User-driven software design for an elderly care service robot

Norman Hendrich, Hannes Bistry, Benjamin Adler, Jianwei Zhang
Informatics Dept, University of Hamburg
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
{hendrich, bistry, adler, zhang}@informatik.uni-hamburg.de

ABSTRACT
This paper describes a service- and scenario-driven software architecture for the ambient assisted living infrastructure currently under development in the Robot-Era project. Involving the end-users from the start, the project integrates an ambient sensor network with an advanced knowledge-representation and planning system and three different robots for outdoor, condominium, and indoor service roles. We explain the design decisions for the user-friendly and medium-cost service robot, with a focus on the integration of the ROS-based sensing and manipulation capabilities with precise indoor navigation and the PEIS middleware for ubiquitous robotics.

Keywords
Elderly care, assisted living, ambient sensor networks, service robotics

1. INTRODUCTION
Due to the increasing number of elderly people, societies in Europe and overseas are facing severe challenges, including an ageing workforce, a growing number of older people living alone and in need of care, but also a number of wealthy senior citizens ready to enjoy their third age. Technology that can support or augment this care at home is often referred to as assisted or independent living technology and is increasingly accepted as a key for the continuous support of our ageing population. Examples are communication aids, remote rehabilitation and telecare systems, vital sign monitors and reminder systems. Smart homes equipped with a multitude of sensors are an active research topic and might become reality soon, and the so-called ambient intelligence provides elderly users and their caregivers with a multitude of information from the collected data, detecting and tracking activities and monitoring health. Also, robotics technologies have significantly advanced, permitting the implementation of service robotic systems able to perform different service tasks also for "ageing well" applications, which are conceived to improve the independent living and quality of life of elderly people and to provide efficient care.

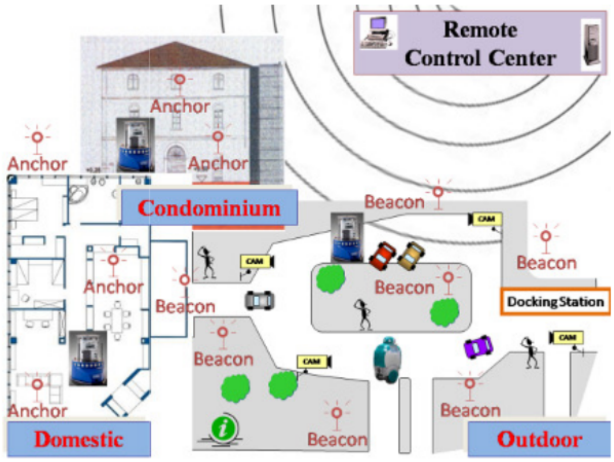

Figure 1: The Robot-Era system for assisted living: different robots for outdoor, indoor transportation, and manipulation tasks provide their services to the users. The apartment and areas are equipped with an ambient sensor network under control of an advanced planner and human control center.

Robotic services can be defined according to the level of autonomy of the end-users. Here, the autonomy not only depends on pathologies and possible motor/cognitive impairments with their obvious impact on the overall health condition, but also on environmental and contextual conditions, e.g. adverse weather. For our software development, three levels of user autonomy have been defined: users with a high level of autonomy do not have particular deficiencies and the robot system should try to keep them active in their daily activities. People with medium level of autonomy are characterized by slight motor or cognitive limitations, while a low level of autonomy applies to persons with bad health and motoric limitations who require regular help for their activities of daily life.

In this paper, we describe the user-driven software design approach and the resulting software architecture developed in the Robot-Era project. The paper is organized as follows: first, related work is summarized in section 2 while an overview of the Robot-Era project is presented in section 3.

PervasiveHealth 2014, May 20-23, Oldenburg, Germany

DOI 10.4108/icst.pervasivehealth.2014.254956

The user-driven software design approach and the resulting architecture is described in section 4. Early results from the recent first experimental phase of the project are also presented. The paper concludes with a summary and the list of references.

2. RELATED WORK

There are many research projects aiming to enable elderly people to live independently in their familiar home as long as possible using technological devices. Typical approaches are developing smart homes, wearable sensors and robot technology, or using a combination of those.

An overview of smart homes and their application in the field of elderly care is given in [4]. The devices in a smart home can be divided into two groups: Some can be classified as traditional home automation (e.g. air conditioning), while others are especially well-suited to monitoring and maintaining the wellbeing of the inhabitants. Typical sensor devices installed in a house to monitor the user include IR-sensors to track their activity [3], weight sensors [6], or usage sensors in typical household appliances to generate activity patterns [23]. In contrast to sensors installed in the house, wearable sensors can directly monitor the health of the user. Typical instances measure vital functions like heart rate and blood pressure [17] or generate activity patterns [12].

Mobile service robots can also be used to assist elderly users in their daily life. The Fraunhofer Institute of Manufacturing Engineering and Automation is working on robots to assist people in their homes. The Care-O-Bot 3 is the third version of their robot platform [21], featuring a sensor head, a manipulator including a gripper and a tray for safe handover operations. The platform is intended to perform fetch and carry tasks including subtasks like navigation, object detection and manipulation. The robot platform is evaluated in the national project WIMI-Care, where it is operated in an old-age home and serves water to the residents.

Mobiserv is another project developing mobile robots for elderly care. A detailed description of the architecture is given in [18]. The mobile robot intended for elderly care is controlled by a tablet PC. The system interacts with several sensors and devices in the smart home, e.g. a blood pressure device or cameras, using different network technologies (Bluetooth, LAN, WAN). CompanionAble is a similar project, also focusing on home assistance for elderly people, combining robot technology and an intelligent environment. A system description and recent progress can be found in [10]. The smart environment provides IR motion sensors to track the position of the users. As these sensors only provide rough information about their position, the mobile robot can be used to determine the position more precisely. One additional focus of this project is on the communication between the user and the robot system [22]. The authors of [13] present evaluation results of the autonomous platforms developed in CompanionAble and Mobiserv. These tests are evaluated under several aspects, including usability and user acceptance. As a result, the authors state that there are several challenges for future assistance-robots, especially in the field of intelligent and context-aware behavior. They further postulate that ambient intelligence in the smart home can contribute to this goal.

In [20] a robot platform used in the ALIAS-project is introduced. The robot provides mainly monitoring and communication services and is intended to improve the quality of life of the elderly users. Similar to Robot-Era, one important aspect of the developed system is the user-acceptance and the creation of an easy to use user-interface. Aim of the Florence-project [15] is to develop a general-purpose mobile robot platform for the well-being of elderly people and therefore targeting a similar approach like the ALIAS-project mentioned above. The main services provided by the Florence-platform are communication, gaming, home monitoring, calendar reminders and fall handling. The platform is built based on low cost components and is evaluated in tests with end-users.

The main difference between Robot-Era and the ALIAS- and Florence-projects is that Robot-Era also focuses on providing complex robot services (involving manipulation, sensing, interaction between different robots) and on the integration into a smart environment.

An architecture for intelligent environments based on OSGi [1] is proposed in [8]. The authors emphasize the advantages of having an infrastructure that is able to support the important standards for home automation, for example KNX. The OSGi middleware is also used in the system for assisted living described in [11]. The authors describe a concept to exchange data between a health institution and different assistance systems at home. Wireless sensor devices can be discovered and connected dynamically during run-time. The system was tested successfully in the three scenarios rehabilitation, service for hearing impaired, and ambient health monitoring.

3. THE ROBOT-ERA APPROACH

Despite significant progress towards ambient assisted living and individual success stories, turn-key systems are not yet available on the market. The aim of the Robot-Era project, started in 2012 and funded by the EC, is to provide elderly people with services conceived to extend their autonomy and independence by improving their quality of life and preserving their health. The main objective is to develop a fully integrated solution, demonstrating the general feasibiliy, effectiveness, social and legal plausibility and, most importantly, the acceptance of complete robotic services by end users. The project is interdisciplinary, bringing together researchers and people from industry, as well as usability experts and operators of elderly care homes.

Concentrating on the services provided by the overall system, common off-the-shelf modules are used for most of the robot and sensor hardware, with research focusing on the software development and integration. The overall system consists of the following components, including three robots with different roles and capabilities:

- the *pilot sites* with dedicated *ambient assisted apartments*, equipped with a variety of networked sensors, including medical monitoring sensors;
- the *domestic robot*, an indoor service robot with sensor head and robot arm that supports object manipulation in addition to carrying tasks;

Service name	devices and location	description
Communication	tablet-PC	Skype (video) call to caregiver or contact person
Reminding	planner	reminding the user, e.g. to take prescribed drugs
Surveillance	any robot, sensors	sensors or robot checks for and identifies people
Walking support	any robot	walking support, tele-operated or autonomous
Escort at night	any robot	walking support and guiding light
Laundry	2 robots	moving laundry to and from apartment
Food delivery	2 robots	bringing food to the apartment
Drug and shop	3 robots	selecting goods for shopping
Garbage collection	3 robots	indoor robot collects garbage and gives to outdoor robot
Object transportation	domestic robot	pick&place tasks performed by the domestic robot
Cleaning	domestic robot	autonomous cleaning performed by the domestic robot

Figure 2: Robot-Era Services for the first experimental phase

- the *condominium robot*, an indoor transportation robot, based on the same mobile platform as the domestic robot, but without the robot arm;
- the *outdoor* robot, a transportation robot;
- a *multimodal user interface* consisting of a tablet-PC with advanced speech recognition and synthesis;
- the *ambient network middleware* that manages the sensors. It combines a knowledge base and an advanced planner for controlling the robots;
- a *control center* for monitoring all sensors and robots, and to evaluate system performance during experiments and tests.

A sketch of the system is shown in figure 1. One key aspect is the availability of different specialized robots, optimized for their specific roles and services. Another is integrating classical in-door activities as well as services that span multiple rooms inside a building as well as outdoor services. For example, the *outdoor robot* is designed as a transport robot with high payload for shopping and carrying services; it operates over a large area ("town") and also provides walking support to users. In contrast, the *condominium robot* navigates the building, exchanging goods with the outdoor robot and bringing them to the individual apartments. The *domestic robot* is designed as a classic service robot with a robot arm to perform a variety of transport and manipulation tasks.

3.1 User-driven Development

From the start, the overall system and software design approach has been based on the actual needs and wishes of elderly people, determined using a variety of techniques ranging from interviews and questionnaires to observation of behavior and evaluation of literature and experiments in previous projects [14]. The project focuces on a specific group of elderly people, male and female over 65 years old, with moderate health problems and minor motor or cognitive deficiencies, living either alone or with their relatives (but without a dedicated caregiver). This choice was motivated by the interest in investigating the interaction between older people and technology and its integration in their daily lives, in order to support and not substitute the relationship between elderly and their family or caregivers. It should be noted that caregivers are also considered as end-users of the system, as they will also interact with its various technologies, influencing their acceptance.

To identify and refine the activities of daily living (ADLs), a study involving 100 elderly people (age 75-90) was performed by project partners in Italy, Sweden, and Germany at the beginning of the project. Usability, acceptance, as well as the expectations and system requirements were studied using interviews, meetings and focus groups with younger elderly people (age 65-75).

From these studies, a set of *robot services* was selected for the system and broken down into detailed *storyboards* that describe the actors (humans, robots, sensors, planners) and their interaction; see figure 8 for an example. The table in figure 2 lists the prototype services defined for the first experimental phase of the project. We expect to update the list and scope of the services based on feedback from the users as well as lessons learned from observations made during the long-term experiments.

3.2 User Interface

To hide the complexity of the overall system, the user-interface was kept intentionally simple and has been tailored towards the needs of elderly users. A common tablet-PC in combination with a webserver provides graphical access to all services (see figure 3), while a speech interface also allows the users to control the system. Both the graphical user-interface and the speech interface are fully localized, and are currently available in English, Italian, and Swedish. A service requested by the user can be canceled at any time, and progress information about task execution is shown to the user. The tablet PC can be kept within reach of the user or it can be mounted easily on the robot. Despite the simple layout of the main user-interface screens with large buttons, many users preferred to use the speech commands. For the case that the tablet PC is not in reach of the user or any malfunction occurs, the robot platform itself provides an emergency stop button that can be used to stop operation.

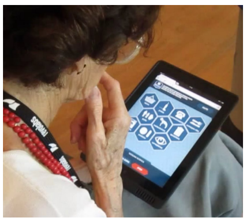

Figure 3: Tablet-PC based user interface

3.3 Ambient Sensor Network

Another key feature of the system is the tight integration between the ambient sensor network and the different robots controlled by the PEIS middleware. PEIS [2] is a decentralized and lightweight framework for the communication between different devices in a smart home (see also section 4). The project has access to two pilot sites for its experimentation, the DomoCasa Lab in Peccioli near Pisa, and the Länsgarden Living Home in Örebro, Sweden. Both sites include several apartments on multiple floors, connected by hallways and elevators accessible to the robots, and are equipped with a variety of fixed and mobile sensors. In addition to standard environmental sensors (e.g. temperature, humidity), apartment doors and windows are partially equipped with switches for monitoring state, while fixed cameras and IR-cameras provide images that can be used for user tracking. RFID tags mounted on the floor and furniture can help with robot localization and navigation, as well as object recognition. Finally, the system comprises a set of wireless health and biomonitoring sensors to be used by the elderly as required. Information by all sensors is published to the PEIS middleware and can be queried by all nodes in the system, including the global planner, all robots, software developers and end-users.

3.4 Condominium and Domestic Robots

Both the condominium and domestic robots are based on a low-cost but validated mobile platform [9]. Due to their small diameter, the robots can pass standard doors and are capable of navigating standard apartments. Yet, the platforms are robust and stable enough to provide walking-support and guidance. Both robots include a moveable sensor head with classic and RGB-D cameras. The condominium robot is equipped with a motorized tray for automatic goods and object exchange between itself and either of the other robots; it is also able to enter elevators and can therefore move through whole buildings. The domestic robot is equipped with a Kinova Jaco manipulator [16], which integrates a 6-DOF robot arm with a three finger robot hand (see figure 4). The arm has a reach of about one meter and supports payloads of about 1.5 kg. Originally designed for wheelchair use, the arm is already certified for medical/health operation and is considered safe even when operated by impaired users. The robot control software, described in detail below, is based on the ROS [19] middleware and the core components are shared between all three robots.

4. SOFTWARE ARCHITECTURE

As described above, our target system includes several robots as well as hundreds of ambient sensors, all part of a network and interacting with the end-users and caregivers. Obviously, the resulting complexity can only be managed by a layered software architecture with carefully designed interfaces, abstractions, and code-reuse wherever possible.

For Robot-Era, the PEIS middleware was selected to control the ambient sensor network, while ROS [19] was chosen for robot control. PEIS is short for "Physically Embedded Intelligent Systems" and is a middleware that aims to ease implementations of ambient intelligence. Because ambient sensors are usually optimized towards small size, cost and power consumption, they are computationally constrained so much that running a common robotics-middleware like ROS is not feasible. PEIS allows for spanning a peer-to-peer network between all these systems and enables information exchange using so-called tuples. A tuple consists of a name (key) and a data field (payload), both of which can be defined by the developer to implement notifications and services particular to a given sensor or actuator. The system provides an auto-discovery service and allows new nodes to enter the network and to create tuples. Nodes can search for tuples based on the publishing node, and can subscribe to tuples originating from a specified source to monitor updates to the tuples.

Figure 4: Domestic robot and the Kinova Jaco arm

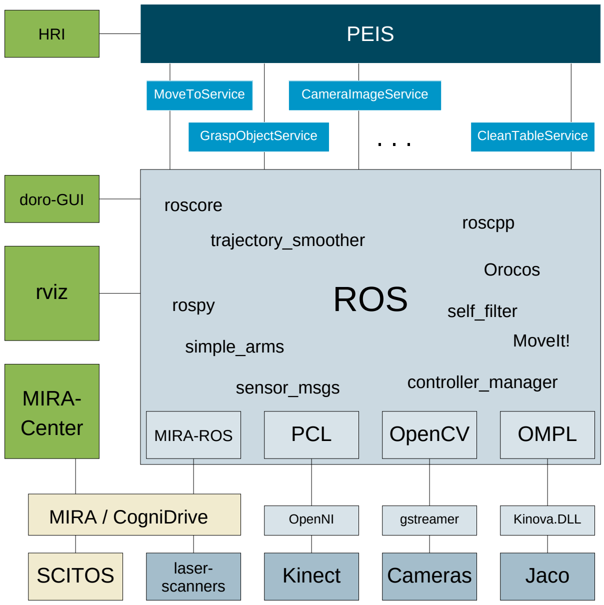

Figure 5: The software architecture of the domestic robot platform and the planning components: The robot software is mainly based on ROS and provides interfaces to the industry-proven navigation stack MIRA/Cognidrive and the planning and ambient intelligence framework PEIS.

Skill	Level	Description
Emergency Stop	L	safe stop of all robot motion
GetCameraImage	L	sends an image from the selected camera to PEIS
GetKinectImage	L	reads one Kinect image and 3D point-cloud data
GetLaserScan	L	reads one laser scan for navigation and docking
MoveTo	L	drives the robot to the given pose (x,y,θ) on the same floor
MovePtu	L	moves the robot head to the given direction
MoveJacoArm	L	moves the robot arm to the given position
RetractJacoArm	L	moves the arm to the save park position
...		
DetectKnownObject	I	find position and orientation of the requested object
GraspKnownObject	I	move the hand to the object and grasp it
PlaceObjectOnTray	I	puts a grasped object onto the robot's tray
MoveHingedDoor	I	arm motion to grasp and open a door
HandoverObjectToUser	I	move the arm towards the user, wait for voice command (confirmation), then release the object
...		
DetectPerson	H	look for and identify a person from camera images
WalkingSupport	H	move towards the user, rotate for easy grasping by the user, then drive the robot according to the user movements.
CleanTable	H	detect all objects on a table, put onto the robot tray, carry to kitchen, then put into kitchen sink.
...		

Figure 6: Example skills defined for the domestic robot. L: low, I: intermediate, H: high-level.

Within just five years since its introduction, the Robot Operating System (ROS) [19] has established itself as a powerful and versatile tool for the implementation of complex robot systems. The basic idea behind ROS is to create complex systems and behavior from a collection of simple individual software modules. It may be noted that this aspect of the ROS middleware is quite similar to PEIS; however, it it offers a far more feature-complete infrastructure and was designed for high throughput and low latency, as required in real-time robotics. The system also provides a powerful mechanism to define and use structured messages with complex data-types, in turn facilitating the development of new ROS modules ranging from low-level device drivers to high-level modules like motion planning and human-robot interaction. Almost all commercially available sensors and most robot arms and platforms are now supported by ROS, and the flexibility of the software implies that new components can be added easily to an existing robot. For example, the software architectures for the WillowGarage PR2 [5] and Care-o-bot [21] service robots are completely based on ROS. Using ROS for all robots in the Robot-Era project permits re-using large amounts of code for solving common problems like perception as well as motion planning and obstacle avoidance for the manipulators.

Figure 5 shows a simplified block-diagram of the software architecture designed for the domestic robot. Most of the high-level functionality including perception and manipulation planning is provided by a large number of ROS nodes and interfaces to key software libraries (PCL for 3D point-cloud processing, OpenCV and GStreamer for image processing, OMPL for collision aware motion planning). The two bottom layers indicate the major hardware components and the corresponding required device drivers. These are the Scitos mobile platform with laserscanners, Kinect 3D camera, standard video-cameras, and the Kinova Jaco robot arm. The software blocks on the left (*doro-GUI*, *rviz*, and *MIRA-center*) indicate the different user-interface components provided for the robot and service developer.

While ROS provides comprehensive support for robot navigation and localization, we currently rely on the vendor's MIRA and Cognidrive modules to actually drive the condominium and domestic robots. MIRA (Middleware for Robotics Applications) is a direct competitor of ROS. While ROS has its roots in the acedemic community, MIRA has a more commercial background and was created to overcome some shortcomings of ROS and guarantee stability and safety as required in industrial and medical applications. Some components — like localization — use the same approaches as found in ROS, while other parts are implemented differently. One such example is motion planning and control: the dynamic window approach [7] used in MIRA's CogniDrive modules exhibits a much more stable and predictable behavior for the differential-drive robots found in the project. Because safe driving turned out to be key to end-user acceptance, it was decided to delegate driving to MIRA's CogniDrive.

4.1 Three Levels of Robot Skills

Of course, neither the end-user nor the PEIS system should ever be aware of the software complexity described above. Instead, the complexity is carefully hidden and all functionality of the domestic robot is encapsulated in a set of basic robot skills. As described below, the ambient sensor network and the multi-robot planner create and update PEIS tuples to trigger and monitor those robot actions. A sequence of basic skills is then combined to provide the high-level services originally requested by the user using spoken commands or the tablet-PC.

Figure 6 lists the basic skills designed and implemented

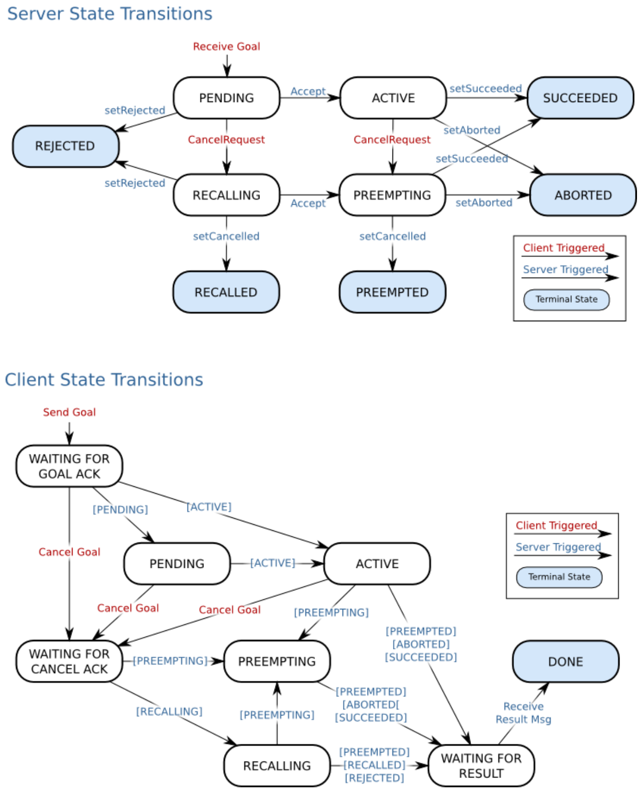

Figure 7: State-machine transitions for each action service (courtesy of www.ros.org).

for the domestic robot. For convenience, they are sorted into three levels of complexity, where the lowest level corresponds to primitive operations that access a single sensor or actuator, like *GetCameraImage* to read one camera image, *MovePtu* to move the robot head via its pan-tilt-unit, *MoveJacoFingers* to open or close the fingers on the Jaco arm, or *MoveTo* to drive the robot to a given location. The *intermediate skills* integrate information from several sensors and provide task-oriented functions, e.g. *DetectKnownObject* to check camera and sensor data for the position and orientation of a given object, or *GraspAndLiftKnownObject* to move the arm towards and then grasp a known object, while *HandoverObjectFromUser* tries to detect a person, and then move the arm to take an object handed to the robot. Finally, high-level services correspond to skills that combine state-of-the-art perception and manipulation functions. For example, *CleanTableService* requests the robot to detect all dishes and cups on a table, to grasp them and to carry them to the kitchen sink for cleaning.

4.2 Actionlib Interface

While some of the robot skills described in the last section execute quickly and reliably, most of the robot actions need several seconds for planning and execution, and there also is a certain chance that a requested action will fail. Fortunately, ROS includes the *Actionlib* software module, which provides a very flexible abstraction of pre-emptable remote procedure calls. Instead of using a simple blocking procedure call, the interaction between and actionlib client and server is split into several phases and can be interrupted by both sides when necessary. This is a great help for the implementation of long-running complex robot tasks.

The client first sends a *request* message to the server, which in turn can accept or reject the request. The request message includes all relevant parameters for the call. Once accepted, the server begins to send customized *feedback* messages which inform the client about the progress of the actionlib call. In this phase, the client can decide to interrupt the ongoing service by sending a *cancel* message, which in turn is accepted or rejected by the server. As soon as the service is complete, a final *status* message is sent back to the client, which also includes the final state, indicating success or detailed reasons for failure. See figure 7 for an overview of the most important state-transitions for a single *Actionlib* service.

4.3 Integration of PEIS and ROS

One of Robot-Era's contributions is the integration of large-scale ambient intelligence with robotics and high-level planning. Since the first two modules are implemented using PEIS and the remaining ones using ROS, a bi-directional bridge was designed. It consists of a set of extendable C++ classes, unifying both frameworks:

- an arbitrary number of so-called *TupleHandler*s can be instantiated and subscribe to the desired tuples with the PEIS ecology. Whenever corresponding PEIS tuples are either generated or updated by any other PEIS node, the *TupleHandler* is notified and can cause any action in the ROS network, like publishing a message or calling a service.
- to accomodate communication from ROS to PEIS, the bridge allows for subscribing to ROS topics and advertising multiple services. When such messages are received (or services invoked), PEIS-tuples can be generated and published.

The interaction between PEIS and ROS is visualized for the bring-object service (see figure 8). The planner communicates with the ambient intelligence and the robot using PEIS-tuples. On the robot, these tuples are received by ROS nodes instantiating tuple-handlers. These ROS-nodes then communicate with other ROS nodes or in case of motion planning and execution with the MIRA framework. Figure 9 lists the corresponding PEIS-tuples relevant for this example.

Using the software components described above, it was possible to successfully operate the domestic robot platform during the first experimental loop (see figure 10). A detailed analysis of the experiments will be presented in an upcoming publication.

5. CONCLUSIONS

We present a novel software design approach for elderly-care service-robots, based on storyboards and services developed in close cooperation with actual end users. In first tests the systems was able to fulfil most of the basic tasks. The studies showed the users to be very interested in robotics and to accept the scenarios well.

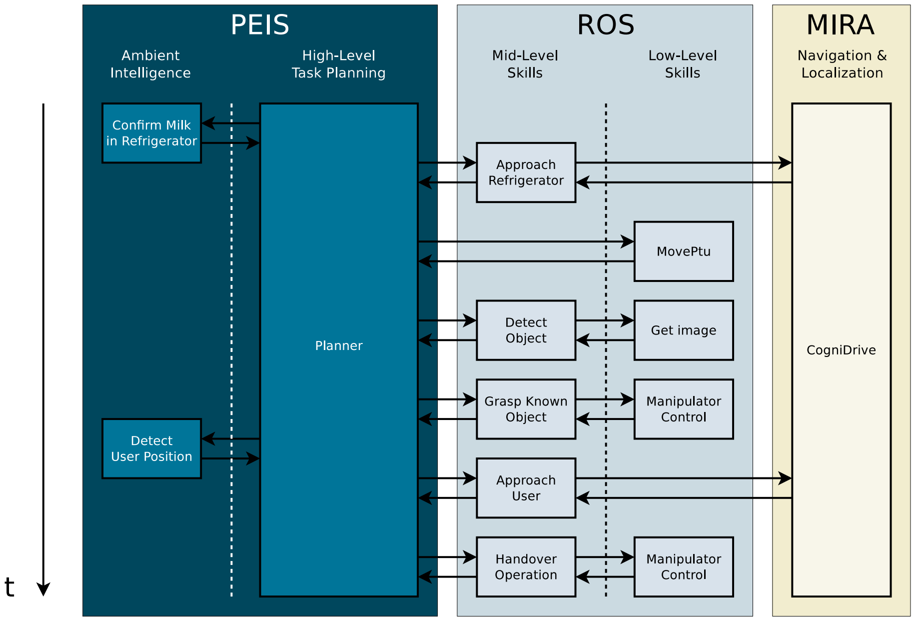

Figure 8: Sequence diagram for the bring-object service

Activity	Tuple
Move_to(location)	robot_name.moveto.id = robot_name.moveto.id.parameters = LOCATION robot_name.moveto.id.command = ON/OFF robot_name.moveto.id.state = IDLE / RUNNING / COMPLETED / FAILED
Detect(object)	robot_name.detect.id = robot_name.detect.id.parameters = OBJECT_NAME robot_name.detect.id.command = ON/OFF robot_name.detect.id.state = IDLE / RUNNING / COMPLETED / FAILED
Grasp(object)	robot_name.grasp.id = robot_name.grasp.id.parameters = OBJECT_NAME robot_name.grasp.id.command = ON/OFF robot_name.grasp.id.state = IDLE / RUNNING / COMPLETED / FAILED
Handover(object)	robot_name.handover.id = robot_name.handover.id.parameters = OBJECT_NAME robot_name.handover.id.command = ON/OFF robot_name.handover.id.state = IDLE / RUNNING / COMPLETED / FAILED
Object location	context.object.name = OBJECT_NAME context.object.pose = <X,Y,Z,R,P,YAW>
Robot location	context.robot_name.position = <X,Y,PHI>
User location	context.user.location.room = ROOM_X

Figure 9: Example for PEIS tuples sent in the object delivery scenario

Figure 10: Photos from the first experimental loop

The moderate-cost domestic robot forms part of a larger system integrating three different robots with a sensor-network for ambient assisted living. Based on an advanced speech and tablet-based user-interface, the system provides support for a variety of monitoring and household activities.

Future work includes the design and implementation of more advanced manipulation capabilities, in particular regarding autonomous cleaning tasks. In addition, lessons learned during the recent first experimental phase of the project are expected to improve the benefit of the robot services for the elderly users.

Acknowledgments This work is partially funded by European project Robot-Era under contract FP7-IP-288 899.

6. REFERENCES

[1] O. Alliance. Osgi-the dynamic module system for Java. *accessed, May*, 25, 2009.

[2] M. Broxvall, B.-S. Seo, and W. Kwon. The peis kernel: A middleware for ubiquitous robotics. In *Proc. of the IROS-07 Workshop on Ubiquitous Robotic Space Design and Applications*, pages 212–218, 2007.

[3] M. Chan, E. Campo, and D. Estève. Assessment of activity of elderly people using a home monitoring system. *International Journal of Rehabilitation Research*, 28(1):69–76, 2005.

[4] M. Chan, E. Campo, D. Estève, and J.-Y. Fourniols. Smart homes – current features and future perspectives. *Maturitas*, 64(2):90–97, 2009.

[5] S. Cousins. ROS on the PR2. *Robotics & Automation Magazine, IEEE*, 17(3):23–25, 2010.

[6] J. Demongeot, G. Virone, F. Duchêne, G. Benchetrit, T. Hervé, N. Noury, and V. Rialle. Multi-sensors acquisition, data fusion, knowledge mining and alarm triggering in health smart homes for elderly people. *Comptes Rendus Biologies*, 325(6):673–682, 2002.

[7] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *Robotics Automation Magazine, IEEE*, 4(1):23–33, 1997.

[8] T. Fuxreiter, C. Mayer, S. Hanke, M. Gira, M. Sili, and J. Kropf. A modular platform for event recognition in smart homes. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, pages 1–6, 2010.

[9] H.-M. Gross, H.-J. Böhme, C. Schröter, S. Mueller, A. König, C. Martin, M. Merten, and A. Bley. Shopbot: Progress in developing an interactive mobile shopping assistant for everyday use. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 3471–3478. IEEE, 2008.

[10] H.-M. Gross, C. Schröter, S. Mueller, M. Volkhardt, E. Einhorn, A. Bley, C. Martin, T. Langner, and M. Merten. Progress in developing a socially assistive mobile home robot companion for the elderly with mild cognitive impairment. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2430–2437. IEEE, 2011.

[11] A. Hein, M. Eichelberg, O. Nee, A. Schulz, A. Helmer, and M. Lipprandt. A service oriented platform for health services and ambient assisted living. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, pages 531–537. IEEE, 2009.

[12] H.-P. Huang and L.-P. Hsu. Development of a wearable biomedical health-care system. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1760–1765. IEEE, 2005.

[13] C. Huijnen, A. Badii, H. van den Heuvel, P. Caleb-Solly, and D. Thiemert. "maybe it becomes a buddy, but do not call it a robot"–seamless cooperation between companion robotics and smart homes. In *Ambient Intelligence*, pages 324–329. Springer, 2011.

[14] C. Huijnen, A. Badii, H. van den Heuvel, P. Caleb-Solly, and D. Thiemert. Improving domiciliary robotic services by integrating the astro robot in an ami infrastructure. In *Gearing up and accelerating cross-fertilization between academic and industrial robotics research in Europe*, pages 267–282. Springer, 2014.

[15] H. Lowet D., Frank. Florence - a multipurpose robotic platform to support elderly at home. In *Proceedings of the 2012 Workshop on Ambient Intelligence Infrastructures (WAmIi)*, 2012.

[16] V. Maheu, J. Frappier, P. Archambault, and F. Routhier. Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[17] D. Marculescu, R. Marculescu, N. H. Zamora, P. Stanley-Marbell, P. K. Khosla, S. Park, S. Jayaraman, S. Jung, C. Lauterbach, W. Weber, et al. Electronic textiles: A platform for pervasive computing. *Proceedings of the IEEE*, 91(12):1995–2018, 2003.

[18] M. Nani, P. Caleb-Solly, S. Dogramadzi, T. Fear, and H. van den Heuvel. Mobiserv: An integrated intelligent home environment for the provision of health, nutrition and mobility services to the elderly. 2010.

[19] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *Open-Source Software workshop of (ICRA)*, 2009.

[20] T. Rehrl, R. Troncy, A. Bley, S. Ihsen, K. Scheibl, W. Schneider, S. Glende, S. Goetze, J. Kessler, C. Hintermueller, and F. Wallhoff. The ambient adaptable living assistant is meeting its users. In *AAL Forum 2012*, 2012.

[21] U. Reiser et.al., Care-O-Bot 3 – creating a product vision for service robot applications by integrating design and technology. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1992–1998. IEEE, 2009.

[22] N. Webb, D. Benyon, J. Bradley, P. Hansen, and O. Mival. Wizard of oz experiments for a companion dialogue system: eliciting companionable conversation. In *LREC*, 2010.

[23] D. Zhang and M. MoUuari. Aware home understanding life activities. *Toward a humanfriendly assistive environment: ICOST*, page 186, 2004.