BRICS – Best practice in robotics

Rainer Bischoff[1], Tim Guhl[1], Erwin Prassler[2], Walter Nowak[2], Gerhard Kraetzschmar[3], Herman Bruyninckx[4], Peter Soetens[4], Martin Haegele[5], Andreas Pott[5], Peter Breedveld[6], Jan Broenink[6], Davide Brugali[7], Nicola Tomatis[8]

[1] KUKA Roboter GmbH, Augsburg, Germany
[2] GPS Gesellschaft für Produktionssysteme GmbH, Stuttgart, Germany
[3] Bonn-Rhein-Sieg University of Applied Sciences, Bonn, Germany
[4] Katholieke Universiteit Leuven, Leuven, Belgium
[5] Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., Stuttgart, Germany
[6] University of Twente, Twente, Netherlands
[7] Università degli Studi di Bergamo, Bergamo, Italy
[8] BlueBotics SA, Lausanne, Switzerland

Summary / Abstract

In the past, the process of developing a new robot application has had more of the design of a piece of artwork or of an act of ingenious engineering than of a structured and formalized process. The prime objective of BRICS is to structure and formalize the robot development process itself and to provide tools, models, and functional libraries, which allow reducing the development time by a magnitude. BRICS is working together with academic as well as industrial providers of robotics “components” (hardware and software), to identify and document best practices in the development of complex robotics systems, to refactor (together) the existing components in order to achieve a much higher level of reusability and robustness, and to support the robot development process with a structured tool chain and code repository. BRICS is a joint research project funded by the European Commission ICT Challenge 2 under grant number 231940. First results include the analysis of existing robot development processes, the first steps towards harmonizing robot control interfaces and component models and the set-up of robot systems for best practice analyses.

1 Introduction

1.1 The robot development process

In computer science the process of developing large software systems is well understood. In March 2009, a search in Scholar Google for the phrase “software development process” resulted in approx. 49.000 hits. Some of the links point to textbooks, which have several thousand citations. At the same time a search for “robot development process” returned a total of 26 hits. This number is an indication that a formal robot development process has to date not been well established as a structured formal process.

Earlier attempts in supporting the “robot development process” did not really focus on improving the process, but on the development and provision of functional libraries, control architectures, software integration frameworks, and communication infrastructure and middleware.

Important aspects were more or less neglected, such as:

- formal models for the various software and hardware components,
- reuse of designs, libraries, and architectural elements between different robot applications,
- methods and rules for the transformation of models between levels of abstraction,
- tools and tool chains which support the verification, transformation and aggregation of models and their compilation into operational components and systems,
- the large value of cross-sectional aspects, such as instrumentation for benchmarking and monitoring, error management and robustness, harmonized interfaces, deployment support, or runtime reconfiguration of services.

1.2 Project objectives

The prime objective of BRICS is to structure and formalize the robot development process itself and to provide tools, models, and functional libraries, which help accelerating this process significantly. More specifically the objectives are:

- to design and implement an integrated development environment for robotics (called “BRIDE”) and an accompanying software repository of best practice robotics algorithms (called “BROCRE”),
- to significantly promote the interoperability of hardware and software components by harmonizing the interfaces and as well as the communication and data exchange between these components,
- to promote and moderate a community-wide discussion on the identification, evaluation and refactoring of best practice in robotics algorithms and software systems through research camps, which will be organized and

hosted by BRICS and to which senior researchers and Ph.D. students from around the world will be invited,
- to survey and analyze completed robot application developments with respect to their deficiencies in terms of tool support, reuse of components, use of harmonized interfaces, and use of functional and model libraries,
- to define and implement showcases, which allow to study and measure the acceleration of the robot development process established by BRICS.

1.3 Related projects and activities

The following projects and activities are related to BRICS:

- *DESIRE* was a cooperative research project funded by the Federal Ministry of Education and Research in Germany [1]. The primary objective of DESIRE was to maintain and build on the leading role held by German researchers and industry in the field of service robotics. The project partners involved in DESIRE could learn valuable lessons as to design robot systems in a structured way.
- *RoSta* (Robot Standards and Reference Architectures) is the main international contact point for robot standards and reference architectures in service robotics. RoSta was a Coordination Action funded within the FP6 of the European Commission [2, 3].
- *ROS* is a new open source robotics framework developed in the United States by Willow Garage. Its objective is to provide the software, robots and tools to create service robotics applications. Several BRICS partners are evaluating and using ROS to be part of their robotics systems [4].
- *OPRoS* (Open Platform for Robotic Service) is a Korean initiative aiming at establishing a component based standard software platform for robots which supports reusability and compatibility in a heterogeneous communication network [5].
- *OpenRTM* is a Japanese initiative to implement the OMG's Robotics Technology Component standard. The BRICS project has invited core OpenRTM developers to give feedback regarding the component model and the toolchain [6].
- *Orocos* was started in 2001 and funded under European Fifth Framework Programme to give shape to an open source framework for robotics, looking both at service robotics and real-time robotics. BRICS project partners are using and evaluating Orocos to be part of their robotics systems [7].

2 BRICS approach

To achieve its objectives, BRICS will cooperate with all interested stakeholders, in an open and constructively critical atmosphere, to develop a design methodology focusing on four fundamental issues:

- provision of hardware components with harmonized interfaces and open-source application programming interfaces (APIs);
- identification of best practice in robot algorithms, software components, and architectures;
- design of an integrated robot development environment that supports rapid and flexible configuration of new robot platforms and the development of sophisticated robot applications;
- cross-sectional activities addressing robust autonomy, openness and flexibility, and harmonization and benchmarking.

2.1 Development of two software packages

BRICS develops and supports two complementary software packages: *BRIDE (BRics Integrated Development Environment)* and *BROCRE (BRICS Open Code Repository; pronounced "broker").*

BRIDE will be based on the widely supported Eclipse platform. Eclipse has already been accepted by many other domains to serve as the integrating vehicle to make the development of complex systems easier and to ensure higher quality. The software engineering methodology behind this is Model-Driven Engineering, which provides developers with best practice models that (i) let developers design with higher level, abstract components, (ii) hide the details of implementations and middleware, and (iii) support automatic verification and code generation, when appropriate. BRIDE could profit from Eclipse-based developments in the domains of embedded systems and the automotive industry, if robotics-specific extensions are added.

BROCRE offers interoperable interfaces and source code components, that implement a lot of robotics functionality, especially for the APIs defined in the project. BRICS will provide an initial repository which will be usable and credible, but which will not cover all possible robotics components. This initial repository is expected to convince other stakeholders of the advantages of using the BRICS methodology and software tools. Their involvement will then result in a gradual extension of the framework.

2.2 Targeted user groups

The following user groups may benefit from the integrated robot software development environment, the interoperable robotics components, and the research platforms provided by BRICS:

- *Robotics researchers* can make use of hardware, which comes with a consistent set of harmonized, documented, public APIs and an integrated software development environment. This enables researchers to develop complex operational robotic systems with minimal effort, to invest their resources in research rather than in the development of research infrastructure, and to avoid so-called "from scratch developments".
- *Hardware manufacturers* can use the harmonized interface descriptions to make their hardware usable in the software development environment without losing access to advanced vendor-specific functionalities.
- *System integrators* – companies as well as research institutes – can utilize the development environment to significantly accelerate the process of integrating com-

ponents into robotic systems and to speed up the development of software for advanced robotic applications.
- *Software developers* have access to a consistent set of harmonized and openly available APIs for which they can offer (commercial or open source) implementations that are directly usable via the integrated development environment.

2.3 Reach out

In the work plan of BRICS we have foreseen so-called research camps to reach out to the robotics developer community. These research camps will be funded by the European Commission and will be accessible to highly qualified Ph.D. students and senior researchers. The character of the research camps will be somewhere between summer schools and the renowned ACM programming contests. Research camps are meetings lasting between one and two weeks, where people gather to jointly do research and develop software. Other than a regular workshop, where people meet to discuss and exchange information, research camps have tangible results, e.g., a software library (contribution to BROCRE) for a specific robotics task.

3 BRICS activities

BRICS is performing Research and Technology Development (RTD) work in the following areas:
- hardware building blocks (see section 3.1)
- architectures, middleware, and interfaces (section 3.2)
- best practice in robot algorithms (section 3.3)
- model-driven engineering and tool chain (section 3.4)
- openness and flexibility (section 3.5)
- robust autonomy (section 3.6)
- harmonized interfaces (section 3.7)

The results from these activities are verified in showcases with exemplary demonstrations (see section 4). For each activity clear resposibilities have been defined (Figure 1).

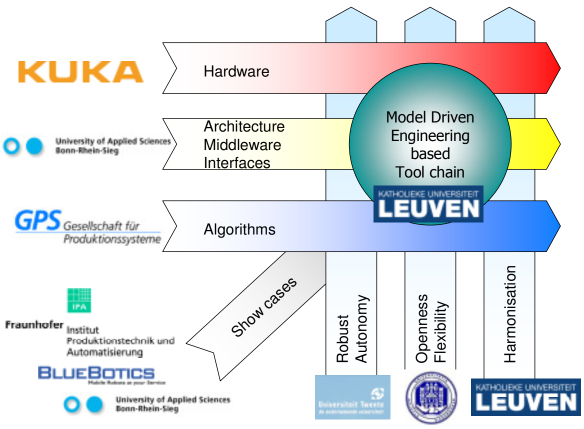

Figure 1: BRICS activities, partners and responsibilities.

3.1 Hardware building blocks

Based on an analysis of the hardware requirements in robotics research and industry the BRICS partners will select Robotics RTD Platforms (RRPs) (Fig. 2) and components:
- for showcases to study, measure, and demonstrate the acceleration of the robot development process;
- to provide interfaces for external collaborators.

This effort focuses on the promotion of the interoperability of hardware and software components by harmonizing the interfaces and as well as the communication and data exchange between these components.

Figure 2: Overview of some of the possible Robotics RTD Platforms (RRPs) to be used in the project (from left to right): KUKA omniRob, mobile base of BlueBotics, IPA Care-O-Bot, BlueBotics Gilberto.

The following scientific and technical key issues will be addressed in this activity:
- concepts for common Robotics RTD Platforms (RRP), which provide open access on different levels of abstraction;
- communication of these concepts to European robotics stakeholders in order to analyze the requirements and possible access models for such research platforms;
- specification and adaptation of existing robotics hardware according to the agreed upon concepts to enable robust and autonomous use of it and to enable integration with other robotics components;
- models of the hardware for easy configuration, simulation, and deployment of this hardware in the context of the BRICS Integrated Development Environment;
- a comprehensive action plan to implement common Robotics RTD Platforms in a sustainable way as an instrument for the European robotics community to more easily conduct and benchmark research and perform technology transfers.

3.2 Architectures, middleware, and interfaces

The objective of this activity is to identify complete robot control architectures or architectural subsystems and patterns, which are attractive for re-use by researchers and application developers. This activity will review and evaluate technologies and methods used in industries dealing with systems similar to robots (automotive, aviation, embedded systems development) and identify those technologies and methods from which robotics would benefit strongly. Outreach to the community is essential for this activity in order to guarantee that we capture all relevant technologies and both the extent and depth of the problems involved. The following scientific and technical key issues will be addressed in this activity:

- Identification of best practice in architecture, middleware, and interfaces: Past research has produced many good ideas, concepts, and even implementations for subsystems or particular aspects of robotics software development. The problem is that most of this work is not reusable. A first challenge of this activity is to make a thorough assessment of previous work and of the state-of-the-art. A second challenge is to define a coherent set of evaluation criteria, with measures and procedures for assessing them.
- Development of a software platform for robotics: The first challenge is to deal with the heterogeneity of hardware devices. This heterogeneity can be dealt with by harmonizing the interfaces to devices. A second challenge is to manage the complexities of communication and distributed systems. This is the core application domain of middleware systems. Middleware for the robotics domain should provide special support for handling soft and hard real-time requirements, communication facilities not supported by current middleware (CAN-bus etc.), and for advanced failure detection and failure handling capabilities to achieve robust autonomy. The third challenge is to handle the heterogeneity of the algorithmic approaches used in complex robotic applications. This requires the harmonization of interfaces of different modules which provide functionality, such as mapping, SLAM, path planning, etc., as well as the harmonization of data structures exchanged between modules. The fourth challenge is to provide appropriate support for tool chain integration, on all levels of the software platform, and both for the development environment as well as the runtime environment.
- Development of functional/control architecture workbench for robotics: Designing and implementing suitable robot control architectures remains a largely open problem. This challenge can be met by developing a control architecture workbench for robotics, which should provide configurable blueprints for well-known robot control architectures, which have been refactored to exploit the robust, state-of-the-art software technologies. The workbench should support re-use of architectural patterns.

3.3 Best practice in robot algorithms

Harmonizing robotics algorithms, so that they are easily replaceable by another, is often not viewed as a fundamental scientific problem, but a software engineering task. However, there are also a number of fundamental issues to address. The BRICS consortium wants to develop a framework that can be applied to a large number of robotic tasks and algorithms, and not only to one or two. The challenge is to make such a framework *sustainable,* so that it accommodates not only existing algorithms, but also future developments, and it is of upmost importance to make this framework really appealing to the robotics community. The following scientific and technical key issues will be addressed in this activity:

- harmonization of operational context and conditions for algorithms;
- specification and harmonization of performance requirements;
- definition of (a language for) abstraction levels and generic descriptions of robot algorithms including description of I/O behavior, module dependencies, timing and hardware dependencies;
- harmonization of data structures and external (and internal) models referred to by an algorithm;
- harmonization of interfaces, and communication mechanisms;
- analysis of real-time requirements;
- definition of test conditions, procedures and benchmarks for comparative evaluation.

To reach the goals of this activity five major contributions are planned:

- The most important contribution is an approach for identification and re-factoring of best practice robot algorithms. This approach should be applicable to any topic in robotics for which competitive and comparable algorithms are developed.
- The approach will be applied to multiple application contexts, resulting in software libraries of interchangeable algorithms. During the runtime of the project a minimum of three libraries will be compiled, among them 'mobile manipulation', '3D perception and modeling', and 'robust obstacle avoidance'.
- The approach will involve a comparative evaluation of algorithms through benchmarks.
- A series of research camps will be organized to reach out to the research community and to get feedback on the developed methodology and framework and to assure acceptance.

3.4 Model driven engineering and tool chain

This activity will focus on:

- creating the Model-Driven Engineering models for robust autonomy in robotic systems;
- the integration of the BRICS best practice results in these domains into the Eclipse tool chain; and
- the integration of existing legacy Computer Aided Engineering (CAE) tools into a BRICS Integrated Development Environment.

The following scientific and technical key issues will be addressed:

- The BRICS Integrated Development Environment: It is planned to create a BRICS Integrated Development Environment, which will be based on Eclipse, remain fully compatible with the ongoing evolutions within the broader Eclipse ecosystem, but that will provide the robotics community with an Model-Driven Engineering (MDE) tool chain that contains the robotics domain specific models and tools to build the BRICS Robotics RTD Platforms, components and interfaces. This challenge contains scientific as well as technical issues and challenges.

- The Model-Driven Engineering MDE models of robotics: At the scientific level, the main challenge is the identification of what MDE models are appropriate abstractions of the complex, autonomous and robust multi-agent systems that are so typical for intelligent robot systems, and that are not well supported by the ongoing activities in the other mentioned domains (automotive, aerospace, aviation, mobile telephony). Technically speaking, the major challenges are to find appropriate granularities for the MDE models, and to package and document them in a way that makes them attractive for reuse and for co-development by all stakeholders in the robotics domain.
- Integration of legacy tools: Being able to integrate already existing models from popular CAE tools (such as Simulink, 20Sim, etc.) will be a key factor for acceptance in the robotics community.

3.5 Openness and flexibility

The IEEE Standard Glossary of Software Engineering Terminology [8] defines flexibility as the "ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed". The overall objective of this activity is to provide the conceptual and technological means that will allow other work packages to develop and re-engineer open and flexible robotic software artifacts, to define metrics to assess their degree of openness and flexibility, and to define design principles and implementation guidelines to improve them. The following scientific and technical issues will be addressed:

- How to predict the class of changes that are likely to occur over the lifespan of the software system? For this purpose, the following issues will be addressed:
  - to identify requirements of a robotic system are more likely to remain stable over time;
  - to identify de facto standards which are enforced in the robotic domain;
  - to analyze the kind of evolution that a robotic system undergoes.
- How to assess the software quality of a robotic system design? The main issue is to define appropriate quantitative metrics.
- Is it possible to identify recurrent design problems and define reusable design solutions (design patterns)? Design patterns let some aspects of system structure vary independently of other aspects, thereby making a system more robust to a particular kind of change.

These challenges are approached as follows:

- identifying Best Practice in system flexibility and metrics to evaluate implementations;
- assessing system flexibility of robotic software artifacts;
- defining design principles, implementation guidelines, and evaluation criteria for enhancing openness and flexibility of best practice robotic software artifacts.

3.6 Robust autonomy

Robust Autonomy (RA) is an ability of the system to react on both explicitly-specified adverse situations and unexpected adverse situations in both the environment and the underlying system (hardware and software) without or with a limited human guidance in order to complete its mission in the best possible (acceptable) way. The robustness of robot autonomy should be such that interactions with the environment based on reasoning remain sufficiently safe, accurate, etc. The only time when an autonomous robot can be provided with resources to achieve robust autonomy is at design time. It is essential to have structured robot development process as well as give robotic appliance all possible resources to ensure robust autonomy of the final design. Therefore BRICS concentrates on steps to be taken during the robot development process and on add-ons which will increase the robustness of the resulting autonomous system.

Robot development should be structured in a way that reveals decisions that affect non-functional requirements such as robust autonomy. BRICS will come up with an ontology for robust autonomy from which a user (designer) can derive which criteria should be taken into account in a specific context. This will help the user to make decisions about the particular demands of the context at hand and to limit the set of possible solutions.

Robust autonomy solutions could be viewed as features included in robot – resources that ensure operation. Utilizing or introducing some type of redundancy into the system could achieve this. As an example let us consider a model of the desired behavior. In the extreme case that the system is described as a "black box", so the model merely consist of its specifications, it is only possible to detect potential failure, but not to resolve it. The more relations are known between the specifications and the internal functions, the more options there are to adapt these functions when parts are failing, such that the specifications are met according to the required level of quality (quality of service), which in turn, can be used as a criterion for robustness.

To reach the goals of this activity the following contributions are planned:

- design principles, implementation guidelines, evaluation criteria, and use case implementations for robust autonomy;
- collection of methods for achieving robust autonomy.

3.7 Harmonized interfaces

This cross-sectional activity starts from the observations that (i) complex robot control systems are built from many components, coming from many different sources, and (ii) it is impossible (at least in the short to medium term) to impose one single standard for these components. BRICS will develop guide-lines, document them, and provide suggestions to the community about how to implement harmonized interfaces. The following scientific and technical key issues will be addressed:

- Multiple levels of APIs: best practice for APIs will be determined taking into account that this has to be done at several levels: (i) the semantic interface of an API has to be considered. A description of the functionality, the objects and the results have to be in a computer-readable form. A clear semantic description may also result fewer mistakes by the developers. (ii) The interfaces have to allow different technical approaches, for example algorithms that optimize different aspects (time, complexity, quality of service, ...) and the information about which solver is provided/needed is to be documented via the semantic interface. (iii) The interfaces have to allow for various computing hardware and programming languages.
- Appropriate decoupling: System developers will only want to reuse each other's components if these components are easily understood and if the functionality they offer is what the developer is looking for. Tradeoffs between functionality and size need to be analyzed.
- Integration in MDE ecosystems: Software systems in robotics need to be readied to be part of a large-scale MDE system such as Eclipse.

The activity will approach these challenges as follows:

- Harmonization of hardware platform interfaces: The goal is to provide interfaces to robotics hardware that are vendor independent, and that have appropriate levels of abstraction and granularity to serve different application needs in the robotics community.
- Harmonization of middleware interfaces: Here the right level of granularity and (de)coupling will be found to make robotic middleware more open.
- Harmonization of algorithm interfaces: The functionality of software systems is mostly encoded in algorithms. BRICS will work together with the community to provide this functionality in a form that is harmonized and that is ready to be integrated in IDEs such as Eclipse.
- Integration of harmonization guidelines in Tool chain: BRICS hopes to contribute to the Eclipse ecosystem, via code and documentation contributions as well as through presentations at relevant MDE conferences to ensure reusability of code is supported to an even grater extend.

4 Showcases

The activity Showcases for Industry, SME, Research, and Education will verify the results from the other activities (as described in section 3) in showcases with exemplary demonstrations. These showcases shall act as vehicles for assessing the suitability and performance of the BRICS hardware interoperability concepts, the BRICS integrated robotic development environment, and methods in typical design processes of advanced specification, design, development and optimization given the needs and interests of key user-groups (industry, research, education).

The goal of this activity is further to introduce and implement a family of BRICS Robotics RTD Platforms (RRPs, including the developed repositories, frameworks, tools and methods) as proof of concepts, cyclic evaluations and entry point for a wider dissemination and adoption of the RRPs into the robotics community beyond the project runtime. The scope of RRP configurations includes, but is not limited to, a set of mobile platforms with one or two handed arms, sensor configurations, and multi-modal interfaces. Possible application domains, which are addressed by the RRPs include robotics in manufacturing, particularly logistics, service or personal robotics, and rehabilitation robotics.

Acceptance of the RRP family depends mainly on its performance, accessibility and cost. An RRP’s functionality has to be specified for the showcases and an RRP should meet the requirements:

- safe operation, given everyday scenarios and disturbance factors (lighting, clutteredness etc.);
- modular and scalable architecture/middleware for a wide number of applications and experiments;
- documented interfaces, component performance (“data sheets”) with responsive technical support;
- use of state-of-the-art mechatronic components including validated simulation models;
- compatibility with state-of-the-art operation systems, middleware(s), computer languages.

4.1 Showcase industry

The family of RRP itself forms the technological basis of future robotic products for emerging markets:

- mobile one or two-armed manipulators for tasks in manufacturing or laboratory automation, particularly in material flow automation and logistics;
- personal or domestic robots as helpers in everyday environments (office, home, public spaces).

The goal of the showcase industry is the implementation of an RRP into a trial system development processes. Of particular interest in this context is the application of the BRICS principles by today’s typical system integrators. For addressing specific needs and engineering cultures of both larger robotics suppliers and spin-off type small and medium robotics industries two approaches are planned:

- A prototype system of a mobile manipulation application will be re-engineered on the basis of the BRICS results. Assessed differences regarding both development processes regarding typical criteria such as reached performance, cost and time advantage, serviceability etc. will be used as a measure for the BRICS RRP optimization.
- In regular assessment workshops teams of robotics engineers apply the BRICS results in defined robotics development tasks. Systematic evaluation results of the BRICS-based hardware, software, tools and engineering process will be used for directing and optimizing the BRICS research and development.

4.2 Showcase SME

The objective of this showcase is to show the efficiency gain and the reduction of development time in the context of a robot application development for an end customer by using the BRICS integrated development environment.

The development of robot applications for end customers is a typical activity of SME system integrators and bears significant potential for innovation. The SMEs involved in BRICS will identify a typical application development activity for one of their (past, current, or future) customers, which has been implemented or will be implemented. Then the SMEs will do a complete redesign of the identified application development using the integrated development environments of the project and the harmonized hardware and software interfaces. They will record the development process, with all advantages and disadvantages and problems and advances, as neutral and detailed as possible.

4.3 Showcase research

The goal of the showcase "research" is the specification, provision and implementation of an RRP and its cyclic assessment with expert members in robotics research:

- to provide access for the European researcher to RRPs in the context of different access-models,
- to implement the MDE tool chain and simulation infrastructure and make it accessible as a complementary component to European research groups, and
- to set-up of a cooperative systems engineering projects between research teams and industry for developing a specified industrially relevant robot assistant (see showcase industry).

4.4 Showcase education

Strengthening the supply of highly qualified robotics researchers and engineers is essential for European robotics research and industry. Thus, this showcase aims at providing access to state-of-the-art robotics research platforms for higher education. An RRP is configured according to the needs of an application scenario like mobile manipulation. Costs will be kept low by using modular and configurable robots, renting robots, or part-time accessing. A repository of teaching material including tutorials, slide sets, and problem sets to be used by instructors will be supplied. With the tools and technology provided by the education showcase, teachers can acquaint their students in the classroom with state-of-the-art technology. Furthermore, joint curricula can be developed between educational robotics programs of multiple universities in different countries.

5 First results

The BRICS project started March 1st, 2009. Initial results include an in-depth analysis of existing robot development processes, a first step towards harmonizing robot control interfaces and component models and the set-up of robot system for best practice analyses.

5.1 Best practice in mobile manipulation

A first prototype of a general methodology for identifying and providing best practice algorithms has been developed in the project. This methodology has been applied to the domain of mobile manipulation, being a mature, but still highly active research area [9]. The first step consisted in exploring the state of the art in the domain itself, including a thorough literature survey as well as the collection and evaluation of existing software libraries and implementations. Afterwards existing algorithms and libraries have been analyzed concerning common structures, such as samplers, collision checkers, etc. Those sub-modules have been identified and their interfaces harmonized in order to allow easy exchangeability. Based on the object-oriented library CoPP (Components for Path Planning) by Morten Strandberg we refactored several probabilistic path planning algorithms and integrated them into one library. This refactoring took place under a software engineering point of view, with the aim of providing a reusable and flexible component-based framework that is supposed to become publicly available in spring 2010.

A proof of concept evaluation of the library has been already undertaken by interfacing it to a newly developed mobile manipulator, allowing the control of its base and arm in a convenient way. In addition first sets of benchmarks could be initiated, comparing mobile manipulation algorithms in varying scenarios. Ongoing work is to further enhance usability and the compatibility with existing frameworks, thus opening up to the robotics community.

One major event in this respect will be the first BRICS research camp with the topic "Mobile Manipulation". It will take place in Malaga, Spain, 24-29 October 2010. 20 robotics Ph.D. students worldwide with a solid background in mobile manipulation are invited to gather for one week in a stimulating environment to explore, revise and improve what are considered the best practice algorithms in mobile manipulation today. The intended outcome of the research camp is an open source library of refactored and harmonized mobile manipulation algorithms. Travel grants, hardware and initial set of software modules for mobile manipulation and 3D perception and modeling will be provided. It is expected from the students that a competitive solution to two given tasks either using the provided or self-developed algorithms for mobile manipulation is created and demonstrated in two competitions on the last day of the research camp.

5.2 BRICS component model

In order to harmonize algorithms and integrate them easily in robotic applications, a common software representation of algorithms is required. The BRICS project is aiming to formalize the interface of algorithms and hardware such that this integration can take place. The BRICS component model defines this exact interface and allows algorithms defined in it, to be deployed to existing robotics software frameworks. BRICS is building a tool chain that integrates and coordinates the necessary steps to generate the code from user specifications. As such, the BRICS component model only serves as a common language between these tools. An Eclipse plug-in is under development that allows a graphical specification of a component or a system of

components. This Eclipse extension is tailored specifically towards robotics and aims to support robot application developers with the integration of their algorithms or hardware into a real robotics system.
The advantages of specifying applications as BRICS components are: allowing to make a late decision on the target platform or robotics framework; having a formal interface description of a component which allows model validation; providing an always up-to-date graphical representation of a component or system of components.
The BRICS component model is defined and defended by a series of use cases that express a commonly experienced action when building a robotic system, e.g., processing sensor data, online parameter tuning or distributed communicating processes. Most modern robotic frameworks support such features, making code generation towards such frameworks straightforward. The present version of the tool chain allows to embed algorithms in components and to specify data flows between these components in a single model, which is used to generate code for Orocos, ROS, and OpenRTM.

5.3 KUKA Fast Research Interface

One goal of the project is to establish the KUKA Lightweight Robot (LWR) as a reference platform for research, in order to make results more comparable and to transfer them to industry. A further goal is the creation of a small-scale controller to make it easier for users and system integrators to integrate the LWR as a component in their own system, e.g., as manipulator for a mobile platform.
As a first step toward reaching this goal, a software interface was created (building on work from the PHRIENDS project [10]) to enable the robot behavior to be very precisely monitored (e.g., position and torque measurement data). Additionally, the robot can be remote-controlled at different levels of autonomy (ranging from discrete events in KRL to quasi-continuous motion at millisecond intervals). The existing control processes of the robot can be utilized for this purpose; the parameters (e.g., stiffness and damping) can be set via the interface (Figure 3).
In a second step, this interface was completely integrated into the KUKA LWR controller, and is now available to research partners as the “Fast Research Interface” (FRI) [11]. Based on a simple UDP protocol, the interface allows the user to control the robot and monitor its status from an

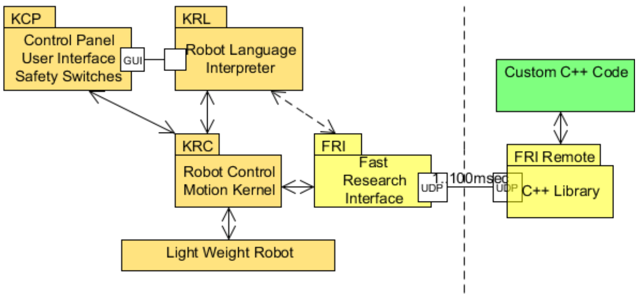

Figure 3: Overview of the Fast Research Interface (FRI) control system architecture (for details see [11]).

external PC. When establishing a connection, the sampling rate of the interface can be freely selected between 1 and 100 ms. Motion commands which are issued more slowly than at millisecond intervals are preprocessed and interpolated by the KUKA controller. As Ethernet UDP is used as the connection technology, the interface can be ported to a wide variety of operating systems and computers.

6 Conclusions

Harmonisation of academic research and technology development is badly needed to create new products, to boost new applications, and to accelerate innovation cycles. The EC funded project BRICS will enable robot manufacturers to collaborate more closely and effectively with academia and to benefit from top European research. Furthermore, system integrators will be able to utilize the developments to significantly accelerate the process of integrating components into robotic systems and to speed up the application development. The EC funded parts of the developments through FP7 – BRICS (ICT-231940).

7 Literature

[1] DESIRE Consortium: URL: www.service-robotik-initiative.de [last accessed on 22.03.2010].

[2] RoSta Consortium: URL: www.robot-standards.eu [last accessed on 22.03.2010]

[3] E. Prassler, H. Bruyninckx, K. Nilsson, A. Shakhimardanov (2010): The Use of Reuse for Designing and Manufacturing Robots. White Paper, Version 0.9, 19 June 2009. URL: www.robot-standards.eu [last accessed on 22.03.2010]

[4] ROS – Robot operating system, URL: www.ros.org [last accessed 22.03.2010]

[5] B. Song, S. Jung, C. Jang, S. Kim (2008): An Introduction to Robot Component Model for OPRoS (Open Platform for Robotic Services). Proceedings of the Intern. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2008), Venice, Italy, 3-4 Nov. 2008, pp. 592-603.

[6] OpenRTM-aist Official Web Site. URL: www.openrtm.org [last accessed on 23.03.2010]

[7] The Orocos Project – Smarter control in R&A. URL: www.orocos.org [last accessed on 22.03.2010].

[8] IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990

[9] E. Milke, S. Christen, W. Nowak, and E. Prassler. Towards harmonization and refactoring of mobile manipulation algorithms. Proc. Intern. Conf. on Advanced Robotics (ICAR 2009), Munich, Germany.

[10] PHRIENDS Consortium: URL: http://www.phriends.eu [last accessed on 23.11.2009].

[11] Schreiber, G.; Stemmer, A.; Bischoff, R. (2010): The Fast Research Interface for the KUKA Lightweight Robot. IEEE ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications. Anchorage, May 2010.