The b-it-bots RoboCup@Home 2009 Team Description Paper*

Dirk Holz, Jan Paulus, Thomas Breuer, Geovanny Giorgana, Michael Reckhaus, Frederik Hegger, Christian Müller, Zha Jin, Ronny Hartanto, Paul Ploeger and Gerhard Kraetzschmar

Bonn-Rhein-Sieg University of Applied Sciences, Computer Science Department,
<first name>.<last name>@inf.h-brs.de

Abstract. This paper presents the b-it-bots RoboCup@Home team and the mobile robot platform *Johnny Jackanapes* for the RoboCup@Home 2009 competitions as well as its robot control architecture. Components for object recognition and manipulation as well as speech recognition are described in detail.

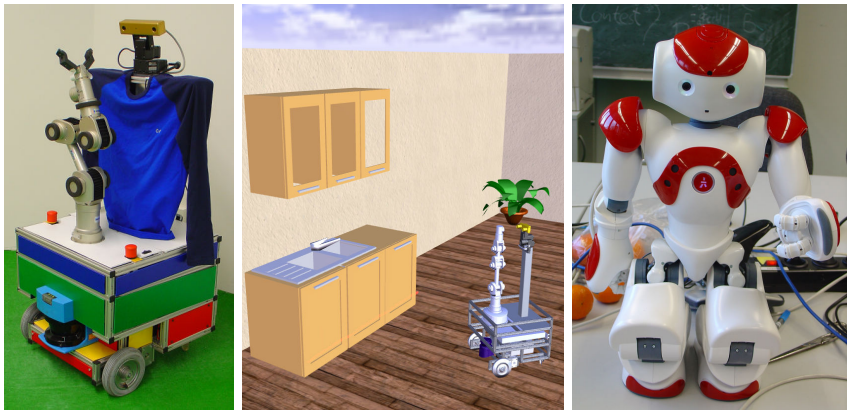
1 Introduction

The b-it-bots RoboCup@Home team at Bonn-Rhein-Sieg University of Applied Sciences has its roots in a former middle-size league team that was known as GMD Robots and AIS/BIT Robots. The b-it-bots team participates in RoboCup-@Home since RoboCup 2007 in Atlanta. The team consists of Bachelor, Master and PhD students, who are advised by two professors from the Bonn-Rhein-Sieg University of Applied Sciences. The results of several research and development (R&D) as well as Master theses projects had already been successfully integrated into the former middle-size league software system. Through this kind of graded course modules our RoboCup team is strongly interwoven with the Master by Research course in Autonomous Systems, which is offered at the Bonn-Rhein-Sieg University of Applied Sciences.

Our main research interests include mobile manipulation, environment modeling, computer vision and human/machine interaction. Our approach is to first identify and evaluate in each subfield the state-of-the-art and the best practice solutions currently available, implement and integrate applicable algorithms, and to develop our own custom approaches.

In this paper we will present our robot control architecture and the underlying concept of autonomous components (ACos). Section 2 will briefly present the robot platforms used in the upcoming events. The architectural concepts are described in Section 3. The ACo for navigation, motion control and simultaneous localization and mapping (SLAM) is presented Section 4, and a detailled description of the approaches used in the ACo for learning, recognizing and manipulating objects is provided in Section 5. Recent results in speech recognition for the Human-Robot-Interaction ACo are presented in Section 6.

* We gratefully mention the continued support of the team by the b-it Foundation.



(a) (b) (c)

Fig. 1. Primary robot platform "Johnny Jackanapes" moving around the BRSU campus (a) and a simulated kitchen environment (b). A Nao is used as secondary platform (c).

2 Robot Platform

Our primary robot is based on a modular mobile platform called *VolksBot* [15], which has been designed for rapid prototyping of robot applications in education, research and industryby the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS).

The team uses a customized variant, see Fig. Figure 1(a), with a Neuronics Katana 6M180 robot arm as an integrated manipulator. It provides five degrees of freedom w.r.t. the gripper's position and orientation in its workspace and has an operation radius of 60 cm. The manipulator has a two-fingered gripper, which is equipped with infrared reflectance as well as force sensors. The arm can handle a maximum payload of 500 g and is mounted in a way to provide good reachability and maneuverability. The primary sensor for perceiving the environment is a SICK LMS 200 laser range finder mounted in the robot's center of rotation. It provides accurate range measurements to surrounding objects intersecting the 2D scan plane in an angular range of 180°. We use an angular resolution of 1° and a continuous sending mode over RS422. In this setup, the laser range finder delivers 2D scans containing 181 range measurements with a frequency of 75 Hz. The drive unit used for locomotion uses a differential drive with two actively driven wheels, powered by two 150 W motors, and two castor wheels to enhance rotating and stability under load. The robot's maximum velocity is 2 m/s.

In addition to our primary platform "Johnny Jackanapes", we plan to use an academic edition of the Nao robot (see Figure 1(c)), that is also used in the RoboCup Standard Platform League.

3 Robot Control Architecture

The current architecture of our robot may be described best as a loosely integrated aggregation of dedicated autonomous components (ACos). This is, on one hand, in contrast to the classical three layer architecture (3T) consisting of controllers (*skill level*), a sequencer (*execution level*) and a planner (*deliberative level*). On the other, it resembles some principles of 3T, see [10]. For example we employ one combined navigation, localization, drive unit (NLD) which works self sufficiently while executing tasks like path planning and following or tracking of a human operator. Similarly we have a combined manipulation / object recognition component and a human robot interaction component which falls into the subcomponents of face recognition and speech synthesis and recognition. These components do not match to any single layer in the 3T architectural pattern since they themselves already comprise several of these levels. The NLD, for example, offers services for low level motor control, guidance or path following yet also contains path and motion planners as well as own deliberation and sequencing components. Furthermore, the NLD maintains different environment representations and contains several components for simultaneous localization and mapping (SLAM). It can work self sufficiently on the achievement of goals (like: move to the refrigerator) and may also decide all by itself when to stop. The very same holds true for our combined manipulation / object recognition / pan-tilt-camera component which takes care of searching, fixating, identifying and grabbing a known object.

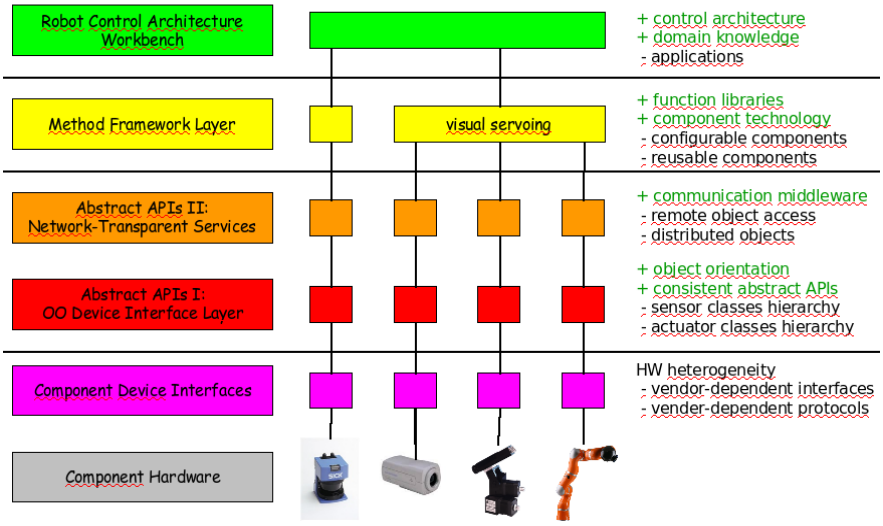


Fig. 2. Architecture

We define an ACo as a unit of composition, containing at least two (possibly all) of the aforementioned classical three levels and working self-sufficiently on goal achievement of its respective task. In our architecture, the integration of and communication between ACos is handled via the ICE middleware[1]. Integration takes place only on the level of ACos (in the *Method Framework Layer*, see Figure 2) and a classical scheduler sequences all operations. As the scenarios in the @Home league are quite fixed and well-defined, fixed behaviors expressed as a simple finite state machines can substitute a centralized planner. This closely resembles the 3T architecture while the introduction of ACo is in contrast to it.

Although this approach together with a *gray box integration* method allowed us a jump start in the league by reusing existing software components, it has severe drawbacks. Right now it is not possible during runtime to share access to all devices from different top level applications. Thus, we are now working on a clean support of all six abstraction levels which is aided by an upcoming EU supported project focussing on the integration of best-in-class algorithms, hardware components and benchmarking methodologies.

4 Navigation, Motion Control and SLAM

All problems related to the platform base are encapsulated in a single ACo. This ACo provides the functionality for autonomously building internal environment representations and localizing the robot in the arena by means of an efficient approach to SLAM, planning and following paths, tracking and following human guides as well as searching the environment and approaching target poses. It is split into three components: a *perception component* for acquiring and processing sensor measurements, an *actuation component* for controlling the robot's effectors and a *task execution component* implementing a multi-layered architecture and a main control loop linking perception and actuation. For a concrete example, consider the task of following a human guide. A virtual sensor within the perception component detects and tracks the human guide, a motion controller activated and run in the task execution component calculates proper vehicle velocities for following the tracked guide which are than transformed to platform-specific motion commands in the actuation component.

In the interface to the robot control architecture and other ACos we distinguish three different types of functions. *Low-level functions* allow for directly controlling the robot's motion, reading odometry measurements and accessing the laser range finder. *Component functions* provide information about the current state of the ACo, e.g. currently executed task, execution state or details on unexpected failures. *Specific ACo functions* allow to select and enqueue tasks for execution. These include navigating to a certain location (e.g. the living room or the refrigerator), approaching a certain pose (e.g. for grasping an object) or accessing the robot's internal environment representation. The latter consists of

[1] ICE is free software and released under the terms of GNU General Public License (GPL). It is available at `http://www.zeroc.com`

an allocentric geometric representation of environmental structures and a topological layer containing, amongst others, a vector of objects and locations with position, orientation, shape and name. Descriptions of the algorithms used in this component can be found in [8] and [9]. It is, furthermore, planned to publish all algorithms used in this ACo possibly helping to establish a common code base in the RoboCup@Home league.

5 Object Recognition and Manipulation

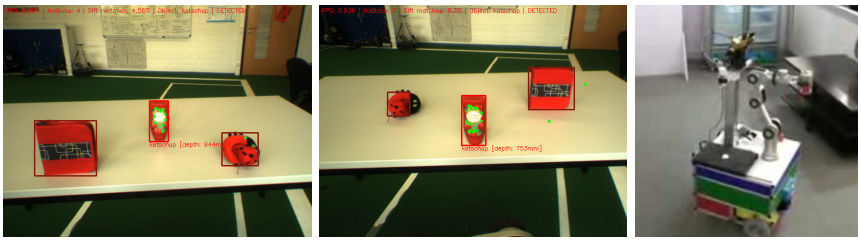
Another ACo is, with reference to Figure 2, *visual servoing*. It provides functionality for learning, detecting and recognizing objects using a stereo camera system as well as for controlling the arm and the pan-tilt the camera is mounted on. This allows the robot to search for objects in a larger field of view. Note that the face recognition component in the HRI ACo, presented in the following section, also uses this functionality to focus on a detected face.

For recognizing objects, we follow a two-step recognition approach by, first, performing a color segmentation of the image (based on the approximate color of the object we're searching for) and, second, extracting and matching features from segmented image regions with those stored in an object database. For each object, learned online or beforehand, we store the object's dominant color (in *YUV* color space) as well as the feature descriptor for the second recognition step. For color segmentation, we apply a linear color thresholding partitioning the *YUV* space with linear boundaries (i.e. planes in 3-dimensional space). A particular pixel is then classified according to the partition it lies in. Extracted regions, resulting from image growing around extracted pixels, are then passed to the feature-based approach.

In the feature-based approach we use *scale-invariant feature transforms* (SIFT), proposed by Lowe in 1999 [11]. SIFT features are based on the texture on an object's surface at particular points of interest. The interesting characteristic of SIFT features in the context of object recognition is that they are invariant to image scale and rotation and locally stable in the presence of noise, partial occlusion and changes in illumination. Extracting features using SIFT consists of the following four steps [12]. In the first step, the points of interest – *keypoints* – are extracted by smoothing the input image using Gaussian filters at different scales and calculating difference images (difference of Gaussians) between adjacent scales. Minima and maxima of the Difference of Gaussians that occur at multiple scales are used the keypoints. In the second step, keypoints and nearby pixels are examined to reject unstable keypoints. In the third step, each keypoint is assigned one or more orientation based on local image gradient directions. This is the key step in achieving invariance to image rotation. In the fourth stage a descriptor vector is computed for each keypoint, too make comparison between keypoints possible. Matching feature descriptors with those that are stored for the objects being searched for, is carried out as an approximate nearest neighbor approach using the *Best-Bin-First* [3] algorithm. Neighbors whose Euclidean dis-

tance to the search descriptor exceed a certain threshold are rejected. A bigger part of false matchings arising from background clutter are thereby discarded.

An example of applying the object recognition approach on real-world objects is shown in Figures 3(a) and 3(b) where the object to grasp is a red ketchup bottle. Once, the object is detected and recognized, we move the pan-tilt unit to center the object in the two camera images and calculate the 3D pose of the object in the robot's coordinate frame and a vehicle pose that allows for grasping. After positioning the vehicle, by invoking the navigation ACo, we perform a blind grasp with the manipulator, i.e. we assume that the position of the object is not changed after calculating its 3D pose and the inverse kinematics for grasping it. For handling situations like that and to increase grasping precision, we are currently working on the integration of a visual servoing approach, i.e. controlling the arm's movement for grasping in closed loop while tracking both, the gripper and the object. After grasping an object the arm is moved into a fixed delivery pose, see Figure 3(c), and the robot can move on to the person who ordered the drink. A detailled description of the approach can be found in [13].



(a) (b) (c)

Fig. 3. Recognizing the "ketchup" bottle from three red objects in (a) and (b). Johnny delivering a coke-can during the open challenge in Suzhou, 2008 (c).

6 Speech Recognition and Synthesis

The ACo for Human-Robot Interaction (HRI) comprises speech recognition and synthesis as well as face detection and recognition. In the context of this paper we will focus on speech-based HRI including components for dialog generation, speech synthesis and speech recognition.

The core of the speech-based HRI components is the dialog system that handles the communication between the robot(s) and human users. In the dialog system, communication can be initiated from both a human user giving a spoken command and the control architecture, e.g. for requesting a confirmation or more information about an assigned task. Whereas the dialog system is integrated into the central scheduling component of the architecture, the components for speech synthesis and recognition are bundled in the HRI ACo together with the components for face detection and recognition.

For the speech synthesis component, we evaluated different approaches with respect to intelligibility and naturalness of their output, namely eSpeak [1], FreeTTS [2], Festival [5], Flite [4] as well as the Microsoft Speech API (using the voices "Sam", "Mike" and "Mary") and the MAC OS X Leopard Speech Synthesis API (using the voice "Alex"). The latter is used in our current setup as it sounds most natural, compared to the others, and is very intelligible. It uses both *ruledriven* and *corpus-based* speech synthesis techniques. During excessive tests in our lab as well as at the GermanOpen and world championship 2008, "Alex" produced good and natural speech output, which is also understandable in the presence of noise.

The speech recognition component should be able to understand speaker-independent speech input, while being robust against noise. To achieve this goals, we evaluated the Microsoft Speech SDK and the CSLU Toolkit [14] w.r.t. their recognition accuracy and reliability. Whereas evaluating the speech synthesis systems above was purely subjective, here we use a metric to measure the recognition accuracy. A common measure for comparing speech recognition system is the *word-error-rate* [7]. This, however, seems to be less appropriate when recognizing complete phrases, since we are not interested in the number of errors in a phrase, but in the number of completely correctly recognized phrases. Instead, we propose a new metric – the *Phrase Recognition Accuracy* (PRA):

$$PRA = \frac{CR}{CR + FR + NS + NR} \tag{1}$$

where CR is the number of correctly recognized phrases, FR the number of falsely recognized phrases, NS the number of recognized phrases when no command was given and NR the number of phrases that were not at all recognized.

Using this new metric revealed that the Microsoft Speech SDK performed better than the CSLU Toolkit, especially in noisy environments and when using long phrases. The configuration of the Microsoft Speech SDK yielding the best results, include system rejection rate of 25 %, starting each phrase with a predefined prefix (here we used the name of the robot, i.e. "Johnny") and using trained profiles of an easy understandable speaker for untrained speakers. Furthermore, the dialog system requests confirmations for irreversible actions in order to avoid the execution of unwanted tasks.

During the GermanOpen and the world championship 2008, this approach has proven well. We achieved good results for both speakers the system was trained for and untrained speakers. Most of the given commands have been correctly understood, even in the presence of noise. Further details of the evaluation can be found in [6].

7 Conclusion

In this paper we presented our loosely coupled control architecture based on autonomous components (ACos). Furthermore, we presented the ACos for object

recognition and speech recognition. For object recognition we applied a two-steps mechanism based on color segmentation and SIFT features in stereovision images. The depth information of the images is then used to calculate the pose of the object being searched for. Our speech recognition component is based on the Microsoft Speech SDK and a custom dialog system.

References

1. espeak text to speech. Available at http://espeak.sourceforge.net.
2. FreeTTS 1.2 - A speech synthesizer written entirely in the Java programming language. Available at http://freetts.sourceforge.net.
3. J. S. Beis and D. G. Lowe. Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1000, 1997.
4. A. W. Black and K. A. Lenzo. Flite: a Small Fast Run-Time Synthesis Engine. In *Proceedings of the 4th ISCA Workshop on Speech Synthesis*, 2001.
5. A. W. Black and P. A. Taylor. The Festival Speech Synthesis System: System documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK, 1997.
6. T. Breuer. Advanced Speech-based HRI for a Mobile Manipulator. Technical report, Bonn-Rhein-Sieg University of Applied Sciences, Germany, 2008. Avaible online in the RoboCup @Home Wiki.
7. G. Damnati. Evaluating speech recognition in the context of a spoken dialogue system: critical error rate. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001.
8. D. Holz, J. Paulus, T. Breuer, G. Giorgana, R. Hartanto, W. Nowak, J. Jackanapes, P. Ploeger, and G-Kraetzschmar. The b-it-bots RoboCup@Home 2008 Team Description Paper. In *RoboCup-2008, Suzhou China*, 2008.
9. Dirk Holz, Christopher Lörken, and Hartmut Surmann. Continuous 3D Sensing for Navigation and SLAM in Cluttered and Dynamic Environments. In *Proceedings of the International Conference on Information Fusion (FUSION)*, 2008.
10. D. Kortenkamp, R. P. Bonasso, and R. Murphy. *Artificial Intelligence and Mobile Robots*. AAAI Press / The MIT Press, 1998.
11. D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, 1999.
12. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
13. J. Paulus. Vision Based Mobile Manipulation. Master Thesis, Bonn-Rhein-Sieg University of Applied Sciences, Germany, 2008. Avaible online in the RoboCup @Home Wiki.
14. S. Sutton, R. Cole, J. De Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y. Yan, B. Rundle, K. Shobaki, P. Hosom, A. Kain, J. Wouters, D. Massaro, and M. Cohen. Universal speech tools: The CSLU toolkit. In *In Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1998.
15. T. Wisspeintner, W. Nowak, and A. Bredenfeld. Volksbot - a flexible component-based mobile robot system. In *Proceedings of the RoboCup Symposium*, 2005.