The RoboEarth Language: Representing and Exchanging Knowledge about Actions, Objects, and Environments (Extended Abstract)*

Moritz Tenorth
University of Bremen, Germany
tenorth@cs.uni-bremen.de

Alexander Perzylo
Technische Universität München, Germany
perzylo@cs.tum.edu

Reinhard Lafrenz
Technische Universität München, Germany
lafrenz@cs.tum.edu

Michael Beetz
University of Bremen, Germany
beetz@cs.uni-bremen.de

Abstract

The community-based generation of content has been tremendously successful in the World Wide Web – people help each other by providing information that could be useful to others. We are trying to transfer this approach to robotics in order to help robots acquire the vast amounts of knowledge needed to competently perform everyday tasks. RoboEarth is intended to be a web community by robots for robots to autonomously share descriptions of tasks they have learned, object models they have created, and environments they have explored. In this paper, we report on the formal language we developed for encoding this information and present our approaches to solve the inference problems related to finding information, to determining if information is usable by a robot, and to grounding it on the robot platform.

1 Introduction

In the Web 2.0, content is now often generated by the users of a web site that form a community of people helping each other by providing information they consider useful to others. Wikipedia grew up to millions of articles, sites like *cooking.com* or *epicurious.com* provide tens of thousands of cooking recipes, and *ehow.com* and *wikihow.com* contain instructions for all kinds of everyday tasks. “Crowdsourcing” the generation of web content made it possible to create large web sites in shorter time with shared effort. In our research, we are investigating how this approach can be transferred to robotics. On the one hand, we aim at enabling robots to use information that can already be found on the Web, for instance by translating written instructions from web pages into robot plans [Tenorth et al., 2011]. On the other hand, we are working towards a “World Wide Web for Robots”, called ROBOEARTH (Figure 1), that is to be a web-based community in which robots can exchange knowledge among each others. Understanding instructions that were made for humans is still difficult, but once the information is made available in a robot-compatible formal language, it should be possible to share it with other robots. These other robots then do not have to go through the difficult conversion process again. We thereby hope to speed up the time-consuming knowledge acquisition process by enabling robots to profit from tasks other robots have already learned, from object models they have created, and from maps of environments they have explored. If information is to be used by robots without human intervention, it has to be represented in a machine-understandable format. In this respect, our system has much in common with the Semantic Web [Lee, Hendler, and Lassila, 2001] that also aims at creating machine-readable web content. An explicit representation of the semantics is important to enable robots to *understand* the content, i.e. to set single pieces of information into relation. Only if they know the semantics of the exchanged information, robots can decide if an object model will be useful to perform a given task, or determine if all required sensors are available. In particular, the representation language provides techniques for describing:

- Actions and their parameters, object poses in the environment, and object recognition models
- Meta-information about the exchanged data, e.g. types, file formats, units of measure, coordinate frames
- Requirements on components a robot needs to have in order to make use of a piece of information
- Self-models of a robot’s components and capability configuration
- Methods for matching requirement specifications to a robot’s capabilities to identify missing components

In this paper, we describe our approach to creating a semantic representation language for the ROBOEARTH system. It is an extended abstract of a paper presented at ICRA 2012 [Tenorth et al., 2012]. A journal version of this paper, with a more in-depth description of the language constructs is to appear [Tenorth et al., 2013]. The main contributions are (1) a semantic representation language for actions, objects, and environments; (2) the infrastructure for using this representation to reason about the applicability of information in a given context and to check if all required robot capabilities are available; and (3) mechanisms for creating and up-

*The paper on which this extended abstract is based was the recipient of the Best Cognitive Robotics Paper Award of the 2012 IEEE International Conference on Robotics and Automation (ICRA) [Tenorth et al., 2012].

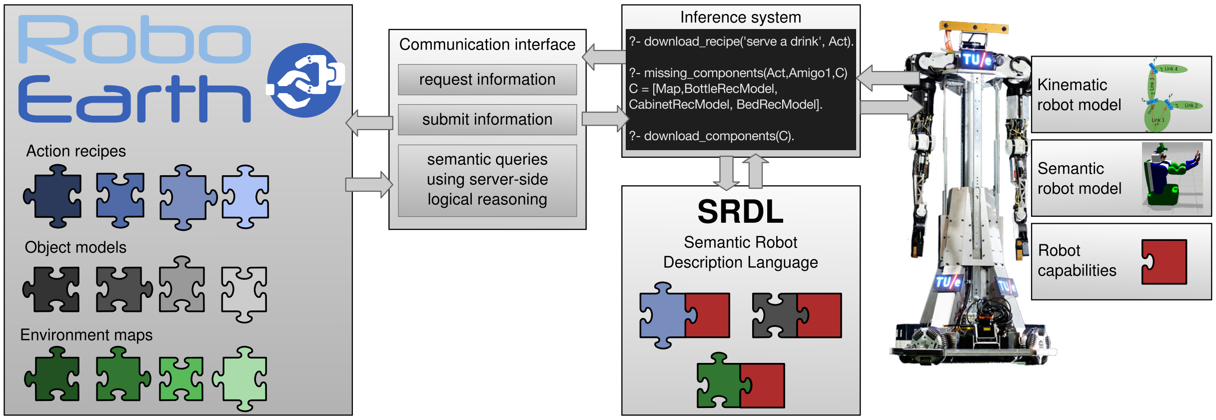

Figure 1: Overview of the ROBOEARTH system: A central database provides information about actions, objects, and environments. The robot can up- and download information and determine if it can use it based on a semantic model of its own capabilities.

loading shared knowledge. These technical contributions are validated by an experiment including two physical robots performing a serving task in two different environments based on information retrieved from and shared via ROBOEARTH.

2 Related Work

As a platform for knowledge exchange between heterogeneous robots, ROBOEARTH requires semantic representations that provide a robot with all information it needs to select information from the knowledge base, to adapt it, and to reason about its applicability in a given situation. Earlier research on knowledge representation for actions or objects usually did not deal with this kind of meta-information, but rather focused on the representation of the information itself, for example in Hierarchical Task Networks (HTN [Erol, Hendler, and Nau, 1994]) and related languages for plan representation [Myers and Wilkins, 1997], workflow specifications [Myers and Berry, 1998], or the Planning Domain Definition Language (PDDL [Ghallab et al., 1998]). Generic exchange formats like the Knowledge Interchange Format KIF [Genesereth, Fikes, and others, 1992] are very expressive and generic languages, but have limited reasoning support. For ROBOEARTH, we chose a shared ontology as pragmatic solution instead of completely self-contained languages. Related work on sharing knowledge among robots focused either on sharing a common belief state in multi-robot systems [Khoo and Horswill, 2003], or on fundamental aspects like how heterogeneous robots can autonomously acquire and share symbols created from perceptual cues [Kira, 2010].

3 The ROBOEARTH System

The language presented in this article is part of the ROBOEARTH system [Waibel et al., 2011] which is a Wikipedia-like platform that robots can use for sharing knowledge about actions, objects, and environments. Most parts of ROBOEARTH have been released as open-source software packages[1] in the ROS robot middle-ware. In this paper, we focus on methods for representing the exchanged knowledge and reasoning about it. Figure 1 illustrates the exchange of knowledge via the ROBOEARTH platform: The central ROBOEARTH knowledge base, depicted on the left, contains descriptions of actions (called "action recipes"), object models, and environment maps. These pieces of information have been provided by different robots with different sensing, acting and processing capabilities, and therefore have different requirements on the capabilities a robot must possess in order to use them. The ROBOEARTH language provides methods for explicitly describing these required capabilities and for matching them against the capabilities that are available on the robot, visualized in the picture by the different shapes of puzzle pieces. Each robot has a self-model consisting of a description of its kinematic structure, including the positions of sensors and actuators, a semantic model of its parts (describing e.g. that a group of parts forms a gripper), and a set of software components like object recognition systems. We apply the Semantic Robot Description Language SRDL [Kunze, Roehm, and Beetz, 2011] to describe these components and the capabilities they provide, and to match them against the requirements specified for action recipes.

The representation language is realized as an extension of the KNOWROB [Tenorth and Beetz, 2013] ontology and described in terms of Description Logic formulas using the Web Ontology Language (OWL). OWL distinguishes between classes, instances of these classes, and properties that can either be described for single instances or for whole classes of things. Classes are arranged in a hierarchical structure, called an ontology, allowing multiple inheritance. KNOWROB's ontology is derived from the OpenCyc ontology [Lenat, 1995]; by staying compatible to this widely used system, we are able to use various tools and interfaces to and from Cyc. We extended the KNOWROB ontology with concepts that are especially required for the exchange of knowledge: Meta-information about the data to be exchanged like units, coordinate systems, its resolution, algorithms that were used for creating data, and requirements that are needed for interpreting it. For the sake of clarity, we will present most of the language constructs in terms of graphical visualizations instead of logical statements.

[1] Available at http://www.ros.org/wiki/roboearth

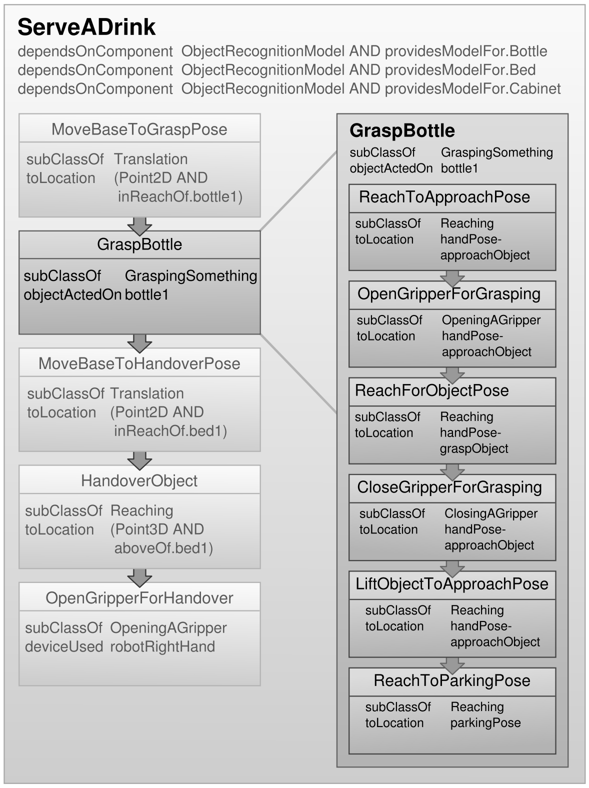

Figure 2: Representation of a "serving a drink" task, called "action recipe" in the ROBOEARTH terminology, which is composed of five sub-actions that themselves can be described by another action recipe.

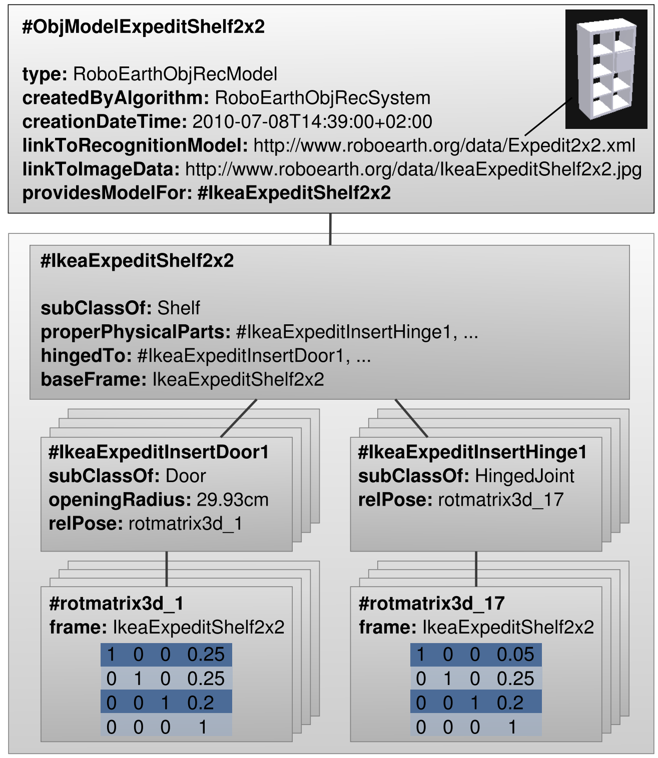

Figure 3: Object model of a cabinet composed of several articulated doors connected with hinges to the cabinet's frame.

4 Overview of the language elements

In this section, we give a brief overview of the language elements that constitute the ROBOEARTH language. Due to space constraints, we have shortened this section and refer to [Tenorth et al., 2013] for details.

4.1 Actions and Tasks

Actions are specified by creating a sub-class of one of the action classes in the KNOWROB ontology and extending the description with task-specific properties like the *fromLocation*, *toLocation* or *objectActedOn*. Figure 2 visualizes an action recipe for serving a drink to a patient in bed. In this picture, action classes are represented as blocks, properties of these classes are listed inside the block, and ordering constraints among the actions are shown as arrows between the blocks. There are three levels of hierarchy: The recipe for the *ServeADrink* action includes the *GraspBottle* action that, by itself, is defined by an action recipe (shown on the right side) consisting of single actions. Both recipes consist of a sequence of actions that are described as subclasses of generic actions, like *Reaching* or *Translation*, with additional task-specific parameters, like the *toLocation* or the *objectActedOn*.

The action recipe lists dependencies on components that have to be available on the robot in order to successfully perform the task, in this example some object recognition models that are necessary to recognize all objects in the task. Additional dependencies are inherited from higher-level action classes, exploiting the hierarchical structure of the action ontology. The dependency on an arm motion capability, for example, is specified for all sub-classes of *Reaching* at once and therefore does not have to be specified in each action recipe. These dependencies correspond to the "puzzle pieces" in Figure 1.

Before execution, the abstract descriptions of objects and locations need to be grounded in concrete locations using the robot's perception methods and its environment model as described in Section 5. The task specification can then be transformed into a robot plan that consists of calls to executable components and parameter specifications.

4.2 Object Models

Object models in ROBOEARTH describe classes of objects by their semantic properties, including information on how to recognize and how to articulate them. Figure 3 exemplarily shows a model of a cabinet in a mock-up hospital room. The upper part describes an instance of an object recognition model, including links to pictures and a CAD model as well as information about the creation time and the algorithm that can use the model. The recognition model instance refers to a description of the object class *IkeaExpeditShelf2x2* (depicted in the lower part) that consists of articulated parts, namely doors connected to its frame via hinges.

4.3 Environment Models

ROBOEARTH supports different kinds of environment maps (Figure 4), some of which are described in the OWL-based

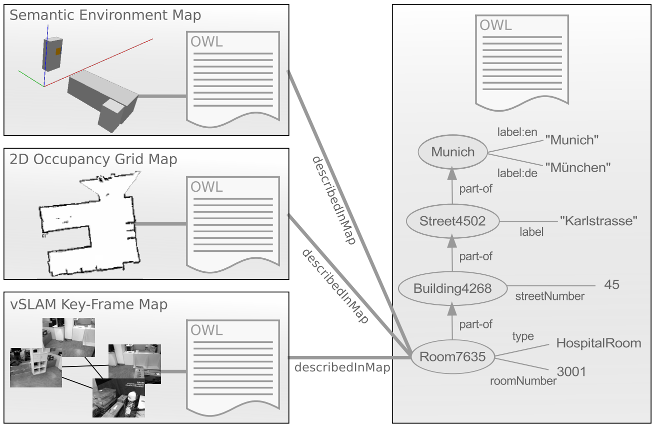

Figure 4: Different types of maps are either described completely in the ROBOEARTH language, or in linked binary file. A spatial hierarchy of room, building, street, city allows to search for maps in ROBOEARTH.

language itself (e.g. semantic maps containing positions of objects in the environment), others (like occupancy grid maps) are linked as binary files. All maps are annotated with an OWL description that specifies their types, some basic properties, and most important the address of the environment that is described in the map. The address is the main information that is used for finding suitable maps in the database.

5 Execution of Action Recipes

Since action recipes are not directly executable code but rather declarative action descriptions, they need to be interpreted by an execution engine in order to be executed on a robot. A reference implementation of an execution engine [di Marco et al., 2012] has been created that is based on the Cognitive Robot Abstract Machine framework (CRAM, [Beetz, Mösenlechner, and Tenorth, 2010]). In this implementation, action recipes are translated into robot plans described in the CRAM Plan Language (CPL). Compared to the OWL-based language for action recipes, which is optimized for reasoning and for integrating information sources, CPL specializes on the robust execution of plans.

6 Evaluation

We evaluated how the ROBOEARTH language can enable robots to perform tasks in previously unknown environments. The experiment included two heterogeneous robot platforms, a PR2 and an Amigo robot, serving a drink from inside a cabinet to a patient in bed at two different locations. Both environments had a different spatial layout but shared common pieces of furniture, which allowed sharing object-related information like the positions of the hinges of the cabinet. Their properties have been estimated by the first robot, uploaded to ROBOEARTH, and used by the second robot to open the door. The upper part of Figure 5 shows the environment maps that were downloaded from ROBOEARTH.

Both robots performed the task using the same execution engine and the same action recipe (shown in Figure 2. The capability matching determined that all required capabilities were available, but recognition models for some of the objects mentioned in the task were missing (namely the bottle and the bed) and had to be downloaded. The bottom two pictures in Figure 5 show how both robots opened the cabinet and grasped the drink inside.

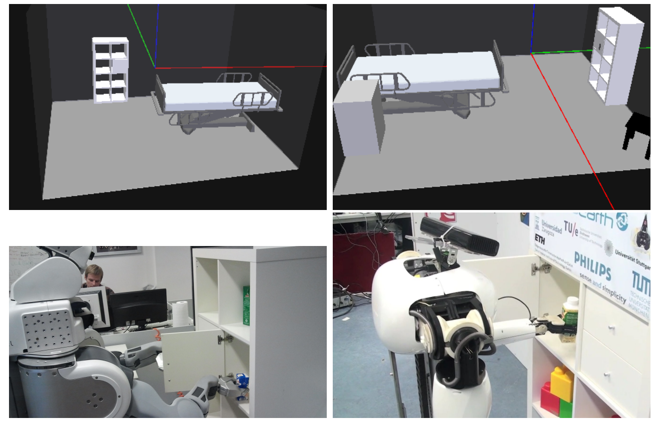

Figure 5: Top: Semantic environment maps of the two hospital rooms, downloaded from ROBOEARTH based on the address and room number. Bottom: PR2 and Amigo robots opening the cabinet and picking up the drink to be served.

7 Conclusions

In this paper, we discussed requirements on a formal language for representing robot knowledge with the intention of exchanging it, and presented our approach to realizing such a language. The language allows to describe actions, object recognition and articulation models, as well as semantic environment maps, and provides methods to reason about these pieces of information. Using the language, robots can autonomously decide if they lack any capabilities that are needed to perform an action, and if so, see whether they can download software to acquire them. ROBOEARTH thereby acts as a complement, not a substitute of existing control structures: If applicable information can be found, it will help a robot with its tasks – if not, its queries will fail and it will be in the same situation as without ROBOEARTH.

The language and the accompanying reasoning methods have successfully been used to exchange tasks, object models, and environment maps among heterogeneous mobile manipulation robots and to execute the abstractly described task. The experiments showed that the presented methods enable robots to download the information needed to perform a mobile manipulation task, including descriptions of the actions to perform, models of the objects to manipulate, and a description of the environment, from the ROBOEARTH knowledge base.

Acknowledgments

This work is supported in part by the EU FP7 Projects ROBOEARTH (grant number 248942) and *RoboHow* (grant number 288533).

References

[Beetz, Mösenlechner, and Tenorth, 2010] Beetz, Michael, Lorenz Mösenlechner, and Moritz Tenorth (2010). CRAM – A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1012–1017, Taipei, Taiwan.

[di Marco et al., 2012] di Marco, Daniel, Moritz Tenorth, Kai Häussermann, Oliver Zweigle, and Paul Levi (2012). Roboearth action recipe execution. In *12th International Conference on Intelligent Autonomous Systems*.

[Erol, Hendler, and Nau, 1994] Erol, K., J. Hendler, and D.S. Nau (1994). HTN planning: Complexity and expressivity. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 1123–1123. John Wiley & Sons LTD.

[Genesereth, Fikes, and others, 1992] Genesereth, M.R., R.E. Fikes, et al. (1992). Knowledge interchange format – version 3.0: Reference manual. Technical report, Computer Science Department, Stanford University Palo Alto, CA.

[Ghallab et al., 1998] Ghallab, M., A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins (1998). PDDL–the planning domain definition language. *AIPS-98 planning committee* .

[Khoo and Horswill, 2003] Khoo, A. and I. Horswill (2003). Grounding inference in distributed multi-robot environments. *Robotics and Autonomous Systems* 43(2): 121–132.

[Kira, 2010] Kira, Zsolt (2010). Communication and alignment of grounded symbolic knowledge among heterogeneous robots .

[Kunze, Roehm, and Beetz, 2011] Kunze, Lars, Tobias Roehm, and Michael Beetz (2011). Towards semantic robot description languages. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5589–5595, Shanghai, China.

[Lee, Hendler, and Lassila, 2001] Lee, T.B., J. Hendler, and O. Lassila (2001). The Semantic Web. *Scientific American* 284(5): 34–43.

[Lenat, 1995] Lenat, D.B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11): 33–38.

[Myers and Berry, 1998] Myers, Karen L. and Pauline M. Berry (1998). Workflow Management Systems: An AI Perspective. Technical report, Artificial Intelligence Center, SRI International. Menlo Park.

[Myers and Wilkins, 1997] Myers, Karen L. and David E. Wilkins (1997). The Act Formalism, Version 2.2. Technical report, Artificial Intelligence Center, SRI International. Menlo Park.

[Tenorth and Beetz, 2013] Tenorth, Moritz and Michael Beetz (2013). KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots. Part 1: The KnowRob System. *International Journal of Robotics Research (IJRR)* . Accepted for publication.

[Tenorth et al., 2011] Tenorth, Moritz, Ulrich Klank, Dejan Pangercic, and Michael Beetz (2011). Web-enabled Robots – Robots that Use the Web as an Information Resource. *Robotics & Automation Magazine* 18(2): 58–68.

[Tenorth et al., 2012] Tenorth, Moritz, Alexander Clifford Perzylo, Reinhard Lafrenz, and Michael Beetz (2012). The RoboEarth language: Representing and Exchanging Knowledge about Actions, Objects, and Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA. Best Cognitive Robotics Paper Award.

[Tenorth et al., 2013] Tenorth, Moritz, Alexander Clifford Perzylo, Reinhard Lafrenz, and Michael Beetz (2013). Representation and Exchange of Knowledge about Actions, Objects, and Environments in the RoboEarth Framework. *IEEE Transactions on Automation Science and Engineering (T-ASE)* . Accepted for publication.

[Waibel et al., 2011] Waibel, Markus, Michael Beetz, Raffaello D'Andrea, Rob Janssen, Moritz Tenorth, Javier Civera, Jos Elfring, Dorian Gálvez-López, Kai Häussermann, J.M.M. Montiel, Alexander Perzylo, Björn Schießle, Oliver Zweigle, and René Molengraft (2011). RoboEarth - A World Wide Web for Robots. *Robotics & Automation Magazine* 18(2): 69–82.