KnowRob 2.0 — A 2nd Generation Knowledge Processing Framework for Cognition-enabled Robotic Agents

Michael Beetz[1], Daniel Beßler[1†], Andrei Haidu[1],
Mihai Pomarlan[1], Asil Kaan Bozcuoğlu[1] and Georg Bartels[1]

***Abstract*— In this paper we present KNOWROB2, a second generation knowledge representation and reasoning framework for robotic agents. KNOWROB2 is an extension and partial redesign of KNOWROB, currently one of the most advanced knowledge processing systems for robots that has enabled them to successfully perform complex manipulation tasks such as making pizza, conducting chemical experiments, and setting tables. The knowledge base appears to be a conventional first-order time interval logic knowledge base, but it exists to a large part only virtually: many logical expressions are constructed on demand from data structures of the control program, computed through robotics algorithms including ones for motion planning and solving inverse kinematics problems, and log data stored in noSQL databases. Novel features and extensions of KNOWROB2 substantially increase the capabilities of robotic agents of acquiring open-ended manipulation skills and competence, reasoning about how to perform manipulation actions more realistically, and acquiring commonsense knowledge.**

I. INTRODUCTION

Robotic agents that are to accomplish goal-directed object manipulation tasks need a lot of commonsense and intuitive physics knowledge. This knowledge is needed to bridge the gap between underdetermined instructions, such as "pick up the cup", and the detailed motion specifications needed to accomplish the action. Many objects have to be grasped and held differently depending on their form, weight, properties (breakable, hot, soft, or wet), the task that is to be performed with them, and the location where they are to be picked up from. Avoiding unwanted side effects might require the robot to use a precision grasp of the handle and keep the cup upright. The appropriate grasp and motion parameterizations depend on background knowledge. Namely that the handle is the object part for holding the cup and that it should be held upright to avoid spilling.

KNOWROB2 is an extension and partial redesign of KNOWROB [1] that provides the knowledge representation and reasoning mechanisms needed to make informed decisions about how to parameterize motions in order to accomplish manipulation tasks. The extensions and new capabilities include highly detailed symbolic/subsymbolic models of environments and robot experiences, visual reasoning, and simulation-based reasoning. Aspects of redesign include the provision of a interface layer that unifies very heterogeneous representations through a uniform entity-centered logic-based knowledge query and retrieval language. In addition, KNOWROB2 is designed to leverage concepts and results from motor cognition and robot control to extend AI reasoning into the motion level and make the robot's reasoning mechanisms more powerful. The new capabilities and functionalities are facilitated through employing modern information processing technologies such as physics simulation and rendering mechanisms of game engines, big data recording, storage, and retrieval technologies, and machine learning. The use the above leading-edge information technology enables robotic agents to acquire generalized commonsense and intuitive physics knowledge needed for the mastery of human-scale manipulation tasks from experience and observation and to make AI reasoning actionable within the perception-action loops of robots.

[1] The author is with the Institute for Artificial Intelligence (IAI), Universität Bremen, 28359 Bremen, Germany

† Corresponding author. danielb@cs.uni-bremen.de

This work was partially funded by Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Center 1320, *EASE*.

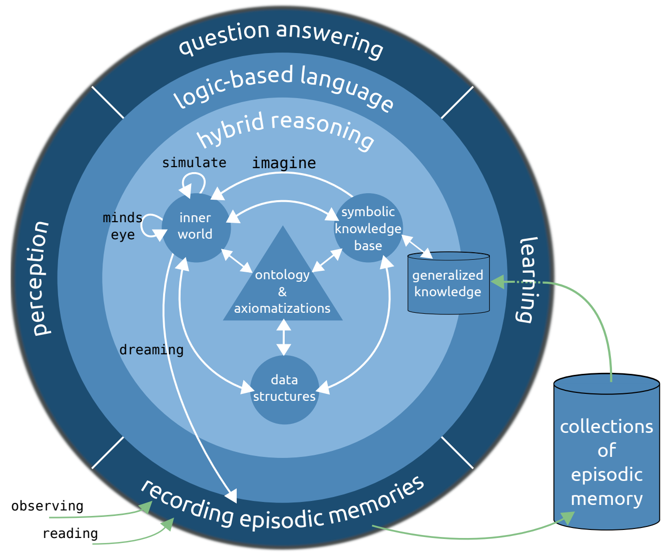


Fig. 1: Software architecture of KNOWROB2.

The purpose of this paper is to give an overview on (•) which additional cognitive capabilities a knowledge processing framework should provide to enable the mastery of human-scale manipulation tasks, (•) how KNOWROB2 integrates these capabilities to provide a fast, robust, and uniform query answering capability for robot control systems that returns results that can be executed by the robot execution system, and (•) how these cognitive capabilities are realized through leading-edge technology for photo-realistic rendering and simulation based reasoning.

Among others, KnowRob2 provides the following cognitive capabilities that go beyond what other robot knowledge processing systems offer. KnowRob2 is be able to

- "reason with its eyes and hands". It is able to imagine a state of the environment at a level of detail that is almost photo-realistic. It is further able to imagine the execution of manipulation actions at the level of motions and force-dynamic interactions of objects, and translate the observed state evolution into a first-order time logic representation that can be used for semantic information retrieval and reasoning. A similar functionality is available for mentally looking at imagined scenes. This capability is currently realized through an additional preliminary knowledge system that is implemented on top of a photo-realistic and physics simulation-based game engine as an additional knowledge system.
- reason about motion parameters such that certain physical effects are achieved or avoided. The goal is to specify actions by their desired physical effects and the motion control system infers the appropriate motion parameterization and adaptation automatically. This is accomplished by learning generalized models of how physical effects of actions vary depending on the motion parameterization.
- learn generalized commonsense and intuitive physics knowledge from experience that is applicable to novel situations and tasks. The information processing mechanisms that realize this capability are inspired by the human memory system and the role that episodic memories play in the acquisition of generalized knowledge [2]. KnowRob2 employs techniques for efficiently recording, storing, maintaining, and semantically indexing huge sensor and motion data streams and data intensive machine learning and data analytics methods to accomplish this.

KnowRob2 is partly available via the knowledge service openEASE [3], a platform through which robots and researchers can upload, access, and analyse episodic memories of robots performing manipulation tasks. The service is publicly accessible at `www.openease.org`.

We evaluate KnowRob2 by demonstrating its comprehensive capability in query answering. We will give examples of queries that are essential for decision making and motion parameterization in robot manipulation that go beyond what can be answered in representations based on the state transition model of robot action. Additional queries are available via openEASE[1].

The remainder of this paper is organized as follows. The next section gives an overview of the KnowRob2 framework and describes its architecture. The subsequent sections then detail the respective reasoning components of the framework. Finally, we describe the logic-based interface language, and how robots can learn from episodic memories.

[1] `https://data.open-ease.org/`

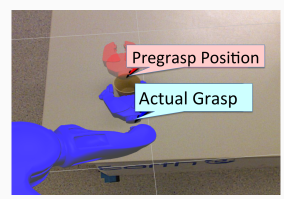
II. Overview

A robot would use KnowRob2 as a query answering system just like today's information agents, such as the iPhone's Siri. The main difference is that KnowRob2 runs within the perception action loop of robotic agents: many queries are connected to the tasks that the agent is executing and refer to images that it captures, motions that it performs, etc. The queries are primary asked to decide on how to manipulate objects or parameterize motions. For example, for a table clearing task typical queries are: *where are the cups and plates*, *where to stand to pick up an object*, *how to reach for the object*, *how to position the grippers*, *how much force to apply*, *how to move the arm to pick it up*, *how to hold it*, etc. To answer such detailed queries KnowRob2 is tightly connected to the control program, reuses some of its algorithms, and shares data structures with it.

A typical inference is depicted in Figure 2. This example is taken from an episode in which a robotic agent grasps a cup to pour out the remains. The robot asks the query *how to pick up the cup to pour out its remains*. The query, stated in Prolog, works as follows: it retrieves a pouring action *Tsk*, that has a sub-action to grasp the source container, which is called *Spt-Tsk*. It then queries the pregrasp *PG* and grasp pose *G* and the grasp force of the respective reaching motion and displays these motion parameters.

Query ?

```
entity(Tsk, [an, action,
 [type, pouring],
 [source, Cup],
 [sub-action, [an, action,
  [type, grasping],
  [name, Spt-Tsk],
  [object-acted-on, Cup],
  [pregrasp-pose, PG],
  [grasp-pose, G],
  [grasp-force, Force]]]]),
entity(Cup, [an, object,
             [type, cup]]),
show([PG, G, Force]).
```




```
Tsk = 'Pouring_0',
Spt-Tsk = 'Grasping_6',
PG = 'Pose_788',
G = 'Pose_685',
Force = 50Nm.
```

! Answer

Fig. 2: A typical query in KnowRob2 on the left, and the inferred answer shown in openEASE on the right side.

There are several aspects that should be discussed about the query. First, the query needs to access data and parameterize data structures and function calls in the perception action loop of the control system; it therefore works in the embodiment of the robot system. Second, the level of detail of the knowledge processing system is much higher than the one typically used in artificial intelligence which considers actions as black boxes and therefore cannot reason about motion parameters and the relation between motion parameters and the effects of actions. A consequence of lowering the abstraction level at which knowledge is represented is a much easier grounding of symbolic expressions. It is easy to see that the information queried, namely the pregrasp and grasp poses, is properly grounded with respect to the perceived scene.

The architecture of KnowRob2 itself is depicted in Figure 1. A unique feature of it is the central position of the symbolic representation of the ontology, even below the data structures of the control system. This enables the programmer to semantically annotate and lets the control system automatically compute the semantic meaning of data structures. Section III gives further details about representations employed by KnowRob2.

Around the ontology is the hybrid reasoning shell. Many of the data structures, representations, parameters of computational processes have associated axiomatizations that declare their meaning with respect to the ontology, allowing the robot control system to use this data as if it were a symbolic knowledge base. The hybrid reasoning shell uses multiple methods for knowledge implementation. Key components are the data structures of the control system and robotics algorithms such as inverse kinematics and motion planning, which, for example, allow the programmer to specify that the robot should believe an object to be reachable if the motion planner can find a collision free path to the position of the object. The hybrid reasoning kernel is discussed in more detail in Section IV.

Another key and novel component of the hybrid reasoning shell is the inner world knowledge base [4]. It is a detailed, and photo-realistic reconstruction of the robot's environment in a game engine with physics simulation and vision capabilities, and adds powerful reasoning methods to the KnowRob2 knowledge processing framework. First, the robot can geometrically reason about a scene by virtually looking at it using the vision capability provided by the game engine, and predict the effects of actions through semantic annotations of force dynamic events monitored in its physics simulation. As Winston [5] would formulate it, it allows the robot to reason with its eyes and hands.

The subsequent interface layer exposes reasoning capabilities of control mechanisms integrated below it through a logic-based language. The language exploits control-level data structures for ad-hoc symbol grounding, and ontologies for unifying these heterogeneous representations. To applications above the interface layer, the hybrid reasoning shell appears to be a first-order logic knowledge base, but it is largely constructed on demand from data structures of the control program, and computed through robotics algorithms. This is discussed in Section V.

Finally, the interface shell provides the question answering, perception interface, experience acquisition, and knowledge learning interface of KnowRob2 that can exploit the rich set of hybrid reasoning mechanisms integrated below the interface layer.

To sum up, the hybrid reasoning shell provided by KnowRob2, which unifies robot control reasoning mechanisms with the inner world knowledge base, experience acquisition and other perception and learning mechanisms, makes KnowRob2 a competent knowledge service for personal robots and, also, unique compared to available knowledge representation frameworks for the same purpose such as [1], [6], and [7].

III. Ontologies and axiomatizations

Action representations used in AI and autonomous agents research typically represent the agent control system at a coarse level of detail, at which actions are described by blackbox models. Instead, KnowRob2 puts ontologies at the core of the control system so that data structures, even those used at lower levels, can be semantically annotated with their meaning according to the core ontologies.

In order to gain a better intuition of the advantages of putting symbolic ontologies at the core of robot control systems consider, for example, the concept of a dynamically changing robot pose. In KnowRob2 we represent this in the form of the *holds* predicate $holds(pose(r,\langle x,y,o\rangle),\ ti)$, which asserts that the robot believes its pose with respect to the origin of the environment map at time instant ti is $\langle x,y,o\rangle$. This relation is defined in the KnowRob2 ontology. Typically, the robot control system would estimate the pose of the robot over time using a Bayesian filter — such as a particle filter for robot self localization. In this case, the robot's belief about where it is is implemented in the probability distribution over the possible robot poses in the environment as estimated through the filter. Then, one can specify the pose where the robot believes to be ($holds(pose(r,\langle x,y,o\rangle),\ ti)$) as the pose $\langle x,y,o\rangle$ with the maximal probability. By specifying rules that ground concepts of the ontology into the data structures of the control systems, substantial parts of the data structures can be turned into a virtual knowledge base where the relevant relations are computed from the data structures on demand.

KnowRob2 employs multiple ontologies. The core ontology is the KnowRob ontology [1], which defines (•) robots, their body parts, how the parts are connected, and sensing and action capabilities, (•) objects, their parts and functionality, and constellations and configurations thereof, (•) robot tasks, actions, activities, and behaviors, and (•) situational context and environment. The ontology together with additional axioms and rules provides background knowledge that is relevant for manipulation tasks. For example, it states that a cup is a vessel that consists of a hollow cylinder and a handle, and that it can be used for drinking, mixing, and pouring substances. It also provides knowledge about the material they are made of, namely metal, wood, porcelain or plastic. A key role of the ontology is also the grounding of the ontology concepts in different components of the control system: the perception, reasoning, and control components.

Finally, KnowRob2 facilitates the use of additional special purpose ontologies for robots to gain more application domain information. Outdoor robots, for example, can use ontologies developed for geo-information systems, such as the ontologies of *OpenStreetMap* tags. Robotic assistants for department stores can use product data from web stores such as *GermanDeli* [2], etc.

[2] http://www.germandeli.com

IV. HYBRID REASONING KERNEL

The hybrid reasoning kernel of KNOWROB2 contains a set of knowledge bases that includes

- the *inner world knowledge base*: a world model composed of CAD and mesh models of objects positioned at accurate 6D poses, and equipped with a physics simulation,
- the *virtual knowledge base*: computed on demand from the data structures of the control system,
- the *logic knowledge base*: abstracted symbolic sensor and action data, with logical axioms and inference mechanisms, and
- the *episodic memories knowledge base*: experiences of the robotic agent.

In general, the knowledge content of these knowledge bases may be redundant or inconsistent. Rather than deriving the correct answer, KNOWROB2 computes multiple hypotheses, which are then checked for plausibility and consistency, like the way the Watson system operates.

To this end, KNOWROB2 employs hybrid reasoning methods. We will sketch the ones that are particular powerful for reasoning about robot manipulation tasks: reasoning based on the inner world model (e.g., simulation-based reasoning), and motion control reasoning.

A. Narrative-enabled Episodic Memories

When somebody talks about the deciding goal in the last soccer world championship many of us can “replay” the episode in our “mind’s eye”. The memory mechanism that allows us to recall these very detailed pieces of information from abstract descriptions is our episodic memory. Episodic memory is powerful because it allows us to remember special experiences we had. It can also serve as a “repository” from which we learn general knowledge.

KNOWROB2 integrates episodic memories deeply into the knowledge acquisition, representation, and processing system. Whenever a robotic agent performs, observes, prospects, and reads about an activity, it creates an episodic memory. An episodic memory is best understood as a video that the agent makes of the ongoing activity coupled with a very detailed story about the actions, motions, their purposes, effects, the behavior they generate, the images that are captured, etc.

We term the episodic memories created by our system narrative-enabled episodic memories (*NEEM*s). A *NEEM* consists of the *NEEM experience* and the *NEEM narrative*. The *NEEM experience* is a detailed, low-level, time-indexed recording of a certain episode. The experience contains records of poses, percepts, control signals, etc. These can be used to replay an episode in detail. *NEEM experiences* are linked to *NEEM narratives*, which are stories that provide more abstract, symbolic descriptions of what is happening in an episode. These narratives contain information regarding the tasks, the context, intended goals, observed effects, etc.

An example of the information contained in a *NEEM* is illustrated in Figure 3. In this episode, a robot cleared a dinner table that had a bowl and a spoon on top. The depicted timeline has marks for some time instants at which the robot

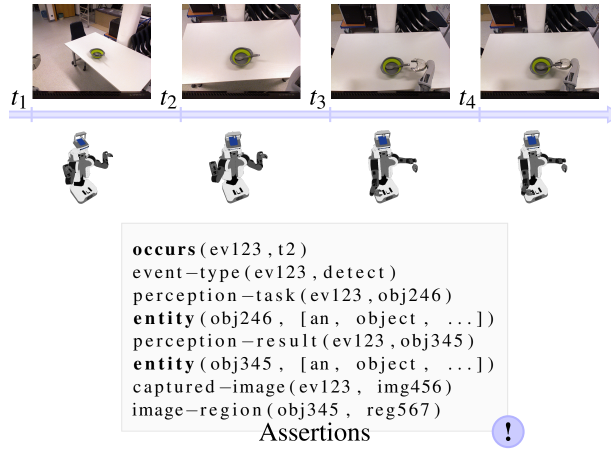


Fig. 3: Illustration of a narrative-enabled episodic memory.

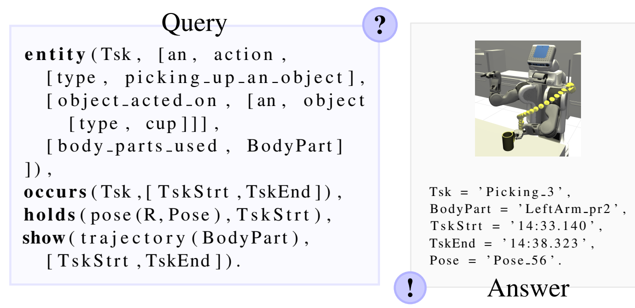


Fig. 4: Example query evaluated on a NEEM on the left, and the respective answer on the right.

started to perform an action. Images that were captured by the robot at these time instants are shown on the top row of the figure. The poses of the robot are shown below of the timeline. The robot navigates to the table at t_1 to perceive objects on top of it at t_2, to establish a pregrasp pose at t_3, and to grasp the spoon at t_4. Some of the corresponding assertions in the knowledge base are shown at the bottom of the figure. These assertions represent, for example, that at t_2 an event ev_{123} occurred, that this event was a detection event with the corresponding perception task obj_{246}, that the perceived object is described by obj_{345}, and that this object corresponds to the image region reg_{567} of the captured image img_{456}. The symbolic assertions are linked to data structures of the control program to enrich the high-level activity description with low-level information such as concrete motions.

*NEEM*s allow to ask queries about which actions the robot performed, when, how, and why they were performed, if they were successful, what the robot saw, and what the robot believed when the action was performed. The robot may ask queries such as: *how did I pick up a cup*, *which body part did I use*, and *how was my pose when picking it up*. These questions map to a query such as the one depicted in Figure 4. Here it searches for *NEEMs* where Tsk is a task where the robot picked up a cup with its body part *BodyPart*, that occurred during the time interval $[TskStrt, TskEnd]$, and at which start time the pose of the robot is described by *Pose*. KNOWROB2 gives the answer to that query in terms of symbol bindings for the free variables in the query, and visually by rendering the scene based on beliefs of the robot.

B. Inner World Knowledge Base

One of the knowledge bases employed by KNOWROB2 is based on the so called *inner world model*. The basic idea is to create a photo-realistic copy of the environment in which the robotic agent is located, composed of symbolically annotated 3D mesh models coupled with a physics engine. While this knowledge base provides all the capabilities of a symbolic knowledge base, it also allows the robotic agent to look at the environment as it believes it to be with its "mind's eye" and mentally execute a manipulation action in order to predict the effects of its parametrization.

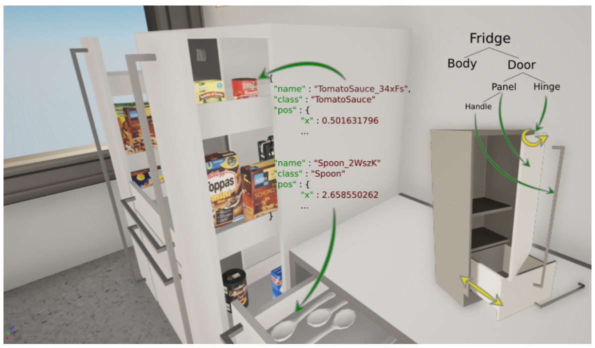


Fig. 5: Inner world model of a kitchen environment.

Figure 5 shows the inner world knowledge base of a kitchen environment that the robot is operating in. We can see that the inner world knowledge base is highly detailed and realistic. The objects in the environment are composed of object parts that have accurate CAD models. Thus, a cupboard consists of a door with a handle and shelves. The cupboard is also equipped with an articulation model that allows the robotic agent to simulate opening the cupboard. Objects such as the cereal box are modelled photo-realistically and stand on the shelf. When pushed, the box would tip over and depending on the exerted force fall down to the floor.

The entities in the inner world knowledge base have symbolic names with assertions that represent their properties. By asserting that an entity in the inner world knowledge base is an entity of the category cupboard that is defined in the KNOWROB2 ontology, the robotic agent can infer background knowledge about the entity. For example it can infer that dirty dishes belong in the dishwasher, perishable items into the fridge, and so on.

KNOWROB2 uses the inner world knowledge base for simulation-based reasoning to make inferences that would be difficult to achieve with purely symbolic knowledge bases (see section IV-C). For example, it can form expectations of how the interior of the fridge will look or whether inserting a certain item might cause another to tilt.

There are two kinds of inner world knowledge bases: first, the one that mirrors the current belief state of the agent about the state of the environment and the ongoing activities and second, ones that are created on demand in order to explore how activities would proceed in hypothetical situations and episodes. The belief state is represented using KNOWROB2 terminology such that it can be trivially mirrored in the inner world knowledge base.

C. Simulation-based Reasoning

An important aspect of performing manipulation actions competently is envisioning their consequences in the inner world to forestall unwanted side effects. This often requires the careful adjustment in the continuous, high-dimensional space of motion parameters including body part poses, velocities, and forces. The only plausible mechanism we see for predicting the motion effects is physical simulation.

Humans have been shown to simulate their actions and its resulting effects, without highly accurate and detailed simulations [8]. For example, when pouring a substance from a container, the human does often not know the viscosity of the substance, or when cutting not the stiffness of the material to be cut. Simulation is a valuable information source despite these limitations because it predicts possible outcomes, which the agent can use to monitor execution and identify the right model at execution time [9].

Simulation-based reasoning is one of the reasoning strategies employed by KNOWROB2. It uses representations on a low abstraction level – poses, perceptions, meshes – to predict outcomes of motion parameterizations and (high-level) plans. These predictions can be used to reason about the correct course of action. Simulation-based reasoning has the benefit of avoiding the frame problem [10] and capturing nuances of physical behavior that would be difficult to formalize in logic-like reasoning systems. Moreover, the knowledge underlying the reasoning – particles, collisions, gravity, etc. – is much more generic and less likely to contain gaps than rules or knowledge we might input manually.

Sensory data, control commands, and higher level events such as contacts between objects, state changes of objects, grasping events etc. are continuously recorded during simulation and asserted to the *episodic memories knowledge base* of KNOWROB2. Since we want to extract meaningful information from these records, we have to abstract away from meshes and poses to for example states and objects. In other words, we create symbolic entities linked to subsymbolic information via procedural attachments. Being in a virtual environment we have access to ground truth data during the whole simulation.

When clearing a table, for example, the robot may have to pour some remaining liquid from a cup into the sink before putting it into the dishwasher, and wants to reason about the tilting angle, and the altitude such that the liquid is not spilled to much in the sink. Such goals are hard to capture in a pure logics formalism, but the fluid dynamics can be monitored in simulation and exposed to the logics-based interface language of KNOWROB2 via *NEEM*s acquired from the data. The *NEEM*s can be queried just like the *NEEM*s from real-world executions:

```
?- entity(Act, [an, action, [type, pouring],
        [success, true], [tilting_angle, Tilt],
        [altitude, Altitude], [spilled_amount, Spill],
        [object_acted_on, [an, object, [type, cup]]],
        [target_location, [an, object, [type, cup]]]]).
```

The query is used to retrieve successful pouring episodes during which some liquid was poured from a cup into another cup. It also retrieves motion parameters (i.e., tilting angle and altitude), and the amount of spilled liquid according to the monitored state of the physics engine. Note that we use simulation both for learning and reasoning. Learning pertains to learning (common-sense) knowledge from *NEEM*s (some of which collected in simulation) and will be described in more detail in Section VI.

Speed is a challenge for simulation-based reasoning. In order to support online decision making, the processes have to be completed within a short timeframe. It can be reasonably expected that the fast development in processor/GPU power and physics engines will help us to significantly improve this aspect. Moreover, by limiting the time over which simulations are run compared to the current state of the real world, e.g. limiting how far one "looks ahead", we can limit how much time simulation-based reasoning requires. Speed is not as critical for learning from simulated episodes, on the other hand. The robot can "dream" overnight while physically inactive by running simulations of activities with varying control parameters.

D. Motion Control Reasoning

KNOWROB2 does not follow the classical approach to reasoning about actions. That is, it does not abstract away the particularities of *how* actions are executed. Instead, it combines action representations with explicit and modular motion descriptions. This tight coupling of motion control and action representations and reasoning is similar in spirit to the research on *combined task and motion planning* [11]. In fact, this tight coupling is a key enabler of competent robot manipulation capabilities.

To establish some intuition of the problem of competent robot manipulation behavior, let us revisit our running example of a robot pouring from a cup. When the robot intends to pick up the cup filled with milk, it has to hold the cup upright to avoid spillage. Holding the cup upright is a motion constraint, while the insight that a tilted cup may lead to spillage is a symbolic conclusion about a causal relationship. Similarly, the robot might want to grasp the cup close to the center of mass to have better control when pouring. From these examples it should be intuitively clear that reasoning about motion control and action execution have to be tightly coupled to achieve competence in manipulation.

By not abstracting away *how* actions are executed, KNOWROB2 avoids the difficult problem of determining an appropriate level of action abstraction. This is indeed a hard problem because what is appropriate depends on the particular task at hand and the physical capabilities of the robot. The problems that arise when tackling manipulation actions with abstract and discrete representations have been well studied in the context of the egg cracking problem [12]. It was shown that the number of assertions and rules needed to state for simple manipulation problems explodes and that the representations are complex and not very intuitive. Often the problems are caused because the continuous behavior and events that are caused by manipulation actions have to be discretized differently for different subproblems.

State of the art methods for robot motion control, on the other hand, only generate the sophisticated motions required to perform manipulation actions. However, these methods lack the representational means and background knowledge to ensure competent action execution. Typically, motions are modelled using complex mathematical *task function* [13]. In a nutshell, for robot motion control, the execution of a manipulation action is that of finding a continuous motion trajectory that optimizes the task function subject to further mathematical constraints. This implies that the control system has no notion that it is performing a pouring action and even worse it does not know how changing the motion parameterization will change the outcome of an action.

In the remainder of this subsection, let us briefly describe and reference two recent extensions of KNOWROB2 that present mechanisms to reasoning about robot motion control.

Tenorth *et al.* [14] present some of the mechanisms to reason about motions. Using the example of pouring liquids, the knowledge base represents and reasons about constraint-based motion descriptions using generic motion templates. Additionally, the inner knowledge base infers the correct placement of control features and frames on the objects involved in the action using CAD-model reasoning.

Fang *et al.* [15] learned models that relate the execution context of pouring tasks to the appropriate constraint-based motion descriptions. To this end, the authors encoded human pouring demonstrations acquired in an interactive physics-based simulation as episodic memories, and applied Random Forest Regression to extract the generalized knowledge required to context-sensitively parametrize pouring motions.

V. LOGIC-BASED LANGUAGE

The representations and mechanisms provided by the hybrid reasoning kernel are very heterogeneous. Also, the accomplishment of more complex reasoning tasks requires the combination of different inference mechanisms. In addition, the representation of knowledge through large unstructured sets of relations or predicates lacks structure and modularity, which makes knowledge engineering difficult.

To facilitate the use of heterogeneous representations and reasoning mechanisms, KNOWROB2 provides a uniform logic interface to the hybrid reasoning kernel. This interface language presents the hybrid reasoning kernel to the programmer as if it were a purely symbolic knowledge base.

The interface language takes an object-oriented view, in which everything is represented as if it were entities retrievable by providing partial descriptions for them (i.e., the *entity* predicate). Entities that can be described include objects, their parts, and articulation models, environments composed of objects, software components, actions, and events.

Particularly important for competent robot manipulation are the events that occur and the state properties that hold at certain stages of action execution (predicates *occurs* and *holds*). The state of the world, and the internal state of the robot continuously changes, and therefore the outcome of all rules attached to this information. The robot may need to plan the next action, for example, during clearing a table, and infers that the remains in the cup need to be poured into the sink, that it has to carry the cup to the sink, and that it has to tilt it and wait a few seconds until the remains were poured into the sink. Temporal predicates express facts about occurrence of events using event logics, and changes in the world state using time dependent relations. The fact that some substance was poured at 11am, for example, can be written as $occurs(pouring_0, 11\text{am})$, or the fact that the cup is empty afterwards can be written as: $holds(empty(cup_0, true), 11\text{am})$.

The task-level control, perception, and motion control components of a robotic agent need to reason about entities with partial descriptions that detail different aspects. This is facilitated by the KNOWROB2 entity description language that allows to partially describe entities in terms of their symbolic, or subsymbolic properties. The description of a cup, from the perspective of the perception component, can be represented in the entity description language as:

```
entity(Cup, [an, object, [type, cup],
    [shape, cylinder], [color, orange]])
```

While the controller-centric representation of the same cup could be written as:

```
entity(Cup, [an, object, [type, cup],
    [proper_physical_parts, [an, object,
        [type, handle], [grasp-pose, G-pose]]]])
```

The interface language is comparable to other query languages for symbolic knowledge bases such as *SPARQL* [16] and *nRQL* [17]. These query languages are quite powerful but lack the integration of sophisticated reasoning methods such as the simulation-based reasoning employed by KNOWROB2.

Many relevant inferences require to combine information of different type, and from different sources. Before grasping a cup, for example, the robot may ask *can I grasp the cup from my current position*, and needs to infer possible grasping points using the perceived object pose, read the position of some joints, and infer the ability to grasp using inverse kinematics. High volume data such as sensor data can not be directly represented in a symbolic knowledge base. Instead, special purpose procedures must be used for querying the data (e.g., reading a sensor value). KNOWROB2 allows to attach procedures to so called *computable relations* [1] that transparently integrate non symbolic data into the reasoning process, and make queries appear to users as if they are working with a symbolic knowledge base. A red cup, for example, can be represented by the entity description $[an, object, [type, cup], [color, red]]$, and its existence can be inferred using the *entity* predicate which internally takes into consideration the robot's internal data structures such as the object hypotheses generated by the perception system.

VI. LEARNING FROM EPISODIC MEMORIES

The *NEEMs* collected in the *episodic memories knowledge base* are used to learn general knowledge. While classical data-centric learning approaches can achieve successful results for some cases (e.g., [18]), the robots are still far from executing human-scale manipulation tasks. One possible approach to achieve this level is to boost existing approaches with symbolic-level structured knowledge so that robots can formalize learning problems by themselves and generate training datasets from the available *NEEMs*.

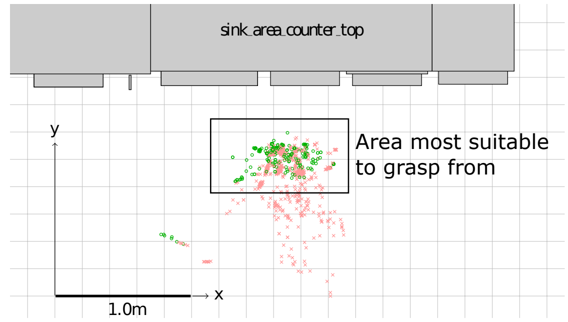


Fig. 6: Robot base poses from episodes during which the robot succeeded (indicated by green dots), and failed (indicated by red dots) to grasp a cup from the counter top of a sink.

If the robot perceives a cup on the table while cleaning the table, it can retrieve a collection of positive *NEEMs* for picking up cups using the following query:

```
?- findall(Act, entity(Act, [an, action,
        [type, grasping_something],
        [object_acted_on, [an, object, [type, cup]]]
   ]), Acts).
```

Then, using the following query, it gets features from these episodes:

```
?-findall(RelPose, (member(Act, Acts),
      entity(Act, [an, action, [object_acted_on, Obj]]),
      occurs(Act, [Begin, _]),
      holds(pose(Robot, RobotPose), Begin),
      holds(pose(Obj, CupPose), Begin),
      transform_relative(RobotPose, CupPose, RelPose)),
      Features).
```

The features in the above query consist the pose information of the robot w.r.t. the cup at the beginning of the grasp.

Given this dataset, the robot can learn in which limited region the grasp is likely to succeed (depicted in Figure 6). To this end, KNOWROB2 offers an interface to the Weka Machine Learning framework [19] and thus to many learning learning algotrihms such as *Gaussian Mixture Models*. This interface can be used to apply standard methods for filtering irrelevant features, training and saving classifiers and regressors, and predicting outcomes given new data.

Having machine learning techniques integrated in a robotic knowledge processing system offers capabilities such as what kind of data is needed for a learning domain problem, how to integrate different datasets together, and how to benchmark the learning results. For instance, it is not always the case that the latest generated trajectory is better than the previous ones in an iterative learning case. Using KNOWROB2, robots can reason about such circumstances and instead of using the latest generated trajectory, use the less-time-taking one. For example, by having some trajectories learned from simulation available, robots can reason and benchmark on them with respect to factors such as durations and lengths, then execute the one that fits the most in the current situation.

VII. Related Work

Knowledge processing for robots has been a research topic for decades. The Shakey robot [20] already had an internal representation of its environment. Since then, robots advanced to perform more realistic tasks that rely on complex scene representations and a massive increase in the knowledge about actions and objects. In the AI community, many methods have been developed that focus on individual inference problems. Allen's time interval algebra [21] focuses on reasoning about time intervals, and languages such as PDDL [22] are designed to reason about plans and goals. Robot centered approaches, such as [23], often lack support for relevant domains such as temporal reasoning, or have incomplete representations of spatial and object information. Encyclopedic knowledge bases, such as Cyc [24], often lack the necessary depth in representing relevant concepts such as the representation of manipulation activities. Other authors tried to automate the acquisition of encyclopedic knowledge by extracting information from web resources [25], and thereby providing knowledge in specialized areas. In the robotics community, much of the research has focused on modelling uncertainty using probabilistic models [26] that are usually tailored to a single modality such as perception, or localization, and lack clear semantics. More sophisticated models combine spatial with grounded semantic representations [27], or geometric and conceptual spatial representations with planning and learning techniques [28].

VIII. Conclusion

In this paper we have given an overview of the knowledge processing framework KnowRob2. Clearly, the performance of the framework should be measured in terms of the reasoning tasks that it can perform and the knowledge it can acquire. We have reported on several reasoning tasks that KnowRob2 can accomplish and that set it apart from other robot knowledge processing systems. These tasks include the simulation-based reasoning, visual reasoning using the inner world model, as well as learning from episodic memories. Taken together these mechanisms can greatly improve the competence of robotic agents in accomplishing manipulation tasks. Many of these capabilities can be tried out using the web-based knowledge service openEASE.

References

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

[2] E. Tulving, "Episodic and semantic memory 1," *Organization of Memory. London: Academic*, vol. 381, no. e402, p. 4, 1972.

[3] 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.

[4] T. Ziemke, "The construction of 'reality' in the robot: Constructivist perspectives on situated artificial intelligence and adaptive robotics," *Foundations of Science*, vol. 6, no. 1, pp. 163–233, 2001.

[5] P. H. Winston, "The right way," *Advances in Cognitive Systems*, vol. 1, pp. 23–36, 2012.

[6] 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.

[7] H. Celikkanat, G. Orhan, and S. Kalkan, "A probabilistic concept web on a humanoid robot," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 2, pp. 92–106, 2015.

[8] G. Hesslow, "The current status of the simulation theory of cognition," *Brain Research*, vol. 1428, pp. 71 – 79, 2012, the Cognitive Neuroscience of Thought.

[9] M. Haruno, D. M. Wolpert, and M. Kawato, "Multiple paired forward-inverse models for human motor learning and control," in *Proceedings of the 1998 conference on Advances in neural information processing systems II*. Cambridge, MA, USA: MIT Press, 1999, pp. 31–37.

[10] P. J. Hayes, *The Frame Problem and Related Problems on Artificial Intelligence*. Stanford University, 1971.

[11] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *International Joint Conference on Artificial Intelligence*, 2015.

[12] L. Morgenstern, "Mid-Sized Axiomatizations of Commonsense Problems: A Case Study in Egg Cracking," *Studia Logica*, vol. 67, no. 3, pp. 333–384, 2001.

[13] E. Aertbeliën and J. De Schutter, "eTaSL/eTC: A constraint-based task specification language and robot controller using expression graphs," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 1540–1546.

[14] M. Tenorth, G. Bartels, and M. Beetz, "Knowledge-based specification of robot motions," in *Proc. of the European Conference on Artificial Intelligence (ECAI)*, 2014.

[15] Z. Fang, G. Bartels, and M. Beetz, "Learning models for constraint-based motion parameterization from interactive physics-based simulation," in *International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, 2016.

[16] E. Prud'Hommeaux and A. Seaborne, "SPARQL query language for RDF," World Wide Web Consortium, Recommendation REC-rdf-sparql-query-20080115, January 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[17] V. Haarslev, R. Möller, and M. Wessel, "Querying the semantic web with racer + nrql," in *In Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL'04*, 2004.

[18] H. Dang and P. K. Allen, "Robot learning of everyday object manipulations via human demonstration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 1284–1289.

[19] G. Holmes, A. Donkin, and I. H. Witten, "Weka: A machine learning workbench," in *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*. IEEE, 1994, pp. 357–361.

[20] N. J. Nilsson, "Shakey the Robot," AI Center, SRI International, Menlo Park, CA, USA, Tech. Rep. 323, 1984. [Online]. Available: http://www.ai.sri.com/shakey/

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

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

[23] M. Thielscher, "Representing the knowledge of a robot," in *International Conference on Principles of Knowledge Representation and Reasoning*, Breckenridge, CO, USA, April 12–15 2000, pp. 109–120.

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

[25] F. Wu and D. S. Weld, "Autonomously semantifying wikipedia," in *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. New York, NY, USA: ACM, 2007, pp. 41–50.

[26] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge: MIT Press, 2005.

[27] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 955–966, 2008.

[28] J. L. Wyatt, A. Aydemir, M. Brenner, M. Hanheide, N. Hawes, P. Jensfelt, M. Kristan, G.-J. M. Kruijff, P. Lison, A. Pronobis, K. Sjöö, D. Skočaj, A. Vrečko, H. Zender, and M. Zillich, "Self-understanding and self-extension: A systems and representational approach," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 282 – 303, December 2010.