Open Robotics Research Using Web-based Knowledge Services

Michael Beetz*, Daniel Beßler*, Jan Winkler*, Jan-Hendrik Worch*, Ferenc Bálint-Benczédi*, Georg Bartels*, Aude Billard#, Asil Kaan Bozcuoğlu*, Zhou Fang*, Nadia Figueroa# Andrei Haidu*, Hagen Langer*, Alexis Maldonado*, Ana Lucia Pais Ureche#, Moritz Tenorth**, Thiemo Wiedemeyer*

***Abstract*— In this paper we discuss how the combination of modern technologies in "big data" storage and management, knowledge representation and processing, cloud-based computation, and web technology can help the robotics community to establish and strengthen an open research discipline. We describe how we made the demonstrator of a EU project review openly available to the research community. Specifically, we recorded episodic memories with rich semantic annotations during a pizza preparation experiment in autonomous robot manipulation. Afterwards, we released them as an open knowledge base using the cloud- and web-based robot knowledge service OPENEASE. We discuss several ways on how this open data can be used to validate our experimental reports and to tackle novel challenging research problems.**

I. INTRODUCTION

One of the main barriers to accelerating the progress in autonomous robot manipulation is the complexity and heterogeneous nature of the tasks involved. To perform a challenging manipulation task such as making a pizza, a robot needs to combine of several complex capabilities. The robot has to recognize objects such as a ketchup bottle in the refrigerator and a spoon in a drawer. It has to monitor how the pizza deforms while it is rolling it and how the surface of the pizza is covered with tomato sauce when pouring sauce onto it. The robot has to be capable of autonomous mobile manipulation: it has to navigate to the fridge, open the door, and fetch the ketchup bottle. The robot also has to reason about how to perform actions. For example, it has to decide whether to use one or two hands for picking up the objects, where to grasp them, and how to hold them.

Only very few research groups can implement robot control systems that have the whole range of capabilities that are required for such manipulation tasks. Most work on small and isolated problem fragments, such as recognition or tracking of textured objects, generation of a control law for pulling a refrigerator open, detection of known objects based on their CAD models, grasp planning for 3D models, etcetera. Consequently, the solutions proposed are often unrealistic with respect to the requirements imposed on the control systems, the benchmarks are not representative for their targeted application tasks, and it is unclear how to transfer the solutions to other tasks, objects, and environments.

Modern technologies in "Big Data" storage and management, knowledge representation and processing, cloud-based computation technologies, and web technology now offer us the opportunity to improve this situation by realizing a new generation of software tools for open research in robotics. We can log the entire body of information that is relevant for achieving robot manipulation tasks including pose data, the images interpreted by the robot perception system, and other sensor and control signal streams, into "Big Data" databases and annotate them with semantic indexing structures that are automatically generated by the interpreter of the robot control system. These episodic memories can help the robotics systems assess what they were doing. The "episodic memories" enable the robots to answer queries about what they did, why they did it, how they did it, what they saw when they did it, and what happened when they did it. Providing a web-based graphical query interface to robots that "know what they were doing" is a powerful enabling technology for open robotics research.

These efforts of providing researchers with semantic access to the experimental and experience data of comprehensive autonomous robot manipulation experiments implements some aspects of Nielsen's vision of "Reinventing Discovery" [1], which suggests new ways of conducting research more effectively through the cooperation facilities provided by modern Internet technology. Inspiring blueprints for such web services that promote open research in other domains include the *Allen Human Brain Atlas* [2] and the *HapMap* project [3] which enables networked science in human genome research.

In this paper we explain how we use the cloud-based knowledge service OPENEASE in order to realize open robotics research. We report on an effort in which we made a comprehensive demonstration that was prepared for a yearly review of a large multi partner project, the EU FP7 project ROBOHOW, available for open research. We will describe how robot demonstrations are logged into hybrid symbolic/"Big Data" knowledge bases and how researchers and reviewers can access the knowledge bases in order to further analyze the experimental data of the experiments and build performance models of the respective system components. The experiment can be accessed and interactively worked with at open-ease.org under the section

*The author is with the Institute for Artificial Intelligence, Universität Bremen, Am Fallturm 1, 28359 Bremen, Germany.

#The author is with the Learning Algorithms and Systems Laboratory, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

**The author is with the Magazino GmbH, Landsberger Str. 234, 80687 München, Germany

experiments (`https://data.open-ease.org/`).

The ROBOHOW demonstration consists of a collection of sub-episodes with different types of manipulation actions, such as autonomous mobile fetch and place and rolling pizza dough. The manipulation actions were performed through hybrid symbolic/subsymbolic control systems with three methods for motion control: constraint- and optimization-based controllers, motions generated from learned dynamic systems, and motion trajectories obtained through motion planning. In addition, there are episodes in which humans train robots by demonstrating pouring actions in a virtual reality simulation. The logged episodes are semantically linked to a common concept ontology and accessible through knowledge formalized in the logic programming language Prolog. The common representation framework facilitates the combination of knowledge from different episodes.

The main contribution of this paper is twofold. First, we describe how a complex real-world robot demonstration can be converted into a knowledge base of a web-based knowledge service in order to facilitate open research in robotics. Second, we show how researchers can conduct open research based on the knowledge bases automatically constructed from the demonstration and the power of the visual inspection and analysis tools that are provided by OPENEASE. We believe that a community effort to make knowledge bases of complex robot experiments available for open research combined with the software tools provided by "Big Data" technology will have a substantial impact on advancing AI-based robotics technology.

The remainder of the paper is structured into three parts. We start with a description of the OPENEASE robot knowledge service and how experiments are transferred into OPENEASE knowledge bases. Then, we give a sketch of the pizza making experiment conducted at the ROBOHOW review and describe how the different sub-experiments were recorded, which information was stored, and which queries can be answered. We also provide an example of how the same infrastructure can support data from other sources on the same actions, in this case human demonstrations in a simulation-based environment. In the final section we outline how remote researchers can work with the experiments, reproduce results, refine the original analysis, and finally conduct their own research with the open data.

II. OPENEASE — A CLOUD- AND WEB-BASED KNOWLEDGE SERVICE

The web-based knowledge service OPENEASE [4] serves as the groundwork for our approach to open robotics research. OPENEASE is a remotely accessible knowledge processing service for robots and robotic researchers. OPENEASE can load knowledge bases represented in the KNOWROB [5] representation language and reason about them. OPENEASE provides the representational infrastructure to make inhomogeneous experience data from robots and human manipulation episodes semantically accessible, and is complemented by a suite of software tools that enable researchers to interpret, analyze, visualize, and learn from the experience data. Using OPENEASE users can retrieve the memorized experiences of manipulation episodes and ask queries regarding to what the robot saw, reasoned about, did, and how actions were performed and which effects occured.

KNOWROB knowledge bases describe relations and are defined by means of Prolog clauses of two types: facts and rules. Prolog rules have the form *Head :- Body* and are read as "Head is true if Body is true". The body of Prolog rules consists of a sequence of calls to predicates connected by logical conjunctions. The logical conjunctions are written as ",". Arguments of predicates can be either values or variables where each argument that starts with an uppercase letter denotes a variable. Facts have the form *hasComponent('robot-Boxy', 'camera1-left')*, which states that the robot with the name *'robot-Boxy'* has a component with the name *'camera1-left'*. Given this fact, one can ask: *?- hasComponent('robot-Boxy', 'camera1-left')*, which results in the answer *yes*.

We can also ask more complex queries such as which components does the robot *'robot-Boxy'* have that are some kind of sensor. This query would then return all sensors of the respective robot as shown below:

```
?- hasComponent('robot-Boxy', Comp),
   isType(Comp, 'Sensor').

 Comp = 'camera1-left' ;
 Comp = 'camera2-right' ;
 Comp = 'infraredCamera' ;
 Comp = 'swissRangerTOF' ;
 Comp = 'videreStereoOnChip' ;
 Comp = 'hokuyo-shoulder' ;
 Comp = 'hokuyo-rear' ;
 Comp = 'hokuyo-front'.
```

Clauses with bodies are called rules. An example of a rule is *sensor(Robot,Sensor) :- hasComponent(Robot, Comp), isType(Comp, 'Sensor')*. Given this rule we could get the same results as above by simply querying *?- sensor('robot-Boxy', Comp).*

More complex queries consist of a conjunction of predicate calls. Such a query is shown in Figure 1. The predicate call *task_goal(Task, Pattern)* binds tasks to the variable *Task* where the given pattern matches. In this example, the pattern matches all *grasp* actions that acted on objects with the type *cup*. *task_outcome(Task, success)* yields all successfully performed tasks. In conjunction with the previous predicate, the query yields all robot tasks that successfully grasped a cup. Each action has an associated start and end time which can be queried by the predicates *task_start(Task, Start)* and *task_end(Task, Start)*. Furthermore, it is possible to query for the end effector of the robot that was used for the grasping action by calling the predicate *task_used_gripper(Task, Gripper)*. Finally, the predicate *add_trajectory(Gripper, Start, End)* is called in order to visualize the trajectory of the endeffector during the grasp action.

Visualization of trajectories for particular actions that were performed during the experiment requires that poses of robot links are logged continuously. Contrary to other knowledge bases, KNOWROB does not perform the abstraction of this data into a symbolic representation before the knowledge

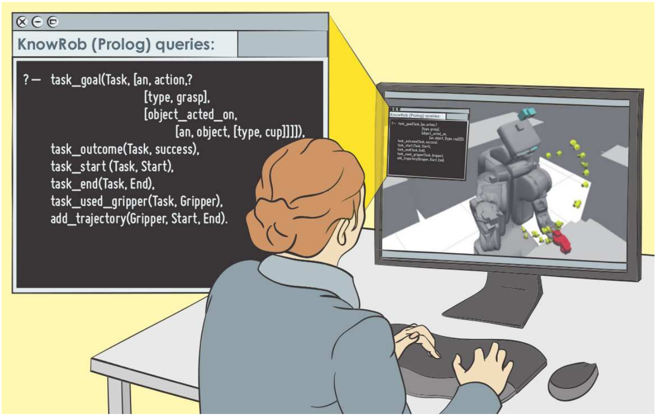


Fig. 1: KNOWROB Query interface.

is asserted into the knowledge base [5]. So called *virtual knowledge bases* are used to compute a symbolic representation of this continuous data on demand if required for answering a query. Those procedural attachments (called *Computable predicates*) allow to include external reasoning sources, such as the perception system, into the reasoning procedure. For example, this allows the definition of a predicate *graspable(O, R)* that yields true if the robot *R* is capable of grasping the object *O* at the most recently perceived object position. Internally, the predicate can be implemented based on the capabilities of the robot, the position, shape and other object properties as well as the kinematic structure of the robot by computing a solution to the underlying inverse kinematics problem. The poses of the robot links and internal data structures of the robot are logged continuously into an unstructured database that is indexed by timestamps. *Computable predicates* are used in order to read the poses and data structures for points in time that correspond to symbolically represented semantic events (such as grasping a cup). By using such external knowledge sources, the knowledge base is extended beyond the symbolically represented content.

The KNOWROB ontologies serve as a uniform symbolic representation for robotic experiments that are published on OPENEASE. Published experiments comprise different environments, everyday activities and robotic as well as human agents. Those experiments can be represented uniformly in the knowledge base and can be investigated with the same set of predefined Prolog predicates.

OPENEASE gives unprecedented access to comprehensive knowledge about leading-edge robotic experiments such as long term pick and place experiments performed by autonomous robots. This framework combines robot and human activity logs in a uniform structure based on the KNOWROB ontologies. The episodic memories include knowledge about when and why actions were performed, the outcome of the actions, the state of the agent during the experiment, perceived objects and internal states of the robot such as information about the control program. This comprehensive data is coupled with an expressive representation and a powerful query language that allows users to flexibly inspect

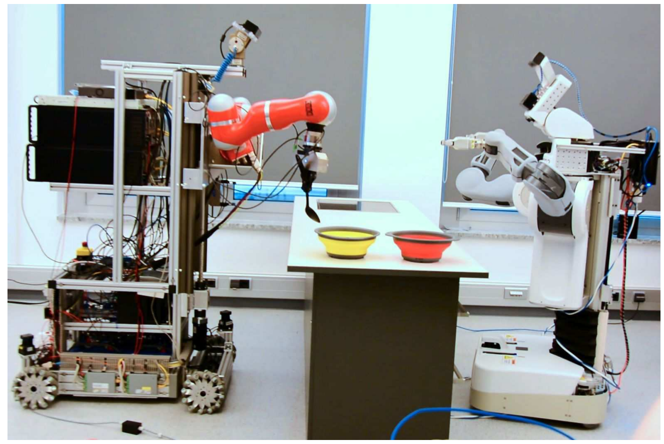


Fig. 3: Both robots (Boxy and Raphael) during the RoboHow pizza demonstration

and process the part of the data of interest. For example, it is possible to query for trajectories of the right arm of the robot where the robot was rolling a dough as a preparation step for making a pizza. In addition, the OPENEASE web service provides a powerful web visualization that allows to visualize parts of the experiment for particular timepoints that correspond to semantic actions such as rolling the dough (see Figure 2). A more complete introduction to the user interface of OPENEASE can be found in the paper [6].

III. THE ROBOHOW PIZZA EXPERIMENT

The RoboHow pizza experiment was shown live during the third-year review meeting of the ROBOHOW project, and consisted of three subactivities centered around preparing a pizza: (1) fetching and placing tools, ingredients and the finished pizza, (2) roll out the dough, and (3) placing the toppings.

These subactivities were done by two collaborating robots in our laboratory shown in Figure 3. The first one is Raphael, an off-the-shelf PR2 mobile manipulation platform from Willow Garage, extended to have a Microsoft Kinect-2 as its main perception device. The second one is Boxy, a robot designed in-house with two KUKA LWR-4+ manipulators, a holonomic platform using mecanum wheels, a movable torso, and two parallel-finger grippers.

In the experiment Raphael was in charge of fetching and bringing tools and ingredients to Boxy, who manipulated the pizza dough, and added the pizza ingredients. After the robots were finished, a person came into the scene, and placed the tray with the pizza into the oven for baking it.

In a second branch of the demonstration we have shown how skill acquisition for such sophisticated manipulation actions can be performed using virtual reality games: adding the ingredients was performed in a simulation-based experiment. Similar tools as used on the physical robots were used for creating and accessing the logs, illustrating how OPENEASE can be used to work with heterogeneous datasets.

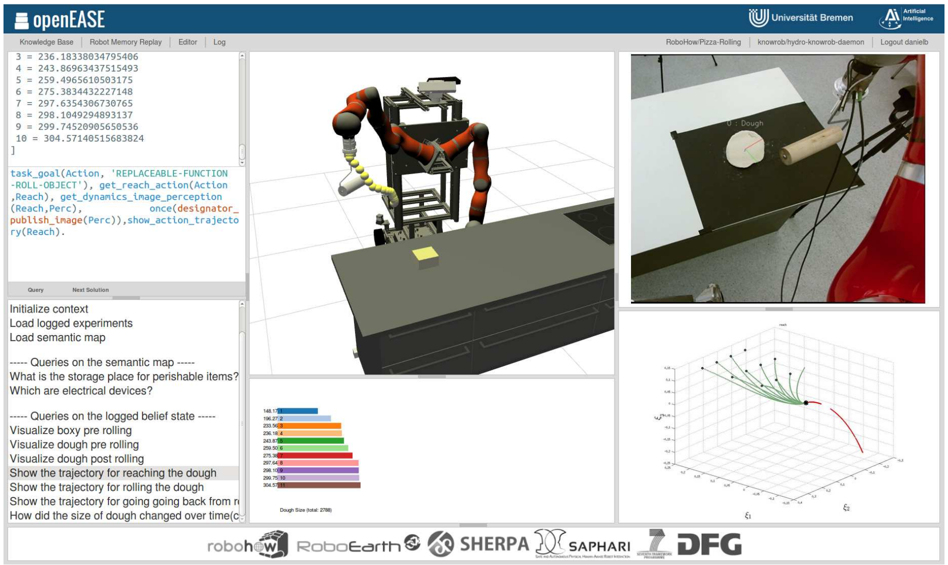


Fig. 2: Web interface of OPENEASE.

In the demonstration the subsystems for the different demonstration components logged episodic memories of the respective subactivities and stored them in the knowledge base of OPENEASE in order to make them publicly available for research usage. The data recorded in the episodic memories consists of symbolic log data from robot plans (performed actions, parameters, failures, results), and subsymbolic sensor data such as robot joint states and images taken by the robot's camera. The images taken during perception actions are semantically labelled by the perception system ROBOSHERLOCK [7] based on what was detected in that moment.

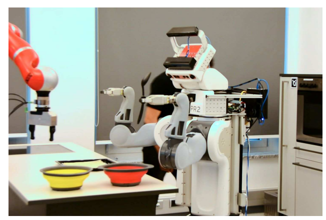
The following sections will briefly describe each part emphasizing the research aspects that should be prepared for open research.

A. Fetch and Place

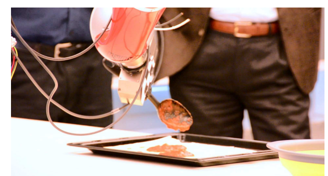
The research aspect of the fetch and place subactivity was to demonstrate that we could design generic plans for fetch and place that enable robot programmers to program the plans compactly by incorporating expressions such as: go to a place where you believe the object to be, position yourself such that you can see the object well, if needed make the object visible, for example by opening a drawer [8]. Figure 5 schematically depicts the fetch and place plan used in the experiment.

In the demonstration most objects required were placed at typical locations and not in reach of the robot performing the cooking task. Within the RoboHow pizza experiment, a PR2 robot was tasked with fetching different objects and bringing them to their appropriate places. These included getting a spoon from a drawer, a ketchup bottle from a fridge, and in the end transporting the tray with the pizza to the oven (see Figure 4).

In all these cases, the same robot plan for fetching and placing objects was used, only differing in parameterization. How to open and close drawers and fridges is dynamically inferred based on knowledge from the semantic environment model. For source or destination locations inside the drawer or the fridge, handling routines for containers are used that are parameterized by the semantic environment model.

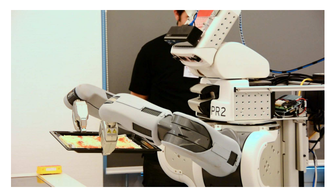


(a) Raphael bringing the spoon for Boxy.



(b) Boxy pouring tomato sauce.

(c) Boxy bringing cheese to the tray.



(d) Raphael bringing the tray to the oven.

Fig. 4: Robots executing parts of the pizza experiment

Using the collected episodic memories, remote researchers conducting research with these data can investigate the robot operation in more detail by stating queries in the web-based OPENEASE interface. Possible queries include *"What did the robot see when it has opened the container?"*, *"Which object types are perceived during the experiment?"*, *"Which path did the robot follow during the pick-and-place task?"* or *"What was the arm trajectory during the putdown action?"*.

B. Dough Rolling

The second component of the experiment was dough rolling. The goal of the robot in this experiment is to flatten out a ball of pizza dough using a wooden roller installed as an end-effector of Boxy as shown in Figure 6.

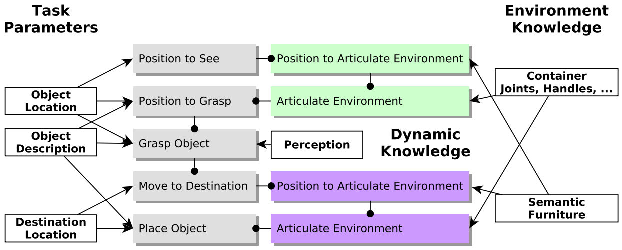


Fig. 5: Schematic depiction of the fetch and place task used in the RoboHow experiment. Environment articulation is only performed where necessary according to object and location parameters.

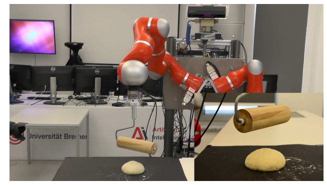
For dough rolling the control system employs a dynamical system that was learned through imitation learning. The first step involves automatic task segmentation and primitive learning. Here we apply a method proposed by Figuerora and Billard [9] which uses a Bayesian Non-Parametric approach for segmentation and clustering, namely an extension of the Beta Process Hidden Markov Model. This method is used to discover the unique primitive motions in a scale, translation and rotation invariant manner, without prior knowledge of the task or the number of primitives involved. For the dough rolling task this resulted in a sequence of three primitive motions (atomic actions): reach, roll and reach back.

We follow by extracting soft task constraints from the discovered primitive motions, as proposed by Pais et al. [10]. These constraints represent low-level knowledge about the task. They describe the variables of interest, the reference frame to be used and the proper stiffness modulation for each action in the given task. Applied to the dough rolling task this method determined a position controller as suitable for the reaching/reaching back phases and a hybrid force-position decomposition for the rolling action. Finally, we learn a set of action models for each primitive motion.

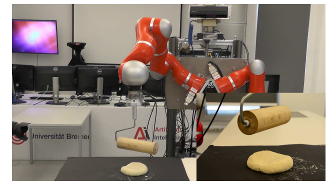
In addition, we extracted high-level knowledge about the task in the form of a success metric. The rolling task presumed multiple iterations of the sequence of atomic actions of reaching, rolling, reaching back. The goal was to obtain a round dough, of a given area. Which is correlated to properly positioning the starting and ending points of the rolling action. Since each rolling deformed the dough in the desired direction, at each iteration we positioned the attractors for the reaching and rolling along the small axis (second principal component) of a fitted ellipse on the dough.

The high-level executive is again programmed as a generic plan that automatically logs episodic memories that include the dynamic system model used, the trajectory that was followed, and the forces that were applied, compared to the original models.

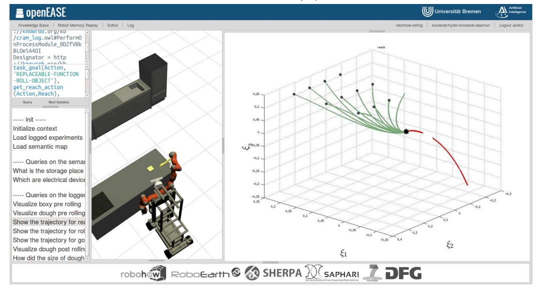
Using the episodic memories, we can infer knowledge such as how the size of the pizza dough is changed after each movement or how the arm trajectory is while reaching the pizza dough.



(a) Before the first movement



(b) After the first movement



(c) Visualization of pizza rolling in OPENEASE

Fig. 6: Dough rolling experiment

C. Preparing the Pizza

Preparing the pizza consisted of taking tomato sauce out of a bowl using a spoon, placing it on the flattened dough, then spreading it. Afterwards shredded cheese should be taken from the spoon and placed over the dough with the tomato sauce. Research aspects that are important in this part of the demonstration are the use of tools that are autonomously grasped in order to perform the manipulation task, the spreading action which is first executed with pushing the spoon into the tomato sauce and then distributing the sauce over the pizza. For the cheese spreading the robot has to scoop out cheese from a bowl and select the position over the pizza from where to drop the cheese. The position of the spoon, the bowls, and the tray with the pizza dough were perceived using the ROBOSHERLOCK system.

Raphael then picked up the tray with both hands and transported it over to the oven lid. It carefully placed the pizza tray onto the lid and backed off, telling the Boxy robot that it finished its task.

For the final stage of the experiment, Boxy started perceiving humans in the scene, tracking a human that went to the oven, put the pizza into the oven, closed the lid, and switched the oven on. After the human left the scene, Boxy informed Raphael about the task being fully completed. The experiment concluded with this action.

D. Preparing Virtual Pizzas

The ROBOHOW experiment also includes episodes where the robots learn capabilities needed to perform complex manipulation tasks from humans demonstrating the tasks in virtual reality environments with an integrated physics simulation. In particular, using virtual games, we set up a scene where we asked players to create a pizza using the following tools and ingredients: a container with tomato sauce, a spoon, two bowls with toppings, and a prepared pizza dough. The player had to pour the tomato sauce from the container onto the dough and spread the sauce as uniformly as possible using the spoon. Afterwards, using the same spoon, they scooped toppings from the given topping bowls onto the dough. The interaction with the virtual environment was done by tracking the users' hands movements and mapping it on a simulated robotic hand, see Fig. 9 (left).

During the gameplay all the simulation data was logged and post-processed to the OPENEASE system using modules similar to the logging used in robot control. Fig. 9 (right) shows an example of querying trajectories from the recorded data. Here the entire trajectory of pouring sauce onto the dough is shown using yellow markers. In addition, the end pose of the hand at the time of releasing the container is visualized, along with other objects that were part of the simulation.

IV. Conducting Open Research

Now, that we have described the experiment knowledge bases that we made available, the knowledge they contain, and some representative queries that can be answered for them let us turn to the topic of discussing how researchers can use the knowledge bases and perform open research.

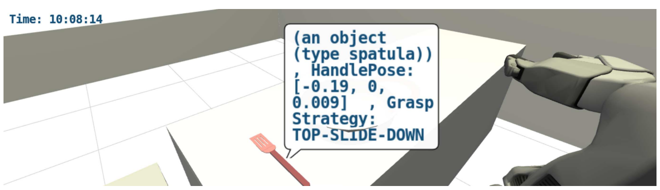


Fig. 7: Frames of a *grasping-a-spatula* video (the robot's view) generated using the automated video generation tool in OPENEASE. The descriptions of *spatula* are denoted as a speech bubble.

First, let us consider how OPENEASE improves the reproducibility of experimental results. To this end, OPENEASE provides various tools for experiment summarization. One of the tools enables researchers to automatically generate videos of the experiment knowledge base. To generate such a video, researchers start with the earliest time instance of an experiment and query the knowledge base for specific information regarding the situation at this time instance. Researcher then can visualize the desired information using speech bubbles and hud texts, advance to the next time instance, and repeat (Figure 7). To generate the individual images of the video users can state arbitrary and complex OPENEASE queries. This functionality is provided by a special interactive web interface that lets researchers visualize the video generation process and parameterize the video generation in various ways such as setting the camera position or the timestamp step size from one image to the next one. For example, researchers can place the point of view to get a good overview of the experiment or one could ask the video generation tool to automatically place the point of view to the current pose of the robot's camera to generate the robot's view on the experiment (Figure 7).

Other experiment summarization tools include the generation of an infographic-like visualization of the experiment, which summarizes the experiment into a single or a small sequence of infographics (Figure 8). These infographics include a 2D sketch of the environment, the robot trajectory and highlighted positions of the robot while accomplishing important tasks. In addition, OPENEASE provides the usual tools for visualizing statistics using bar charts and pie diagrams that allow researchers to visualize failure distributions in the experiment or metering the resources needed by different information processing steps (Figure 8 (right)).

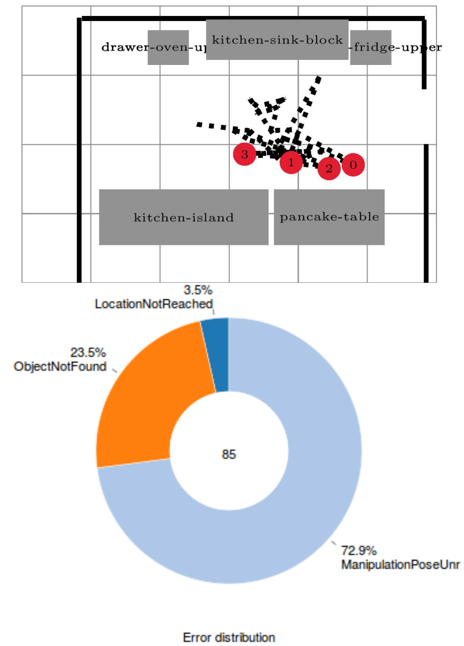


Fig. 8: An infographic in which the robotic agent performed multiple pick-and-place actions (top). The locations where it performed a *PUTDOWN* action are highlighted. An error distribution statistics of a pick-and-place experiment (bottom).

Besides experiment summarization OPENEASE also allows to have a more thorough look at the experiment conditions. For example, using the OPENEASE query interface researchers can ask which objects the robot picked up and placed somewhere else or could inspect the images that were used in order to recognize, localize, and reconstruct the perceived objects for manipulation.

But besides the assessment of the conditions under which actions were performed researchers can also analyze the employed methods more carefully in order to understand their strengths and weaknesses. A simple means is to learn classifiers that predict whether methods succeed or fail based on their retrieved results in the respective experiment. For example, by learning decision trees for predicting the success of object recognition tasks based on a large variety of context condition that are recorded by the system. The rules that are learned could then state findings such as that a object recognition method can robustly detect and recognize objects unless they are located in the refrigerator, which would suggest to further analyze the methods under the respective lighting conditions and certain characteristic views and types of occlusions that are typical for refrigerators.

OPENEASE also supports the investigation of perceptual

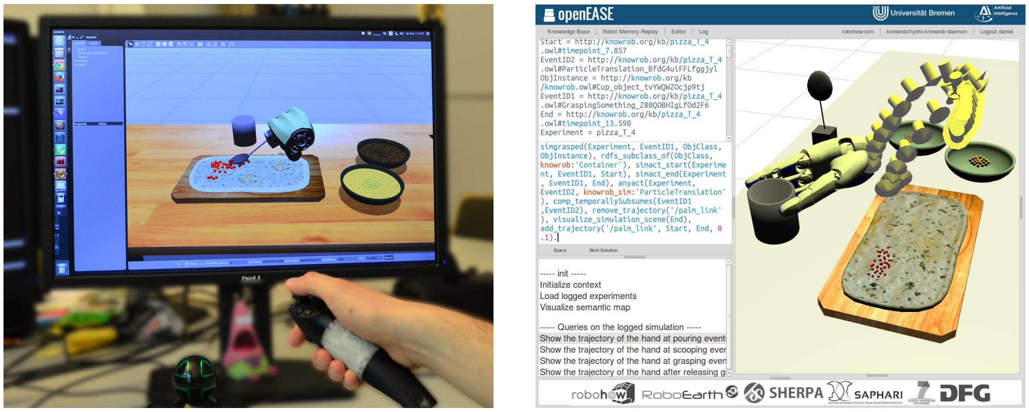


Fig. 9: Making pizza in Virtual Games. The user interaction with the game (left) and the corresponding pouring trajectory in OPENEASE (right).

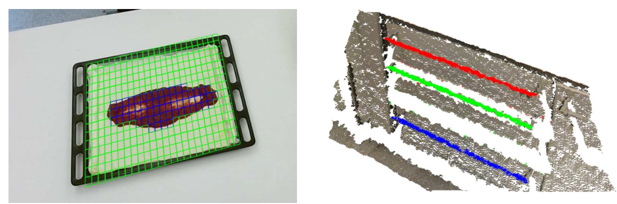
capabilities of robotic agents, in particular when used in combination with the perception framework ROBOSHERLOCK. ROBOSHERLOCK [7] is a knowledge enabled perception system, that is based on the Unstructured Information Management (UIM) paradigm [11]. It takes advantage of the ensembles of experts approach and is equipped with a library of state-of-the-art perception methods which can be used as experts. In ROBOSHERLOCK perception tasks are solved by reasoning about the perception task that is formulated by the higher level planning and adapting perception pipelines during run-time based on the results of this. This is made possible through two defining concepts of the framework: (1) the use of knowledge available to the robot operating in environments over long periods of time and the use of this knowledge for simplifying perception tasks (e.g. localization in a semantic map), and (2) the perceptual capabilities of the framework are modeled so that it can autonomously make decisions about which perception algorithm to run in order to successfully accomplish the given task.

OPENEASE has been extended so that researchers can send perception queries to ROBOSHERLOCK which internally configure the perception routines that are applied to captured images and run these perception routines through OPENEASE. Thus, researchers can select images captured during the experiments and process them again using ROBOSHERLOCK. To this end, OPENEASE provides the option to generate new pipelines or modify the exisiting pipeline, using Prolog queries. Reasoning about the components of a pipeline happens internally in ROBOSHERLOCK but users can see these results by formulating *build_pipeline* queries where the parameters are either a visual description or the instance of an object expressed through KNOWROB entities. This allows for comparing alternative perception algorithms on image data that was captured during real robotic experiments instead of having image data from static camera setups. An example of two queries is depicted in Figure 10, where specialized perception pipelines are internally generated in order to find drawer handles and the tomato sauce on the pizza dough.

OPENEASE also allows for creating benchmark sets for perception tasks. For example, researchers can query for

```
(detect
 (an ingredient
  (type tomatosauce)
   (on (an object (type pizza)
    (on (an object (type tray)))))))
```

```
(detect
 (an object
  (in kitchen)
  (color grey)
  (type handle)))
```



(a) Sauce on Pizza

(b) Detect Handles

Fig. 10: Queries and results of a ROBOSHERLOCK pipeline

images that contain three objects that are at a distance of at least 3 meters and at most 4 meters to the robot. This enables researchers to test their algorithms on data which was not only recorded for a specific purpose.

As mentioned above, OPENEASE can also be used to find and inspect weaknesses of an algorithm. Consider a robot that was supposed to pick up a red cup, but mistakenly picked up a plate. Researchers can query for the situation when the robot picked up the plate which it thought to be a red cup. Researchers will get the information related to that scene, like region of interest of the object and the entire image, and can re-run the perception pipeline to figure out why the plate was mistakenly recognized as a cup or try other perception routines on the same scene in order to find an algorithm which is more suitable under the given circumstances.

V. RELATED WORK

For interfacing web applications with cutting-edge robotic soft-/middleware, Alexander et al. offer *Robot Web Tools* [12] which make ROS accessible through JavaScript and HTML5. The *Robot Web Tools* are used by our OPENEASE web service. In the context of web-based knowledge processing systems, Wielemaker et al. propose a system called ClioPatria [13]. Using ClioPatria, users can send queries to a static knowledge base. Instead of allowing users to write Prolog queries they are using an SQL-like query language SPARQL that is internally mapped to Prolog queries. Saxena et al. introduce a knowledge engine called *RoboBrain* [14]. *RoboBrain* incorporates multiple data modalities including symbols, natural language, haptic senses, robot trajectories and visual features where the knowledge is acquired from sources such as physical interactions, web knowledge bases and learned representations. Goldberg et al. reviews ways that cloud computing in robotics has potential to improve performance [15]. According to the authors the five ways are: Providing access to global image, map, and object data libraries; Parallel computing on demand for demanding tasks; Sharing of outcomes, trajectories, and dynamic control policies; Sharing of source code, data, and designs for programming; On-demand human guidance for exception handling and error recovery.

VI. Conclusion and Future Work

Our RoboHow Pizza Making Experiment features the whole chain of robot control: Using high-level action plans, an autonomous robot gathers necessary ingredients from a kitchen environment out of drawers and fridges, manipulating its environment and perceiving objects in a semantic fashion, and brings them to a second robot that prepares a pizza. Using force-torque based low-level manipulator control, pizza dough is rolled out while its shape is monitored and compared to a desired size. The same semantic perception system that monitors the dough shape inspects where sauce is missing on a flat pizza, and the robot spreads the sauce accordingly. A human monitoring component then tracks a human that switches on the oven and completes the task.

The whole experiment is logged into a "big data" database using our extensive logging system, and is made ready for analysis within the openEASE system. With the openEASE system researchers can store extensive and comprehensive episodic memories of robot experiments in a cloud-based storage system for inspection by humans and robots alike. Using its open architecture, new experiment types can be added to the database and queried for information using a flexible logic programming interface. openEASE allows access to logged experiments in a very detailed manner, opening them up for in-depth analysis for researchers world-wide without the need to conduct the experiments themselves.

As shown, openEASE comes along with powerful tools for analyzing, re-running and summarizing experiments performed on real robots and in simulation. Using openEASE researchers have the opportunity to access semantically annotated data from real world scenarios. This is a major advantage compared to making only raw data public. Researchers can easily compare algorithms at specific situations, learn decision trees for predicting the success of grasping actions during pick-and-place experiments or the success of object recognition tasks based on context conditions, figure out weaknesses of a system, and reproduce experiments due to the fully semantically labeled data.

We plan to provide an easy-to-use interface for integrating new types of experiments and constantly add new analysis modules to the openEASE system. Our growing database of fundamentally different datasets shares the same logic programming interface for each type and will be extended beyond pure robot experiments, covering more demonstrations by humans whose intentions are not inspectable from the outside.

Acknowledgements

This work is supported by the EU FP7 project RoboHow (Grant Agreement Number 288533).

References

[1] M. Nielsen, *Reinventing discovery: The New Era of Networked Science*. Princeton University Press, 2012.

[2] M. J. Hawrylycz, E. S. Lein, A. L. Guillozet-Bongaarts, E. H. Shen, L. Ng, J. A. Miller, L. N. van de Lagemaat, K. A. Smith, A. Ebbert, Z. L. Riley, *et al.*, "An anatomically comprehensive atlas of the adult human brain transcriptome," *Nature*, vol. 489, no. 7416, pp. 391–399, 2012.

[3] R. A. Gibbs, J. W. Belmont, P. Hardenbol, T. D. Willis, F. Yu, H. Yang, L.-Y. Ch'ang, W. Huang, B. Liu, Y. Shen, *et al.*, "The international hapmap project," *Nature*, vol. 426, no. 6968, pp. 789–796, 2003.

[4] M. Beetz, M. Tenorth, and J. Winkler, "Open-EASE – a knowledge processing service for robots and robotics/ai researchers," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, finalist for the Best Cognitive Robotics Paper Award.

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

[6] G. Bartels, D. Beßler, M. Beetz, M. Tenorth, and J. Winkler, "How to use openease: An online knowledge processing system for robots and robotics researchers," in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Istanbul, Turkey, 2015.

[7] M. Beetz, F. Balint-Benczedi, N. Blodow, D. Nyga, T. Wiedemeyer, and Z.-C. Marton, "RoboSherlock: Unstructured Information Processing for Robot Perception," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, best Service Robotics Paper Award.

[8] M. Beetz, D. Jain, L. Mösenlechner, M. Tenorth, L. Kunze, N. Blodow, and D. Pangercic, "Cognition-enabled autonomous robot control for the realization of home chore task intelligence," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2454–2471, 2012.

[9] N. Figueroa and A. Billard, "Discovering hierarchical structure in heterogenous and sequential task demonstrations," *Journal of Machine Learning Research*, 2015, in preparation.

[10] A. L. Pais, K. Umezawa, Y. Nakamura, and A. Billard, "Task parametrization using continous constraints extracted from human demonstrations," *IEEE Transactions on Robotics (TR-O)*, 2015, submitted.

[11] D. Ferrucci and A. Lally, "UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment," *Natural Language Engineering*, vol. 10, no. 3-4, pp. 327–348, 2004.

[12] B. Alexander, K. Hsiao, C. Jenkins, B. Suay, and R. Toris, "Robot web tools [ros topics]," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 4, pp. 20–23, 2012.

[13] J. Wielemaker, W. Beek, M. Hildebrand, and J. van Ossenbruggen, "Cliopatria: A logical programming infrastructure for the semantic web," *Semantic Web Journal*, 2015, under review.

[14] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula, "Robo brain: Large-scale knowledge engine for robots," arXiv, Tech. Rep., 2014, http://arxiv.org/abs/1412.0691.

[15] 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. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-5.html