motion Files.

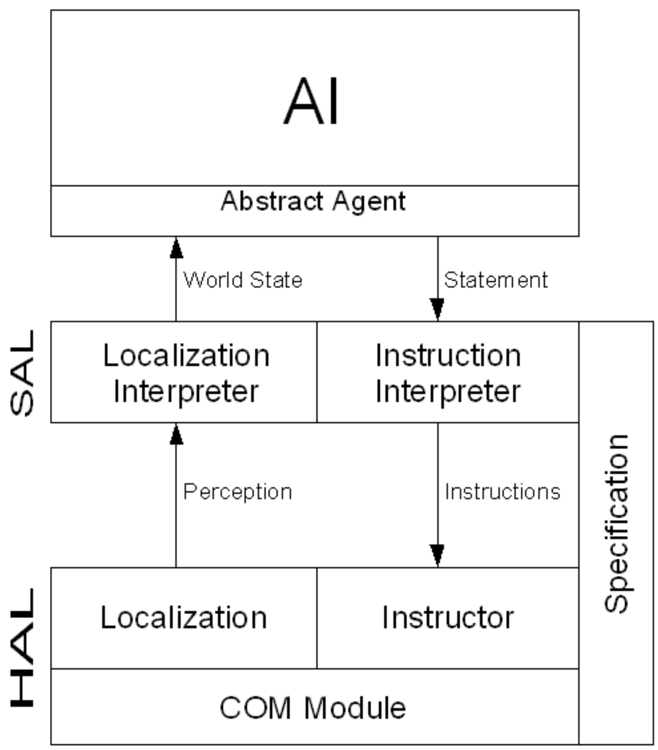


**Fig. 3.** Software Architechture