



**ICT Call 7
ROBOHOW.COG
FP7-ICT-288533**

Deliverable D1.3:

Evaluation criteria for robot knowledge processing systems



Januar 31st, 2014

Project acronym: ROBOHOW.COG
Project full title: Web-enabled and Experience-based Cognitive Robots that Learn Complex Everyday Manipulation Tasks

Work Package: WP 1
Document number: D1.3
Document title: Evaluation criteria for robot knowledge processing systems
Version: 1.0

Delivery date: Januar 31st, 2014
Nature: Report
Dissemination level: Public (PU)

Authors: Moritz Tenorth (UNIHB)
Daniel Nyga (UNIHB)
Michael Beetz (UNIHB)

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n^o288533 ROBOHOW.COG.

Contents

1	Evaluation Criteria	5
2	Example evaluation queries	6
2.1	Queries on object information and perception tasks (WP1, WP4)	6
2.2	Queries on task descriptions and recorded experience data (WP1, WP6)	7
2.3	Queries on observed human demonstrations (WP2)	8
2.4	Queries on motion models learned by observation or demonstration (WP5)	9
2.5	Queries on the constraint-based motion specifications (WP3)	9
3	Knowledge-based evaluation of complex robot behavior	10
4	Open knowledge bases and query interfaces as means for evaluation	11

Summary

This deliverable proposes evaluation criteria for robot knowledge processing systems. Evaluating the content of a knowledge base and the associated inference procedures is a challenging task. We propose to use a set of benchmark queries that cover a wide range of knowledge areas and inferences to characterize the strengths and limitations of a system.

In the following sections, we first discuss the general issue of evaluation robot knowledge bases and then present a catalog of evaluation queries that cover the full range of information that is produced or consumed by the different work packages in RoboHow. This is particularly important to ensure that the needs of all other components can be satisfied by the developed methods and representations. We then discuss the use of knowledge-based methods not only for the evaluation of the knowledge base itself, but for characterizing the performance of the overall robot system by using comprehensive logging techniques. In the last chapter, we argue that making the representational structures, the inference methods and the knowledge content available to the public in terms of documented open-source code and models improves the reproducibility of results and assessment of the system's performance.

Chapter 1

Evaluation Criteria

Evaluating the performance of a knowledge processing system that enables robots to competently perform manipulation tasks is a difficult task, and it is not sufficient to treat it as a black box: By just looking at the quality of the generated robot behavior, one would not be able to distinguish contributions made by the knowledge processing from other aspects. This is because many measurable performance factors (e.g. the time needed to complete a task, success rates for tasks, etc.) also depend on other factors such as the quality of sensors and actuators or the experience of the programmer.

The research focus of WP1 in RoboHow, however, is rather on the reasoning and knowledge representation problems that have to be solved in order to enable the robot to competently master a set of tasks. Since the plans in RoboHow are derived from vaguely formulated, underspecified instructions, the knowledge processing methods play a crucial role in identifying and filling information gaps. A question to be answered is thus which kinds of information can be provided and if the resulting plans are sufficient for successful execution in a real-world environment.

We therefore propose to measure the performance of the knowledge processing system to be developed as part of this work package in terms of its knowledge and information content and the range of queries it can answer. The information content of a robot control system can be assessed through query-based benchmarking. To this end, we will develop structured libraries of queries in a formalized representation language that require different kinds of cognitive capabilities.

When presenting the queries and their results, we will explain in detail which of them could successfully be answered, which ones could not, and if not, if this results from a technical or principled issue or only from a lack of coverage of the knowledge base. Since robots operating in the real world impose real-time constraints on the query-answering procedures, we will also give information on how fast the different kinds of queries can be answered.

Chapter 2

Example evaluation queries

The benchmarks for the RoboHow knowledge processing system can be specified by a set of queries the robot can answer that will be selected from the following list of candidates. The queries will be formulated in a Prolog-based query language, but are given here in plain English for better readability.

The following list is clearly no exhaustive compilation of all queries that may be interesting or relevant for an autonomous robot. Rather, it is intended to demonstrate the abilities of the system by listing example queries that stand representatively for a whole range of similar inference tasks. We have selected these queries by their expected usefulness for a robot and by the range of information sources and inference methods that are required for answering them.

Since the knowledge representation and inference methods will play a crucial role in RoboHow as a semantic integration layer for knowledge exchange among the modules, we have sorted the queries regarding the type of knowledge they cover and the work packages they interact with. While WP1 provides the representation in the knowledge base and the resulting inference capabilities, the information content covered by these queries (e.g. a concrete motion description) will be developed in the respective work packages.

Note that the queries have mainly been selected to showcase and benchmark the capabilities of the developed representations, inference methods and knowledge content. They therefore describe the developer's perspective and intend to make the internal processes transparent. Another perspective is that of a robotic agent that uses the system during operation to infer answers to queries that are needed for planning and performing its actions. Such a robotic agent may ask different queries and formulate them in a different way. While the latter perspective is also investigated in the RoboHow project, this deliverable focuses on the former one.

2.1 Queries on object information and perception tasks (WP1, WP4)

This group of queries is concerned with information about objects – either about general properties of object classes or concrete information about the objects that have been detected by the perception system.

- **Which tools are required for a task, including those that are not mentioned in the instructions?** This involves reasoning about the relation between actions and objects in the instructions as well as the inclusion of background knowledge about actions and their properties.
- **Consider a scene consisting of a pan, a spatula, and a bottle, which is the object that likely contains pancake mix and is to be used for a pouring action?** Selecting the right object requires reasoning about the properties of objects, their roles in actions, and the relation between stuff and objects.
- **Which part of an object is to be used for grasping, which one for e.g. opening a bottle?** Reasoning about this topic includes affordances and part-based object representations.
- **Explain how an object looks like, i.e. which parts it is composed of, where they are located and what they can be used for.** This requires the robot to have geometric, semantic and affordance knowledge about object parts.
- **Did the grasp of the object become unstable during the execution of the task?** This query requires information about perceptual events that have been registered (e.g. the instability of a grasp) and temporal reasoning for relating the event to concurrent tasks.
- **What have you been asked for when getting a perception task and what was your response?** Perception tasks are formulated in terms of (incomplete) object descriptions that are to be filled and grounded using the perception methods. The information gained by the perception procedure can be measured by the difference between the information in the request and the response.

2.2 Queries on task descriptions and recorded experience data (WP1, WP6)

The Web instructions are converted into a formal representation in the robot's knowledge base and are therefore available as information source. In addition, the robot records comprehensive log files during execution that represents instantiations of the abstract action templates described in the instructions.

- **Which actions is a task composed of, which properties does it have, which motions are part of the task?** Much of this information will come from importing the Web instructions and formally representing them in the knowledge base, while these instructions will have to be complemented by learned models of the corresponding action verbs.
- **Are any action steps missing in the instructions?** Based on qualitative process models (which are required to be available for the task at hand), the robot can reason about action consequences and processes in the environment and compare the predicted with the desired result. If they do not match, the instructions may lack important steps like switching on the stove before use.
- **Which robot components does an action depend on, and are they available on a given robot?** Using explicit representations of its own hardware components and capabilities

as well as dependencies of actions on them, the robot can reason about which components are needed and which of its parts could be used to perform which parts of the task.

- **For which of the motions do we have models (learned or programmed) that allow this motion to be executed?** Being aware of the models that exist allows the robot to reason about whether information is missing, if models have to be acquired, or which of the models is to be used in a particular situation.
- **How well did that action work in practice?** This kind of query is to be answered based on the logged execution data and characterizes the performance as well as failures that may have occurred.
- **How long did it take on average?** From multiple executions of a task, a robot can learn prediction models that describe for instance how long different actions take.
- **Which were common problems, and which were suitable solutions?** Looking at which actions fail in which way and which strategies for failure recovery worked well under which circumstances could allow the robot to focus attention on difficult parts or select appropriate ways how to react.
- **What was the force at the time when the spatula touched the pan for being pushed under pancake?** Since the robot does not only record symbolic plan events but also traces of lower-level sensory data, it can extract information like forces at particular points in time for analysis, diagnosis or failure detection.
- **What are possible effects of an action?** By performing the task at hand in a physical simulation and interpreting the results, the robot can come up with a set of possible outcomes of an action and classify them into desired and undesired effects.

2.3 Queries on observed human demonstrations (WP2)

The observations of human activities generated by WP2 are represented in the knowledge base and are therefore available for answering queries. They provide information about action instances as well as the configuration of objects during the task.

- **For which of the action classes that are part of a task do we have experience knowledge?** This experience knowledge can origin from either own task executions or observed demonstrations given by humans.
- **How did the hand and the objects move during the observed task?** This query reads information on the position and orientation of objects over time.
- **Which hand and which hand pose was used to grasp an object?** Besides the position of the hand, the system also has information about the articulation of the fingers.
- **What was the pose of the bottle at the end of the pouring action?** As the observations are time-indexed, the poses can be related to semantically described higher-level information.

2.4 Queries on motion models learned by observation or demonstration (WP5)

The motion models learned from observation are explicitly represented in the knowledge base as part of the `ROBOTEVAREP`. While some aspects of the models are not interpretable by symbolic reasoning (e.g. the numerical information in Gaussian Mixture Models), other aspects can be queried:

- **How many motion phases does the model consist of, and which kinds of models describe each phase?** Different kinds of models are available, and being able to explicitly query for their types allows to automatically set up the controllers.
- **Which quantities are controlled by this model?** The learned models can flexibly describe combinations of positions, velocities, forces, stiffnesses et cetera. The information about which of these quantities are described by which part of the model is stored explicitly and can for example be used to decide whether a model is suitable for a robot (e.g. one that does not have a force sensor).
- **Which is the object to be manipulated, and where on this object is the attractor for this motion?** Information about which object is to be used is needed to perform the action in the physical world.

2.5 Queries on the constraint-based motion specifications (WP3)

Using the constraint-based motion description formalism, robot actions can be described in terms of relations between parts of a tool and parts of the world. These relations can be of different types, imposing constraints on for example the distance, height or relative orientation of the tool- and world frame. These explicit motion descriptions are available for reasoning about different aspects of the motion:

- **Which types of constraints have been used to describe a motion?** The types of constraints used indicates which aspects of a motion are important and could give a first idea if conflicting constraints or freedom in some dimensions is to be expected.
- **Which parts of the objects are relevant for a particular movement?** Looking at the motion specifications, the system can read which object parts are important and make sure these have been reliably identified.
- **How well did a motion follow the constraints?** If the fulfillment of the motion constraints can be recorded, this could give information about how well a motion followed the specification, if this may have caused errors, or if performing the task slower would help to improve success.
- **How flexible is the given motion with respect to some relation between two object parts, e.g. the height above the pan?** The range of values that are allowed indicates how strict parts of the motion have to be followed. If the robot has problems with staying inside this interval, one could increase it; if the task is not successful, they may need to be narrowed.

Chapter 3

Knowledge-based evaluation of complex robot behavior

Besides the evaluation of the robot's knowledge processing system, also the overall evaluation of the generated functionality is a challenging task. Just measuring the overall task achievement does not give a clear and detailed enough picture of why tasks fail, which components lead to failures, or which kinds of individual failures result in a failure of the complete task. Recording more comprehensive evaluation data requires sophisticated logging mechanisms that do not only store the result of a task, but also which tasks have been performed with which parameters, which objects were manipulated, which percepts were made, which raw sensor data resulted in these percepts, which kinds of failures have occurred et cetera. Ideally, this logged data should enable a programmer (or, ultimately, the robot itself) to reconstruct the robot's belief state and the incoming information from the outer world at each time as accurately as possible.

We intend to use a knowledge-based framework for recording log data of robot tasks that is as comprehensive as possible and includes information such as the internal state of the robot's control program, its symbolic and geometric belief state, perception results, sensor data, and the robot's pose information. Since this data will be integrated into the robot's knowledge base, it will be available for queries. The representation will be based on the same language that is also used for describing the robot's run-time information about objects, the environment, the actions to be performed as well as its own hardware and software.

Using the knowledge base for this task has several advantages: The explicit representation of the semantics of the recorded data allows to categorize on the fly along different dimensions, e.g. to read "all perception tasks", "all actions for opening containers" or "all pick-up actions performed on cups". The taxonomy of failure classes allows to filter, select and categorize, while grouping similar kinds of problems. Knowledge about the task tree allows to reason about the relations of actions, the propagation of failures to super-actions, the success of different kinds of recovery strategies, etc.

Chapter 4

Open knowledge bases and query interfaces as means for evaluation

While we consider evaluation queries as the most suitable means for evaluating robot knowledge bases for publications, reports and presentations, they can only demonstrate parts of the capabilities of a system. Showing the full range of possible queries and the degree of variation that is allowed is difficult as part of a text. We are therefore currently investigating methods to create Web-based query and visualization tools that allow Web users to interact with the knowledge base. While still requiring users to have a certain proficiency in using the requires tools (e.g. Prolog as query language), this would allow them to better assess the knowledge content and to explore novel ways in which it can be used.

Creating a fully-functional Web interface poses several complex and challenging problems in areas such as robustness, scalability and security that are not part of RoboHow and that we therefore do not expect to solve during the project. We plan, however, to investigate the possibilities in the context of student projects and evaluate whether these techniques can help with the evaluation of robot knowledge bases.

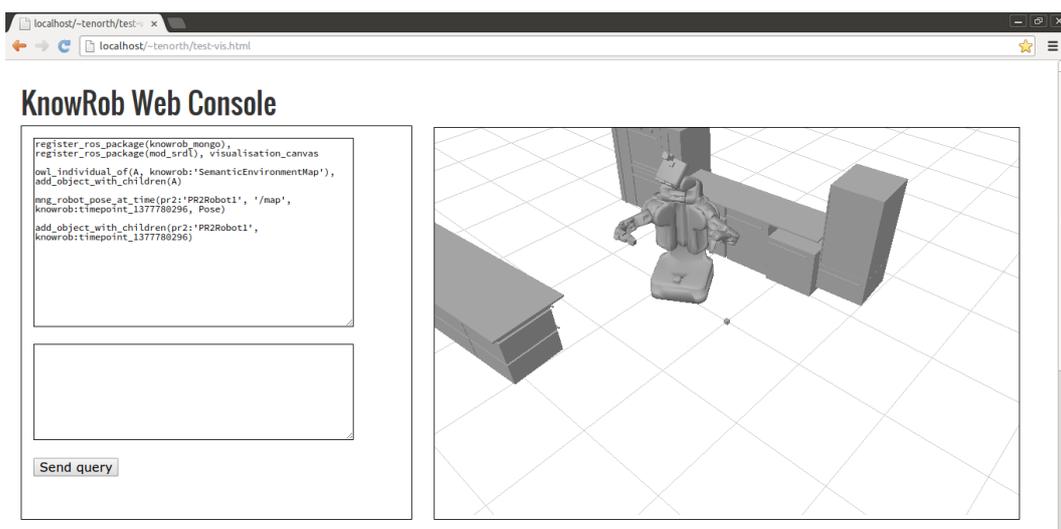


Figure 4.1: Experimental prototype of a Web console for interacting with the KnowRob knowledge base. Results of the queries can be visualized in the 3D canvas on the right-hand side.