The DESIRE Service Robotics Initiative

Paul G. Plöger, Kai Pervölz, Christoph Mies, Patrick Eyerich, Michael Brenner, Bernhard Nebel

We present some advanced hardware units and an appropriate component based SW architecture for DESIRE. As an example we describe the integration of a enhanced AI task planner which allows for higher flexibility and dependability during complex task execution.

1 Introduction

In 2006 a consortium of twelve German partners -including some leading providers of industrial robotic technology components (RTCs)- teamed up with some research institutes and universities to work on a joint publicly funded four year project called DESIRE. It is focused on service robots, which shall offer their services under unconstrained conditions in immediate proximity of users in everyday situations. The dedicated goal was to address the extraordinary high demands on functional competence ("capability"), reliability ("dependability") and an acceptable price ("affordability").

In this article we focus on the open issues of defining a suitable software architectural framework on one hand in combination to the use of an advanced planer to cope with the high complexity demands of service robotics on the other. We report on the current status of the DESIRE work, especially the architecture and the planer component.

For the architecture side it is a widely accepted fact that in the large a major revision of the methods and technologies currently used for the implementation of robotic software is called for. Like in many other areas the construction of robot control software should focus more on the deployment of common off-the-shelf software and components in combination with a conceptually clean integration of them. This topic has drawn major attention of a large number of researchers [1], conferences session and workshops and also some major organizations initiated and funded respective activities [7]. The issues are much under debate and far from being settled.

The integration of a task planner into an autonomous robot increases the robot's level of intelligence and flexibility by altering the way the robot is controlled, moving away from predefined sequences of detailed user instructions to a more sophisticated target oriented approach. It is not longer required to provide the robot with a fully worked out description of its task (e.g., "Go to the big table, take the plate, come back to me an give the plate to me!") but rather to state some declarative targets (e.g., "Give me the plate!") and leave it to the robot to find a suitable way to achieve them on its own. Accordingly, planning can be understood as reasoning on a human level of abstraction.

Task planning itself is a thoroughly investigated subfield in artificial intelligence [3]. However, in a robotics context, one has to deal with aspects complicating the application of task planning, some of which are: Imperfect knowledge of the surroundings, non deterministic changes, and user interaction. One of the main goals of this project, as far as the planning part is concerned, is to make task planning more suitable for everyday use in a robotics context.

The remainder of this article is structured as follows. The section 2, which gives some general aspects of the overall project, is followed by two sections dealing with deduction of architectural requirements from the given hardware components and how we define the architecture of so called *Autonomous Robotic Components (ARC)* to address the special needs induced by this approach. This is followed by section 4 which describes the use of an advanced planner which is aggressively used and thus builds an important improvement in the technology demonstrator. We close with some remarks on lessons learned in the penultimate section 5.

2 DESIRE Overview

DESIRE is an acronym for "Deutsche Servicerobotik Initiative". As such it is a collaborative research project funded by the German Federal Ministry of Education and Research (BMBF). The main objective is to deliver those key functions and components in hardware and software which will achieve a technological leap towards the use of service robots in everyday scenarios. Project activities have been grouped into three action-lines namely the creation of a reference architecture for mobile manipulation, the promotion the convergence of technologies through integration into one common technology platform and finally pre-competition research for new products to enable technology transfer into start-up enterprises in the field of service robotics.

The consortium is composed of 14 partners, four of which are research institutes, three universities and three larger and four small and medium sized enterprises. Some partners take a lead in respective work packages, see Table 1, where ALU abbreviates "Albert Ludwigs University" and FHG "Fraunhofer e.V.". For all detailed names and contact informations please refer to the project website [4]. Since the overall project scope is very wide and the number of partners is large we can only give a more detailed account on the architectural and the task planning aspects in sections 3 and 4. Other areas of research are briefly sketched in the following three subsections.

2.1 Technology Platform

The overall hardware system is depicted in Fig. 1, the main software components are described in Table 1. The most outstanding characteristics of the robot are:

- two 7 degree of freedom light weight arms (LBRs) prototyped by DLR and now being produced by KUKA. LBRs have a payload to kerb ratio of 1.
- two anthropomorphic hands from Schunk (SAH) with four self-sustaining fingers. Compared to human hand the SAH is over-sized by a factor 1.5.

Both components constitute landmarks of RTCs currently produced in Germany. Underneath the redundant drive unit has four omni-directional wheels and houses the necessary control cabinets.The dimension of the undercarriage is mainly determined by size requirements for electric control cabinets and static stability for the LBR arms.

Perception is provided by a combination of a high resolution color stereo camera and a common 3D depth in field camera both mounted on a pan tilt unit. Additionally a fixed standard laser scanner in front of the robot is used for mapping, obstacle avoidance and an advanced algorithm for people tracking.

2.2 Mechatronical Key Components

One of the most obvious skills of a service robot is to grasp objects robustly in cluttered and/or crowded environments. To this end two prerequisite have to be met, firstly the robot has to be endowed with suitable hands and versatile arms. Secondly the perception, especially object recognition and localization has be robust. Such components have not been available on the market till now.

To control a grasp robustly it is fundamental to apply compliant stiffness control to each finger. Controllers which optimize grasp forces are state of the art. The partners KUKA Roboter(lead), Schunk, DLR, FZI Karlsruhe and RUB Bochum tackle these problems in two work packages. The solution is based on the LBR arm and the Schunk hand and couples both by means of a newly developed tool changing system. Because of an adjustable, passive stiffness control this solution will react robustly to collisions with the environment. In Figure 1 you can see the SAHs mounted to the LBR arms. Extended neural control paradigms have been studied on this HW to solve the problem of robust grasps in cluttered environments, see [6].

2.3 Perception

Beside the basic tasks of localization and navigation the mobile service robot needs object recognition, obstacle avoidance and interaction with persons also when operating under uncooperative conditions like occlusions or unfavorable lighting. Before objects may be grasped they first have to be spotted and identified then exactly localized. This is called scene analysis. Finally detecting and recognizing people as well as reliably identifying gestures are key to being able to understand situations and intentions. This is called situation analysis.

In DESIRE both analysis are tackled from algorithmic side, all used sensors (3D camera, hires stereo camera and laser scanner) are standard. For the scene analysis essential progress in performance and robustness has to be expected from a combined, simultaneous utilization of different feature types. They originate naturally from above mentioned different cameras with their different modalities. To enable efficient processing and further combination of these sources, all are represented in a common and appropriate data framework. A probabilistic formalization has been developed which starts out from object features initially stored in a database used by the various recognizers. Methods to fuse perceived features, to optimally choose features as well as to reach optimized sensor positions during active perception are based on this probabilistic data representation. Finally the representation allows explicit and consistent treatment of uncertainties. Tracking of people over extended periods of time becomes possible. The architecture of the perception component is open to a variety of feature types and therefore supports multi-sensors and multi-cueing.

Figure 1: The DESIRE technology demonstrator with two LBR arms (7 DOF) including the four finger anthropomorphic hands (SAH).

RTC	lead partner	topics	OS
HMI	Univ. Bielefeld	Language understanding	Linux
Head	FHG IPA	3D modeling	Wind.
Planner	ALU	Ontologies	Linux
Sequencer	FHG IAIS	Architectures	Wind.
Platform	FHG IPA	Mechatronics	Linux
Perception	Siemens	Scene Analysis	Wind., Linux
Mobile manipulation	KUKA	Flexible Control	VxWorks, Linux
Eigenmodel	IAIS and ALU	Self Diagnosis	Linux

Table 1: Name of robotic technology components (RTCs), partners and research focus, for details see [4].

3 SW Architecture

In contrast to other projects and as it is illustrated in the previous chapter the overall DESIRE platform is composed from fairly advanced robotic technology components (RTCs).

For example the undercarriage drive unit natively offers a built-in point to point movement command with collision avoidance using the laser scanner in combination with a loadable environment map. An independent path planner is included and external clients may subscribe to the acquired laser scans. So the drive unit already comprises and solves a good cross-section of classical problems in mobile robotics namely from low level control, over guidance to navigation, trajectory generation, trajectory following and obstacle avoidance all by itself. Many architectures tackle these tasks on separate layers using dedicated functional units for each subtask.

All in all the RTCs from Table 1 clearly advocated for the realization of a SW architecture based on component-based software engineering principles. Our approach connects an hierarchical hybrid robot control with a component oriented software architecture. A high-level sequencer controls the execution of sensor-actor feedback loops encapsulated in so called *Autonomous Robotic Components*, s.b.

At the time being none of the available architecture candidates like Player/Stage, CARMEN, MCA, Orocos, GeNome from LAAS fitted all these requirements, see [7], link "Middleware" for an up to date comprehensive overview of all these approaches.

Autonomous Robotic Components

Our architecture defines an *Autonomous Robotic Components (ARC)* as a coarse granular unity which encapsulates algorithms and states. An *ARC* offers *services* via well-defined public interfaces to their basic operations and commands. For each service, preconditions and effects are specified to provide the semantics of the service. An *ARC* may, but does not necessarily need to be, hosted on one specific hardware RTC. As an example the manipulation *ARC* plans, controls and monitors grasp movements simultaneously. Usually an *ARC* knows very little about the whole system due to the intended loose coupling.

Each *ARC* can communicate with other components via a CORBA based middleware [8]. Two types of standard services have been defined: *command* and *operation*. The former one is an asynchronous call while the later is used for short task, where the caller is blocked until the callee finishes. Building up a framework is eased by an interface description language which is used to validate signatures of components already on model level. Stub code can be generated and the final connection to the middleware is accomplished automatically.

Wish Lists

ARC may return data of type *Wish List* because up to a certain degree an *ARC* is able to reason about the nature of failures of tasks in its own domain autonomously. Basically, a *Wish List* gives a hint how a failed goal can still be reached. For example this can be a recommendation to change the global state of the overall robot and move sideways to get occluded objects in back in sight.

The implementation of *Wish Lists* use a qualitative calculus to enable *ARCs* to communicate with one another. Reconsidering the example: if the *SceneAnalysis* has the task of exploring the table and only parts of it are visible from the current camera position a useful *Wish List* is to signal repositioning of camera to the sequencer. This in turn decides if the wish can be fulfilled. If so a repositioning of the sensor is initiated to allow a changed perception and thus to complete an update of the world model. To resolve qualitative to quantitative wishes, a qualitative calculus has been implemented.

There can occur several wishes from different components at the same time. The sequencer reasons about the fulfillment of the wishes. No component can rely on the fulfilling of its wish thus the control of the whole system still is executed by the sequencer. If *Wish Lists* are not contradicting, several wishes can be fulfilled in parallel. Basically, *Wish Lists* can occur nested (another *Wish List* can occur while the current wish is fulfilled), sequential and parallel as well as contradicting or not. Thus, the reasoning about the fulfilling of the *Wish Lists* must consider the current resource allocation in the system when scheduling execution.

4 Planning

To incorporate a planning system, a robot has to provide basically three kinds of information: A description of the abilities of the robot (the *actions* it is able to perform), knowledge about the *current world state*, and some adequate representation of the desired *goal state*.

It is assumed that the robot is not able to learn new behaviors on-line, so the set of possible actions is treated as static during runtime. The prevalent planning language, the *Planning Domain Definition Language PDDL* [2], is used as representation language for the definition of actions. Thereby, each action consists of a precondition, stating what has to be satisfied in a given world state in order to make the action applicable, and of effects, stating in which way the world changes after the application of the action. However, to enable the use of automatic ontology reasoning procedures such as Fact++ and convenient modeling tools such as Protégé, PDDL is not used directly for the purpose of stating domain details such as type hierarchies and certain properties of types. Instead, the Web Ontology Language *OWL* is used for that purpose and an automatic translation of the parts necessary for planning to PDDL is performed. Additionally, a method to integrate PDDL action descriptions and OWL ontologies which is able to reason about subsumption relations between actions has been developed.

The distributed architecture of the whole system is strongly reflected in the way the planner perceives the current world state: Information from different components has to be collected and integrated into one coherent abstract state. To achieve this, each component providing information necessary for the planner uses a proxy to transfer its relevant knowledge. The planner then collects all this information and generates an abstract world state out of it.

While a human usually uses imperative commands (e.g., "Take a plate!"), a planner needs a logical formula representing the goal state (e.g., $\exists x(plate(x) \wedge holding(x))$). The planner generates its goals out of the output of the natural language processing tool from the HMI component. Roughly speaking, the planner searches for some key words in the output which are bound to actions in the planning domain and generates the goal condition out of the effects of these actions. During the generation process, the planner detects determinate objects among the objects referred to in the instruction (e.g., within the instruction "Give me *the* plate!" the user rather refers to one distinguished plate than to an arbitrary one). Then it plans separately for all combinations of objects currently in the abstract world model which fit to the type constraints of the distinguished objects. In that way, the planner is able to discover some forms of ambiguity. If such an ambiguity is detected, the planner informs the sequencer, which in turn tries to dissolve the ambiguity, e.g., by checking if there is a user pointing gesture to some of the objects.

The base planning architecture is a slightly modified version of *FF-Module*, which in turn extends the well known FF [5] planner by metric fluents and external modules. Especially external modules provide great potential for planning in a robotic context, since they allow to source out time intensive computations to external instances where they can be handled in a much more efficient way.

5 Lessons Learned and Resume

In this article we outlined the research goals, approaches and solutions of the service robotic project DESIRE. The successful integration of the distributed RTCs into an high level architecture of *ARCs* has been achieved including necessary extensions like *Wish Lists*, and an *Eigenmodel* offering system-wide services like health and failure monitoring and alive management. On component level we reported about significant progress during user interaction due to recent advancements in AI planning. A thorough and comprehensive benchmarking of all components and their interplay is under way right now.

Project progress and debugging worked best when all partners collaborated in presence of the shared and unique platform, which took place in quaterly workshops. To sustain a high testing efficiency we introduced a remote testing strategy using VPN, see [8]. Another substantial productivity push would become possible by the use of a reasonably exact simulation model of the whole system upfront. This never-ending need for realistic modeling will be tackled next.

Acknowledgment This work was funded as part of the research project DESIRE by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01-IAIS and -ALU.

References

[1] Davide Brugali, editor. *Software Engineering for Experimental Robotics*, volume 30 of *Springer Tracts in Advanced Robotics*. Springer - Verlag, Berlin / Heidelberg, April 2007.

[2] Maria Fox and Derek Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *JAIR*, 20:61–124, 2003.

[3] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practise*. Morgan Kaufmann Publishers, San Fransisco, CA, 2004.

[4] GPS Gmbh. Desire web side. http:// www.service- robotik-initiative.de/.

[5] Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *JAIR*, 14:253–302, 2001.

[6] I. Iossifidis and G. Schöner. Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for a redundant robot arm. In *Proc. of the International Conference On Intelligent Robots and Systems (IROS)*, Beijing, China, 2006. IEEE.

[7] Erwin Prassler and Azamat Shakhimardanov. RoSta - Robot Standards and Reference Architectures. http://www.robot-standards.eu/.

[8] Ulrich Reiser, Christoph Mies, and Christian Plagemann. Verteilte Software-Entwicklung in der Robotik - ein Integrations- und Testframework. In *Proceedings of ROBOTIK2008*, 2008.

Contact

Prof. Dr. Paul-Gerhard Plöger
Fraunhofer Institute IAIS
Schloss Birlinghoven, D-53754 Sankt Augustin
E-mail: paul-gerhard.ploeger@iais.fraunhofer.de

Bild

Paul G. Plöger received his Ph.D. from the Technical University of Brandenburg. In 2004 he was appointed full Professor at Bonn-Rhein-Sieg University of Applied Sciences. His research focuses on control and perception algorithms in robotics especially neural network based. Currently he is involved as a work package coordinator in project DESIRE (BMBF) and XPERO (EU). **Kai Pervölz** received his Diploma in Mechatronics from the Aachen University of Applied Sciences in 2003. From the Bonn-Rhein-Sieg University of Applied Sciences he received in 2005 his Master of Science in Autonomous Systems. In 2005 he started his PhD at the Fraunhofer Institute IAIS with the focus on fault tolerance in mobile robot systems. **Christoph Mies** received his Diploma in Computer Science from the Aachen University in 2007. Afterwards he got a research position at the Fraunhofer Institute IAIS with the focus on multi-agent systems in simulation and robotics.

Bild

Bernhard Nebel received his PhD from the University of Saarland in 1989. Between 1982 and 1993 he worked on different AI projects at the University of Hamburg, the Technical University of Berlin, ISI/USC, IBM Germany, and the German Research Center for AI (DFKI). From 1993 to 1996 he held an Associate Professor position at the Computer Science Department of the University of Ulm. Since 1996 he is a full professor at University of Freiburg and head of the Laboratory for Foundations of Artificial Intelligence, which is one of the world's leading research groups in AI Planning. The group won several prizes and participated in many projects like CogViSys, CoSy (EU), DESIRE (BMBF) and also SFB TR/14 Automatic Verification and Analysis of Complex Systems (AVACS), and SFB TR/8 Spatial Cognition. **Michael Brenner** is a PhD candidate in the "Foundations of Artificial Intelligence" group working on techniques for planning in multiagent and robotics environments. He has worked on the award-winning planner IPP and was co-teamleader of the "ResQ Freiburg" RoboCup team (world champion 2004). **Patrick Eyerich** received a diploma in computer science in May 2007. Currently, he is a PhD candidate in the "Foundations of Artificial Intelligence" group. His research interests include AI Planning and Knowledge Representation.