Making High-Tech Service Robot Platforms Available

Florian Weisshardt[1], Ulrich Reiser[1], Christopher Parlitz[2] and Alexander Verl[1]
[1] Fraunhofer IPA, Robot Systems, Stuttgart, Germany
[2] SCHUNK GmbH & Co. KG, Lauffen, Germany

Abstract

The development of reliable, robust and capable service robot platforms is expensive and time consuming. On the other hand, stable hardware is a prerequisite to be able to concentrate on the targeted research area. This paper proposes a strategy to make such high-end service robot platforms available as research platform, by means of the example of Care-O-bot® 3 developed at Fraunhofer IPA. Several methods to provide robotic hardware for research are introduced, starting from the maintenance of a common open-source repository, to the making available of simulation models of the hardware components and the possibility of remote access via a web-interface.
It is shown, that Care-O-bot® 3 is particularly suited as research platform through the modular structure of its hardware components. Thus, a very flexible set-up for Care-O-bot® 3 is possible, making it easy to adapt to the specific requirements of different robotic research fields.

1 Introduction

Recently, some capable robot platforms have been built: Care-O-bot® 3 [1], DESIRE [2], Armar III [4], Asimo [13], Friend [11] etc. However, up to now, they are unique platforms usually available only at one place to a very limited clientele.
Furthermore, the high integration effort for components into different hardware platforms and different component frameworks hampers the all over progress of service robot capabilities. Hence, the authors of this paper are convinced that distributed development on common hardware platforms could be a considerable driver for the development of service robot technologies and applications.
This paper therefore proposes a methodology to make high-tech platforms available to the general robotics research community. The methodology consists of three major steps:

- Provide a common open-source repository for the platform's hardware
- Provide simulation models of hardware components
- Provide remote access to the platform's hardware

The second and third step is largely motivated by the fact that most platforms contain expensive hardware components, not allowing every interested institute to buy such a robot. Thus, we obtain different access layers to the hardware according to Figure 1. Using the mobile platform Care-O-bot® 3 by way of example, this paper is organized as follows: after shortly introducing the modular Care-O-bot® hardware in the next section, the open-source repository of Care-O-bot® 3 is presented in section 3. The forth section describes the simulation packages available for the robot, and the fifth section describes the use of a Web Portal [3] for remote access and organization of distributed development. Care-O-bot® 3 was chosen, because it is a complex mobile manipulation service robot with multimodal sensory input and more than 25 individual degrees-of-freedom, offering a variety of possible research areas and so interesting for various research institutions.

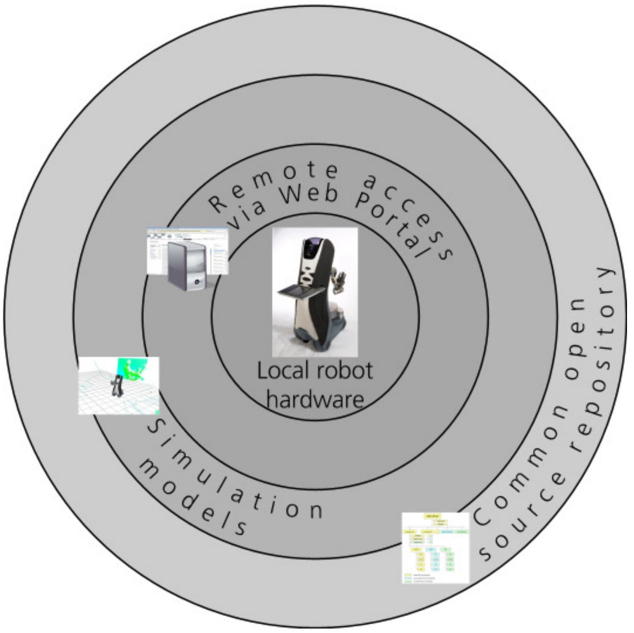

Figure 1 Different access models on hardware platform.

2 Hardware

In order to evaluate the proposed strategy of making service robot platforms available to the community, it is currently implemented on Care-O-bot® 3 [12]. The following sub sections therefore introduce the hardware of Care-O-bot® and motivate for the use of modular industrial grade components.

2.1 Care-O-bot® 3 Set-Up

Figure 2 shows the major independent hardware components of Care-O-bot® 3. The robot is driven by four wheels. Each wheel's orientation and rotational speed can be set individually. This gives the robot an omni-directional drive enabling advanced movements and simplifying complete kinematic chain (platform-manipulator-gripper) control. The base also includes the Li-Ion battery pack (50 V, 60 Ah) for the robot, laser scanners and one PC for navigation tasks. The size of the base is mainly defined by the required battery space. Nevertheless, the maximal footprint of the robot is approx. 600 mm and the height of the base is approx. 340 mm. The torso sits on the base and supports the sensor carrier, manipulator and tray. It contains most of the electronics and PCs necessary for robot control. The base and torso together have a height of 770 mm. The manipulator used is based on the Schunk LWA3, a 7-DOF light-weight arm. It has been extended by 120 mm to increase the work area so that the gripper can reach the floor, but also a kitchen cupboard.

It has a slim quick-change system between the manipulator and the 7-DOF Schunk Dexterous-Hand. This robotic hand has tactile sensors in its fingers making advanced gripping possible. The force-torque sensor is used for force controlled movements like opening draws and doors, but also for teaching the robot new tasks by physical interaction with the human. The quick-change system allows the use of other grippers and robotic hands like Schunk Anthropomorphic-Hand. Special attention was paid to the mounting of the arm on the robot torso. The result is based on simulations for finding the ideal work space covering the robot's tray, the floor and area directly behind the robot following the 'two sides' concept developed. By this concept, working side and human interaction side of the robot are separated to back and front of the robot [8]. Since the manipulator has a hollow shaft no external cables are needed. The tray is the main human-robot interface attached to the robot. It serves for the passing of objects, but also by integrating a touch screen for traditional human-computer interaction. If the tray is not used it can be retracted so that the robot is as compact as possible in stand-by.

The robot has a sensor carrier carrying high-resolution firewire stereo-vision cameras and 3-D-TOF-cameras, enabling the robot to identify, to locate and to track objects and people in 3-D. These sensors are mounted on a 5-DOF positioning unit allowing the robot to direct his sensors in any area of interest. It is very important in our concept not to create a human-like face with these sensors and this is very difficult to avoid. However, this set-up can be easily adapted to special needs by the concept of modular components.

2.2 Modular Components

Building complex service robots is only possible by choosing a modular approach. Modules with local "intelligence", i.e. sensory data preprocessing and actuator control are combined into a motion of the robot system.

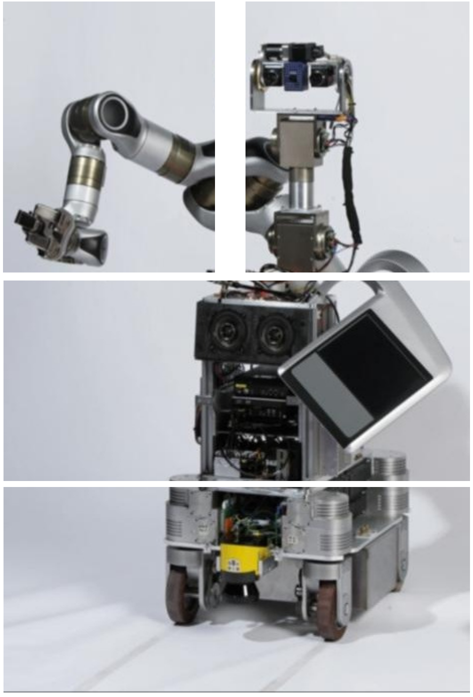

Figure 2 Independent main units of Care-O-bot® hardware set-up: mobile base (bottom), torso (middle), arm (top left) and sensor head (top right).

A module is defined as the smallest autonomously working unit of a robot, taking aside common resources like power. These units are not necessarily only actuated modules like e.g. a mobile base, but can also be other functional units like a head with visual sensors. Ideally these modules can be configured into any robot system because they have common interfaces and can therefore be easily exchanged for other hardware implementations with the same or even different functionality. The approach "from individual modules to complex robot structure" forms the basis for economical concepts of service robots suitable for daily use in research and development.

This development methodology was not only used for the main units of Care-O-bot® as seen in Figure 2, but also in the actual functional units as well. The mobile base of Care-O-bot® uses 4 identical drive units. A new version with hub drive is currently being developed and will be able to be integrated into the existing robot with minimal further changes to hardware or software. The manipulator of the robot also uses individual modules. The joint is the main concept of the manipulator. Instead of traditional linked design, joint modules are combined into a motion system. A module is defined as the smallest autonomously working unit of the manipulator. These units are not necessarily only joint modules, but can also be other functional units like force-torque-sensors or visual sensors. These modules can be configured into any kinematic chain system because they have common interfaces. They need to be combined by mechanical brackets. To be able to easily combine such modules they need a direct interface to PC control, as seen in Figure 3, allowing to access all arm modules with high accuracy without relying on an external arm controller like often seen in industrial robot manipulators. The manipulator can be reconfigured and reused for further applications. The manipulator works with 24 V DC power and can therefore be used in mobile battery powered applications

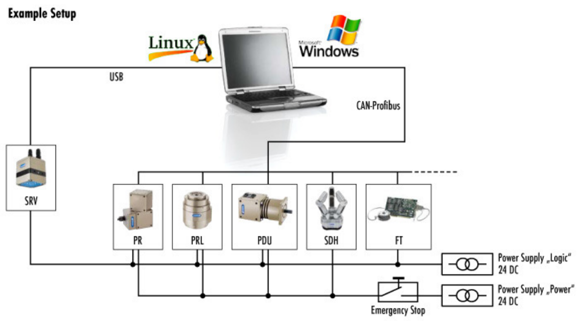

Figure 3 Different access models on hardware platform.

3 Open-Source Repository

In the first place, the open-source repository provides drivers for the Care-O-bot® 3 hardware components, including drivers for sensors and actuators e.g. laser scanners, cameras, and complex components like the Schunk dextrous hand SDH, the redundant Schunk LWA3 manipulator or the 8 DOF omnidirectional base. Figure 5 shows some of the currently available components inside the open-source repository and their dependencies. Besides the driver stack there's mainly a stack for simulation components and one with sample applications.

These components are implemented in multiple widespread open-source frameworks such as OROCOS [5] and ROS [6] in order to keep the integration effort into high level algorithms as minimal as possible. The goal is to provide open-source as close to the hardware as possible, but in a layered architecture making higher level control algorithms independent from hardware. Furthermore, the repository is structured in decoupled components which are connected via common interfaces. In the case of the ROS implementation these common interfaces are messages exchanged via topics or services. The message type is chosen according to the common ROS message definitions and interaction API. This reduces the efforts of exchanging software on different hardware platforms and offers the possibility to easily substitute or integrate new components.

Figure 4 shows the structure of the open-source repository. Currently two open-source frameworks, OROCOS and ROS, are supported, but the structure is open to support more frameworks. The openness for other frameworks is enabled by separating each component into a common part where the pure algorithm is located and framework specific parts which are mainly wrappers for the communication. The common part contains the actual functionality of each component. Additionally to the packages for the hardware drivers, there are common used utility packages which can be used by other components. Thus, the repository can be very easily enhanced for other frameworks and other hardware platforms.

Independently from available hardware, developers can work jointly in an open-source repository. This is often sufficient for components largely independent from the actual hardware set-up of the service robot (e.g. speech recognition modules). More hardware dependent developments like manipulation controllers can be tested in the next access level, as shown in Figure 1, the simulation environment. After successfully passing this level and further testing is required, a remote access to real hardware can be realized by a Webortal.

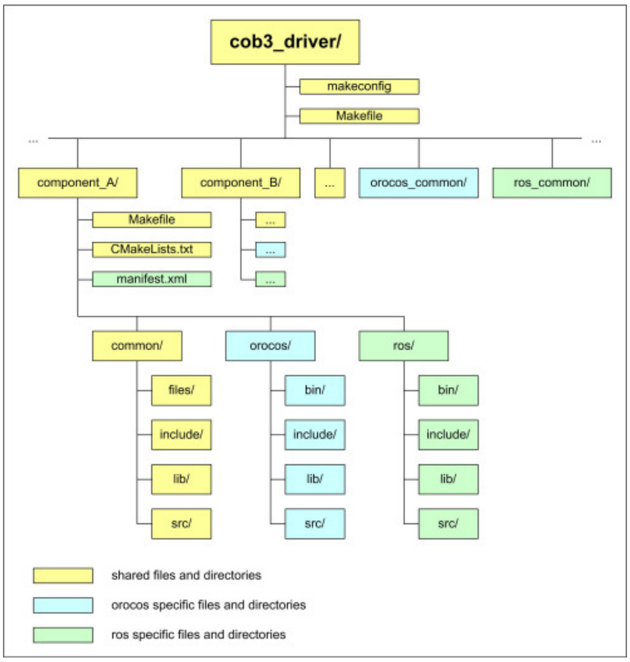

Figure 4 Structure of the open-source repository.

4 Simulation Models

Most hardware components of Care-O-bot® 3 are available as simulation components. The interfaces of the simulated components do not differ from the real hardware drivers. By this prerequisite simulated drivers and real hardware drivers can be substituted without changing the rest of the software system. Due to modelling and computing performance different simulation layers are needed. On the one hand, an overall robot and environment simulation on a scenario level for general robot activities is needed. To satisfy this kind of application a simulation model of the Care-O-bot® 3 is developed for the simulation environment Gazebo [7]. On the other hand, more detailed simulation models are necessary for developing components which are closer to the hardware, e.g. arm controllers.

Developing in simulation is cheap because no expensive hardware is necessary. It is safe because nobody is hurt by the virtual hardware and the hardware damages neither itself nor other objects. Working with simulation is comfortable because simulation models can be easily distributed over multiple developers without having to ship hardware or travelling to the hardware's location. Finally, the development or testing on a real hardware system is necessary to verify the functionality of the developed components.

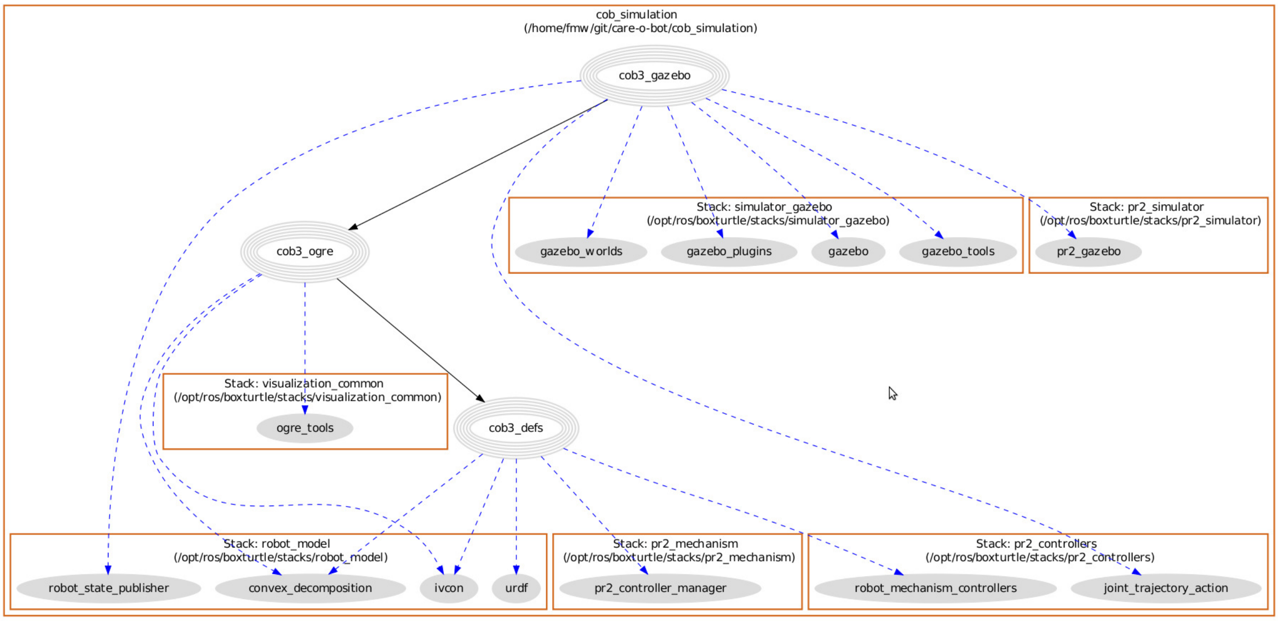

Figure 5 Example for a dependency graph of different ROS packages for the simulation stack of the Care-O-bot® repository.

The Care-O-bot® 3 Gazebo simulation model includes kinematical and dynamical models of each component as well as a composed model for the whole robot as you can see in Figure 6. Most sensors and actors are available with the same interfaces like the real hardware, e.g. messages for laser scanners or joint angles of a manipulator.

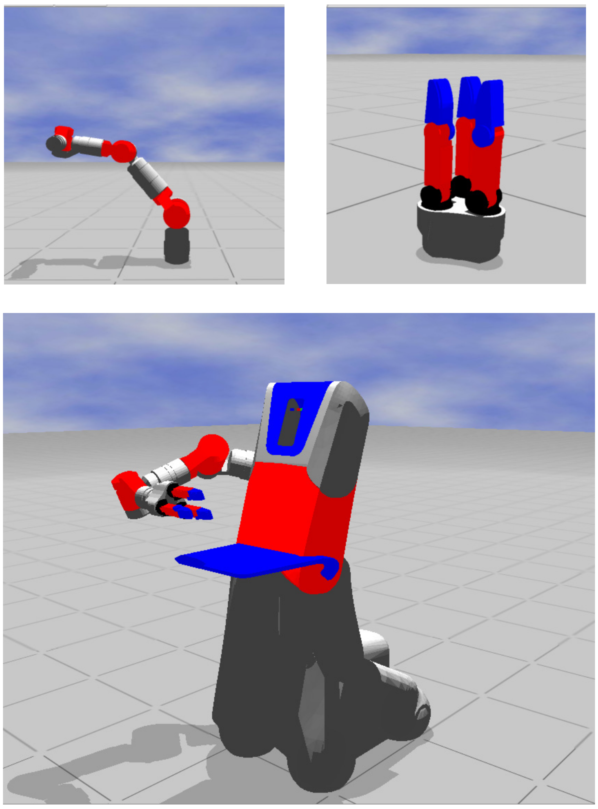

Figure 6 Simulation models of single hardware components e.g. the Schunk Light Weight Arm (top left) or the Schunk Dextrous Hand (top right) and a whole robot model of Care-O-bot® 3 (bottom).

From the view of a high level component the interface look alike, therefore the simulated components can be seamlessly substituted by real components. Having the same interfaces in simulation and on the real hardware the high-level components can be easily ported and run on a real hardware. Thus, the development process can be accelerated by combining the advantages of developing in simulation and on real hardware.

5 Remote Access via Webportal

The webportal developed within the project DESIRE [3] constitutes an important link between the open-source repository and the actual robot hardware. It is based on the collaboration platform trac [9] and offers support during all phases of software development from specification over design and implementation to integration and tests as visualized in Figure 7. It provides remote access to the hardware by permitting control, configuration and debugging of software components on real hardware. In addition, it offers an automatic build and test system for distributed development. By checking in a component into the repository it automatically becomes available on the robot by an automatic deployment and testing system.

Thus, the webportal constitutes a powerful tool for distributed development on one common hardware platform and provides extensive remote support and maintenance possibilities. In particular, it provides the possibility to test software components or simple applications that were developed e.g. on the basis of the available simulations, on real hardware without the need of local access and the connected travelling efforts and costs.

The benefits of the webportal for distributed development could be assessed in DESIRE, where 12 partners collaborated in the development of software on one common hardware platform. Partners distributed all over Germany were able to set up and implement a complex service robot system by accessing the hardware remotely, reducing the must to travel by a great extent.

In the context of this paper, especially the building blocks “remote access” and “process control” will be highlighted in the following.

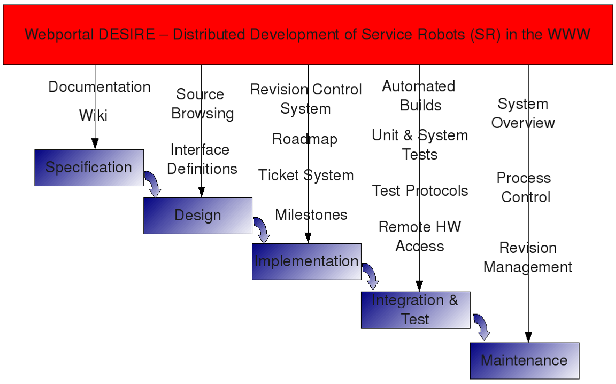

Figure 7 Supportive tools comprised in the webportal with respect to the different development phases.

5.1 Remote Hardware Access

The webportal runs on a dedicated server, which is connected both to the internet and to the platform computers as shown in Figure 8. This allows various scenarios of distributed development and testing: when, e.g. a couple of developers (C) are testing their components locally connected to the platform, developers A and B located at different sites can participate transparently of their location at the integration and testing session via the webportal. As the remote operation of hardware components like robot manipulators or mobile platform is commonly very critical, access control can be assured by the Trac authentication module, which grants access to components relevant to security only to authorized persons.

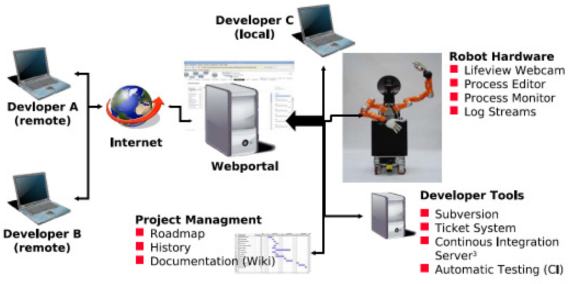

Figure 8 Typical distributed development set-up in a service project with centralized hardware [2].

5.2 System Overview

The ultimate goal of remote access is to provide the remote accessing person the same information about the current state of the platform as the local developers have. Therefore, a plug-in was developed that permits centrally managed configurations, control and monitoring of software component processes. The developer is able to perceive at one glance, which components are online and their current states via web log streams. Graphical applications such as simulations or visualisations are supplied through a graphical desktop sharing system (e.g. tightvnc [10]). For the video surveillance two network cameras can be installed in the test area where the robot hardware is placed. The video recordings are also available through the web interface.

5.3 Process Control

In addition to monitoring of the current running components, developers are able to start and stop components remotely.

Note that this is possible without knowledge of the physical location on the computing hardware. New components can be integrated easily into the web interface: the administrator of a component merely needs to provide the following configuration information once:

- deployment of the component (host address, operating system, authentication information)
- scripts to start and stop the component
- log file name

As component control is residing on the process level of the operation system, it is abstract from the specific implementation of the components and thus very flexible. Currently, Linux and Windows operating systems are supported. Once this information is given, the component can be operated without knowledge of the insides and the deployment.

6 Conclusion

In this paper the combination of using one common open-source repository, providing simulation models and tools for remote hardware access and distributed development was proposed to make high-end hardware platforms available. As example, the service robot platform Care-O-bot® 3 was described along with the concrete implementations of the presented concept, i.e. open-source repository of hardware drivers, ROS Gazebo models of hardware components and the webportal.

Future work concentrates the integration and tests of already available higher-level open-source components for the Care-O-bot® hardware. Components that are compatible to the hardware are planned to be included and maintained in a second repository or made available in a Care-O-bot® specific compilation, respectively.

The ultimate goal is to provide application developers the tools and software components to create complex and robust applications in a fast and flexible manner. These activities will be embedded into the Care-O-bot research initiative [12].

7 Acknowledgement

This work was conducted in the department Robot Systems at the Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Stuttgart, Germany. It was funded as part of the EU Project BRICS (Best Practice in Robotics – www.best-of-robotics.org) under grant no. 231940.

8 Literature

[1] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hägele, A. Verl: Care-O-bot 3 – Creating a product vision for service robot applications by integrating design and technology. In: The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1992-1997, October 11-15, 2009 St. Louis, USA

[2] Deutsche Service-Robotik Initiative (DESIRE), [Online]. www.service-robotik-initiative.de, 2009

[3] U. Reiser, R. Klauser, C. Parlitz and A. Verl: DESIRE WEB 2.0 – Integration Management and Distributed Software Development for Service Robots. In: Proceedings of the 14th International Conference on Advance Robotics, June 22-26, Munich, Germany

[4] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, R. Dillmann: ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In: IEEE-RAS International Conference on Humanoid Robots, 2006, Genova, Italy

[5] H. Bruyninckx: Open robot control software: the OROCOS project. In: IEEE International Conference on Robotics and Automation, vol.3, 2001, pp. 2523-2528

[6] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng: ROS: an open source Robot Operating System. In: ICRA International Conference on Robotics and Automation, May 12-17, 2009 Kobe, Japan

[7] Player/Stage/Gazebo: Free Software tools for robot and sensor applications, http://playerstage.sourceforge.net/

[8] C. Parlitz, M. Hägele, P. Klein, J. Seifert, K. Dautenhahn: Care-O-bot 3 – Rationale for human-robot interaction design. In: International Federation of Robotics u.a.: ISR 2008: 39th International Symposium on Robotics, 15.-17. Oct. 2008, Seoul, Korea, pp. 275-280

[9] Edgewall Software, The trac integrated scm and project management, visited March 2010. [Online]. Available: http://trac.edgewall.org

[10] TightVNC Software. [Online]. Available: http://www.tightvnc.com/

[11] Friend, http://www.amarob.de

[12] Care-O-bot Research, http://www.care-o-bot-research.org

[13] Asimo, http://asimo.honda.com