KnowRob$_{SIM}$ — Game Engine-enabled Knowledge Processing Towards Cognition-enabled Robot Control

Andrei Haidu, Daniel Beßler, Asil Kaan Bozcuoğlu, Michael Beetz
{ahaidu, danielb, asil, beetz}@cs.uni-bremen.de

***Abstract*— AI knowledge representation and reasoning methods consider actions to be blackboxes that abstract away from how they are executed. This abstract view does not suffice for the decision making capabilities required by robotic agents that are to accomplish manipulation tasks. Such robots have to reason about how to pour without spilling, where to grasp a pot, how to open different containers, and so on. To enable such reasoning it is necessary to consider how objects are perceived, how motions can be executed and parameterized, and how motion parameterization affects the physical effects of actions. To this end, we propose to complement and extend symbolic reasoning methods with KnowRob$_{SIM}$, an additional reasoning infrastructure based on modern game engine technology, including the subsymbolic world modeling through data structures, action simulation based on physics engine, and world scene rendering. We demonstrate how KnowRob$_{SIM}$ can perform powerful reasoning, prediction, and learning tasks that are required for informed decision making in object manipulation.**

I. Introduction

Goal-directed manipulation of objects and substances is a hallmark of intelligent agency [1]. In human evolution, the size of the human brain had to increase drastically to meet the requirements for competent object manipulation. To meet the new challenges, the human brain also had to develop new cognitive capabilities and substantially advance existing ones. Examples of such cognitive capabilities include the development of more powerful representations of actions [2], the co-development of language and action [3], and more powerful mechanisms for the mental imagination of actions [4].

We believe that symbolic reasoning is necessary but not sufficient to realize the full range of reasoning capabilities needed for mastering object manipulation tasks. Thus, we propose to complement the symbolic knowledge representation and reasoning system with an additional knowledge system that can perform subsymbolic reasoning tasks including visual imagination, mental simulations of actions, learning from observation, and semantic retrieval of subsymbolic information about objects, substances, actions, motions, and their physical effects (see Figure 1).

The key strength of the KnowRob$_{SIM}$ knowledge processing system is that it unifies a collection of bleeding edge reasoning mechanisms through symbolic representations at a level of detail which enables control-level reasoning for embodied agents. Robotic agents are supposed to assert their belief state about the world as a KnowRob$_{SIM}$ 'world state' in a game engine by accessing the engine's world state data structure, annotating relevant data structures with symbolic names, and asserting symbolic facts about these data structures and their relations. The reasoning mechanisms can view the world state as a (virtual) symbolic knowledge base, where the mechanisms of the game engine, namely data structure retrieval, physical simulation, and rendering, are the essential reasoning mechanisms.

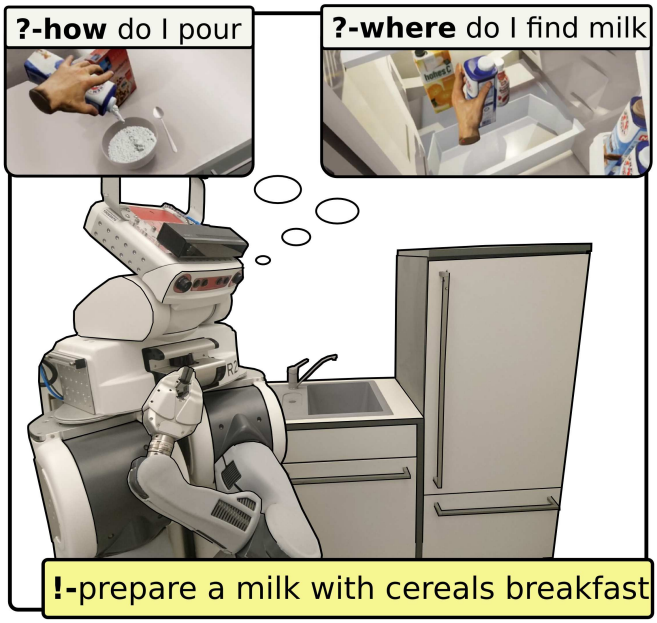

Fig. 1: PR2 using KnowRob$_{SIM}$ to learn how to prepare breakfast.

To realize this motor cognition reasoning system, this paper makes the following key contributions:

- a computational infrastructure, called the KnowRob$_{SIM}$ inner world, that can generate and maintain an approximate, photorealistic and physics enabled, copy of the robot's environment. This data structure is then further used as a hybrid knowledge base with a semantic information retrieval language. The infrastructure provides and maintains semantic environment and scene models with detailed position, orientation, and state information. The infrastructure also realizes a virtual hybrid symbolic/subsymbolic knowledge base for the motion and image level of robot control using the world state data structures.

Authors are with the Institute for Artificial Intelligence, University of Bremen

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

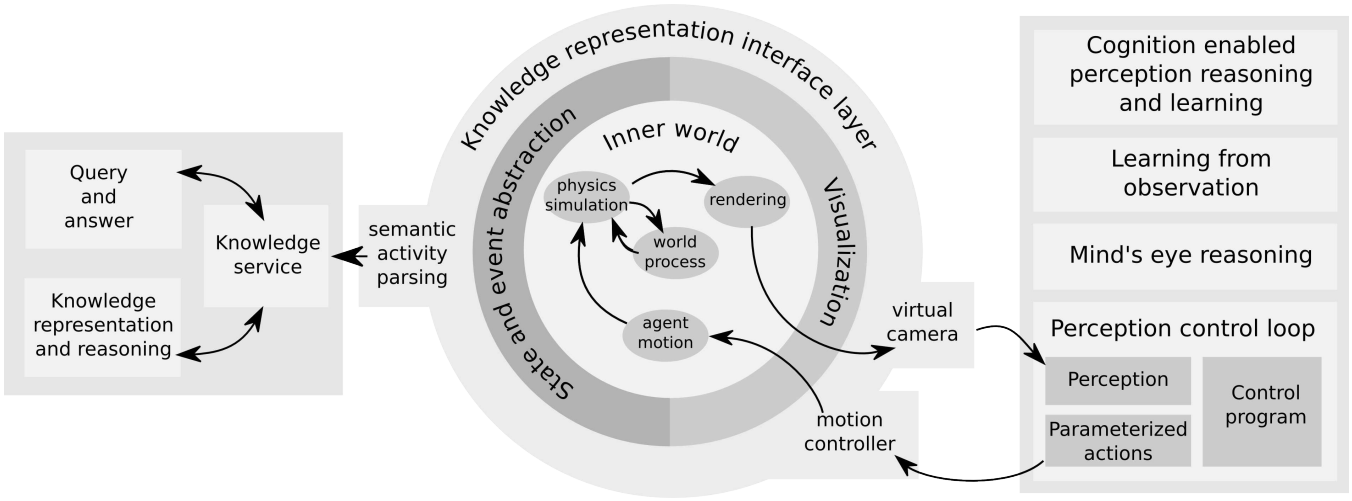

Fig. 2: Software architecture of the game engine-enabled knowledge processing.

- a mechanism for parsing episodes in the KNOWROB$_{\text{SIM}}$ inner world into hierarchical symbolic action models represented in a first-order time interval logic.
- an infrastructure for capturing images from an envisioned world with the intention of performing various reasoning tasks, such as occlusion analysis and other similar visual aspects.
- an interface for executing vision-guided control programs in the KNOWROB$_{\text{SIM}}$ inner world using programs similar to those executed on real robots.
- a knowledge acquisition component that can learn commonsense and naïve physics knowledge from human action demonstrations in a virtual environment.

II. OVERVIEW

Figure 2 shows the software architecture of the game engine-based system which is part of the KNOWROB 2.0 knowledge processing framework [5]. For the purpose of this paper the game engine handles one or more agents that can be controlled from outside, and an environment with objects and substances that evolve according to the laws of physics, as implemented through the rigid body and particle based physics engines.

The core component of the knowledge processing system, called *inner world* performs a *basic loop* with the following steps: (1) update the agent's dynamic state (e.g., the control signals send to the joint motors of a robotic agent), possibly there will be a world process that also changes the state of the world (e.g., other agents not under control), (2) the world state is updated based on the current state and the control inputs generated by the agent and the world process (according to the laws of physics implemented by the physics simulation), and (3) visually rendering the updated world state. This loop evolves the virtual world state, which is accessible through the application programming interface of the game engine.[1]

[1]Our system is implemented using the game engine Unreal Engine 4[6].

The core is enclosed by the *state and event abstraction layer*. The state and event abstraction layer abstracts the world state into a representation that facilitates naive physics and qualitative reasoning [7]. To do so, the layer automatically computes physical and spatial relations as well as force-dynamic events. An example of a naive physics relation is the *supported-by* relation. An object O is supported by object S if O is physically stable, in contact with, and above S. Other relations are, doors and containers being open or closed, substances being spilled, etc. Touching is an example of a force-dynamic event, which happens when the hand makes contact with another object. The detection of force-dynamic events is essential for the recognition and the segmentation of actions into motion phases. For example, grasping and lifting an object is characterized by the hand touching an object, keeping contact with the object, and the object losing the contact to its supporting surface.

State and event abstraction is used by the *knowledge representation interface layer* that provides modules for *activity parsing and recording*, *query answering*, *semantic environment model extraction*, and *virtual image capturing*. The activity parsing and recording module takes the stream of time-stamped world states together with the abstract world states, force-dynamic events and motion events from the agent and generates a symbolic activity representation stated in a first-order time interval logic. The symbolic activity representation is time synchronized with subsymbolic stream data that includes the agents and objects poses and shapes, and images. The semantic environment extraction module maps over the data structures of the world state and asserts for each relevant object and its parts symbolic names, the label category, the part-of hierarchy, the articulation chains and models, and other relevant symbolic relationships (Figure 3). The virtual image capturing module can place cameras in the game environment, access their scene's built in (deferred) rendering information, such as color, depth, specular etc. image data, and further extend it by segmenting the image into objects and labeling them with their corresponding

symbolic names.

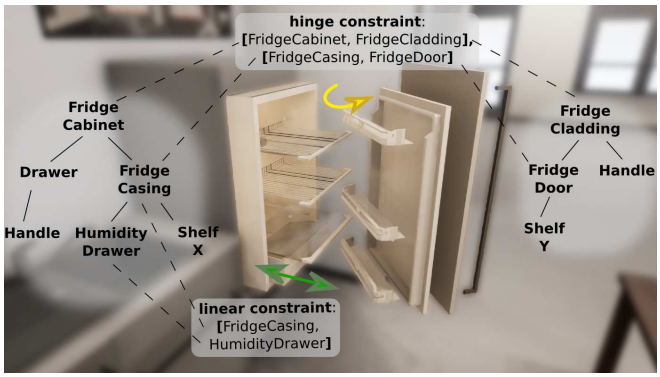

Fig. 3: Semantic environment model of a fridge.

Around the knowledge representation module layer is the cognitive capabilities layer. This layer includes the *KnowRob query answering service*, the *robot perception component*, a component for the *mental simulation of actions*, a component for *learning from virtual reality demonstrations*, and another for *learning action models from virtual experience data*.

In the rest of this paper we explain the components of game engine-enabled knowledge processing in more detail.

III. Inner World Implementation

Modern game engine technology has reached an unprecedented level of sophistication and efficiency. This technology, used in domains such as computer graphics, video games, or animation movies, typically employs physics engines: software providing an approximate simulation of certain physical systems, such as rigid body dynamics (including collision detection), soft body dynamics, and particle simulation. Physics engines such as Nvidia's PhysX and FleX have already reached a level of performance that allow us to simulate manipulation actions with an accuracy and realism sufficient to develop software for many subproblems in robot control in simulation rather than requiring real physical robot experiences [8]. Nowadays, game engines can simulate and render complex scenes with update rates of up to 90*hz* (typically required by VR applications), these simulated environments can be extended to large open worlds (hundreds of square kilometers) while maintaining the accuracy of being able to show realistic leaves on trees and single blades of grass on fields [9]. Finally, the computational resource requirements of physics engines have dropped such that they can run on devices such as smart phones and within modern web browsers.

The combination of these developments have brought us to a point where it is easily possible to run physics engines at execution time as components of robot control programs. This makes it possible for robots to maintain a photorealistic model of their environment (see Figure 3) with approximate physics simulation. Having access to the data structures of such a model, it allows robots to retrieve detailed subsymbolic information about their world, to mentally look at scenes, and to simulate action executions. The size of the environments, the number of objects and the level of detail are modeled in a way that goes far beyond what symbolic knowledge services could provide so far.

An advantage that we gain by combining game engine-based knowledge processing with symbolic knowledge processing is that robots can construct problem specific abstractions on the fly. For example the position of the cup can be retrieved as a contact with a supporting surface (e.g. it is on the table). Or, if required, the knowledge processing system can infer the detailed pose, weight, fill level, etc. of the cup.

A limitation of using physics engines from game engines is having many simultaneous physically stable contacts, as required for grasp stability analysis [10]. For such applications due to the approximations of the effects simulations can easily cause numerical instabilities. Precise simulations also require very accurate models of articulated objects (e.g. robots). Inaccurate modeling can quickly lead to numerical instabilities, for example due to self collisions. On the other hand, since technology leaders [8] start to target robot simulation and learning as application domains, it promises substantial advances in robot specific simulation requirements.

IV. State and Event Abstraction

In the virtual world, the physics governing the evolution of the world is implicit: in each simulation cycle numerical physics laws are applied to everything to compute the state of the next time step. In contrast, the physical interpretation of perceived scenes is deeply integrated into the perception system of humans. If we observe a heavy object without support not falling down, we immediately conclude the object to be fixed to the wall behind it. If an object is on top of another one without moving we conjecture that it is supported by the other object. We understand the world by generating explanations about the forces that interact between objects and agents.

The role of force interaction between entities is also put forward in cognitive linguistics by Talmy [11] who proposes to characterize situations and actions through concepts such as the exertion of force, resistance to such exertion and the overcoming of such resistance, blockage of a force and the removal of such blockage, and so forth. Force dynamics analyzes "causing" into finer primitives and sets it naturally within a framework. Thus, force dynamic states and events can build a strong foundation of a naive physics understanding of the world.

To realize a similar conceptual apparatus in our knowledge processing system we have to automatically assert the respective relations based on the monitoring of what happens in the physics simulation. For this we have various nodes built in the game world which continuously monitor such relations. When a corresponding event happens these will trigger an assertion of the symbolic representation of the event into the knowledge base.

One basic monitor node is listening for contact events, namely, whenever relevant objects are beginning or ending a physical contact. This node can be further extended for more

specific events, such as the *supported-by* event, which additionally checks if the supported object is stable and on top of the supporting one. For grasping events, the monitoring node checks for contacts between the palm and/or fingers and the grasped object and if the object is secured in the hand. Another example is a monitor for the states of articulation models, for example triggering whenever a door/drawer gets opened/closed, or a knob/button gets turned/pressed etc.

Other mechanisms for translating data into symbolic abstractions are the interpretation of hand-object interactions, the classification of grasp types, the interpretation of motion patterns (e.g., walking), and the abstraction of motion features (e.g., keeping containers upright). In addition, objects and their structures need to be interpreted. This includes the inference of part-of hierarchies of objects (partonomies) and the articulation models of objects such as doors, drawers, knobs, lids, and the like.

V. The Knowledge Representation Interface Layer

The next layer builds up the more complex hybrid knowledge representation structures and the mechanisms for query answering.

A. Semantic Activity Parsing

A key mechanism of the game engine-enabled knowledge system is the automatic translation of actions and events simulated in the inner world into the appropriate hybrid, symbolic-subsymbolic activity and event representations.

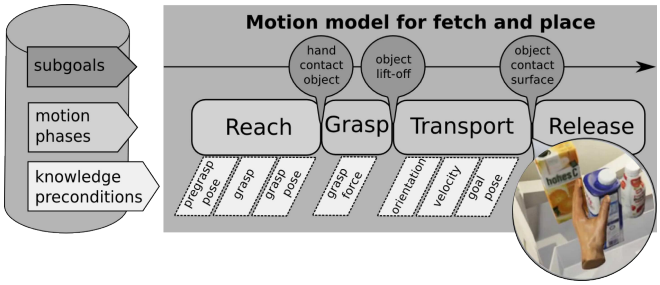

Fig. 4: Flannagan's fetch and place motion model.

Cognitive scientists study the organization of actions in terms of motion phases and how expectations about perceived events and prediction enable human motor cognition help to accomplish manipulation tasks so competently. Figure 4 shows the model proposed by Flannagan et *al.* [12] that forms the basis of our knowledge representation. It structures actions into motion phases, where the phases have subgoals, which are force dynamic events that also generate distinctive sensory feedback. Motor cognition and robot control conceptualize the parameterization and optimization of motion parameters that enable the robot to cause the desired physical effects and avoid the unwanted ones.

Furthermore, a study in cognitive linguistics [11] suggests that the semantics of action verbs can be grounded in force dynamic events and that the semantics of action verbs can be defined through verb-specific temporal patterns of force

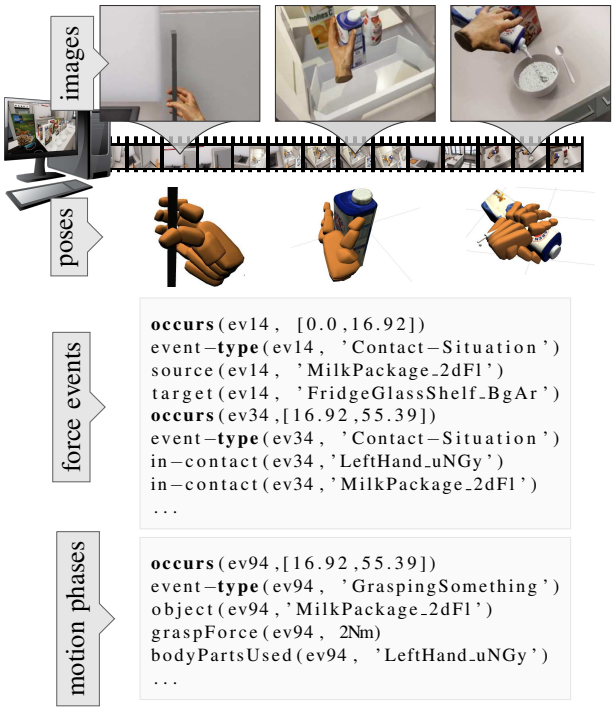

Fig. 5: Representation of a fetch and place activity episodes in KnowRob$_{SIM}$.

dynamic events. This means by detecting force dynamic events and classifying them an activity parser interpreting the evolution of the inner world can automatically recognize actions, categorize them, and decompose them into their motion phases.

The inferred action models are then represented as statements in a first-order time interval logic. An activity representation for a fetch and place episode is depicted in Figure 5. It shows a human operator in a virtual kitchen environment fetching milk from a fridge, pouring it into a bowl, and putting it back into the fridge again.

The activity semantics can not be monitored directly. Instead, we detect force dynamic events in the physics engine of the virtual reality such as the human hand getting in contact with the fridge door handle, the milk package ending contact with the shelf, and milk particles moving from the container to the bowl. The monitored events yield assertions in the symbolic knowledge base about the occurrence of events, their types, and entities that were involved such as the objects in contact or the milk particles that were poured into the bowl. Facts about events may refer to symbols that describe the virtual reality world such as the objects present, their parts, and the self model of the human. The first-order logics based representation of virtual worlds directly maps to data structures used by the game engine which allows to synchronize the virtual world with its semantic description, and to spawn new worlds according to facts in the knowledge base.

Temporal patterns of memorized force dynamic events determine the motion phases of the recorded activity. Reaching motions are indicated by objects getting in contact with

the hand, and a (successful) grasping motion occurred if the object stays in contact to the hand while leaving its supporting surface. The human operator transports the milk to pour from it such that the first transport motion phase is determined according to the force dynamic event that corresponds to the first particle leaving the milk package. Further, we can state that a pouring motion occurred starting from the first particle, and ending with the last particle leaving the milk package. Finally, we can state that the second transport motion phase ends with the milk package being in contact with the supporting plane of the shelf again.

The symbolic representation of motion phases is used as a search index into a noSQL data base that stores high volume data such as poses of objects and images generated by the rendering engine. High volume data pieces that correspond to motion phases can be accessed by navigating through the symbolic representations. To this end, we use procedural hooks in the symbolic knowledge base that define particular ways of abstracting the high volume data on demand for reasoning tasks at hand. This is, for example, useful for qualitative spatial reasoning where the existence of some spatial relation is only relevant for a particular time interval (e.g., during some motion phase). This data can further be used as an evidence for more specific classification of the recorded activity (e.g., if the particles were poured or scooped).

B. Query answering

Another essential information service provided by KNOWROB$_{SIM}$ is the answering of semantic queries. Many queries have a similar structure. They infer an action (or object) using a symbolic high-level query, then they use the name of the action to infer the time instant when the action started and terminated and use the time instants in order retrieve subsymbolic information (such as the pose of objects and agents at the respective time instants or motions during the respective time intervals).

Let us consider the following example query to get a better intuition of the nature and the reasoning power of these queries:

```
:- entity(FP, [an, event, [type, 'Contact-Situation'],
     [in-contact, [an, object, [type, 'EndEffector'], EE]],
     [in-contact, [an, object, [name, O]]]]),
   entity(_, [an, event, [type, 'Transportation'],
     [object, [an, object, [name, O]]], [during, FP]]),
   occurs(FP, [Begin,End]),
   pose_at_time(EE, Begin, [Pos,Rot]).
FP = 'Contact-Situation_ud4f', O = 'MilkPackage_2dFl',
EE = 'LeftHand_uNGy', [Begin,End] = [16.92, 55.39],
[Pos,Rot] = [[0.1,0.3,1.2],[1.0,0.0,0.76,0.0]] ...
```

This query navigates through experienced situations FP that are subsumed by the partial description provided, and retrieves the time interval during which the situation occurred. It is important to note that partial descriptions expand to first-order logic formulas, and that possible bindings for occurring variables are searched such that the corresponding formula holds (i.e., entities that are subsumed by the description). The subsumption hierarchy further allows to query for more general concepts, such as end effector, rather then for more specific ones, such as the simulated hand.

Entity descriptions include an ordered sequence of symbolic constraints that must be satisfied by matching entities. In this case, matching entities must be a contact situation between some object O and the end effector in contact during which the object was transported. The *pose_at_time* predicate represents the subsymbolic data (the pose) of the end effector at the given timestamp. The temporal predicate *during* shows another strength of query answering in KNOWROB$_{SIM}$: its existence is not known in advance but can be proven at query time using rules that operate on facts in the knowledge base (in this case using Allen's interval algebra [13]). Predicate computation rules do not necessarily need to strictly follow first-order logic formalism and may only do a general computation but in the end need to abstract some data to a predicate symbol in the knowledge base. This mechanism allows KNOWROB$_{SIM}$ not only to reason with heterogeneous and high-volume data sources, such as the stream of rendered virtual reality images or the recorded pose of the human operator over time, but also to interface reasoning techniques such as temporal reasoning, or semantic activity parsing.

The example query can also be interpreted as a parser for fetch and place activities. It is surely a simplified view to classify any contact situation between an end effector and an object during which the object was transported as fetch and place activity, but it already captures the characteristic force dynamic events. Such simple rules serve the purpose to define some rather abstract facts about occurring situations, while more specialized activity parsers can refine these general descriptions to more specific ones that describe finer details. We could, for example, state that fetch and place actions during which some particles were transported from one container to another can be classified as pouring activity. This activity parser could be written as query in the following way:

```
?- entity(FPP, [an, activity, [type, 'FetchAndPlace'],
     [object, [an, object, [name, Container],
                           [type, 'Container']]]]),
   entity(_, [an, event,
     [type, 'ParticleTransportation'], [during, FPP],
     [from, [an, object, [name, Container]],
     [to,   [an, object, [type, 'Container']]).
```

One of the strong aspects of query answering in KNOWROB$_{SIM}$ is that knowledge can be inferred on demand from raw data available (such as the record of hand and object poses over time). During pouring activities, for instance, the milk package needs to be tilted and held such that the fluid can flow into the bowl without spilling it onto the counter top. Motion parameters such as the tilting angle are not explicitly logged but can be computed at query time with some spatial reasoning hooks into the symbolic knowledge base. The tilting angle is computed as the angle between container and horizontal plane, and for specific time instants (in this case the end of the pouring motion) it can be computed with following query:

```
?- entity(Pouring, [an, event, [type, 'Pouring'],
     [object, [an, object, [name, Container]]]]),
```

```
    occurs(Pouring, [_,End]),
    holds( tiltingAngle(Container,D), End ).
D = 43.75^Degree, ...
```

C. Virtual Camera

Another workhorse for reasoning about manipulation actions is the KNOWROB$_{SIM}$ virtual camera infrastructure. This infrastructure enables the robot to assert a 6d pose for a virtual camera and capture an image of the rendered virtual world. As depicted in Figure 6, the camera returns not only the captured image but also the ground truth segmentation for each object depicted in the scene together with the respective symbolic object name. An example is shown in Figure 6, which depicts the captured image (scene), the ground truth of object segmentation (object masking), the overlapping segments of the milk carton and the bottle in front of it (overlapping area), and the non-occluded part of the milk carton (visible texture). These different views of captured images are computed automatically and very efficiently by the rendering mechanism of the game engine and can be greatly accelerated through the use of GPUs.

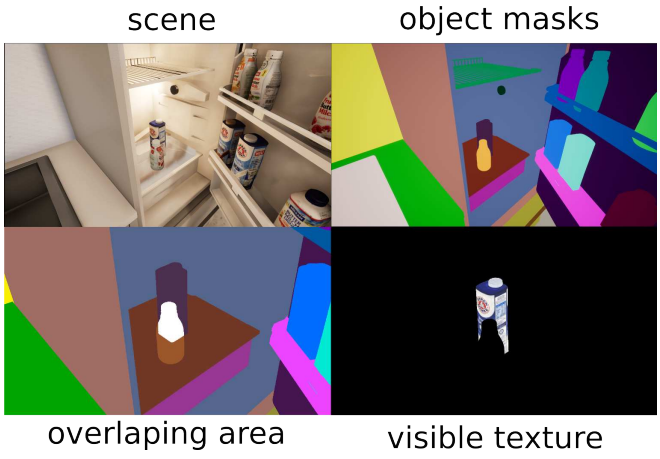

Fig. 6: Virtual camera image of the milk carton in the fridge.

Using the virtual camera infrastructure KNOWROB$_{SIM}$ can answer queries such as suppose we place the camera at position x and point it into the direction d, then KNOWROB$_{SIM}$ can answer queries such as (*) which object occludes the milk carton?, (*) how much of the milk carton is visible?, (*) could the milk carton be detected with a SIFT feature based object detector?, (*) is the text on the milk carton readable?, and so on.

D. Executing plans in the inner world

The fourth capability is the percept-guided execution of robot plans. Where we consider plans to be robot control programs that can be executed, reasoned about, and modified [14]. KNOWROB$_{SIM}$ provides a user-provided robot plan interpreter with a stream of images captured with the virtual camera being positioned at the respective pose of the real robot camera. This way KNOWROB$_{SIM}$ can mimic the images captured by the robot in the real world. Further the percept-guided execution can issue parameterized motion specifications that can be realized through the agent motion control of the inner world. Thus, by issuing a parameterized motion control command, executing a simulation step, rendering the resulting world state, and capturing the image from the robot's camera point of view, the mental plan simulator can run a very detailed and realistic perception action loop of the robot. In addition, the evolution of the world state when executing a plan can then be segmented and interpreted through the semantic activity parsing described in Section V-A generating a symbolic knowledge base of the imagined activity.

In the current state of implementation, we can only execute simplified versions of mental simulation, namely the controlled motion of an object and the the control of a simple robot hand. We are currently implementing simulation and virtual control models for complete and complex mobile robot manipulation platforms.

VI. REASONING, PERCEPTION, AND LEARNING

Let us now consider how we can realize cognitive capabilities with the KNOWROB$_{SIM}$ representation and reasoning infrastructure.

A. Learning from Observation

Observing human-scale manipulation tasks in game environments enables the possibility to collect a variety of commonsense and naive physics knowledge. This knowledge is intensively used by humans to accomplish their tasks successfully, and can be used by robotic agents to fill knowledge gaps caused by incomplete instructions. For example, the robot can learn the motion constraints for carrying opened containers without spilling their content. These constraints can be further parameterized to depend on the fill level of the containers, or on the tasks being executed.

By monitoring and observing the force-dynamics states and events we can learn generalized action and motion models. Such as, the standing location of the robot in order to successfully pick up an object, by running multiple simulations and learning a classifier that predicts success [15]. The recognition of force interactions patterns between entities leads to the characterization and segmentation of various actions. For example a generic fetch pattern, from a fetch-and-place action, would have the following interaction sequence: (1) object in contact with a surface, (2) hand in contact with object, (3) object attached to hand, (4) contact broken between the surface and the object. We can recognize the action of getting the milk out of the fridge by extending the aforementioned sequence with opening and closing the fridge door interaction. By computing multiple sequences of relevant force dynamic states and events, the reasoning infrastructure can recognize more complex actions and segment them into relevant motion phases.

B. Mind's eye reasoning

1) Learning action-related concepts: Robots that do everyday manipulation tasks can immensely benefit from being able to predict consequences of their actions just before the

execution. An advantage of having a "mind's eye" simulation with the inner world model is being able to predict outcomes before execution. The main requirement for this is being able to operate in a physics-enabled and photorealistic inner world model. Currently, there are still ongoing developments on generating robot models with complex kinematic structures in game engines. In [16], the authors demonstrate how a robot can formalize its simulation goals using PROLOG queries inside KNOWROB$_{\text{SIM}}$ and, in response, the knowledge base spawns the simulation with desired world and parameters, and executes the corresponding plan.

Representing self and being able to operate with own control and planning systems in an inner world model has various advantages for high-level planners. First it can give hints about which parameters will lead to a successful outcome before the actual execution. Second, since robots use similar control executives, similar episodic memories are generated from simulation for later reasoning. Lastly, it can generate data for training of a classifier or other machine learning tools.

2) Robot perception: Robots can also make use of their "mind's eye" to boost perception abilities and increase resolution of their object belief-states. Spawning and maintenance of object belief states in gaming environments will offer many possibilities such as having a constant connection between how a scene looks like after manipulation and how a robot "imagines" it to look like. Additionally, it can also estimate object poses where perception fails due to reasons such as occlusion or losing sight.

VII. EVALUATION, DISCUSSION, AND RELATED WORK

A. Evaluation

As an AI knowledge representation and reasoning (kr&r) system, an adequate evaluation for KNOWROB$_{\text{SIM}}$ is to assess it with respect to the desired properties of such kr&r systems, to show where KNOWROB$_{\text{SIM}}$ improves on these properties, and explain why. Empirical evaluations are not suitable means for evaluation because they evaluate the tools that KNOWROB$_{\text{SIM}}$ is built upon, that is the simulation and rendering methods, but not KNOWROB$_{\text{SIM}}$ itself. Frequently used desired properties of kr&r systems are: representational adequacy, inferential adequacy, inferential efficiency, and acquisitional efficiency. Representational adequacy assesses the ability of kr&r systems to represent the knowledge needed for manipulation control; inferential adequacy is concerned with the ability to infer answers to relevant queries from the represented knowledge; inferential efficiency assesses the computational resources needed to answer queries, and finally, acquisitional efficiency considers how well the kr&r system supports the acquisition of new knowledge.

KNOWROB$_{\text{SIM}}$ is representationally more adequate than AI action representations, because AI representation systems make the so-called atomic state-transition system assumption [17]. KNOWROB$_{\text{SIM}}$ also represents the "continuous" flow of the underlying dynamic system, including motions, instantaneous force-dynamic events, as well as the variations of physical effects caused by variations of motion parameterizations. The representation of these aspects are essential for competent reasoning about object manipulation.

KNOWROB$_{\text{SIM}}$ is also inferentially more adequate because we can formulate and answer queries about manipulation actions that can not be handled by AI action representations. We will give two examples of such queries but any query regarding motions generated by manipulation actions, force dynamic events, and the relation between motions and their effects can not be handled in a kr&r system making the atomic state transition system assumption.

An example, we collect negative training set for a pouring trajectory generator. We can consider a pouring trajectory as a *failure* when there is a spillage over the table. In KNOWROB$_{\text{SIM}}$, we can detect such a spillage by checking whether there exists a contact between a fluid particle and dining table during pouring. Thus, the query in PROLOG syntax is as follows:

```
?- entity(G, [an, activity,
     [type, 'FetchAndPlace'],
     [object, [an, object, [name, O],
                           [type, 'Container']]]]),
   occurs(G, [_, End]),
   entity(E, [an, event, [type, 'Contact-Situation'],
     [in-contact, [an, object,
         [type, 'LiquidTangibleThing']]],
     [in-contact, [an, object,
         [type, 'DiningTable']]]]),
   occurs(E, [SpillTime, _]),
   SpillTime < End, Start < SpillTime.
```

Another example was shown earlier related to Figure 5.

KNOWROB$_{\text{SIM}}$ is also inferentially efficient as it uses the physics simulation and rendering engines as inference mechanisms. Physical simulation and rendering are gpu-accelerated inference mechanisms that scale much better towards realistic action projection than AI approaches that have been shown to generate huge search spaces in state transition graphs. KNOWROB$_{\text{SIM}}$ is also acquisitionally efficient because we can extent creation process for game environment such that it automatically creates the KNOWROB$_{\text{SIM}}$ environment representation as a side effect.

B. Discussion

A number of researchers question the usefulness of simulations as a model for robot behavior and a prediction model for physical effects because simulations are not considered as accurate enough [18]. We believe that this conclusion is not valid for several reasons.

First, humans also often make informative predictions without detailed knowledge of the physics simulation parameters. For example, when humans predict the effects of pouring they do not need to know the viscosity of the fluid inside a bottle. Second, it is often possible to learn or make predictions in state spaces that do not require detailed knowledge of physics. For example, instead of predicting the effects of pouring pancake mix onto an oven in terms of fluid viscosity we can express the prediction model in terms of how the shape of the pancake evolves. In this case, the viscosity is compiled into the growth rate of the pancake size.

Another possibility to transfer knowledge from simulation and apply it in the real execution context is to identify the proper models at execution time. The usage of game engine environments for learning real world robot control is also starting to gain momentum [19].

C. Related Work

Polceano and Buche [20] give a comprehensive review of computational mental simulation, which categorizes proposed approaches with respect to their functional roles and link them to cognitive science research. Ullman et al. [21] propose game engines for intuitive physics, which take distributions over natural scene descriptions, scene images, memories, etc, simulate the physics for a set of scenarios sampled from the respective distribution and generalizes knowledge from the distribution of simulations. Their approach is conceptually the closest to ours. In contrast, KNOWROB$_{\text{SIM}}$ realizes a wider range of cognitive reasoning capabilities and provides a proper integration into a symbolic reasoning framework. Feldman and Narayanan [22] propose the neural theory of language (NTL) that aims at providing a simulation based interpretation of natural language sentences that is based on mentally simulating action verbs. KNOWROB$_{\text{SIM}}$ shares the concentration on actions but uses a much more fine-grained simulation model that is based on physics simulation rather than a Petri net simulation. Billing et al. [23] propose robots that generate an internal simulations of the sensory-motor interactions with the environment and use these internal simulations to generalize and reproduce the demonstrated behavior through imitation learning. Again, KNOWROB$_{\text{SIM}}$ provides a broader range of cognitive reasoning capabilities and a proper interface to logic-based knowledge representation. Wächter et al. [24] propose a framework allowing robots to solve complex tasks from natural language in dynamic environments. The framework provide predicates to abstract away sensory data into discrete symbolic one, describing states as an AI representation. Our predicates contain subsymbolic data as well, such as the pose of the hand when grasping, the physical force values, furniture states include opening angles, the amount of a substance present in a container.

VIII. Conclusions

In this paper we have proposed KNOWROB$_{\text{SIM}}$, a knowledge processing infrastructure that enables robotic agents to reason, plan, and learn at the image and motion level of object manipulation. We propose to use the data structures of game engines as an implementation basis for the knowledge system. Data structures are annotated with symbolic names, which are linked ontologies and symbolic background knowledge. At the same time the data structures associated with the symbolic names allow access to subsymbolic information. In addition, the physics engines of the game engines are instrumented to detect force dynamic events that are necessary for the automated recognition of actions and their segmentation into motion phases. We have demonstrated through example queries that the proposed knowledge system can answer queries that are essential for the competent execution of manipulation actions that, to the best of our knowledge, cannot be answered by other robot knowledge systems.

References

[1] M. Ersen, E. Oztop, and S. Sariel, "Cognition-enabled robot manipulation in human environments: Requirements, recent work, and open problems," *IEEE Robotics Automation Magazine*, vol. 24, no. 3, pp. 108–122, Sept 2017.

[2] G. Rizzolatti and L. Craighero, "The mirror-neuron system," *Annu. Rev. Neurosci.*, vol. 27, pp. 169–192, 2004.

[3] M. A. Arbib, *Action to language via the mirror neuron system*. Cambridge University Press, 2006.

[4] G. Hesslow, "The current status of the simulation theory of cognition," *Brain research*, vol. 1428, pp. 71–79, 2012.

[5] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, "Knowrob 2.0 – a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018.

[6] P. Oliver, "Unreal engine 4 elemental," in *ACM SIGGRAPH 2012 Computer Animation Festival*. ACM, 2012, pp. 86–86.

[7] J. Siskind, "Reconstructing force-dynamic models from video sequences," *Artificial Intelligence*, vol. 151, no. 1, pp. 91–154, 2003.

[8] "Isaac: Virtual simulator for robots," https://www.nvidia.com/en-us/deep-learning-ai/industries/robotics/, accessed: 2018-02-9.

[9] G. Moran, "Pushing photorealism in "a boy and his kite"," in *ACM SIGGRAPH 2015 Computer Animation Festival*, ser. SIGGRAPH '15. New York, NY, USA: ACM, 2015, pp. 190–190. [Online]. Available: http://doi.acm.org/10.1145/2790329.2790332

[10] L. Han, J. C. Trinkle, and Z. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Trans. Robotics and Automation*, vol. 16, no. 6, pp. 663–674, 2000.

[11] L. Talmy, *Toward a Cognitive Semantics*, ser. Bradford book. MIT Press, 2000, no. v. 1. [Online]. Available: https://books.google.de/books?id=g7IoanNUNksC

[12] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, "Control strategies in object manipulation tasks," *Curr. Opin. Neurobiol.*, vol. 16, no. 6, pp. 650–659, Dec 2006.

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

[14] D. V. McDermott, "Robot planning," *AI Magazine*, vol. 13, no. 2, pp. 55–79, 1992.

[15] F. Stulp, A. Fedrizzi, L. Mösenlechner, and M. Beetz, "Learning and Reasoning with Action-Related Places for Robust Mobile Manipulation," *Journal of Artificial Intelligence Research (JAIR)*, vol. 43, pp. 1–42, 2012.

[16] A. K. Bozcuoglu and M. Beetz, "A cloud service for robotic mental simulations," in *International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.

[17] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning and Acting*, 1st ed. New York, NY, USA: Cambridge University Press, 2016.

[18] E. Davis and G. Marcus, "The scope and limits of simulation in cognitive models," *arXiv preprint arXiv:1506.04956*, 2015.

[19] S. Qiao, W. Shen, W. Qiu, C. Liu, and A. Yuille, "Scalenet: Guiding object proposal generation in supermarkets and beyond," in *ICCV*, 2017.

[20] M. Polceanu and C. Buche, "Computational mental simulation: A review," *Computer Animation and Virtual Worlds*, vol. 28, no. 5, 2017.

[21] T. D. Ullman, E. Spelke, P. Battaglia, and J. B. Tenenbaum, "Mind games: Game engines as an architecture for intuitive physics," *Trends in Cognitive Sciences*, vol. 21, no. 9, pp. 649–665, 2017.

[22] J. Feldman and S. Narayanan, "Embodied meaning in a neural theory of language," *Brain and Language*, vol. 89, pp. 385–392, 2004.

[23] E. A. Billing, H. Svensson, R. Lowe, and T. Ziemke, "Finding your way from the bed to the kitchen: Reenacting and recombining sensorimotor episodes learned from human demonstration," *Frontiers in Robotics and AI*, vol. 3, p. 9, 2016.

[24] M. Wächter, E. Ovchinnikova, V. Wittenbeck, P. Kaiser, S. Szedmák, W. Mustafa, D. Kraft, N. Krüger, J. H. Piater, and T. Asfour, "Integrating multi-purpose natural language understanding, robot's memory, and symbolic planning for task execution in humanoid robots," *Robotics and Autonomous Systems*, vol. 99, pp. 148–165, 2018.