Creating and Accessing Knowledge Graphs for Action Parameterisation

Michaela Kümpel[1], Jan-Philipp Töberg[2]

[1]Institute for Artificial Intelligence, University of Bremen, Bremen, Germany
[2]Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, Bielefeld, Germany

Abstract

One of the visions in AI based robotics are household robots that can autonomously handle a variety of meal preparation tasks. Based on this scenario, we present a best practice tutorial on how to create actionable knowledge graphs that a robot can use for execution of task variations of cutting actions. We implemented a solution for this task that integrates all necessary software components in the framework of the robot control process. In the context of this tutorial, we focus on knowledge acquisition, knowledge representation and reasoning, and simulating robot action execution, bringing these components together into a learning environment that – in the extended version – introduces the whole control process of Cognitive Robotics. In particular, the Tutorial will detail necessary concepts a knowledge graph should include for robot action execution, how web knowledge can be automatically acquired for the domain of cutting fruits, and how the created knowledge graph can be used to let robots execute tasks like slicing a cucumber or quartering an apple. The learning environment follows an immersive approach, using a physics-based simulation environment for visualization purposes that helps to illustrate the concepts taught in the tutorial.

Tutorial resource: https://github.com/Food-Ninja/Tutorial_ESWC_HHAI

Keywords

Knowledge Representation, Cognitive Robotics, Web Knowledge, Actionable Knowledge

1. Introduction

We envision household robots that can be placed in any kitchen to then be given a random recipe from the Web that they can understand and parse into action plans that can be broken down into executable body motions that can be performed with available objects in the environment. For this, robots need to be enabled to perform meal preparation tasks with any tool, on any available object and for a variation of tasks. This tutorial is based on prior research that proposed a methodology for creating actionable knowledge graphs [1], a knowledge engineering methodology that is more specifically aligned to creating ontologies for meal preparation tasks that can be used to parameterize robot action plans in order to perform task variations of cutting actions [2] as well as previous tutorials on the topic [3, 4]. These tutorials focus on creating knowledge graphs that link object to action and environment information, thus making them *actionable*.

There has been lots of research on creation of knowledge graphs, which has led to many domain knowledge graphs that have proven to be good in answering questions about that domain. Usually, these knowledge graphs contain object information (e.g. about food objects, recipes, people, books). To make such knowledge graphs *actionable*, it is important to link the contained **object knowledge** to **environment knowledge**. If robots shall use the knowledge graphs for action execution, they need to further include **action knowledge**. This idea is based on the basic perception action loop of agents, proposed by Russel and Norvig [5] and visualized in Figure 1.

As a step towards embodied AI, environment and action knowledge need to be linked in the knowledge graph in order to make the contained knowledge applicable in agent applications. The object knowledge that these graphs contain aims not at perfectly capturing the real-world object but focus on creating an

EKAW 2024: EKAW 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024), November 26-28, 2024, Amsterdam, The Netherlands
michaela.kuempel@uni-bremen.de (M. Kümpel); jtoeberg@techfak.uni-bielefeld.de (J. Töberg)
0000-0002-0408-3953 (M. Kümpel); 0000-0003-0434-6781 (J. Töberg)

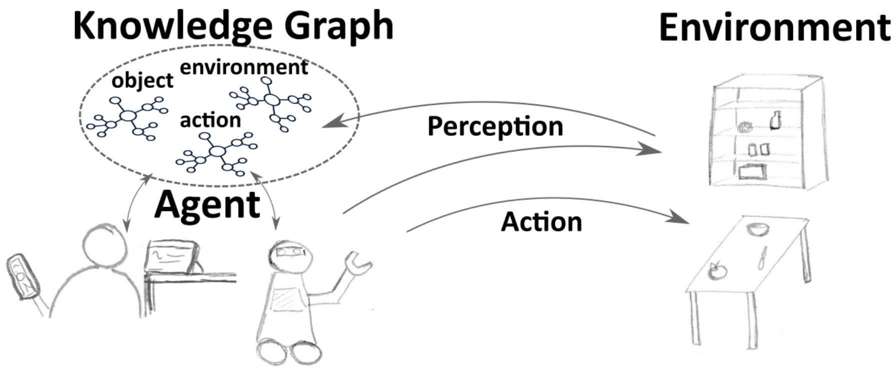

Figure 1: Perception Action Loop of an agent, including a knowledge based component.

abstraction that is suitable for the task at hand by reusing existing knowledge sources. This tutorial will detail the necessary concepts for creating an actionable knowledge graph for the example domain of *Cutting Fruits and Vegetables*, which shall be used by robotic agents to be able to infer the correct body motions for quartering an apple or dicing a cucumber.

2. Structure of the Tutorial

If we consider a random part of an exemplary recipe instruction, such as *"Peel and core the apple to then cut it into thin slices"* for creating "Grandma's best apple pie", we notice that recipe instructions are very complex. They are written for humans and thus have specific assumptions about the commonsense knowledge inherent in these recipes, but even humans have difficulties understanding some recipe instructions. We want to enable robots to understand such instructions, so that they can translate the instructions to movements of their body that result in achieving the desired outcome. But how can we create actionable knowledge graphs in such a way that a robot knows what to motions to perform for achieving a desired result? These are the three main factors that come to mind:

1. Motion parameters for action parameterisation: For successful action execution we need motion parameters that can translate knowledge into body movements. Example motion parameters are *angle, duration, position, number of repetitions*. Motion parameters depend on the actual action to be executed, making them *task-specific*.
2. Action verbs: What are the actions that a robot should be able to execute? We have to reduce the scope and look at one action category to tackle this problem, and we start with the action category of *cutting*. In order to find out action verbs in that category, we can look at lexical resources to find out synonyms and/ or hyponyms of verbs.
3. Objects and object properties: The last thing we need to also consider are objects and their properties, since they heavily influence action execution. Cutting a soft bread results in a different motion than cutting a cucumber, the existence of a peel or core also influence action execution.

There already exist approaches to tackle these problems and create graphs containing the mentioned information. But with all this information in a graph, one main question remains:

How do we make this work?

Let us iterate over our main factors again and focus on how these need to be changed in order for a robot to use the knowledge for action parameterisation:

1) **Motion Parameters**: When taking a closer look at motion parameters in our action category of *cutting*, we notice that different action verbs result in different motion parameters. In [2], we developed the concept of Action Groups (AGs), which are a more specialized group of action verbs in an action category that result in the same motion parameters and action output. For example, we differentiate between the AGs of *dicing* and *slicing*.
2) **Action Verbs**: The action verbs we collected need to be grouped into AGs based on the defined motion parameters. For example, we argue that cubing, chopping and mincing belong to the AG of dicing, since they all result in cube-shaped objects. The AGs then also need to be linked to the motion parameters.
3) **Objects and Object Properties**: There are many sources available on the web that offer object knowledge. But in order to translate object knowledge into actionable directives, we need to collect information and concrete values for the task-specific object properties that influence the action execution. We use the concepts of object affordances and dispositions to link objects and tools, but we also introduce the concept of edibility, which is used as an indicator if a fruit part can be removed but doesn't need to be, should be removed before consumption due to its taste, or if it should be removed because its either poisonous or can not be eaten.
4) **Making it Work**: Last but not least, the knowledge has to be linked in such a way that a robot can easily call a simple query to understand what needs to be done and how to parameterize its action plan.

After working through these problems, the tutorial is presenting the knowledge engineering methodology introduced in [2] and its application on the exemplary task of *Cutting Fruits & Vegetables* [3]. In general, the methodology consists of five steps to create actionable knowledge graphs that a robot can employ to handle manipulation tasks, as can be examined in Figure 2 and as explained in [3, 4].

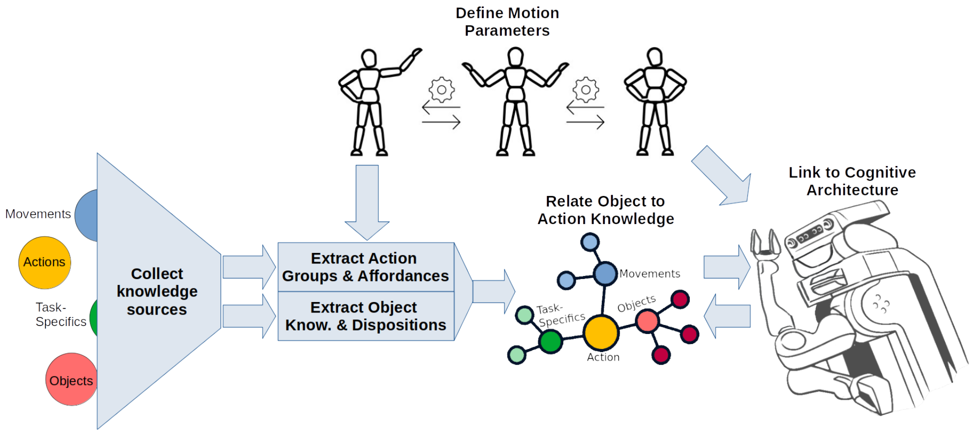

Figure 2: The Knowledge Engineering Methodology proposed in [2] we use as the foundation for the tutorial.

3. Tutorial Material

For the tutorial, we made our implementation available in Jupyter Notebooks that can be found in a GitHub repository[1]. Participants are encouraged to download the notebooks and follow along, but the notebooks are presented in depth during the talks, so actual hands-on experience is optional.

Additionally, a simulation environment for testing the query execution is available in the virtual research building[2]. Here, you can either just use the dropdowns to query the knowledge graph directly and see the resulting action parameters, or use one of the buttons provided to inspect the knowledge graph using SPARQL queries. Additionally, you can choose parameters and run the demo yourself -

[1]https://github.com/Food-Ninja/Tutorial_WebKGs4PlanParam
[2]Cutting Task Execution in Simulation: https://vib.ai.uni-bremen.de/page/labs/actionable-knowledge-graph-laboratory/

this will open a dockerized jupyter notebook where you can run the simulation of the selected robot, in the chosen environment, performing the selected task.

Acknowledgments

The tutorial is organized by the SAIL Network in collaboration with the Joint Research Center on Cooperative and Cognition-enabled AI (CoAI JRC). The research towards this Tutorial has been partially supported by the German Federal Ministy of Education and Research; Project-ID 16DHBKI047 "IntEL4CoRo - Integrated Learning Environment for Cognitive Robotics", University of Bremen as well as the German Research Foundation DFG, as part of CRC (SFB) 1320 "EASE - Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject R04 "Cognition-enabled execution of everyday actions".

References

[1] M. Kümpel, Actionable Knowledge Graphs - How Daily Activity Applications Can Benefit from Embodied Web Knowledge, Ph.D. thesis, Bremen University, Bremen, Germany, 2024. URL: https://doi.org/10.26092/elib/2936.

[2] M. Kümpel, J.-P. Töberg, V. Hassouna, P. Cimiano, M. Beetz, Towards a Knowledge Engineering Methodology for Flexible Robot Manipulation in Everyday Tasks, in: ESWC 2024 Workshops and Tutorials Joint Proceedings, Heraklion, Crete, Greece, 2024. URL: https://ceur-ws.org/Vol-3749/akr3-04.pdf.

[3] M. Beetz, P. Cimiano, M. Kümpel, E. Motta, I. Tiddi, J.-P. Töberg, Transforming Web Knowledge into Actionable Knowledge Graphs for Robot Manipulation Tasks, in: ESWC 2024 Workshops and Tutorials Joint Proceedings, CEUR-WS, Heraklion, Crete, Greece, 2024. URL: https://ceur-ws.org/Vol-3749/akr3-tutorial.pdf.

[4] M. Beetz, P. Cimiano, M. Kümpel, E. Motta, I. Tiddi, J.-P. Töberg, Translating Actionable Knowledge Graphs into Robot Action Execution, in: Proceedings of the Workshops at the Third International Conference on Hybrid Human-Artificial Intelligence Co-Located with (HHAI 2024), CEUR-WS, Malmö, Sweden, 2024.

[5] S. J. Russell, P. Norvig, E. Davis, Artificial Intelligence: A Modern Approach, Prentice Hall Series in Artificial Intelligence, 3rd ed., Prentice Hall, Upper Saddle River, 2010.