Team Zadeat 2010
— Application for Participation —

Alexander Ferrein[1], Tim Niemüller[2], and Gerald Steinbauer[3]

[1] University of Cape Town, Robotics and Agents Lab, Rondebosch, South Africa
alexander.ferrein@uct.ac.za
[2] RWTH Aachen University, Knowledge-Based Systems Group, Aachen, Germany
niemueller@kbsg.rwth-aachen.de
[3] Graz University of Technology, Institute for Software Technology, Graz, Austria
steinbauer@ist.tugraz.at

1 Statement of Participation

As in the years 2008 and 2009, team ZaDeAt is herewith committing itself to participate in the RoboCup 2010 competition of the Standard Platform League, if selected for participation. Our research groups have a strong interest in continuing our work on the Nao. The RoboCup 2010 is an excellent opportunity to keep in touch with the Nao community. Furthermore does a participation support our several research and community activities, as laid out below. Our general commitment to the SPL is substantiated by the fact that two of the authors served as RoboCup 2009 General Chair and SPL Local Chair, respectively.

2 Team Structure

The team consists of the three research groups: (1) The Robotics and Agents Research Lab, University of Cape Town, South Africa, (2) the Knowledge-Based Systems Group, RWTH Aachen University, Germany, and (3) the Institute for Software Technology, Graz University of Technology, Austria. The members of the team consists of academic staff, PhD students, and MSc and BSc students. Each local group has a sub-team leader who is responsible for co-ordinating the own group locally, and for co-ordinating with the other groups. Inside each group there is a good mix of experienced researchers with a long record in RoboCup and students who participate for the first or second time in a RoboCup event. The South African members are: Gareth Priede (PhD candidate), Jack Liu (BSc.Hons. candidate), Alexander Ferrein (research staff and head); the German team consists of Tim Niemüller (MSc. candidate and head) and Gerhard Lakemeyer (research staff); and the Austrian members are: Christof Rath (MSc. candidate and head), Tobias Kellner (MSc. candidate), Patrick Podbregar (BSc. candidate), Darshaka Pathirana (BSc. candidate), and Gerald Steinbauer (research staff).

3 Research Focus and Planned Activities

3.1 The Fawkes Framework

The experience of around six years of RoboCup participation were amalgamated into our new software framework Fawkes[4]. This new framework, which will also be deployed for the AllemaniACs Middle-size and RoboCup@Home robots, aims at providing a light-weight, fast, versatile, and easy-to-use software framework for mobile robots and agents.

Besides general software characteristics with features such as providing a (robot) hardware abstraction, being extensible and scalable, causing a limited run-time overhead, or a proper documentation, we particularly set our goals to providing tools and methods for controlling robots with today's hardware specifications. Some of Fawkes' design ideas are given in the following: (1) *Component-based Design.* The software is logically structured into separate exchangeable components operating in a unified environment. (2) *Multi-Threading.* The system is designed with modern multi-core CPUs in mind. Multiple threads can process data in parallel, but it is still flexible and efficient enough to operate on the Nao. (3) *Data Sharing.* A corner stone for a software architecture is the way different components communicate with each other. We decided to use a hybrid blackboard and messaging architecture, where different components provide data by means of well-defined interfaces and accept commands via message queues. It has proved to be simple to use, fast, robust, and easy to monitor and log, because all the important data is stored at a single place and events can be triggered whenever data is modified. The blackboard is realised by shared memory segments. (4) *Program Flow.* One of the major decisions made was to eliminate the former multi-process architecture and replace it with a single-process multi-threaded environment. Especially since the advent of multi-core CPUs with reduced power consumption it was a hard criterion to exploit this with a massively multi-threaded architecture. Experience has shown that most of the computational work is done in a sequential recurring loop at least roughly following the classical sense-think-act cycle. But within one of the sequential step the work can usually be done in parallel. (5) *Runtime Reconfigurability.* For an efficient development environment components must be able to be loaded and unloaded at runtime. It avoids the need to reload the whole software every time, possibly going through lengthy (hardware) initialisation. It also allows to exchange components with similar functionality for easy testing. (6) *Script-based Behaviours.* As mentioned in the introduction, an important topic in robotics besides the low-level system of a robot is behaviour control. This ranges from reactive components controlling execution of a certain sub-task like locomotion to a specific destination to deliberative agents planning the overall strategy for the robot. In the future the scripting language Lua with its run-time system is used to ease this task [1], allowing for a flexible operation and providing the full power of a complete programming language.

[4] Fawkes is available as Open Source Software at http://www.fawkesrobotics.org

3.2 High-level Reasoning with Golog

It was shown before (e.g. [2]) that logic-based robot control languages such as Golog can be deployed successfully for robotic soccer activities. Although the soccer domain is quite restricted in terms of the required knowledge representation (which is a stronghold of logic-based robot control languages), the challenges lie on the side of reactivity and computational restrictiveness. Therefore, one of our research goals is to deploy Golog-type languages for the high-level control of the robot Nao. As Golog implementations are usually done in Prolog, which is not available on Nao yet, we started to reimplement Golog in the scripting language Lua. Lua has the advantage that it is one of the fastest interpreted scripting languages around, and it comes with a very small memory footprint. Therefore it is well suited to be deployed on the Nao.

We just finished our first prototype implementation of Golog in Lua [3], which will be presented at the AAAI Spring Symposium on Embedded Reasoning. The first preliminary results are very promising. The integration of a Golog-based high-level agent with the Nao system is facilitated by our component-based software design approach with Fawkes and the fact that our behaviour engine is also implemented in Lua. A seamless integration will easily be possible. As another interesting side aspect of this research one has to state that a Non-Prolog implementation of Golog might convince researcher outside the Cognitive Robotics community to try this kind of approach, as it allows to combine planning with programming, which is particularly interesting in restricted domains like robotic soccer, where the programmer has some good knowledge how the robot can achieve its goals while still being able to take advantage of a planning system.

3.3 Motion Tracking

Estimating good error measures is a crucial task particularly in a robotic system with a high number of degrees of freedom. To do so, one needs to combine the local data, i.e. the sensor values the robot has, with some external reference ground truth data. Getting good ground truth data is a non-trivial task. Preferably, one would take the sensed data of each joint of the robot and enrich these data with ground truth data of each limb in 3D or even 6D coordinates (Euclidean coordinate plus Euler angles). These data help to improve the motion system of the Nao as one can use the outer reference value for estimating the error of the computed kinematics. But not only for motion are these data useful, the testing of all other software components also benefit from complete sensor/ground truth data sets. For example, the stream of camera images from a recorded scene together with the ground truth data of the robot and the other objects can be used to improve the vision system or the ball tracker.

At the University of Graz, a motion tracking system is available that can track objects in 6D with high precision. Our aim is to provide ground truth data together with the robot's sensor data. We will track data in several typical scenarios on a SPL pitch, which shall resemble real tournaments situations. Therefore we equip the Nao with up to 42 markers which will be attached at the limbs and head. During each run we will record all the sensor values of the robot together with its camera images. As we can stream the camera images only with 15 Hz in full VGA resolution, we plan to replay each scenario also with 30 Hz in 320×240 resolution. The idea is of course not new, standardised data sets are wide-spread especially in the computer vision field, and also [4] tracked the Nao's pose successfully. However, we hope that we can improve these data sets with the 6D coordinates of the limbs rather than just giving the pose of the robot. We hope to be able to provide these data for the community till the time of the RoboCup Symposium.

4 Summary of Own Relevant Work and Publications

In this section we give a short overview of the relevant work that was published during the last year or will be published in the next year. Earlier important and relevant publications can be found partly in the references or at the author's homepages. We are currently working on a number of projects inside the team, some of which were already mentioned above.

- On the low-level control side, we are working on an improved *navigation and collision avoidance* algorithm based on an any-angle Phi* approach [5] as an extension of our behaviour engine, which was presented in [1].
- On the high-level decision making side we are planning to have a revised version of our Golog interpreter in Lua ready and running for the competition. We also hope that we are able to incorporate simple soccer moves as envisioned in [6] and are able to integrate a reactive control mechanism into Golog as presented in [7].

References

1. Niemueller, T., Ferrein, A., Lakemeyer, G.: A Lua-based Behavior Engine for Controlling the Humanoid Robot Nao. In: Proc. RoboCup 2009: Robot Soccer World Cup XIII. (2009)
2. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. Journal Robotics and Autonomous Systems **56**(11) (2008) 980–991
3. Ferrein, A.: golog.lua: Towards a non-prolog implementation of golog for embedded systems. In Beetz, M., de Kleer, J., Frank, J., Hoffmann, G., Ingham, M., Kuhn, L., McGann, C., Tomlin, C., eds.: Proc. of the 2010 AAAI Spring Symposium on Embedded Reasoning: Intelligence in Embedded Systems, AAAI Press (2010) accepted for publication.
4. Laue, T., Röfer, T.: Pose extraction from sample sets in robot self-localization - a comparison and a novel approach. In Petrovic, I., Lilienthal, A.J., eds.: Proc. 4th European Conference on Mobile Robots (ECMR 2009). (2009)
5. Nash, A., Koenig, S., Likhachev, M.: Incremental phi*: Incremental any-angle path planning on grids. In Boutilier, C., ed.: Proc IJCAI-09, AAAI Press (2009) 1824–1830

6. Dylla, F., Ferrein, A., Lakemeyer, G., Murray, J., Obst, O., Röfer, T., Schiffer, S., Stolzenburg, F., Visser, U., Wagner, T.: Approaching a formal soccer theory from behaviour specifications in robotic soccer. In Dabnicki, P., Baca, A., eds.: Computer in Sports. WIT Press (2008)
7. Ferrein, A., Schiffer, S., Lakemeyer, G.: Embedding fuzzy controllers into golog. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'09), IEEE (2009) 498–509