Team Zadeat 2009
— Team Report —

Alexander Ferrein[1], Gerald Steinbauer[2],
Graeme McPhillips[3], Tim Niemüller[1], and Anet Potgieter[3]

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

1 Introduction

In this paper we present the state of the development of the Team Zadeat, after a year of being engaged in RoboCup's Standard Platform League. The most important milestone in the last year clearly was the participation in RoboCup 2008 in Suzhou, where we could make first experiences of the stability of the hardware platform, and the bunch of problems that exists beyond our own software project. The bottom line of the state of the team Zadeat is that we actively started our cooperation over the quite long distance between the tip of Africa and central Europe.

The achievements we made during that year, besides providing the ground for a research cooperation on an administrative level, are, that our robot control software framework *Fawkes* is up and running, and that low-level modules for vision and object detection as well as localisation are prototypically implemented. As soon as the new hardware platform will be available, we will concentrate on the motion design of the robot. The integration of these patterns into the software will be rather easy as one focus of Fawkes lies on the execution and monitoring of basic behaviours. Besides a brief report on what we achieved so far, we also want to give some outlook on future work here. Our research interest are on combining logical reasoning with reactive control mechanisms on the robot platform.

2 The Team

Active team members are (in alphabetical order): Alexander Ferrein, Tobias Kellner, Greaeme McPhilips, Stephen Marais, Tim Niemüller, Anet Potgieter, Chritof Rath, Gerald Steinbauer. The working group consists of the Knowledge-based Systems Group of RWTH Aachen University [1] with their RoboCup team "AllemaniACs" [2], the Institute for Software Technology from Graz University of Technology [3] with their RoboCup team "Mostly Harmless" [4], and the Robotics and Agents Lab at the University of Cape Town [5]. In the following, we give a brief overview of the different institutes working in this research cooperation: The *Knowledge-based Systems Group* focuses, among other things, on intelligent high-level control of robots and agents acting in dynamic domains. With their team "AllemaniACs", the group participates in RoboCup tournaments in the Middle-Size League, the 2D and 3D Simulation League, and in the RoboCup@Home League since 2003. Notable results were achieved in the RoboCup@Home league with 1st places at the 2007 and 2008 German Open, the 2006 and 2007 RoboCup World Championships, and a 2nd place at the RoboCup 2009. The *Institute for Software Technology* is engaged in the field of intelligent robust control of autonomous systems with a best paper award received for their robotics research in that field [6]. Their team "Mostly Harmless" participates since 2003 in the Middle-Size League, but also in 2D Simulation League and in RoboCup@Home. Furthermore, the institute organised the Austrian RoboCupJunior participation in RoboCup 2007 and 2008. In 2007 the RoboCup@Home team came second at the German Open. The *Robotics and Agent Research Lab* at the University of Cape Town is engaged in hardware development and control of all kinds of different robots. Current projects include inter alia rescue and duct inspection robots. Also small-size robots have been developed, though, they did not participate in international RoboCup tournaments up to now. Moreover, there is experience with the development of autonomous agents in the field of sensor networks including RFID tags and video surveillance, social networks, and gaming.

3 The Fawkes Framework

The experience of around six years of RoboCup participation were amalgamated into our new software framework Fawkes. This new framework, which will also be deployed for the AllemaniACs Middle-size robots, aims at providing a lightweight, fast, 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) *Multi-Threading.* The system is designed with modern multi-core CPUs in mind. Multiple threads have to be able to process data in parallel. (2) *Data Transport.* A corner stone for a software architecture is the way different components communicate with each other. We decided to use the blackboard architecture, where different components write to a central storage system via well-defined interfaces. It has proven to be easy to use, fast, robust, and easy to monitor and log, because all the important data is stored at a single place. Other components can access the information in the same well-defined way. The blackboard is realised by shared memory segments. (3) *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. (4) *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. (5) *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 subtask like locomotion to a specific destination to deliberative agents planning the overall strategy for the robot. The experience with behaviour programming in the past has shown, that C++ is not a suitable language for this task. In the future the scripting language Lua with its run-time system is used to ease this task.

4 Future Work

4.1 High-level Reasoning and Reactive Behaviours

One accepted paradigm for the high-level control of autonomous mobile robots is the paradigm of knowledge-based AI. Many successful applications prove the usefulness of this class of approaches to exhibit goal-directed and intelligent behaviour of a mobile robot. The robot is equipped with a program which allows for goal-directed behaviours. Some approaches rely on programming techniques such as [7, 8], other approaches combine programming with planning (deliberation) algorithms (e.g. [9, 10]). These approaches have in common that the robot has a representation of the goal which it should fulfil. These methods often make use of logical formalisms to represent the goals of the agent or robot and to reason about the world the robot is acting in. On the other side of the spectrum are behavioristic approaches following the lines of [11–13]. These approaches are mostly biologically inspired, and the idea is that intelligent behaviours emerge from the system. Unlike the knowledge-based approaches has the single agent no representation of the overall goal, goal-directed behaviours emerge from the acting of the single robots without a particular representation of the goal. The advantage of these approaches is that they are more reactive as they do not rely on a complex reasoning machinery. Finally, for many applications, where knowledge about the application domain exists, it seems to be beneficial to make use of it. Here, the representation of the domain knowledge in the reactive decision mechanism (e.g. state machines, Bayes nets) will improve the behaviour of a single agent; seen from a different perspective can behavioristic techniques also be beneficial for logic-based high-level control strategies of a robot as they improve the reactivity of the decision-making.

We aim at studying the border line between both approaches and how both worlds can be connected with each other. In particular, we aim at extending the logic-based robot programming and planning language Readylog [14, 15] with reactive approaches. Readylog was developed as the high-level control language for robots acting in dynamic real-time domains like robotic soccer. While until now the focus of the development of Readylog was on deliberative decision-making, we now want to integrate behavioristic methods into the robot programming and planning framework as a means to approach reactive control. First ideas how deliberation can be brought together with reactive control were already presented in [16]. In a recent paper we presented how fuzzy control techniques could be integrated into the logical formalism of the situation calculus [17]. Along these lines, the combination of reactive control and deliberation seems to be possible.

4.2 Improved Odometry

Another important aspect we will work on in the next season is to provide the Nao with a more accurate odometry. A precise odometry can form the foundation for a improved self-localisation and can be used for better low-level behaviours like ball dribbling. The idea is to investigate and evaluate the potential of the sensors of the Nao in a first step. We believe that the build-in sensors like the gyros and the accelerometer can be used to estimate the motion of the robot. In order to evaluate the performance of the sensors we plan extensive motion experiments with the robot using a high-precision three-dimensional tracking system. The collected ground truth data will be the basis for the evaluation of the sensors itself and will be used as a test suite for the odometry methods we will develop. In a second step we will use the collected data to develop an improved version of the odometry. In particular we have two ideas how we can tackle the problem to estimate the motion of the robot based on the data form the robots internal sensors.

The first idea is to use techniques from machine learning in order to learn the odometry. In detail we have to learn a mapping between the inputs of the sensors and the true motion of the robot recorded in the tracking experiments. In previous work we obtained very good results in the prediction of time series using biologically inspired neural networks [18]. An import question we have to answer in this context is, if we are able to learn a good general odometry suitable for a greater population of Nao robots. Here we have to keep an eye on the generalisation capabilities of our used learning methods. The second idea is to use technique from signal processing. Because of the nature of a humanoid robot like the Nao we expect that the signals form the sensor will be noisy and uncertain. For instance we expect spikes in the signal of the accelerometer if the robot puts a foot on the ground when taking a step. We will work out how filtering techniques from the domain of signal processing can improve the quality of the data provided by the sensors.

4.3 Self-Localisation and Object recognition

A precise estimation of the position of the robot itself and the other objects like the ball in respect with the field is a very important requirement for a good robot soccer game. Because of the computational restrictions and the different kinematic of the Nao robot we have to adapt our methods for self-localisation and object recognition we have developed for our teams in the middle-size league.

Therefore, we will work on two aspects in this context. The first aspect is to provide the computer vision module with a good estimation of the position and orientation of the head of the robot which carry the camera. Such a good estimation of the posture of the camera enables a precise mapping of the coordinates in the camera image to the coordinates in the real world which is used in object recognition and self-localisation. For this estimation we will develop a method which is based on an inverse kinematic model of the robot and estimation of the position where the feet of the robot touch the ground.

Moreover, we will work on a computational inexpensive self-localisation approach feasible for the Nao. The idea is to use mainly pose tracking with Kalman filter instead of the more expensive particle filter. For the required measurement or update step of the filter we will adapt the ideas of scan matching previously used in localisation approaches using laser range finder [19]. In combination with the improved odometry we believe that this computational cheaper method will be accurate enough for an autonomous soccer game. For the initial estimation of the pose we work on a triangulation-based approach.

5 Conclusion

In this paper we briefly described the work done so far as well as sketched some ideas for future work. While some of the work will be finished until the next RoboCup 2009 in Graz, other projects have a longer horizon. The practical implementation work of the Fawkes framework done so far as well as the planned improvements on the odometry and the localisation and object recognition are due to the next tournament, while the combination of reactive control and deliberation are longer term funded research projects. With this very brief overview of the state of the development and the future research directions we presented a programme for the next two to three years. With it we hope to have shown our sincere interest to further participate in the Standard Platform League.

References

1. http://kbsg.rwth aachen.de: Knowledge-Based Systems Group, RWTH Aachen University (2007)
2. Ferrein, A., Lakemeyer, G., Schiffer, S.: AllemaniACs 2006 Team Description. In: International RoboCup Symposium., Bremen, Germany (2006)
3. http://www.ist.tugraz.at: Institute for Software Technology, Graz University of Technology (2007)

4. Steinbauer, G., Brandstöter, M., Buchleitner, M., Galler, S., Jantscher, S., Krammer, G., Mörth, M., Weber, J., Weiglhofer, M.: Mostly Harmless Team Description 2006 - Robust Control of Mobile Robots. In: International RoboCup Symposium., Bremen, Germany (2006)
5. http://www.ral.uct.ac.za: Robotics and Agents Research Lab, University of Cape Town (2007)
6. Steinbauer, G., Mörth, M., Wotawa, F.: Real-Time Diagnosis and Repair of Faults of Robot Control Software. In: Proceedings of the International RoboCup Symposium., Osaka, Japan (2005)
7. Myers, K.: A procedural knowledge approach to task-level control. In Drabble, B., ed.: Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96), AAAI Press (1996) 158–165
8. Firby, R.: Task networks for controlling continuous processes. In Hammond, K., ed.: Proceedings of the Second International Conference on Artificial Intelligence Planning Systems (AIPS-94), AAAI Press (1994)
9. McDermott, D.: A reactive plan language. Technical Report YALEU/DCS-RR-864, Yale University, Department of Computer Science (1991)
10. Beetz, M.: Structured reactive controllers. In Etzioni, O., Müller, J.P., Bradshaw, J.M., eds.: Proceedings of the Third International Conference on Autonomous Agents (Agents-99), ACM Press (1999) 228–235
11. Rosenblatt, J.K., Payton, D.W.: A fine-grained alternative to the subsumption architecture for mobile robot control. In: Proc of the IEEE Int. Conf. on Neural Networks. Volume 2., Washington, DC, IEEE Press (1989) 317–324
12. Maes, P.: A bottom-up mechanism for behavior selection in an artificial creature. In: Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats, Cambridge, MA, USA, MIT Press (1990) 238–246
13. Brooks, R.A.: Intelligence without representation. Artifical Intelligence **47**(1-3) (1991) 139–159
14. Ferrein, A.: Robot Controllers for Highly Dynamic Environments with Real-t ime Constraints. Doctoral dissertation, Knowledge-based Systems Group, Computer Science Department, RW TH Aachen University, Aachen Germany (2008)
15. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. J. Robotics and Autonomous Systems, Special Issue on Semantic Knowledge in Robotics (2008) to appear.
16. Dylla, F., Ferrein, A., Lakemeyer, G.: Acting and deliberating using golog in robotic soccer – a hybrid architecture. In: Proc. CogRob02, AAAI Press (2002)
17. Ferrein, A., Schiffer, S., Lakemeyer, G.: A fuzzy set semantics for qualitative fluents in the situation calculus. In: Proc. of ICIRA 2008, Part I. Volume 5314 of Lecture Notes in Computer Science. (2008) 498–509
18. Burgsteiner, H., Kröll, M., Leopold, A., Steinbauer, G.: Movement prediction from real-world images using a liquid state machine. Journal of Applied Intelligence **26**(2) (2007) 99–109
19. Gutmann, S., Weigel, T., Nebel, B.: Accurate, and robust selflocalization in polygonal environments. In: Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. (1999) 1412–1419