OPEN-EASE —
A Knowledge Processing Service for Robots and Robotics/AI Researchers

Michael Beetz[1], Moritz Tenorth[1] and Jan Winkler
{ beetz, tenorth, winkler }@cs.uni-bremen.de

Abstract— **Making future autonomous robots capable of accomplishing human-scale manipulation tasks requires us to equip them with knowledge and reasoning mechanisms. We propose OPEN-EASE, a remote knowledge representation and processing service that aims at facilitating these capabilities. OPEN-EASE gives its users unprecedented access to the knowledge of leading-edge autonomous robotic agents. It also 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 and robots to interpret, analyze, visualize, and learn from the experience data. Using OPEN-EASE users can retrieve the memorized experiences of manipulation episodes and ask queries regarding to what the robot saw, reasoned, and did as well as how the robot did it, why, and what effects it caused.**

I. INTRODUCTION

Within the next years autonomous mobile manipulation robots will be increasingly requested to accomplish human-scale manipulation tasks: autonomous warehouse robots will be asked to fetch the items on a given order list and pack them into a box, and household robots will be tasked to clean the table and unload the dishwasher [1], [2]. It is the nature of such human-scale manipulation tasks that they are incompletely specified and that they require reasoning from background knowledge to be accomplished successfully. Warehouse robots have to infer how to grasp, handle, and place the objects they are to collect and pack. They also have to reason about whether to pad the items, wrap them, and possibly even about how the center of mass of the box might change by packing it in different ways. When cleaning a table, the robot has to reason about the state of objects, whether they are clean or dirty, filled or empty, and handle them accordingly.

Different experimental autonomous robot control systems have been proposed that enable robots to accomplish such tasks by employing knowledge- and reasoning-enabled control, e.g. [3], [4], [5], [6]. However, equipping autonomous robots with comprehensive knowledge and the corresponding reasoning capabilities is a difficult and tedious programming task that might require proficiency in AI reasoning methods and non-standard AI programming languages. For teams not having a background in this field, the barriers for equipping their robots with "intelligent" problem-solving capabilities are often very high.

[1] Author names in alphabetical order. All authors are with the Institute for Artificial Intelligence and the TZI (Center for Computing Technologies), University of Bremen, Germany.

We propose OPEN-EASE[1], a remote knowledge representation and processing service that aims at facilitating the use of Artificial Intelligence technology for equipping robots with knowledge and reasoning capabilities. OPEN-EASE provides its users with unprecedented access to the knowledge of leading-edge autonomous robotic agents performing human-scale manipulation tasks. It includes

1) knowledge about the robot's hardware, its capabilities, its environment and the objects it manipulates;
2) memorized experiences of manipulation episodes that allow to reason about what the robot saw, reasoned, and did, how it did that, why, and what effects it caused; and
3) knowledge obtained from training episodes in which humans demonstrate skills that the robot can learn from.

This information can be retrieved by queries formulated in PROLOG, a general-purpose logic programming language. These queries can be sent either by humans via a web-based graphical interface, or by robots that access OPEN-EASE via a webservice API. This way, they can query and use OPEN-EASE's background knowledge to provide semantic meaning to their sensor data and to the data structures they use for control purposes. We plan to extend the system to let researchers and robots upload their own data structures and execution log files, to declare their elements as PROLOG rules, and thereby to convert their data files into virtual OPEN-EASE knowledge bases.

OPEN-EASE can also be viewed as a means for promoting open research in the domain of AI-enabled autonomous robot manipulation. After entering the website *http://www.open-ease.org*, researchers have complete access to comprehensive data sets of robots performing fetch-and-carry tasks and to human demonstrations of some of these tasks. A standardized semantic retrieval language provides full access to all data and enables researchers to combine individual sources of information. Sophisticated software tools enable researchers to visualize and analyze data through the web-based interface. This way, researchers in machine learning will be able to create realistic and highly relevant robot learning problems. Researchers in computer vision will be able to turn real perception tasks and the corresponding sensor data into benchmark problems.

Our efforts in developing OPEN-EASE and making it publicly available can be considered to be in the spirit of Nielsen's vision of "Reinventing Discovery" [7], which promotes new ways of conducting research more effectively

[1] EASE is the abbreviation of Everyday Activity Science and Engineering.

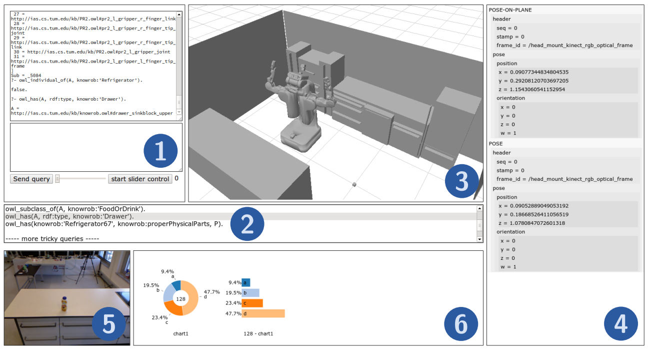

Fig. 1. Web interface of OPEN-EASE.

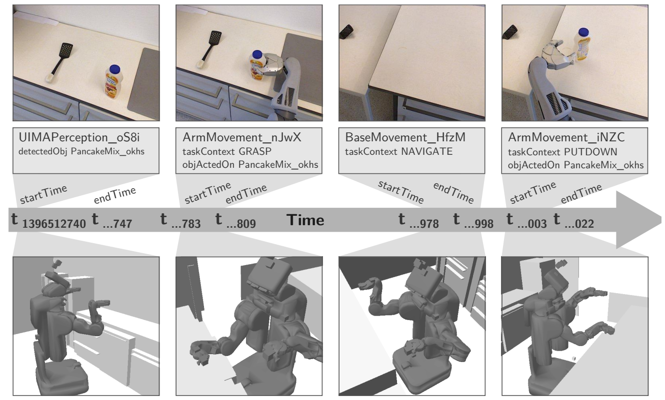

Fig. 2. Time interval view of the log data of a grasping action.

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* [8] and the *HapMap* project [9] which enables networked science in human genome research.

The scientific and technical contributions of the paper are the (1) comprehensiveness with which real execution data of modern autonomous manipulation robots is logged, stored and made openly accessible to the research community; (2) the representational infrastructure that is provided to make very inhomogeneous experience data from different robots and even human manipulation episodes semantically accessible in a uniform and standardized concept vocabulary; and (3) a suite of software tools that enable researchers and robots to interpret, analyze, visualize, and learn from the experience data.

The remainder of the paper is organized as follows. We start with a description of OPEN-EASE from a user perspective. Then, we give an overview of its functional components, explain its implementation in detail, and present some exemplary use cases. We finish with outlining some projected applications, with a discussion of related work, and our conclusions.

II. A GLIMPSE AT OPEN-EASE

To the human user, OPEN-EASE presents itself through the web-based interface depicted in Figure 1. The web interface includes panes with different purposes. The *Prolog interaction pane* (1) allows the user to type Prolog queries and commands and to see the answers to these queries. A list of prepared queries with English translation is provided in the *query list pane* (2). The *3D display pane* (3) can visualize the robot and its environment. Other information such as trajectories, robot and object poses can be added and highlighted. The *belief pane* (3) enables the user to inspect the internal data structures of the robot's beliefs including object, action, and location descriptions used by the robot. Finally, there is the *image pane* (4) for displaying images captured by the robot's camera, and the *visual analytics pane* (5) which can visualize statistical data as bar charts and pie charts.

The conceptual view that OPEN-EASE imposes on the log data of manipulation activity episodes is that of a first-order time interval logic [10], [11], [12] as shown in Figure 2, sometimes also referred to as a "chronicle representation" [13]. *Time intervals* are specified through the *time points* at their start and end. These are linked to the corresponding Unix time stamps, which allows synchronization with other recorded data such as captured images and robot poses. *Events*, such as a reaching motion, *occur* over time intervals. Actions are considered as events that are caused by an agent to achieve some goal. *Occasions*, representing for instance the state of an object being at some location, *hold* over time intervals. Instantaneous events and continuous states occur at a *time instant* t_i, which is a time interval with the duration 0. As an example, consider the time chronicle representation of a fetch-and-place task depicted in Figure 2. The figure includes the events that occur during the episode that are asserted in the representation language through a set of facts including the following ones:

```
occurs('ArmMovement_nJwX', [t6,t8]).
task_type('ArmMovement_nJwX', 'Grasping').
object_acted_on('ArmMovement_nJwX', 'PancakeMix_okhs').
category('PancakeMix_okhs', 'PancakeMix').
belief_at(robot('pr2_base','Pose_423'),t7).
occurs('UIMAPerception_oS8i', t2).
category('UIMAPerception_oS8i', 'ObjectDetected').
...
```

This representation allows us to ask sophisticated queries that combine information from these logical facts with continuous and geometric aspects such as robot poses in three-dimensional space. We can for instance query for a task *Tsk* with the goal of grasping an object of type *cup*, and retrieve the *Pose* of the robot in terms of global map coordinates *'/map'* at the end time point *End* of the grasping action. This query is answered based on the logged experience data; its result is depicted in Figure 3 (left).

```
?- task_goal(Tsk, [an, action,
                   [type,grasp],
                   [object_acted_on, [an, object,
                                      [type,cup]]]]),
   task_end(Tsk,End),
   robot_pose_at_time('PR2', '/map', End, Pose).
```

In addition to queries for individual time points, we can also retrieve trajectories of arbitrary parts of the robot while performing an action. That is, we first retrieve the time interval between *St* and *End*, which denote the beginning

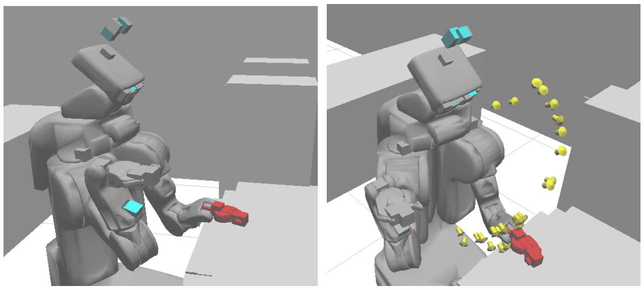

Fig. 3. Visual results of queries on logged robot experiences.

and the end of the grasping action, read which gripper was used for that action from the log data, and add the trajectory between these times to the visualization:

```
?- task_goal(Tsk, [an, action [type,grasp]]),
   task_outcome(Tsk,success),
   task_start(Act,St),
   task_end(Act,End),
   task_used_gripper(Act, Grp),
   add_trajectory(Grp,St,End).
```

Queries are not limited to the logged experience data, but may also include general background knowledge. The following two queries use the robot model to highlight components connected to the left arm (in red) and all cameras of the robot (in blue), respectively. The results are also shown in the figures above.

```
?- sub_component(pr2:pr2_left_arm , Sub),
   highlight_object(Sub).

?- owl_individual_of(Cam, srdl2:'Camera'),
   highlight_object(Cam).
```

III. Overview of Open-EASE

Open-EASE can be considered as a huge, remotely accessible knowledge service that consists of

1) ***a big-data database*** storing comprehensive data about episodes in which humans and robots perform complex manipulation tasks;
2) ***an ontology***, i.e. an encyclopedic knowledge base, that provides a conceptual model of manipulation activities;
3) and ***software tools*** for querying, visualizing, and analyzing the manipulation task episodes.

A. Databases of Manipulation Episodes

The data provided by Open-EASE comprises (∘) "raw" sensor data and the results of their interpretation by the robot, (∘) logged robot behavior including pose data, (∘) the robots' plans and their interpretation, (∘) a structured, semantically labeled environment model, and (∘) objects and their poses in the scene. Logged plan interpretation data, the environment model and object detections are represented in the Web Ontology Language OWL [14]. This representation can be loaded into the knowledge base and is available for reasoning using temporal logics as described in the next section. Sensor data and robot pose data, however, are usually of much higher volume, and storing them in OWL would lead to significant overhead. These kinds of data are therefore stored in MongoDB [15], an efficient schema-less, high-volume

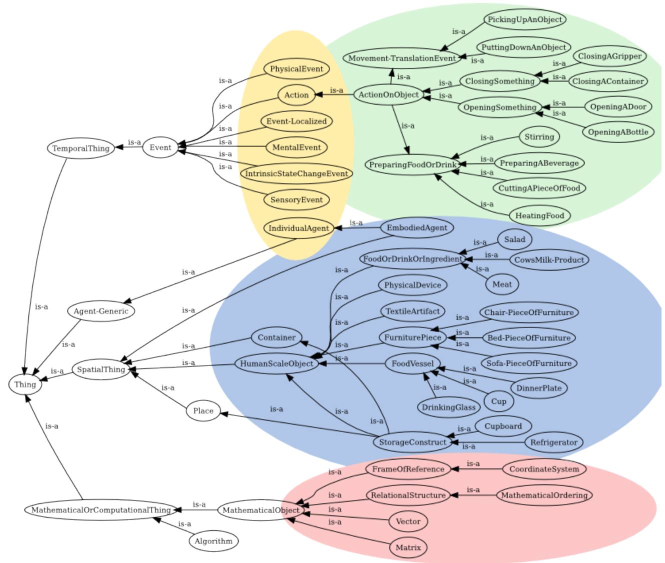

Fig. 5. Upper concept taxonomy used in Open-EASE.

database. The data can be accessed from the knowledge base by special predicates that, using procedural attachments, transparently load the required information and relate it to the semantic model. This approach enables Open-EASE to reconstruct the state of the robot including its pose and the computational state of the plan interpretation at any time. In addition, all images that have been used for perception tasks are stored, together with the results that the perception algorithms computed and the poses of all objects that are relevant for the manipulation tasks.

B. Open-EASE *Concept Vocabulary*

Open-EASE provides standardized semantic access to the database described in the previous subsection. The approach combines an ontology, which defines a conceptual model of manipulation activities, with a set of query predicates for reasoning about this conceptual model. The KnowRob ontology [16] that is used by Open-EASE and depicted in Figure 5 defines events and temporal things, actions, spatial things including objects, stuff, and agents, as well as mathematical concepts as its main concepts. The complete taxonomy counts about 8.000 classes, including about 130 action classes, 7000 object types and 150 robot-specific concepts, that can be described by over 300 kinds of properties. These classes and their instances are defined through a set of assertions and can be linked to subsymbolic data that describes them in more detail. For example, object classes can refer to 3D models of their geometry and their composition from functional components, as shown in Figure 6 for a bottle of pancake mix. These combined semantic-geometric models can be generated automatically from common 3D models as they can be found in public database in the Internet [17].

These concepts provide the vocabulary that is used by a set of predicates for representing and reasoning about robot activity episodes, the most important of which are listed in Figure 4. They are thematically grouped into predicates about

Predicates on occasions, beliefs, and events	
holds(Occ,T_i)	The occasion (fluent) *Occ* holds in the time interval T_i
belief_at(Occ,T_i)	The robot believes at T_i that the occasion Occ holds at T_i
occurs(Ev, T_i)	Event *Ev* occurs in time interval T_i
Occasion types	
loc(Obj, Loc)	Location of an object
object_visible(Obj)	Object is visible to the robot
robot(Part,Loc)	Location of the robot part *Part*
Event types	
object_perceived(Obj)	Object has been perceived
image_captured(Img)	Image has been captured
Object descriptions	
desig_type(Desig,Tp)	Type of designator
desig_prop(Desig,Prop,Val)	Property values of designator
desig_pose(Desig,Pose)	Pose of perceived object designator
matches(Desig,Descr)	Match designator to description

Predicates on plans and plan interpretation	
task(Tsk)	Tasks on the interpretation stack
task_type(Tsk, Type)	Type of this task element
task_goal(Tsk, G)	Goal of task
task_start(Tsk,T_i)	Start time of task
task_end(Tsk,T_i)	End time of task
task_used_gripper(Tsk, Grp)	Gripper that has been used for a *Tsk*
subtask(Tsk,Sub)	Task is a parent of Subtask
subtask$^+$(Tsk,Sub)	Task is an ancestor of Subtask
task_outcome(Tsk,Res)	Result of task (success or fail)
task_failure(Task,Failure)	Failure of a task
failure_type(Failure,Type)	Type of failure
failure_attribute(Failure, Name, Val)	Failure attribute (e.g. error message)

Fig. 4. Predicates for reasoning about the memorized experiences.

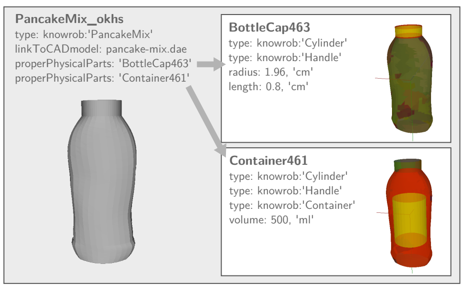

Fig. 6. Conceptual knowledge about a bottle of pancake mix.

occasions, beliefs, and events; predicates about plans and plan interpretation; event types; occasion types; and object descriptions. While these predicates provide the user with a uniform, logical view on the data, they may be computed at query time from different information sources.

We call this concept a "virtual knowledge base" that is created on top of the semi-structured and often high-volume log data [18]. The content of the symbolic knowledge base is computed on demand as an abstracted "view" on the subsymbolic data and is therefore fully grounded in the robot's internal data structures. Having a common symbolic representation allows to easily combine data from different sources and to answer the same queries on data sets of different structure. For example, some robots may not be controlled by a sophisticated plan-based controller and have less information in the log files. However, as long as the following predicates can be implemented on top of the logged data, the same queries can be answered.

The predicates *holds*, *believes_at*, and *occurs* represent different aspects of a changing world state: *holds(Occ,T_i)* is true iff the occasion type *Occ* is true throughout the time interval T_i. Occasions (also called fluents [19], [20]) represent time-varying states of the world such as the locations of objects. Formally, an occasion is a functional term in logic that maps an occasion type such as *empty(cup-23)* into the time intervals in which *cup-23* was empty. Thus, the assertion *holds(empty(cup-23),t-46)* is true if *t-46* is a sub-interval of a time interval in which the cup was empty.

The predicate *believes_at(Occ,T_i)* is similar to *holds(Occ,T_i)*, with the difference that *believes_at* represents a belief of the robot rather than the true world state. Typically, we distinguish between *holds* and *believes_at* only if we have external devices for observing manipulation episodes that can provide us with ground truth data, and return identical values otherwise. The predicate *occurs(Ev,T)* states that event *Ev* occurs throughout the time interval *T*. For example, the assertion *occurs(contact-13,t-31)* states that the contact event *contact-13* has occurred in the time interval *t-31*.

In addition to changes of the outside world, the episodes also describe the process of plan interpretation. In this context, we mean by a task *tsk* (assertion: *task(Tsk)* the intention of the robot to execute a piece of the control program *Exp*. The tasks that a robot generates when executing its plan are organized in a hierarchical task network (because plan steps can be executed concurrently). The task hierarchies are represented through the predicate *subtask(tsk_1,tsk_0)* stating that tsk_1 is a subtask of task tsk_0. Thus, if we are interested *how* a task was executed, we have to explore the subtask relations. If we want to infer *why* a task was executed, we have to analyze its super tasks.

The representation further includes sensory events: Whenever the camera driver gets the command to capture an image, an event of the form *occurs(image_captured(i), T)* is automatically asserted, and the corresponding image is stored as an effect of the event. Related to this are the object descriptions the robot uses to manipulate objects which are called "designators" in our system. In the initial plan, the descriptions might be abstract such as "the green cup on the kitchen counter". While the robot searches for and finds the cup, this abstract description will be updated with the data extracted from captured images: the exact pose of the cup, its size, a bounding box, and so on. The parameterization of reaching and grasping actions heavily depends on the descriptions of the objects to be manipulated, so their values and the evolution over time are contained in the episodes. Information about designators can be retrieved using the occasions *desig_pose* and *desig_attribute*. A snapshot of these descriptions is stored whenever they are updated, allowing the system to reason about the evolution of the robot's belief over time by comparing the descriptions before and after an action. The differences are usually those attributes that can be inferred from perception, such as the pose and size of the object. OPEN-EASE provides the occasion *matches(Desig,Descr)* to reason about object descriptions *Descr* and whether or not the robot believes that they are satisfied by objects in the world (*Desig*). The logical expression *believes_at(matches(Desig,* [*an, object,* [*type, cup*], [*color, red*]]*), t)* is true for every object description *Desig* that the robot believes to represent a red cup.

This logical query language is complemented by a set of predicates for loading and reasoning about lower-level data and for visualizing the results of queries. When dealing with geometric information that changes over time, sophisticated methods for transforming poses between coordinate frames are required. We extended the widely-used *tf* library in ROS to operate on the database of logged data, offering the same interface that is used for runtime operation.

C. Software Tools for Recording, Querying, Visualizing, and Analyzing Episodes

The OPEN-EASE system comes with a suite of software tools for logging data from robot manipulation episodes, for reasoning about them, and for visualizing the results. Our robots perform their tasks under the supervision of the CRAM executive [21] that automatically records comprehensive log data as described in [18]. The approach is not limited to robots running CRAM, and the same analyses can be performed on very different log data as long as predicates in the previous section can be computed from it. However, the data produced by CRAM is much more comprehensive and semantically rich than logs of other executives such as SMACH [22]. The system, including the query-answering modules and the web interface, can either be used as a hosted cloud service, or be downloaded as open-source software[2] and installed locally.

IV. IMPLEMENTATION — KNOWROB$_S$

The OPEN-EASE system has been implemented in a cloud-based version of the KNOWROB robot knowledge base [16]. KNOWROB provides expressive representations and sophisticated reasoning methods that are tailored to the needs of autonomous robots. Low-level data from robot and human activities are logged into a "big data" database using an extension of the *mongodb_log* tool [23]. The methods for recording and reasoning about higher-level experience data are based on our own prior work [18]. The communication between the browser and the ROS system in the cloud, as well as many of the graphical elements in the query frontend, have been built using the *robotwebtools* framework [24].

OPEN-EASE has to fulfill an important technical requirement, namely to equip each user with her own individual knowledge base. This is necessary because users have to load and unload knowledge bases from their own and from common repositories to perform their experiments, and will also assert additional facts and rules to work with the knowledge base. This capability is provided by KNOWROB$_S$, a Software-as-a-Service cloud application which offers KNOWROB functionality to remote users that can connect to a WebSocket [25] using the rosbridge [26] protocol. Web sockets are supported by most modern browsers, but can also easily be implemented as part of a client application on a robot. KNOWROB$_S$ uses the highly efficient virtualization techniques of the Docker framework[3] to create separate virtual knowledge bases for each user. Instead of emulating a computer's hardware, Docker isolates processes which still run on the same Linux kernel w.r.t. process IDs, memory and storage resources, computing time, network interfaces, user rights etc. These capabilities allow us to provide individual knowledge bases to different users without prohibitive usage of memory and computing resources.

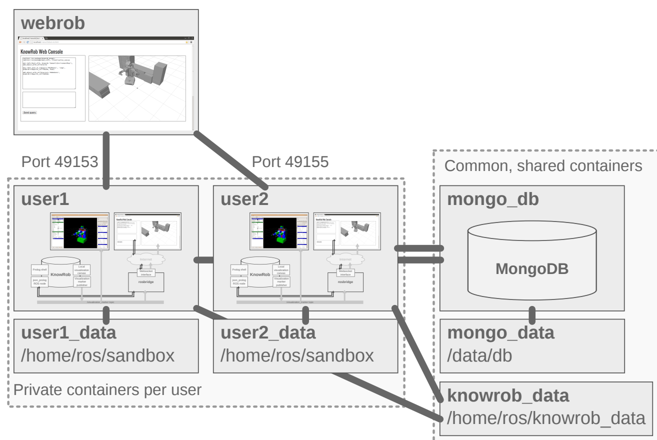

Fig. 7. Structure of the proposed system. Each user has a private knowledge base container, but can transparently access shared datasets of robot data.

Figure 7 visualizes the architecture of the KNOWROB$_S$ system. The web-based frontend manages the different containers and assigns them to users once they log into the system. Each user has one container with a complete KNOWROB system, plus a container for persistent data storage. In addition, we have containers that are shared among all users, such as a common database with logged high-volume robot data and a common repository of memory episodes.

V. USE CASES

All of the following use cases have been realized using the same approach. 1) A computer system is generating or observing manipulation activities. 2) The system is extended through a "big-data" logging mechanism that logs the high-volume and in particular subsymbolic system state data as comprehensively as possible without slowing down the system operation. 3) The logged data are symbolically annotated and interpreted as instances of the symbolic concepts in the KNOWROB taxonomy such that they can be semantically indexed. 4) We can then use the Prolog language together with

[2] Installation instructions: http://knowrob.org/doc/openease

[3] https://www.docker.com/

the predicates listed in Figure 4 to reason about manipulation episodes and answer queries about them.

A. *Working with Robot Manipulation Episodes*

The combination of a powerful representation and logic-based query language with comprehensive geometric information enables robots to reconstruct the state of the world as the robot believed it to be at a semantically described point in time, for example at the moment when grasping a cup (Figure 8 left). This allows the a-posteriori analysis of failure situations, which can be very helpful in case of incidental problems that are very difficult to trace otherwise. By testing new algorithms on the logged sensor data, one could test whether they would have performed better in that respective situation. By setting up a simulator with the logged world state, one could even combine these perception results with (simulated) robot actions.

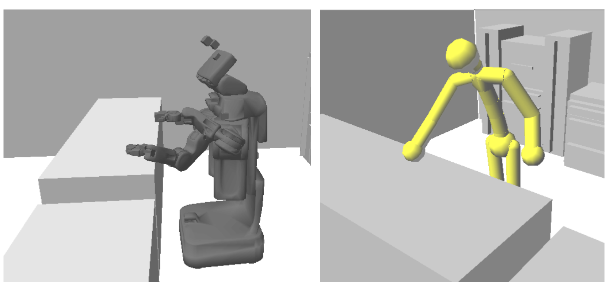

Fig. 8. Example use case of reading subsymbolic information on poses and trajectories based on logical queries. Both robot log data and observations of human actions can be queried in the same way.

B. *Working with Human Demonstration Episodes*

The same queries that can be answered on logged robot experiences can also be answered based on observations of human activities (Figure 8 right). The obvious restriction is that information about the plans and intentions of the human is not available, but only the external observation of the resulting behavior. If we annotate the observations, either manually or using automated activity recognition methods, we can use the same methods for semantic retrieval of observed data that could e.g. help with selecting data for analysis or learning purposes.

C. *Working with Collections of Episodes*

Having not only one, but a collection of episodes allows to compute statistical information that can be used for evaluation of the robot's performance and for learning prediction models. This helps to answer questions such as how long actions take on average, how reliable they are, and which failures occur most frequently. Figure 9 shows two examples: The majority of the errors is of type *ManipulationPoseUnreachable*, as can be seen in the chart on the left, which suggests that improvements of this component can have a strong impact on the overall performance. The average duration of tasks, shown in the chart on the right, gives robots and humans information on how long actions typically take, e.g. for scheduling purposes.

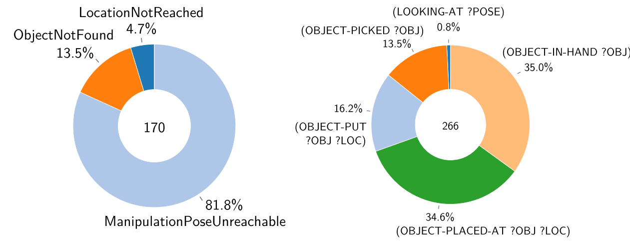

Fig. 9. Statistics computed from 25 manipulation episodes. Left: Distribution of failure types. Right: Average duration of common subtasks.

D. *Robots Using* OPEN-EASE

While we present the results of queries as graphical visualizations in the web-based query frontend, the same interface can be used by robots to send queries to a KNOWROB instance in the cloud. In a local ROS setup, robots send Prolog queries to KNOWROB encoded in the JSON format via the *json_prolog* service. KNOWROB$_S$ provides the same interface, tunneled via a WebSocket connection.

VI. PROJECTED APPLICATIONS

Besides being a potentially powerful remote knowledge processing service for AI / Robotics researchers and autonomous robots, we currently investigate a number of possibly high-impact applications of KNOWROB$_S$, namely using KNOWROB$_S$ for realizing

1) an eLearning tool in AI-based robotics. We are using OPEN-EASE as a web tool for teaching a course in intelligent robotics to let students explore the hardware of robots, their sensors and effectors, and to get better intuitions about the data that sensors generate ("Can you detect the handles of cups using images where the camera is positioned at least 1,5m away from the cup?", or "Which objects or object parts in the kitchen environment cannot be detected with the Kinect sensor of the robot?"). In addition, we let the students do students exercises with real robot data, such as learning object classifiers for the objects that stand on the kitchen counter during a set of manipulation episodes.

2) a tool for reproducible experimental data. OPEN-EASE improves the reproducibility of experimental data. Consider the case where an experimental evaluation of a scientific publication has to be extended after some time. Rerunning the experiments is tedious and time consuming and requires a similar hardware setup. The comprehensive log data collected and the semantic retrieval facilities supports researchers to add missing evaluations on the existing experimental data. Researchers can even give reviewers and readers access to the data through OPEN-EASE, which allows them to clarify questions regarding the experimental setting (e.g. where the robot stood when the object recognition mechanisms succeeded, or in which scenes an object could not be recognized).

3) a tool for open robotics research. Recently, progress in many research fields has been fueled by making large volumes of data available and by running data and information analytics tools on them, or by simply visually browsing and searching through these data. Examples of such initiatives are the Allen Brain Atlas, geographic data, etc. OPEN-EASE currently provides data from robotic agents performing fetch and place tasks in a kitchen environment, users demonstrating pancake making in a virtual reality game, people setting the table and cleaning up (from the TUM kitchen dataset [27]). We plan to include experience data from robotic agents performing chemical experiments, human-robot cooperation, and others. This makes OPEN-EASE one of the most comprehensive and detailed activity knowledge bases relevant for autonomous robotics research.

3) a tool for creating realistic benchmark problems for machine learning and robot perception. Another possible usage of OPEN-EASE is the creation of realistic benchmark datasets. Consider the case where you want to create a realistic set of robot perception tasks in order to test some newly developed robot perception method. To do so, you could take characteristic everyday activities such as setting the table and first query OPEN-EASE for the set of perception tasks that a robot issues in order to perform such an activity. This analysis will result in some understanding of which types of perception tasks are important and which ones are not. We can then generate "ground-truth" data by finding the time instant where the robot has the most complete and detailed information about a specified scene, such as the objects on a table. If needed, the user can assert additional knowledge or correct information by retracting and asserting KNOWROB facts. Finally, the user can make up situations in which the perception tasks are to be performed.

4) a tool for grounding and assessing the assumptions and inference mechanisms of action formalizations in knowledge representation. Most knowledge representation languages and methods for symbolically reasoning about actions and change are based on modeling assumptions. OPEN-EASE gives researchers in these fields the opportunity to test to what extent such assumptions are valid for autonomous manipulation robots, and to what extent the inferences performed by these formalisms are valid with respect to the behavior and the physical effects that robotic agents generate.

VII. RELATED WORK

OPEN-EASE is positioned in the intersection of intelligent information systems and remote software services for robots.

In robotics, the *Robo Brain*[4] project led by Saxena and his colleagues is most closely related to OPEN-EASE. *Robo Brain* is a large-scale computational system that learns from publicly available Internet resources, computer simulations, and real-life robot trials. It accumulates everything into a comprehensive and interconnected knowledge base. OPEN-EASE differs from *Robo Brain* in several aspects. OPEN-EASE incorporates data from different sources into a common, *formal knowledge representation language* with powerful *inference mechanisms*. The data are *automatically generated* through robots and observation systems rather than human computation methods. Unlike in the *Robo Brain* project, however, which already broadly applies learning to the data, the learning efforts in OPEN-EASE have not started yet.

OPEN-EASE follows up on research that aims at providing knowledge bases for robots in the world-wide web. Those research efforts include RoboEarth [28] which investigates how robots can share their knowledge by providing a meta representation language for robot plan schemata and knowledge bases that allows robots to upload, find, and download available knowledge and for checking whether they can employ the respective knowledge [28], [29], [30].

OPEN-EASE is a cloud robotics application [31], [32], [33] that is specialized for providing robots with knowledge. Other examples of cloud services are Mujin[5], providing client robots with motion planning capabilities, and Google goggles [34] that can retrieve web pages matching captured images.

Web services are more common in the area of intelligent information systems. Here, WordNet[6], a dictionary knowledge base, OpenCyc[7], an encyclopedic knowledge base, and the OpenMind and the OpenMind Indoor Common Sense [35] common-sense knowledge bases, the Google Knowledge Graph [36] and DBpedia [37] are popular examples. However, these knowledge services focus primarily on text and symbolic data, while OPEN-EASE contains large amounts of sensory data of many modalities.

Finally, there are also a number of data repositories for human manipulation activity data offering relevant data, though not as formally represented knowledge nor as a ready-to-use web service. Such data repositories include the TUM kitchen dataset [27], the MPII Cooking Activities dataset![38], and the CMU MMAC dataset [39].

VIII. CONCLUSIONS

In this paper we have described and discussed OPEN-EASE, a remote knowledge representation and processing service for human researchers and robots. OPEN-EASE enables its users to interpret, analyze, visualize, and learn from the experience data from robots and human manipulation episodes. Using OPEN-EASE users can retrieve the memorized experiences of manipulation episodes and ask queries regarding what the robot saw, reasoned, and did as well as how the robot did it, why, and what effects it caused.

OPEN-EASE is unique because of (1) the comprehensiveness with which real execution data of modern autonomous manipulation robots is logged, stored and made openly accessible to the research community; (2) the representational infrastructure that is provided to make very inhomogeneous experience data from different robots and even human manipulation episodes semantically accessible in a uniform and standardized concept vocabulary; and (3) a suite of software tools that enable researchers and robots to interpret, analyze, visualize, and learn from the experience data.

Projected applications of OPEN-EASE include its use an eLearning tool in AI-based robotics, a tool for making

[4]http://robobrain.me

[5]http://mujin.co.jp/en/

[6]http://wordnetweb.princeton.edu/perl/webwn

[7]http://sw.opencyc.org/

reproducible experimental data accessible and for enabling semantic information retrieval, and a tool for open robotics research. OPEN-EASE can be accessed through the web page *http://www.open-ease.org*.

ACKNOWLEDGMENTS

This work has been supported in part by the EU FP7 Project *RoboHow* (grant number 288533). We would like to thank Arne Stefes, Asil Kaan Bozcuoglu and Daniel Beßler for their help with the visualization tools and the query library.

REFERENCES

[1] SPARC – The Partnership for Robotics in Europe, "Strategic Research Agenda for Robotics in Europe 2014 – 2020," euRobotics aisbl, Tech. Rep., 2014, http://www.eu-robotics.net/cms/upload/PPP/SRA2020_SPARC.pdf.

[2] J. Hollerbach, M. Mason, and H. Christensen, "A Roadmap for US Robotics – From Internet to Robotics," Computing Community Consortium (CCC), Tech. Rep., 2009.

[3] 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.

[4] B. C. Williams, M. Ingham, S. H. Chung, and P. H. Elliott, "Model-based Programming of Intelligent Embedded Systems and Robotic Space Explorers," *Proceedings of the IEEE: Special Issue on Modeling and Design of Embedded Software*, vol. 9, no. 1, pp. 212–237, January 2003.

[5] R. Alami, R. Chatila, A. Clodic, S. Fleury, M. Herrb, V. Montreuil, and E. A. Sisbot, "Towards human-aware cognitive robots," in *AAAI-06, Stanford Spring Symposium*, 2006.

[6] K. Okada, M. Kojima, S. Tokutsu, Y. Mori, T. Maki, and M. Inaba, "Integrating recognition and action through task-relevant knowledge for daily assistive humanoids," *Advanced Robotics*, vol. 23, pp. 459–480(22), March 2009.

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

[8] 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.

[9] 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.

[10] D. McDermott, "A Temporal Logic for Reasoning About Processes and Plans," *Cognitive Science*, vol. 6, no. 2, pp. 101–155, 1982.

[11] ——, "Reasoning about plans," in *Formal Theories of the Commonsense World*, J. R. Hobbs and R. C. Moore, Eds. Norwood, NJ: Ablex, 1985, pp. 269–317.

[12] J. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[13] M. Ghallab, "On Chronicles: Representation, On-line Recognition and Learning," in *International Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 1996, pp. 597–607.

[14] 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.

[15] K. Chodorow and M. Dirolf, *MongoDB: The Definitive Guide*. O'Reilly & Associates, 2010.

[16] 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.

[17] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz, "Decomposing CAD Models of Objects of Daily Use and Reasoning about their Functional Parts," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo Big Sight, Japan, November 3–7 2013, pp. 5943–5949.

[18] J. Winkler, M. Tenorth, A. K. Bozcuoglu, and M. Beetz, "CRAMm – memories for robots performing everyday manipulation activities," *Advances in Cognitive Systems*, vol. 3, pp. 47–66, 2014.

[19] J. McCarthy and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," in *Machine Intelligence 4*, B. Meltzer and D. Michie, Eds. Edinburgh University Press, 1969, pp. 463–502.

[20] R. Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT press, 2001.

[21] M. Beetz, L. Mösenlechner, and M. Tenorth, "CRAM – A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22 2010, pp. 1012–1017.

[22] J. Bohren and S. Cousins, "The SMACH High-Level Executive," *IEEE Robotics and Automation Magazine*, vol. 17, pp. 18–20, 2010.

[23] T. Niemueller, G. Lakemeyer, and S. S. Srinivasa, "A Generic Robot Database and its Application in Fault Analysis and Performance Evaluation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2012*. Vilamoura, Algarve, Portugal: IEEE/RAS, 2012.

[24] 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.

[25] I. Fette and A. Melnikov, "The WebSocket Protocol," RFC 6455 (Proposed Standard), Internet Engineering Task Force, Dec. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6455.txt

[26] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[27] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition," in *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*, 2009.

[28] 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.

[29] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz, "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, May 14–18 2012, best Cognitive Robotics Paper Award.

[30] 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.

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

[32] 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

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

[34] 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.

[35] R. Gupta and M. J. Kochenderfer, "Common sense data acquisition for indoor mobile robots," in *Nineteenth National Conference on Artificial Intelligence (AAAI-04*, 2004, pp. 605–610.

[36] J. Eder, "Knowledge graph based search system," 2012, uS Patent App. 13/404,109. [Online]. Available: http://www.google.com/patents/US20120158633

[37] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," *The Semantic Web*, pp. 722–735, 2007.

[38] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, United States, June 2012.

[39] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, and J. Macey, "Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database," CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, Tech. Rep., 2009.