Team Description Paper 2010
Austrian Kangaroos

leader@austrian-kangaroos.com

April 2010

Abstract

The *Austrian Kangaroos* are a young team, which participated in RoboCup for the first time in 2009. Hence, the team's primary concerns were good enough running software for its soccer robots. However, this report summarizes the work done by the team in 2009, and also provides a brief outlook on things planed for the coming RoboCup season. Contributions to RoboCup are divided into three main categories: research, competition, and community. The *Austrian Kangaroos* contribute in all of them. The team consists of three institutes of the two biggest technical universities in Vienna. The covered research topics in the fields of robotics aim at the fields of computer vision, computational intelligence, but also at social and management aspects in mid to large scale project teams.

1 Introduction

The *Austrian Kangaroos* are a joint team of Vienna University of Technology (VUT) [1], and *University of Applied Sciences Technikum Vienna* (UASTV)[2]. Based on successful work during the season 2009 (the *Austrian Kangaroos* reached the WC's quarter-finals) the team will participated again in RoboCup 2010 within the Standard Platform League. The team is a joint effort of researchers, lecturers, and students of both universities. At Vienna University of Technology two institutes are directly involved: *The Automation and Control Institute* (ACIN)[3], and the *Institute of Computer Languages* (Compilers and Languages Group) (COMPLANG)[4].

2 Software Architecture

This sections provides information on the Austrian Kangaroos' efforts dedicated to software architecture, and robotic frameworks.

Existing expertise at the COMPLANG group covers static (timing) analyses, compilation techniques for embedded systems, middleware synthesis, and software architectures, all mainly in the fields of automotive and avionic systems, and signal processing. One of the team's goals is to adapt this know-how to the robotics domain, hence contributing for future technologies.

[1] Vienna University of Technology website: `http://www.tuwien.ac.at`
[2] University of Applied Sciences Technikum Vienna website: `http://www.technikum-wien.at`
[3] ACIN website: `http://www.acin.tuwien.ac.at`
[4] COMPLANG website: `http://www.complang.tuwien.ac.at`

However, the work done in 2009 was mainly targeted at a running system and is far away from the team's vision of a robust, real-time framework. Nevertheless, the experience gained with the team's first steps provides vital information on robotic middleware and middleware constraints,which will be incorporated into the upcoming new design for the *Austrian Kangaroos* real-time robotic middleware.

2.1 2009

In the beginning of 2009 student members of the *Austrian Kangaroos* started to develop a general software framework. This version (which is still in use in a slightly modified form) is mainly driven by the middleware architecture of NaoQi: application modules are connected via SOAP mechanisms; orchestration can be achieved within the modules but also from outside via a Python scripting layer. Both, scripting, and SOAP (without a RT OS or at least RT middleware) turned out to be sub-optimal for efficient and predictable control, and hence led to serious problems in terms of robustness and reliability.

Figure 1 depicts a high-level view of the *Austrian Kangaroos'* 2009 software architecture. Four main classes of components were defined (additional two classes for external monitoring devices have been implemented during the season). All components of the same class have been packaged into modules, which serve as static units of deployment. The used component classes (corresponding to modules) are:

Sensor: The sensor module contains all components that are related to the robot's sensing. In detail, this module contains a component for computer vision (see Sect. 3.1), inter-robot communication (in 2009 only WLAN was used), self localization (an implementation of a monte-carlo localization was used [1], but has been disabled during the world cup due to stability problems), and sensor fusion.

Motion: The motion module holds one component responsible for motion planing, and one for the execution and monitoring of single motions. The motion planing component calculates sequences of atomic motions to achieve goals, identified by the high-level AI. The executer component's responsibility is to run single atomic motions (via NaoQi) and to monitor the progress of execution (motion exceptions like falls or collisions are handled here). Due to missing RT functionality within NaoQi, monitoring was only used on a coarse grained level.

Control: The control module (its name is rather misleading and was replaced by end of season 2009) contains components of the high-level AI. The strategy selection component implements goal driven behaviors, based on predefined (statically assigned) roles. To reach a specific (sub-)goal, complex motion commands are issued to the motion module that calculates and executes motion sequences. The game controller component mainly accounts the global game state, and accordingly controls all other modules.

Shared Memory: Each robot holds his own local world database. The database is stored in shared memory (based on the NaoQi mechanisms). All components access this database within their own threads of execution, and keep values of interest up to date. At the beginning, the publisher-subscriber mechanism of NaoQi looked promising and hence was heavily used. Unfortunately, it turned out to be not reliable, and finally led to numerous data misses. For the team's future framework, this mechanism will be avoided wherever possible.

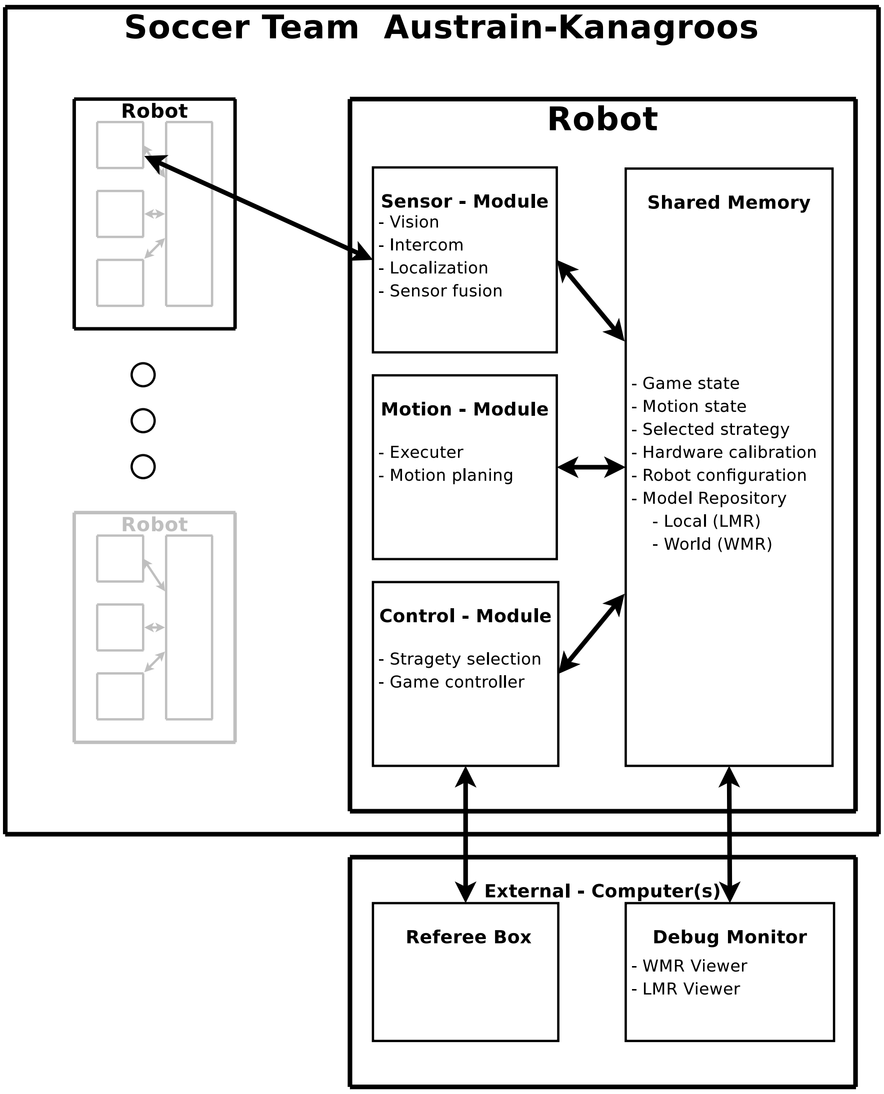

Figure 1: Our system architecture

In fact, inter-module communication of the 2009 framework was realized via NaoQi. This approach turned out to be rather error prone and resource consuming: profiling and debugging exposed NaoQi's SOAP (incorporating tiny XML) to consume unexpected high amounts of valuable resources when doing XML encoding and decoding.

2.2 2010

For 2010 the team has completely refactored the existing software architecture, and is aiming at a robust, self-repairing system. The application components will be decoupled from NaoQi, while a bridge component will be put in place to encapsulate robotic middleware. Live sanity checks, as much as RT-monitors will come to use to increase the overall stability of the robots' software.

Timing-aware software composition similar to the one described in [2, 3] will be incorporated into the *Austrian Kangaroos'* engineering methodology. This will initiate a middle term goal: designing an analyzable and certifiable software architecture.

3 Vision & Sensor Fusion

This section covers work related to the pre and post processing of the robots sensor readings which also includes the vision processing.

3.1 2009

As a new team, the *Austrian Kangaroos* relied on existing solutions wherever possible, like the color segmentation software CMVision [5] available form *Carnegie Mellon University* (CMU). The given package provides a code base to segment images in regions of interest. This regions were assigned probabilities, calculated by heuristics, which resulted from research within the ACIN group. Figure 2 depicts this regions and the assigned probabilities as seen by the robot.

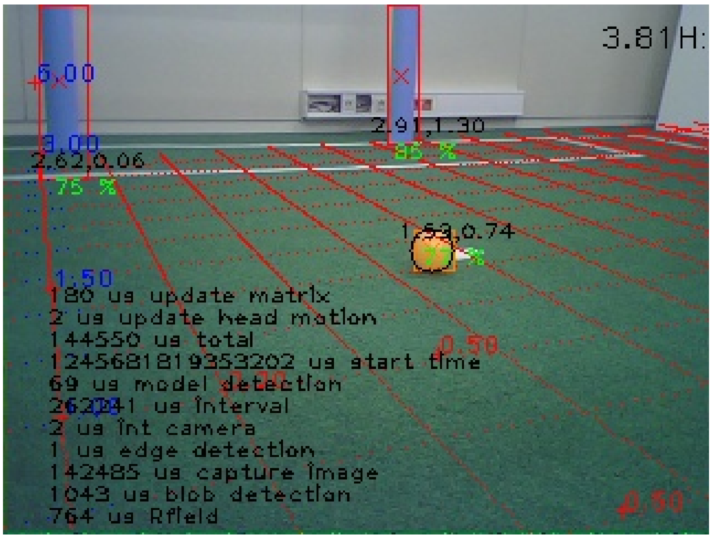

Figure 2: Regions, objects, and probability

In addition, a more precise camera pose relative to the ground plane was calculated, taking knowledge of the robot's kinematic and its internal sensor reading into account. As a result, the robots are able to estimate the distance to regions of interest, and to verify the accuracy of observations that are based on object sizes in the image plane. Data representation on a projected spherical coordinate system fostered post- and pre-processing of incoming sensor readings. In addition, timed memory introduced temporal calculations between new and aged sensory data. The hereby gained multilayered potential field is consequently used to efficiently propagate errors, and to hold multi-hypothesis information.

To increase the robots world knowledge, a global world model was implemented, and integrated into the robots world model via the multi-hypothesis approach. Every robot shared his knowledge (especially his own position and the position of the ball) with the full team. Due to a low reliability, resulting from an unstable implementation, this functionality was disabled during the world cup in Graz.

[5] CMVision website: `http://www.cs.cmu.edu/~jbruce/cmvision/`

Sonar was not used by the *Austrian Kangaroos* in 2009, but will be covered in coming seasons.

3.2 2010

For the next season the *Austrian Kangaroos* plan to develop an auto-calibration tool to overcome manual color calibration, so far required before every game, as much as auto-adjustment during a running game. Auto-calibration will be based on Graph-Based Image Segmentation [4]. Its realization will in addition help to enhance the team's new color-based vision, which will rely on a shape based approach for object verification by color and shape[5].

The new system will be able to integrate sensor readings from other robots, which allows the construction of a common world model.

A vital new issue is line detection for playground lines, where the implementation will be based on a line detector [6]. Lines are also used to increase the accuracy of goal detection.

In general, the development of the team's sensory framework will become context sensitive (dynamic weighting of object probabilities with respect to global states and goals), and more robust.

4 Education

Another topic of interest within the *Austrian Kangaroos* is teaching and education. Hence, the members of the team are heavily involved in the RoboCup-Junior scene in Austria.

4.1 2009

Members of the *Austrian Kangaroos* helped to establish the community and build up 7 regional centers in Vorarlberg, Tyrol, Salzburg, Upper Austria, Carinthia, Styria and Vienna. In 2009 they organized the 2^{nd} RoboCup-Junior Austrian Open in Vienna, the Austrian qualification for the RoboCup 2009 in Graz.

In the direction of team management all RoboCup teams have similar problems that are namely fluctuation of students, loss of knowledge, and team organization. To focus these topics the members of the *Austrian Kangaroos* developed a strategic concept based on the Vienna Cubes Rotation concept published in [7]: All students participating in the *Austrian Kangaroos* project are divided into three categories: Junior, Core, and Senior.

1. First, students enter the Junior phase as new undergraduate members. They learn from Core or Senior members and must build up knowledge about the team's tool chain, the game rules and the system. They take part at the German Open but will not take part at the RoboCup WC. One of the most important technical responsibilities of Juniors is extensive system testing combined with code reviews.

2. After spending at least one year as Junior, students may be promoted to Core team members, who lead the project and drive the technical requirements and implementation. They also recruit, motivate and teach new Junior members. The Core team members also participate at the German Open and the RoboCup WC.

3. After leaving the active team, former Core members keep a status as Senior members. The *Austrian Kangaroos'* Senior members are supposed to recruit, motivate and teach

new Junior members by giving presentations in undergraduate classes. Senior members may function as consultants to the Core team, however they are not supposed to work actively in technical tasks.

4.2 2010

Austrian Kangaroos members run the RoboCup-Junior regional center in Vienna and offer programming and robotics courses with schools, teachers and pupils as a preparation for the next Austrian Open 2010. Several presentations with the topic 'Why do robots play soccer?' have been held in schools in Vienna and Lower Austria. Currently the regional center Vienna has about 25 LEGO Mindstorm sets with extra sensors especially for Soccer, Rescue and Dance applications that children can borrow for participation in RoboCup-Junior.

5 Summary

To finally conclude the season 2009, the *Austrian Kangaroos* spent most of their efforts to get a running system for the team's first year in RoboCup SPL. The software architecture developed by student members of the team was based on (and mainly driven by) NaoQi, the robots' pre-installed middleware. Although the team's software was modular and component based, it contained tight coupling between modules and components, and did not provides means of real-time support. The vision system was based on CMVision, and integrated a multi-hypothesis approach. A module for probabilistic self-localization was developed, but has not been used at tournaments. At education and management level, the *Austrian Kangaroos* successfully supported national efforts in RoboCup Junior. A rotation concept for human resources has been established, and will be monitored for the next view years, in order to prove its effectiveness.

References

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* MIT Press, Cambridge, Mass., 2005.

[2] Dietmar Schreiner and Karl M. Göschka. Explicit Connectors in Component Based Software Engineering for Distributed Embedded Systems. In *SOFSEM 2007: Theory and Practice of Computer Science, Proceedings*, volume 4362 of *LNCS*, pages 923–934. LNCS, Springer, January 2007.

[3] Dietmar Schreiner, Markus Schordan, and Jens Knoop. Adding Timing-Awareness to AUTOSAR Basic-Software – A Component Based Approach. In *Proceedings of the 12th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2009)*, pages 288–292. IEEE, IEEE Computer Society, March 2009.

[4] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 3. November 2004.

[5] M. Bader, M. Albero, R. Sablatnig, J.E. Simo, G. Benet, G. Novak, and F. Blanes. Embedded Real-Time Ball Detection Unit for the YABIRO Biped Robot. In *Intelligent Solutions in Embedded Systems, 2006 International Workshop on*, pages 1–9, June 2006.

[6] Paul L. Rosin and Geoff A. W. West. Nonparametric segmentation of curves into various representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(12):1140–1153, 1995.

[7] H. Wahl, R. Pucher, A. Hofmann, A. Mense, H. Gollner, and F. Schmöllerbeck. Handle team changes in student projects – the example of the vienna cubes. In *Proceedings of the International Conference PBL2006 ABP*, 2006.