Carpe Noctem 2009

T. Amma, P. Baer, K. Baumgart, P. Burghardt, K. Geihs, J. Henze, S. Opfer, S. Niemczyk, R. Reichle, D. Saur, A. Scharf, J. Schreiber, M. Segatz, S. Seute, H. Skubch, S. Triller, M. Wagner, A. Witsch

Distributed Systems Group, University of Kassel, D-34121 Kassel, Germany

Abstract. Carpe Noctem is a Mid-Size League RoboCup team at the University of Kassel. It is part of the Distributed Autonomous Systems Laboratory of the Distributed Systems Group which is well known for its research contributions on middleware platforms, distributed system management, and software technologies for distributed systems. Carpe Noctem is a team of researchers and students who collectively aim at competing in the RoboCup championships. Several undergraduate students are involved in the research as part of their bachelor or master thesis, and the achieved results are directly integrated into the overall system. Carpe Noctem successfully developed a modular and platform-independent communication middleware for autonomous robots in the past years. The main research focus has now shifted towards representation and robust execution of cooperative strategies in dynamic domains. As such, RoboCup is again an ideal domain to evaluate our approach.

1 Introduction

The RoboCup team Carpe Noctem was established in 2005 at the Distributed Systems Group at the University of Kassel. The Distributed Systems Group has an international reputation for its successful research in areas such as middleware platforms [1] and distributed system management [2]. Recent research projects have addressed model-driven architecture (MDA) approaches for context-aware distributed computing [3], agent-based self-managing architectures, efficient secure communications in mobile systems, and evolutionary programming of sensor networks. Carpe Noctem is part of the Distributed Autonomous Systems Laboratory (DAS-Lab) which is a research project focusing on techniques and solutions for autonomous cooperation in mobile systems.

For the Distributed Systems Group at the University of Kassel the Carpe Noctem robot project is a research platform for the exploration of adaptive and autonomous mobile systems and for the development of a lean and efficient robot middleware in tight collaboration with other researchers [4]. Apart from this, Carpe Noctem research also targets the development of a model-based high-level specification approach for goals and behaviours of autonomous robots and the investigation of learning techniques for abstract team strategies.

Carpe Noctem easily meets the basic requirements for a successful participation in RoboCup tournaments. The robots are able to detect, follow, and acquire the ball, even under sub-optimal lighting such as daylight and shadows on the playing field. They are able to localise themselves based on line features. Obstacles on the playground are detected and avoided. Abstracted sensor data are communicated so that the robots inform each other about their beliefs and internal states. This allows them to estimate each other's decisions and react dynamically to these estimations.

In the following, we present an overview of our robot design. This paper is organised as follows: Section 3 gives an overview of the robot hardware. In Section 4, we describe our approach for a lean robot software framework, followed by an introduction to our communication approach in Section 5. Section 6 gives an overview of the behaviour engine employed, which controls the action of the corresponding robot. In Section 7, we discuss how this engine handles cooperative tasks.

Section 8 discusses our contributions to vision processing in the RoboCup domain. The paper closes with conclusions and outlook in Section 9.

2 The Carpe Noctem Team

The robotic soccer team employed by Carpe Noctem consists of five field players and one goalie. Two field players have been newly constructed in 2007, the rest of the team was built in early 2006. All robots share the same fundamental design, see Section 3 for details. However, due to more powerful motors employed, the two new robots are slightly shortened for better stability at high speeds.

Even in its early days in 2005, Carpe Noctem could already benefit from many years of experience in RoboCup, as two of the key architects have been involved in the The Ulm Sparrows RoboCup team at the University of Ulm since 2001. This was one reason why Carpe Noctem was ranked 7th at the RoboCup championships 2006 in Bremen after only 10 months of development.

In summary, Carpe Noctem participated succesfully in the following RoboCup events:

- RoboCup 2006, Bremen – ranked 7th
- Technical Challenge RoboCup 2006 – 10 out of 10 points
- German Open 2007 – ranked 5th
- German Open 2008 – ranked 4th

3 The CN2009 Robot Platform

The robot platform was designed from ground up to be as light-weight and functional as possible. It is a modular construction with four main functional parts: motion, kicker, control, and vision devices. The robot base plate comprises a triangular shape with concave sides. There is no preferred direction. Thanks to the concave shape, the ball is very near to the centre of the robot, hence allowing for tight manoeuvres while handling the ball.

Motion Device The motion device is a three-wheeled omni-directional drive which has become the de facto-standard in the RoboCup Mid-Size league. Our first generation robots rely on 70 W Faulhaber motors and Fraunhofer TMC200 motor controllers. The second generation is based on 150 W Maxon motors controlled by Fraunhofer VMC controllers. Both controllers are operated by a custom firmware developed in the Carpe Noctem team. It provides three effective degrees of freedom to influence the movement of the robot. The wheels have been developed by Carpe Noctem, specifically for application in the RoboCup domain. Figure 1 shows a mounted wheel. Currently, the wheel design is being evaluated and likely to be employed by *1. RFC Stuttgart* and the *Tribots* from University of Osnabrück.

Fig. 1. Omniwheel designed and built by Carpe Noctem

Kicking Device The kicking device is a turnable solenoid kicker. It may be rotated so that is can operate in three different directions. This is why a Carpe Noctem robot only has to rotate itself about 30 degrees in average to reach the ball.

The goalie employs a pneumatic kicker, since pneumatic devices are used for its extensions as well

Control Device The control device is a standard off-the-shelf notebook (Lenovo X60s/X61s). It communicates with its sensors and actuators via Firewire and Ethernet. We have further developed a small and energy-efficient circuit board named *Eth2CAN* that translates Ethernet to CAN and RS232, connecting motorcontroller and kicker-hardware to the notebook. It further serves electric decoupling purposes for connected devices.

Vision Device The vision device is an omni-directional camera (The Imaging Source DFK 21AF04). This approach is also the de facto-standard in modern RoboCup robots. The image processing is taken care of by the control device. The omnidirectional mirror is made of polished aluminium.

We are currently testing additional directed cameras and will most likely use at least one additional camera on the goal keeper, providing it with hybrid stereovision.

Other Sensors Besides the camera, the TMC200 and VMC motor controllers provide the robots with odometry data. Directional data is provided by an electronic compass. The fusion of vision data, odometry, and directional data allows for a very strong and robust localisation.

Figure 2 shows a Carpe Noctem robot from 2008. The picture was taken during the RoboCup GermanOpen 2008.

Fig. 2. Carpe Noctem roboter facing difficult lighting conditions

4 The Robot Software Framework

Each robot runs Ubuntu Linux 8.04 with standard packages. Most of the software is written in C#, which is eases rapid prototyping and teaching endeavours compared to native languages. Performance critical components such as image processing, however, are written in native C++ for efficiency reasons.

Every logical component of our software platform is implemented as an independent software module, for example the vision system, the motion, and the decision making process. The inter-module communication as well as the inter-robot communication is handled by the middleware framework Spica [5, 6]. Spica is one of the first finished research projects in the DAS-Lab. The next section gives a brief overview of the communication architecture and design goals of Spica

5 Communication Middleware

Spica handles communication in a transparent and efficient way, allowing for easy development of cooperative tasks. Technically speaking, communication is based on UDP, either in a unicast or multicast setup. There is no explicit differentiation between local or remote operation.

Messages and the communication infrastructure are specified in an abstract modelling language. The Spica Modelling Language (SpicaML) is a platform-independent, domain-specific modelling language that addresses communication in heterogeneous groups of distributed agents. The MDSD-based Spica approach specifically facilitates rapid prototyping and interoperation of heterogeneous software components. Ready-to-run source code is generated from a model automatically. The Spica appriach is introduced in [6, 7]. Communication channels are established dynamically by adopting a service discovery mechanism. This further enhances the robustness and interoperability of the framework [8].

6 Behaviour Modelling and Execution

In 2008, Carpe Noctem and the DAS-Lab started a new research project, aiming at a comprehensive teamwork model for autonomous agents acting in highly dynamic domains. The project so far resulted in a new specification language. ALICA [9] (A Language for Interactive Cooperative Agents) is a highly expressive language that features complete formal semantics. Publications are currently under peer review. The developed language is based on the teamwork model STEAM [10] and the BDI language 3APL [11].

One of our research goals is to provide means to model complex team behaviour in an intuitive way and to support reusability and platform independence through a model-driven design. This so far has yielded a graphical editor for ALICA strategies. It will be available as open source software after the RoboCup World Championships 2009. The editor relies on the Eclipse Framework [12], thus facilitating easy modifications and extensions through a plugin system. It serialises ALICA programs in a platform independent and interoperable XMI representation. Performance critical components of the language, such as utility functions and runtime conditions are automatically transformed into code in a model-oriented development fashion.

Modelled behaviours are executed by a prototypical implementation of ALICA's operational semantics. The one-on-one correspondence between semantics and implementation allows for direct evaluation of the theory in experimental settings such as the RoboCup championship. Preliminary tests have shown the robustness of this approach against sensor noise and unreliable communication, while providing means to react swiftly on dynamic changes in the environment.

7 Team Behaviour and Cooperation

ALICAallows cooperative strategies to be modelled directly from a global perspective. These strategies are executed directly by the robots, without an intermediate agent representation. Each robot estimates the decisions of its teammates and bases its decision on these estimations. Periodic communication of sensor data – for example the ball position – and internal states allows to correct both estimations and decisions dynamically. These internal states are defined by the ALICA semantics and represent intentions within the BDI model of each robot. Through special language constructs, namely *synchronisations*, these intentions can be raised to joint intentions [13]. This enables us to model the degree of commitment directly within the language. For instance in the RoboCup domain, a pass requires a commitment of both involved robots under tight time constraints, while an agreement on which robot attacks and which defends is less time critical and can even be done without explicit communication.

8 Image Processing

The Carpe Noctem approach for image processing is designed to be robust against changes in lighting conditions and to avoid extensive calibration tasks. The first important module to achieve robustness is a Gain Regulator for the camera used for omnidirectional vision. The *Gain Regulator* updates the gain settings of the camera based on estimations of the illumination on the camera lense and on different areas in the surroundings. This allows deriving appropriate gain settings even if the field is illuminated very inhomogeneously. The appropriateness of a setting is estimated based on the success of the localisation module, which highly depends on gain settings as line points are detected as contrast changes on scan lines in the greyscale image. This way, feedback from the localisation module is used to stabilise the gain settings.

In order to avoid time consuming calibration tasks for colour segmentation almost all calculations are done on the greyscale image. The only exception is the ball detection approach, which relies on a so-called *ROI channel*. High values represent interesting colours, less interesting colours are weighted lower. The ROI channel can be adjusted manually by roughly specifying interesting areas in the YUV colour space but also automatically by providing some sample images of the ball. A colour histogram is calculated from the sample images. After smoothing, it can be used as a lookup table to calculate the ROI channle almost without further modification. An attention control approach is used to focus on the most interesting areas of the ROI channel and finally, the ball is detected by applying a very simple but effective template matching approach on the gradient image of the ROI channel. This approach also proved to be very appropriate to detect balls with arbitrary colours.

Apart from being able to detect an arbitrarily coloured ball under different lighting conditions, a further challenge is to precisely estimate the 3D position of the ball. In particular, this is very important for the goal keeper. For this purpose the vision module applies a simple multi-hypothesis tracking and realises a two-fold sensor fusion approach to combine the information gathered from the omnidirectional and the directed vision systems. O the one hand,3D ball positions are derived from each camera separately by estimating the size of the detected object on the image. On the other hand, a 3D ball position is derived by considering the two cameras as hybrid stereo vision. The final 3D position estimation of the ball is calculated by fusing all these information.

Apart from the detection of basic features and objects, the vision module is also responsible for the self-localisation and tracking of moving objects like the ball and other robots.

9 Conclusions

The Carpe Noctem Mid-Size RoboCup team of the Distributed Systems Group at the University of Kassel has a research focus on lean software architectures, model-driven software design, and cooperative artificial intelligence. We use the RoboCup scenario as a testbed for our research as well as education and teaching efforts. Our robots and the robot control software were designed from ground up with modularity and extensibility in mind.

We will evaluate our new team cooperation approach during the German Open 2009 and incorporate the results for the RoboCup championship. In the future, we will further extend ALICA towards heterogeneous teams, ideally supporting different behaviour modelling approaches. Furthermore, we will examine possibilities to learn team strategies in a symbolic manner. For both research goals, evaluation data gathered from the two tournaments is highly important.

Bibliography

[1] Geihs, K.: Middleware challenges ahead. Computer **34** (2001) 24–31

[2] Weis, T., Ulbrich, A., Geihs, K., Becker, C.: Quality of service in middleware and applications: A model-driven approach. In: EDOC '04: Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International (EDOC'04), Washington, DC, USA, IEEE Computer Society (2004) 160–171

[3] Geihs, K., Barone, P., Eliassen, F., Floch, J., Fricke, R., Gjørven, E., Hallsteinsen, S., Horn, G., Khan, M.U., Mamelli, A., Papadopoulos, G.A., Paspallis, N., Reichle, R., Stav, E.: A comprehensive solution for application-level adaptation. Software Practice & Experience (2008) Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/spe.900.

[4] Utz, H., Sablatnog, S., Enderle, S., Kraetzschmar, G.K.: Miro – middleware for mobile robot applications. IEEE Transactions on Robotics and Automation, Special Issue on Object-Oriented Distributed Control Architectures **18** (2002) 493–497

[5] Baer, P.A.: Platform-Independent Development of Robot Communication Software. PhD thesis, University of Kassel (2008)

[6] Baer, P.A., Reichle, R., Zapf, M., Weise, T., Geihs, K.: A generative approach to the development of autonomous robot software. In Sterrit, T.B.R., ed.: Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems, IEEE, IEEE (2007) 43–52

[7] Baer, P.A., Reichle, R., Geihs, K.: The Spica Development Framework – Model-Driven Software Development for Autonomous Mobile Robots. In Burgard, W., Dillmann, R., Plagemann, C., Vahrenkamp, N., eds.: IAS-10, IAS Society, Proceedings of the 10th International Conference on Intelligent Autonomous Systems (2008) 211–220

[8] Baer, P.A., Weise, T., Geihs, K.: Geminga: Service Discovery for Mobile Robotics. In: The Third International Conference on Systems and Networks Communications, IEEE Computer Society Press (2008)

[9] Skubch, H., Wagner, M., Reichle, R.: A Language for Interactive Cooperative Agents. To appear as Technical Report (2009)

[10] Tambe, M.: Towards flexible teamwork. Journal of Artificial Intelligence Research **7** (1997) 83–124

[11] Hindriks, K.V., Boer, F.S.D., Hoek, W.V.D., Meyer, J.J.C.: Agent programming in 3APL. Autonomous Agents and Multi-Agent Systems **2** (1999) 357–401

[12] Beaton, W., d. Rivieres, J.: Eclipse Platform Technical Overview. Technical report, The Eclipse Foundation (2006)

[13] Levesque, H.J., Cohen, P.R., Nunes, J.H.T.: On Acting Together. In: Proc. of AAAI-90, Boston, MA (1990) 94–99