Carpe Noctem 2011

T. Amma, P. Burghardt, F. Gawora, K. Geihs, J. Henze, T. Heumueller, D. Kirchner, N. Kubitza, C. Luehring, S. Opfer, S. Niemczyk, R. Reichle, D. Saur, J. Schreiber, 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.

The Distributed Systems Group at the University of Kassel views the Carpe Noctem robot project as a research platform for the exploration of adaptive and autonomous mobile systems. Carpe Noctem research targets the development of a model-based high-level specification approach for goals and behaviours of autonomous robots, robust and adaptive execution of such specifications under time constraint, 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 four field players and one goalie. Two field players have been constructed in 2007, two new robots currently being constructed.

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.

3 The CN2010 Robot Platform

In 2010, Carpe Noctem began to construct a new robot platform, first to be employed in 2011. Its fundamental design principle is geared towards stronger cooperative play. As such, the platform allows for fast flat kicks. Moreover, two new actuators are planned, allowing the robot to get the ball under control easier and faster. These hardware features are meant to foster and employ a dynamic pass-based game style.

The robot platform was designed from ground up to be as robust and functional as possible. It is a modular construction with four main functional parts: motion, kicker, control, and vision devices.

Motion Device The motion device is a four-wheeled omni-directional drive which has become the de facto-standard in the RoboCup Mid-Size league. The CN2010 robots rely on 200 W Maxon brushless DC-motors, controlled by Maxon EPOS2 motor controllers. The wheels have been developed by Carpe Noctem, specifically for application in the RoboCup domain. Figure 1 shows a mounted wheel. The first generation wheels have been used in 2009 by the *1. RFC Stuttgart*, and served as a basis for the wheel design of the new *Tribots* robot developed in 2008/9. Based on the earlier and successful design, a new, lighter wheel was designed in 2010. Due to it being made almost entirely out of Polyoxymethylene (POM) it weights almost half of the earlier design.

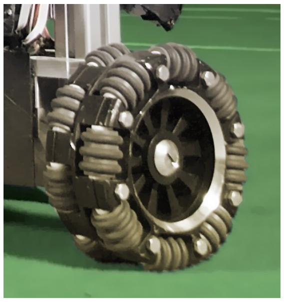

Fig. 1. Omniwheel designed and built by Carpe Noctem

Kicking Device The kicking device was also reworked in 2010. The new platform employs a fixed solenoid-based kicker with two shovels, one for flat kicks, one for high kicks. The goalie employs a pneumatic kicker, since pneumatic devices are used for its extensions as well.

Control Device The control device is a standard IPC featuring a modern Intel Core i7 dual core processor, and two independent Firewire 1394b controllers. This allows it to process data from multiple cameras. Actuators are connected via CAN-to-USB and CAN-to-Ethernet interfaces.

Vision Device The vision device is an omni-directional camera (A PointGrey Flea2). 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. Additional directed vision devices are planned but not yet employed on the field players.

Our goal keeper is using an additional directed camera, providing it with hybrid stereovision and a much longer field of view.

Other Sensors Besides the camera, the 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.

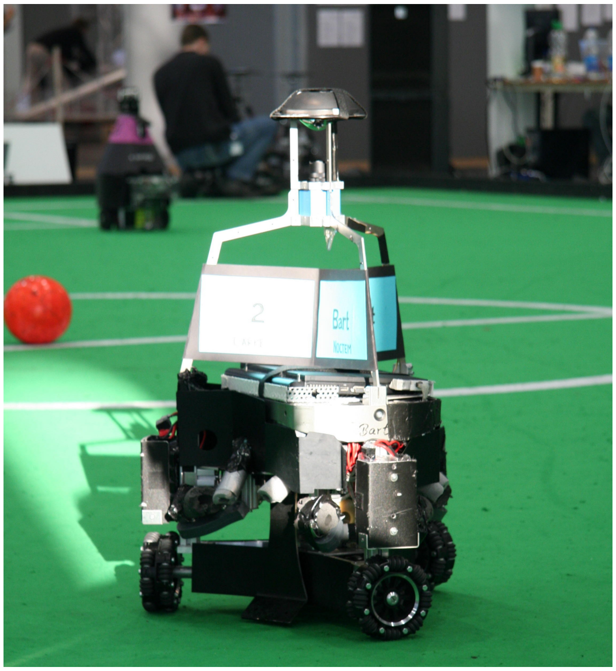

Fig. 2. Carpe Noctem roboter facing difficult lighting conditions

4 The Robot Software Framework

Each robot runs Ubuntu Linux 10.10 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 [4, 5]. 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 approach is introduced in [5, 6]. Communication channels are established dynamically by adopting a service discovery mechanism. This further enhances the robustness and interoperability of the framework [7].

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 [8] (A Language for Interactive Cooperative Agents) is a highly expressive language that features complete formal semantics. Emperical results obtained during the RoboCup 2009 and German Open 2010 were published in [9, 10, 11]. The developed language is based on the teamwork model STEAM [12] and the BDI language 3APL [13].

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 yielded a graphical editor for ALICA strategies. It is available as open source software on our website (http://carpenoctem.das-lab.net/research/alica). Future releases are under active development. The editor relies on the Eclipse Framework [14], 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 an 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. An evaluation during the RoboCup 2009 has 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

ALICA allows 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 or 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 [15]. 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.

Current research focusses on more expressive language elements, which allow for a distributed representation of constraint satisfaction problems to model the team's behaviour.

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 channel 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. On 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 evaluated our new team cooperation approach during the tournaments in 2009, and were able empirically show the robustness of our approach. Currently, we further extend ALICA towards more expressive modelling elements, such as constraint satisfaction problems.

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

[5] 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

[6] 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

[7] 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)

[8] Skubch, H., Wagner, M., Reichle, R.: A language for interactive cooperative agents. Technical report, University of Kassel (2009)

[9] Skubch, H., Wagner, M., Reichle, R., Triller, S., Geihs, K.: Towards a comprehensive teamwork model for highly dynamic domains. In: Proc. of the 2nd Int. Conf. on Agents and Artificial Intelligence. (2010)

[10] Skubch, H., Saur, D., Geihs, K.: Resolving conflicts in highly reactive teams. In: KIVS 2011, Open Access Series in Informatics (In Press)

[11] Skubch, H., Wagner, M., Reichle, R., Geihs, K.: A modelling language for cooperative plans in highly dynamic domains. Mechatronics **In Press, Corrected Proof** (2010) –

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

[13] 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

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

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