Building Knowledge-enabled Cloud Robotics Applications using the Ubiquitous Network Robot Platform

Moritz Tenorth*

tenorth@cs.uni-bremen.de

Koji Kamei†

kamei@atr.jp

Satoru Satake†

satoru@atr.jp

Takahiro Miyashita†

miyasita@atr.jp

Norihiro Hagita†

hagita@atr.jp

Abstract—In this paper, we discuss how networked robot architectures can facilitate the development, deployment, management and adaptation of distributed robotic applications. Our aim is to modularize applications by factoring out environment-, task-, domain-, and robot-specific knowledge components and representing them explicitly in a formal knowledge base that is shared between the robots and service applications. Robot control decisions can then be formulated in terms of inference tasks that are evaluated based on this knowledge during task execution. The explicit and modular knowledge representation allows human operators with different areas of expertise to adapt the respective parts of the knowledge independently. We implemented this concept by integrating knowledge representation methods of the ROBOEARTH project with the distributed task execution capabilities of the Ubiquitous Network Robot Platform.
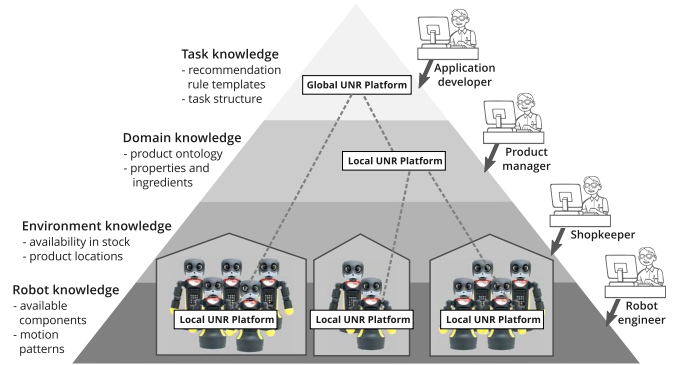
Fig. 1. The modular structure of the proposed system allows different groups of operators to focus on their area of expertise: Robot programmers implement the basic functionality, application developers combine these modules to provide useful services, while shop operators provide environment-specific information needed to execute the applications.

I. INTRODUCTION

With "Cloud Computing" becoming more and more popular, there has been increasing interest in applying similar concepts to robotics: Performing complex computations or storing large-scale knowledge bases can often be done more efficiently on dedicated server hardware, requiring less computing power, less memory and therefore also less battery capacity on the robot itself. Applications of these concepts to robotics are commonly referred to as "cloud robotics" [1]. There have been several efforts to move parts of the robot control program into the cloud, each focusing on different aspects like storing and sharing knowledge, off-loading complex computations, coordinating distributed robot teams, or remotely operating partly autonomous robots.

Kamei et al. [2] introduce the Ubiquitous Networked Robot Platform (UNR-PF) as a framework for distributed task coordination and control. The UNR-PF abstracts away from the robot's concrete hardware and offers a generic interface that can be used by application developers to create hardware-independent robotic services. A developer can request components that fulfill a given specification, and the UNR-PF will then assign suitable devices that can be controlled remotely. The UNR-PF supports a hierarchy of local and global platforms that allow to remotely start, supervise and coordinate tasks that are jointly performed by a set of components on different physical robots. Taken together, the hardware abstraction and the networked control

architecture facilitate the creation of re-usable distributed applications that can run on very different robot platforms.

While such applications are hardware-independent and reusable, they do not easily adapt to novel situations. Since the control programs are compiled code, all control decisions, the complete task structure and all interactions with objects and the environment need to be included into the program. As a result, all required information needs to be known and all decisions need to be taken at compilation time. It will be difficult to foresee all circumstances the robot will face during operation, so the control program has to actively tackle this open-world challenge and be designed in a way that additional knowledge can be acquired and used for accomplishing the robot's tasks.

In this paper, we investigate how a knowledge-enabled and cloud-based approach can help to make the robot control program more flexible and adaptable (Figure 1). We use ROBOEARTH, a cloud-based knowledge base designed for the exchange of knowledge between robots, to encode the task structure, environment information, object descriptions and general common-sense knowledge as a formal knowledge base. Using this knowledge base, the system can flexibly infer control decisions based on the most up-to-date information the robot has at that time. For example, the decision where to search for objects in an environment is formulated as query to the knowledge base that is evaluated based on the information available at the time the information is needed. Instead of programming in detail what to do in which situation, the application developer specifies which information is required in terms of a query, and the knowledge base will then use all the knowledge and inference methods

* Moritz Tenorth is with the Institute for Artificial Intelligence and the TZI (Center for Computing Technologies), University of Bremen, Germany.
† Koji Kamei, Satoru Satake, Takahiro Miyashita and Norihiro Hagita are with the Intelligent Robotics and Communications Laboratories, ATR, Kyoto, Japan.

that are available to compute answers to it.

Other than a compiled program, knowledge is composable and can be represented in a modular fashion. This facilitates the independent update of different parts of the knowledge base (like task specifications and environment maps) by experts in the respective fields (Figure 1). Since the central knowledge base is in the Cloud, these experts can remotely edit its content and deploy the updates to a large number of robots in different locations. In an open world, robots will eventually face the problem of missing information that would be required to accomplish a task. Using the cloud-based system, they can ask human operators to provide the missing pieces of information.

In our experiments, we applied the developed methods to robots that interact with customers in a convenience store (Figure 6) to answer questions about product locations or ingredients and to recommend alternatives if a product is not available. This scenario illustrates how actual robot systems can benefit from the aspects described above. Efficiently managing a large number of such robots in different locations requires generic robot-independent applications (e.g. for product recommendation) that, however, need to be parameterized with the spatial arrangement of products in the market and knowledge about the product types and their properties (e.g. their prices and ingredients). Parts of the required knowledge can be shared among robots of the same type (motions and hardware interaction), that operate at one location (environment map) or in stores of the same brand (product catalog and properties), respectively.

The main contributions of this paper are (1) the knowledge-enabled and cloud-based distributed robot control framework, (2) techniques for integrating human operators with different expertise as knowledge sources, and (3) the application of the methods to robots interacting with customers in a convenience store. In the following sections, we start with an overview of related work on cloud robotics, and then present the architecture of our system and its main components. We then describe our experimental setup and the capabilities exhibited by the knowledge-enabled system and finish with a discussion of our approach.

II. RELATED WORK

Multiple systems have been proposed that address different aspects of the Cloud Robotics vision: Some of them focus on remote sensor data processing [3] or on implementing computationally expensive algorithms in the cloud [4]. The "PR2 Remote Lab" investigates robot teleoperation and remote control via the Internet using a Web browser [5]. The Ubiquitous Network Robot Platform (UNR-PF) deals with distributed task execution and supervision [6] on multiple robots and sensorized devices at different locations. The ROBOEARTH project develops a web-based knowledge base through which robots can share information they have obtained [7]. Other attempts try to make existing web- and cloud-based resources available to robots. While robot-specific applications will first need to be established and filled with content, many applications originally developed
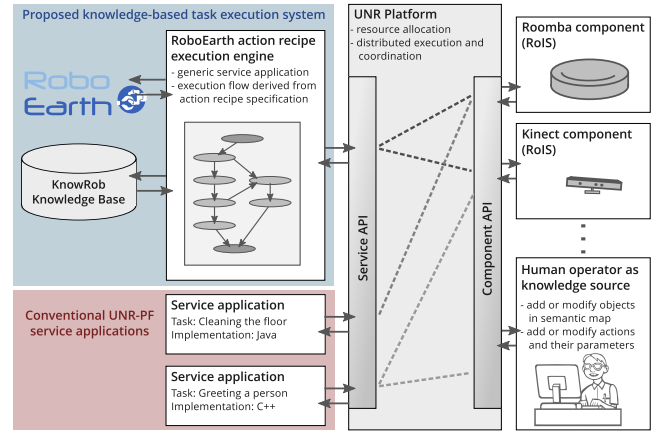

Fig. 2. Structure of the proposed system. Other than common task-specific robot applications, the generic execution engine can be parameterized with knowledge-based task descriptions. Control decisions are defined in terms of queries to the knowledge base that are answered based on the robot's background knowledge and belief state. A human operator can be contacted to provide missing information.

for humans do already provide information that can be useful for robots [8]. Examples are cloud-based object recognition systems like Google Goggles [9], on-line image and object model repositories like the 3D Warehouse [10], and product information from shopping websites as well as task instructions and cooking recipes [11]. A recent survey by Goldberg gives a good overview of cloud robotics approaches [12]. So far, all of these systems have focused on single aspects like distributed task execution, cloud-based information exchange, and shared-autonomy tele-operation. In this paper, we show how their combination can enable novel functionality.

III. SYSTEM OVERVIEW

The system presented in this paper combines the distributed task execution methods of the UNR-PF with the knowledge-sharing techniques of ROBOEARTH. Figure 2 gives an overview of its main components. The UNR-PF is used as communication middle-ware and abstraction layer between the hardware components and robot applications. A special execution engine can interpret task specification shared via ROBOEARTH and execute them on the platform, acting as a generic robot application that can be parameterized with different task descriptions. The execution interacts with the system's knowledge base to resolve abstract specifications in the task descriptions to concrete parameters that are needed for executing the actions. A graphical interface for a human operator facilitates the inspection, modification and creation of knowledge.

A. The Ubiquitous Network Robot Platform (UNR-PF)

The UNR-PF acts as an interface layer between hardware- and software components on the one side and robotic service applications on the other side. The components implement well-defined interfaces (e.g. for a *PersonIdentification* or a *Reaction*) and offer this functionality to the platform via the component API. Applications compose useful functionality out of these basic building blocks. Since all dependencies

are defined in terms of the abstract interfaces, robot applications are agnostic of how the functionality is provided or which components provide it. The *MovingPlatform* interface, for example, could be implemented using a wheeled or a legged platform, both providing the functionality of moving the robot. Before executing a command, an application requests components that have the required properties from the platform and, if suitable ones are available, interacts with them via the service API. The UNR-PF draws upon different standardized platforms and representations: The component interface of the UNR-PF is based on the RoIS standard [13]. Spatial information is encoded following the Robotic Localization Service (RLS) standard [14] and the CityGML language [15]. The UNR-PF is publicly available as open-source software[1].

B. The ROBOEARTH knowledge base

ROBOEARTH aims at building a "World Wide Web for Robots", a web-based Wikipedia-like platform for sharing knowledge about actions, objects, and environments between robots (Figure 3). All pieces of information stored in the knowledge base are annotated with their requirements in terms of robot capabilities which are checked when downloading them. Using these requirement specifications, a robot can determine whether it will be able to use this information as it is or if additional information is needed. The client-side reasoning system is built using the KNOWROB knowledge processing system [16] which is used for storing knowledge, drawing inferences, and offering a query interface to the execution engine.
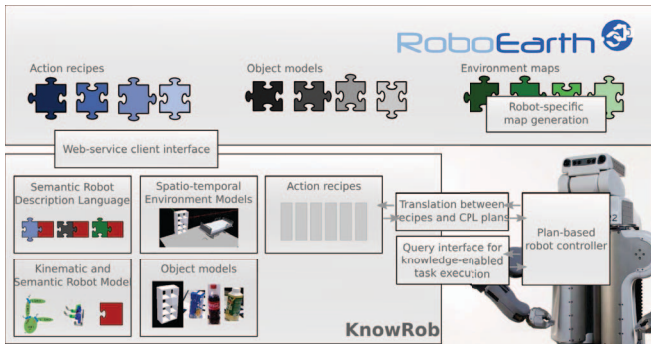

Fig. 3. Overview of the ROBOEARTH system for exchanging information about robot actions, object models and environment maps between robots. The KNOWROB knowledge base is an important part of the system.

The knowledge stored in the ROBOEARTH system is described in a formal language [17] that is implemented as an extension of the Web Ontology Language (OWL, [18]). The ROBOEARTH ontology defines the concepts and properties that are available for representing knowledge. It is derived from the KNOWROB ontology [19] which, by itself, is partly derived from the OpenCyc ontology [20] that became a quasi-standard in robot knowledge representation. By re-using these existing ontologies, we ensure compatibility with parallel developments in this research area and can incorporate results of other projects more easily.

[1]http://www.irc.atr.jp/std/UNR-Platform.html

C. Knowledge-enabled task execution

Task descriptions in the ROBOEARTH language are not directly executable but, similar to cooking recipes, describe which actions need to be performed in which order and with which arguments. In order to execute them on the robot, they need to be interpreted by an execution engine that translates them into calls to the respective robot components and that supervises how the task is performed. Action recipes are composed of action classes, for example *LocalizingAPerson*, which are linked to the corresponding components in the UNR-PF. Before execution, the recipes are loaded and the system checks if all required components are available, then generates a state machine from the task description in the recipe, and starts the execution by calling the respective UNR-PF components as specified in the recipe. During this project, we have represented all components that have so far been implemented in the UNR-PF in the ROBOEARTH language as an extension of the action ontology. The UNR-PF is more focused on human-robot interaction, which had not been modeled in sufficient detail in ROBOEARTH, so the ontology has been extended with these kinds of actions.

Each action is transformed into one state in the state machine that first requests and binds all required components using the UNR-PF, then calls the respective commands, and finally returns their results. Depending on the results, the execution transitions to the next regular state or to a specified error state. Perceptual components, e.g. for person localization or identification, are interfaced using "perception actions" that wait for events generated by the components. The recipe can specify when these actions are to return, for example once the first event message has been received, or the first n messages, or after a specified condition evaluates to $true$ (e.g. once a specific person has been recognized). These conditions are described using OWL restrictions that are evaluated on the robot's knowledge base to check whether the condition is fulfilled.

To account for the heterogeneous nature of actions and their respective requirements in terms of information interchange with other actions, we developed a flexible information passing scheme using the local knowledge base. All information gained by executing an action is represented in the local knowledge base using the ROBOEARTH language, including the results of sensing actions, the outcome of manipulation actions, and information about which actions have been performed with which parameters. This approach closely links the knowledge-based task instructions to the robot's belief state and allows robotss to reason about and to integrate different sources of information like the instructions, the object ontology, and the semantic environment map. It also facilitates the exchange of information via the ROBOEARTH platform since all information in the system is already explicitly represented in this language.

D. Interactive knowledge editor interface

We combine the execution engine with a graphical user interface that allows human operators to extend and correct the robot's knowledge base, to add new objects to a map
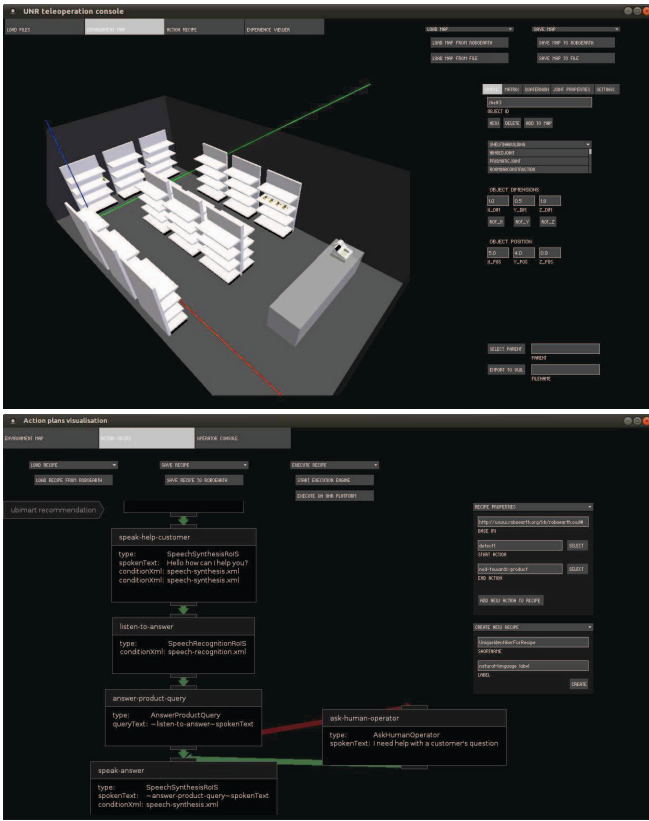
Fig. 4. Top: Semantic map visualization and editor. Bottom: Editor for defining the task structure and interactive execution interface.

or new actions to a task, and to start and supervise the task execution on a remote robot. The UNR-PF thereby serves as distributed platform for the run-time coordination between the operator, the task-level controller, and the different (robot) components that perform the task. ROBOEARTH complements this by providing the robot and the human operator with a shared knowledge base about the task to be executed and the environment the robot(s) operate(s) in.

The two main components of the user interface are visualizations and editors for semantic environment maps [21] and for robot task specifications [17]. Both editors can either load specifications from the robot's local knowledge base or import them from ROBOEARTH. The map editor can visualize and edit environment maps that describe the spatial arrangement of objects around the robot (Figure 4 top). In these maps, each object is described as an instance of an object class at a specified 6D-pose in space. This instance inherits all properties specified for its object class, and can be further annotated with additional properties. Using the editor, objects can be added to, deleted from and moved around in the map. The task editor can be used to create, visualize and edit task descriptions by adding or removing actions, changing their properties, and specifying transitions between actions. The user can specify in detail how the task shall react in nominal and error cases using conditions for action transitions (indicated by the differently colored arrows in Figure 4 (bottom)). Green arrows indicate a transition in case of successful execution, dark red ones are in case of an error, light and dark blue transitions are chosen depending on the

outcome of a decision node, etc. During task execution, the action editor doubles as supervision interface: The currently executed action is highlighted, action parameters can be inspected, and the task execution can be started and aborted.

E. Integration of human operators as knowledge sources

A robot can call a human operator during task execution and ask for information that it found to be missing. The communication is performed using the UNR-PF by adding the human operator interface as a component that can be commanded to acquire information. If the robot application (which is supervising the task execution) notices a problem, it executes the command for asking the operator for help. The human can then download the relevant information from the ROBOEARTH knowledge base, investigate the problem, update the information, upload it to ROBOEARTH again, and notify the application that the information is available.

The ability to ask for help raises the questions *when to ask* and *what to ask for*. We consider two main cases in which interaction with an operator seems necessary: when a query for information unexpectedly gave no results (e.g. when a question asked by a customer cannot be answered) or when an action failed or produced inappropriate behavior. The former case can be handled in the context of an ongoing task by transitioning to an "interaction state" that sends a support query to the operator, blocks until an answer has been received, and returns to the same action to try again. This interaction scheme is indicated in the right part of Figure 5. The latter is a more exceptional case that often requires modifications of the task definition itself. It may be detected by errors thrown by the UNR-PF, by checking if the outcome of an action is as expected (e.g. a customer that does not move away even if the robot considers a dialog to be complete), or if customers or the shopkeeper complain about the robot's behavior. In these cases, the task specification needs to be updated and the task needs to be restarted.

IV. EXPERIMENTS

We implemented the experimental scenario of robots interacting with customers in a convenience store. With respect to this scenario, we build upon prior work on establishing a ubiquitous-sensing infrastructure in a convenience store and using this infrastructure for guiding customers [22] and for recommending products [23]. The mock-up store is equipped with a ubiquitous sensing infrastructure, including laser scanners for tracking customers and RFID tag readers for detecting if objects have been picked up [24], as well as several interactive robots (Figure 6). With this scenario, we explore how semantic representations can yield greater flexibility in customer interaction, and how the proposed distributed infrastructure helps to create, deploy and maintain robot applications. A video showing the editor interfaces for semantic maps and action recipes as well as the execution of the task on the robot can be found online[2] as well as the
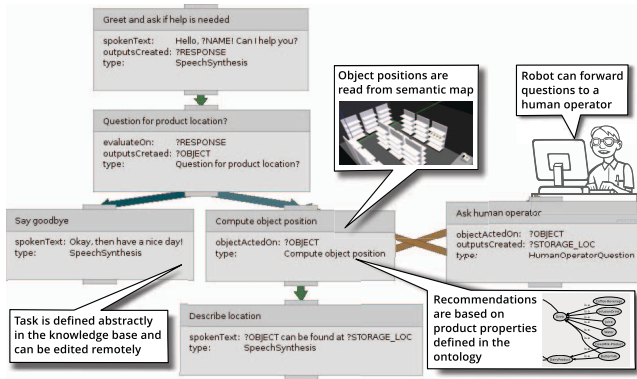
---

[2]http://vimeo.com/70096817

Fig. 5. Example task specification for recommending a product. The call-out boxes indicate where the proposed system contributes to the task execution.

OWL files with the class definitions for UNR actions and the action recipe for this experiment[3].

Figure 5 shows the definition of a recommendation task in the knowledge base and explains in which parts the contributions of this work come into play. The robot first greets the customer and asks if help is required. It then decides whether the customer asked for the location of a product (light blue arrow) and, if so, tries to compute its location based on the semantic environment map and explain how to get there. If no answer can be found, the question is forwarded to a human operator that updates the knowledge base with the required information. The accompanying video explains in more detail how knowledge can be edited and how the remote task execution and supervision interface can be used to control the robots.

A. Knowledge-based customer dialog

The robot can use its semantic and spatial knowledge about products in the environment to answer questions posed by customers. Questions can either be asked using a speech recognition system or (for development purposes) using a chat-based interface; both are implemented as *SpeechRecognition* component on the UNR-PF. Variations of the following questions can currently be understood:

- *Where can I find A [that (does|does not) contain B]?* This question type combines spatial information about the locations of products in the environment (from the semantic map) with semantic information about their ingredients (from the product ontology).
- *How much is A?* This question type reads the price property of a product class from the ontology. If the query asks for the price of a generic class (e.g. "How much is green tea?"), it reads the prices of all subclasses and returns the price range.
- *Is A x ?* This question type asks for product properties *x*, for example if *A* is edible, drinkable, perishable or vegetarian.

B. Spatial knowledge for guiding customers

A semantic map [21] contains instances of products and the locations where they can be found in the environment.

[3]http://knowrob.org/ontologies

When asked for the location of a product, the robot reads the location from the semantic map, computes the position relative to itself and points towards the objects in addition to the spoken answer. The map can be coupled with sensors in the environment, for example RFID tag readers, that update the information in the map and provide the robot with an up-to-date view of which products are still available. A graphical editor enables operators without robotics expertise to update the environment model if the shop layout has changed.

C. Semantic representations for product recommendation

Using semantic information like an ontology of products and their properties [8], a robot can flexibly answer questions about these products. The hierarchical structure of the ontology provides the robot with information which products belong to a category like *Food* or *Stationery*. Based on the represented product properties, it can answer questions about ingredients the customer may be allergic against. By computing which products are close in the ontology, it can find semantically similar alternatives if a product is not available any more [25].

D. Human operator as fall-back knowledge source

While the robots operate autonomously most of the time, there may be situations in which they cannot answer a customer's question. In this case, they can forward the question to a human operator that can update the robot's knowledge, for example by adding or removing products in the map or by changing their properties. The human and the robot share the same environment model that is distributed via the ROBOEARTH platform; when the human updates the map in ROBOEARTH, it is directly available to the robot as well. The required interaction with the human is included into the task specification as a special action that is triggered if a question cannot be answered.

E. Remote adaptation and deployment of task specifications

In some cases, the task specification itself needs to be updated, for instance to change the robot's recommendation behavior or to fix flaws in the task definition. This adaptation can be done in a centralized fashion since the tasks the robots perform are themselves described in the knowledge base and shared via ROBOEARTH. The operator can adapt the specifications remotely using a graphical editor interface and upload the updated version to the ROBOEARTH knowledge base, thereby making it available to all robots in the system.

V. DISCUSSION AND CONCLUSIONS

In this paper, we discussed how a system for knowledge-enabled distributed task execution can be built on top of existing, common platforms in order to increase modularity, flexibility, and adaptability of robot applications. By consequently separating generic functionality from environment-, domain- or task-specific knowledge, we intend to achieve a high degree of reusability as well as improved adaptability to novel situations. Different kinds of knowledge are explicitly represented and can be edited independently by the respective
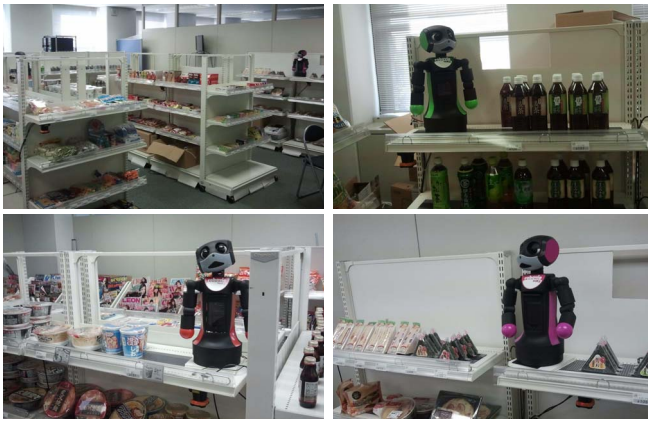
Fig. 6. Recommendation robots in the convenience store experiment setup.

domain experts. Control decisions are formulated as inference tasks that are evaluated on the robot's knowledge during execution.

Our approach involves abstraction along multiple dimensions: Abstraction from the robot hardware and the execution context is achieved by the components, services, and remote execution capabilities of the UNR-PF. Abstraction from the environment is obtained by encapsulating all environment-related knowledge in a semantic map that combines spatial and semantic information about objects. Abstraction from the executed tasks is done by a generic execution engine that can perform arbitrary tasks defined as combinations of basic functionality building blocks. Abstraction from the application domain can be achieved by parameterizing generic functionality with an abstract domain ontology. We expect this abstraction to increase re-usability of components in different tasks and environments since individual parts of the knowledge base can be exchanged independently. The same task description for serving a drink, for example, should work in different hospital rooms, kitchens or offices as long as an appropriate environment model is available.

VI. Acknowledgments

References

[1] E. Guizzo, "Robots with their heads in the clouds," *Spectrum, IEEE*, vol. 48, no. 3, pp. 16–18, 2011.

[2] K. Kamei, S. Nishio, N. Hagita, and M. Sato, "Cloud networked robotics," *IEEE Network Magazine*, vol. 26, no. 3, pp. 28 –34, May-June 2012.

[3] R. Arumugam, V. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. Kong, A. Kumar, K. Meng, and G. Kit, "DAvinCi: A cloud computing framework for service robots," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 3084–3089.

[4] D. Hunziker, M. Gajamohan, M. Waibel, and R. DAndrea, "Rapyuta: The roboearth cloud engine," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[5] B. Pitzer, S. Osentoski, G. Jay, C. Crick, and O. Jenkins, "Pr2 remote lab: An environment for remote development and experimentation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 3200–3205.

[6] J. Furrer, K. Kamei, C. Sharma, T. Miyashita, and N. Hagita, "Unr-pf: An open-source platform for cloud networked robotic services," in *System Integration (SII), 2012 IEEE/SICE International Symposium on*, dec. 2012, pp. 945 –950.

[7] M. Waibel, M. Beetz, R. D'Andrea, R. Janssen, M. Tenorth, J. Civera, J. Elfring, D. Gálvez-López, K. Häussermann, J. Montiel, A. Perzylo, B. Schießle, O. Zweigle, and R. van de Molengraft, "RoboEarth - A World Wide Web for Robots," *Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011.

[8] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz, "Web-enabled Robots – Robots that Use the Web as an Information Resource," *Robotics & Automation Magazine*, vol. 18, no. 2, pp. 58–68, 2011.

[9] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[10] U. Klank, M. Z. Zia, and M. Beetz, "3D Model Selection from an Internet Database for Robotic Vision," in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2406–2411.

[11] M. Tenorth, D. Nyga, and M. Beetz, "Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, May 3–8 2010, pp. 1486–1491.

[12] K. Goldberg and B. Kehoe, "Cloud robotics and automation: A survey of related work," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-5, Jan 2013.

[13] OMG, *Robotic Interaction Service (RoIS) 1.0*. Object Management Group, Feb. 2013. [Online]. Available: http://www.omg.org/spec/RoIS/1.0/

[14] ——, *Robotic Localization Service (RLS) 1.1*. Object Management Group, Aug. 2012. [Online]. Available: http://www.omg.org/spec/RLS/1.1

[15] OGC, *City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium, 2012, http://www.opengis.net/spec/citygml/2.0.

[16] M. Tenorth and M. Beetz, "KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots. Part 1: The KnowRob System," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566 – 590, April 2013.

[17] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz, "Representation and Exchange of Knowledge about Actions, Objects, and Environments in the RoboEarth Framework," *IEEE Transactions on Automation Science and Engineering (T-ASE)*, vol. 10, no. 3, pp. 643–651, 2013.

[18] W3C, *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax*. World Wide Web Consortium, 2009, http://www.w3.org/TR/2009/REC-owl2-syntax-20091027.

[19] M. Tenorth and M. Beetz, "KnowRob – Knowledge Processing for Autonomous Personal Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4261–4266.

[20] D. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[21] D. Pangercic, M. Tenorth, B. Pitzer, and M. Beetz, "Semantic object maps for robotic housework - representation, acquisition and use," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, October, 7–12 2012.

[22] K. Kamei, T. Ikeda, M. Shiomi, H. Kidokoro, A. Utsumi, K. Shinozawa, T. Miyashita, and N. Hagita, "Cooperative customer navigation between robots outside and inside a retail shop – an implementation on the ubiquitous market platform," *Annals of Telecommunications*, vol. 67, pp. 329–340, 2012.

[23] H. Kidokoro, K. Kamei, K. Shinozawa, T. Miyashita, and N. Hagita, "You stopped by there? I recommend this: changing customer behaviors with robots," in *Proceedings of the 13th International Conference on Ubiquitous Computing*. ACM, 2011, pp. 569–570.

[24] K. Kamei, K. Shinozawa, T. Ikeda, A. Utsumi, T. Miyashita, and N. Hagita, "Recommendation from robots in a real-world retail shop," in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ser. ICMI-MLMI '10. New York, NY, USA: ACM, 2010, pp. 19:1–19:8.

[25] M. Schuster, D. Jain, M. Tenorth, and M. Beetz, "Learning organizational principles in human environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012.