Abducing hypotheses about past events from observed environment changes

Ann-Katrin Becker[1], Jochen Sprickerhof[1], Martin Günther[1,2], and Joachim Hertzberg[1,2]

[1] Knowledge Based Systems Group, Osnabrück University,
Albrechtstr. 28, 49076 Osnabrück, Germany
firstname.lastname@uni-osnabrueck.de
http://www.informatik.uni-osnabrueck.de/kbs/

[2] DFKI Robotics Innovation Center, Osnabrück Branch,
Albert-Einstein-Straße 1, 49076 Osnabrück, Germany,
firstname.lastname@dfki.de

Abstract. Humans perform abductive reasoning routinely. We hypothesize about what happened in the past to explain an observation made in the present. This is frequently needed to model the present, too.
In this paper we describe an approach to equip robots with the capability to abduce hypotheses triggered by unexpected observations from sensor data. This is realized on the basis of KnowRob, which provides general knowledge about objects and actions. First we analyze the types of environment changes that a robot may encounter. Thereafter we define new reasoning methods allowing to abduce past events from observed changes. By projecting the effects of these hypothetical previous events, the robot gains knowledge about consequences likely to expect in its present. The applicability of our reasoning methods is demonstrated in a virtual setting as well as in a real-world scenario. In these, our robot was able to abduce highly probable information not directly accessible from its sensor data.

1 Motivation

Imagine Calvin the delivery robot tasked with delivering a set of packages. When it passes by Martin's office door early in the morning, it notices that the door is still closed, so it continues on its round without stopping to check whether Martin is inside. Half an hour later, Calvin observes from the end of the corridor that the door is now open, so it concludes that Martin is in his office, and the package can be delivered (see Figure 1).

This paper considers the question of how we can enable a robot to draw this kind of conclusion automatically in a wide variety of situations, based on abductive reasoning on its internal knowledge base.

In order to do so, we build upon the KnowRob Knowledge Processing Framework [9]. This provides a large amount of basic knowledge of objects and actions, as well as methods to reason about their effects. In the past, it has been shown

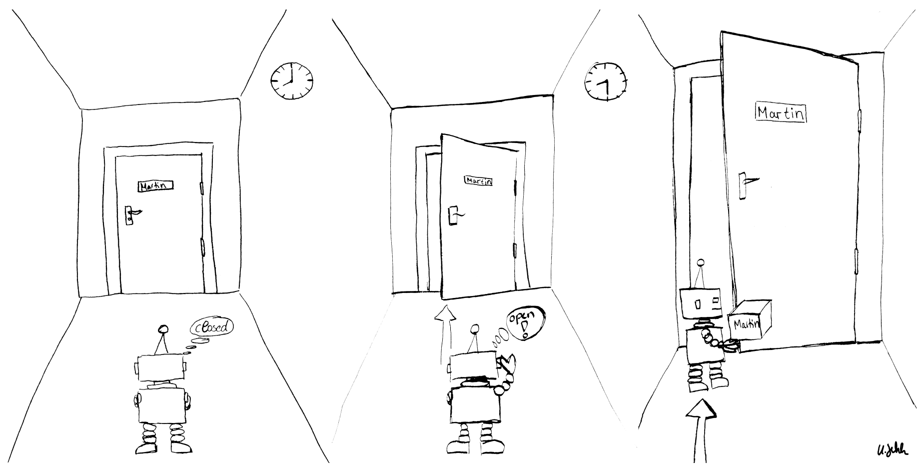

Fig. 1. Via abductive reasoning our robot is able to hypothesize that Martin is in his office by observing the open office door only.

that this framework may be utilized for planning [8]. That is, knowing the current and provided a desirable future environment state, the robot is able to derive a sequence of actions that it must perform to invoke the desired state.

Yet, reasoning about past events, which are not explicitly recorded in the knowledge base, has not been a topic of robotics research so far, although this is desirable in a variety of situations. In the above example the robot may open the office door to check if Martin is present, but explicitly checking such facts is usually expensive and sometimes even impossible. Instead, abducing a plausible sequence of events that explain the open door (e.g., Martin, who has the key, opened the door) is much faster and suggests the fact of Martin's presence, too. In general reasoning about observed environment changes or in other words hypothesizing about events that caused them, poses additional hypotheses about the current state of the environment. Thus reasoning about the past helps a robot to keep its knowledge base up to date and as complete as possible.

In the following section we present the techniques used to enable our robot to perform abductive reasoning about past events responsible for certain environment changes. The applicability of our methods is demonstrated in the delivery scenario described above as well as in a real world experiment. Afterwards we compare our approach to the literature and discuss open questions to be tackled in future research on this topic.

2 Abducing the Cause of Environment Changes

At first we need to examine the changes that a robot may encounter in its surroundings. Usually, these result from actions performed by other individuals sharing the environment, while the robot only observes their effects. Classifying

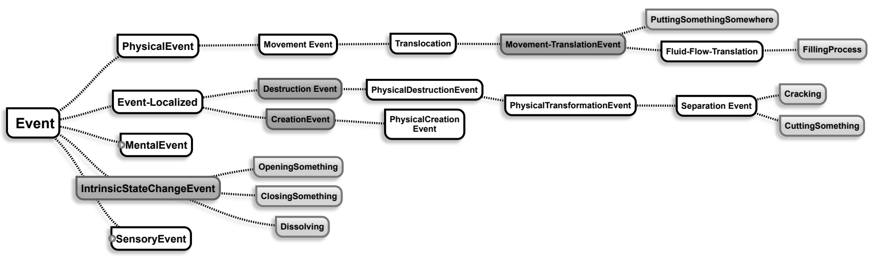

Fig. 2. Excerpt of KnowRob's basic ontology illustrating different types of events [9].

the different actions recorded in KnowRob's basic ontology (for an excerpt refer to Figure 2) by these effects, we identified five categories.

First, there are the so-called State Change Events, which change one or more properties of a known object instance; this is an instance that has been encountered by the robot previously, such as a specific office door. Second and third are Creation and Destruction Events. The former leads to the appearance of a new object instance, while the latter results in an object disappearing from the location that it previously occupied. These categories comprise events such as *CuttingSomething*, which leads to the appearance of at least one additional instance of a certain object, but also several *MovementTranslationEvents*, for example, *PuttingSomethingSomewhere*. This is modeled as from an external, decoupled view of an observer who did not perform the movement himself nor observed its execution: a new object instance and one that was moved from somewhere are not distinguishable. The last two categories we call Partial Creation and Destruction Events. These add or remove some part of a known object instance such as a *FillingProcess* that adds some new fluid to a known container object. Note that these categories are not pairwise disjoint. *CrackingAnEgg*, for instance, is a Destruction as well as a Creation Event, as during its conduction an egg gets destroyed, but instances of egg yolk and egg white are created.

This categorization allowed us to specify the input for our abductive reasoning mechanisms. Remember that our goal is to abduce a set of actions that could be responsible for an observed environment change. As in KnowRob reasoning is performed via Prolog [7], we defined Prolog predicates for each category of events. All information available on the observed change is provided as input. For example, if the state of a known object instance has changed, the instance the property of which changed, and the previous and current property values are provided. If, on the other hand, a new object has appeared, only this is passed to the corresponding predicate.

Upon reasoning depending on the method, all actions acting on a certain object or producing a specific output are tested for their effects. This means, we check if applied to the previous world state executing an action results in the current world state. If this is the case, the action is added to the result set, which contains all actions that could be responsible for the observed changes. Note that if the object acted upon is not specified via the input parameters, such as for an observed state change, the result set is comprised of pairs of action and object acted-on.

In the introductory example the delivery robot may send the following query to the knowledge base upon observing the changed state of Martin's office door:

```
?- find_cause_of_stateChange('Door-1', stateOfObject, 'ObjectStateLocked',
                             'ObjectStateOpen', ?ResultSet).
ResultSet = ['OpeningALockedDoor'].
```

This results in the conclusion that the door must have been unlocked previously. The knowledge base additionally tells us that Martin is the only one possessing a key. Thus, the robot now hypothesizes that Martin is inside his office, as entering the office is an further effect of the *OpeningALockedDoor* action, and consequently proceeds delivering the package to him.

3 Real World Experiment

One of the real-world scenarios to test our reasoning methods on-line using real sensor data was generating hypotheses about the contents of a drinking glass. These hypotheses could not have been retrieved by mere visual perception of the filled glass. This shows the ability of our abductive reasoning techniques to generate additional hidden knowledge.

For the experiment we placed a robot that monitors its surroundings via a Kinect-like depth sensor in front of a table. This contained an empty drinking glass and several juice Tetra Paks, whose labels were not readable from the robot's point of view. At this point whenever asked for the contents of the visible drinking glass the robot correctly identified it as empty. Now we took one of the Tetra Paks, let us call it *TetraPak-1*, and poured juice into the glass until it was completely filled, see Figure 3. Our robot correctly identified the performed *FillingProcess* and recorded it in its knowledge base. After this, asking the robot for the content of the drinking glass resulted in the following feedback.

First, it correctly identified the juice color as yellow, for example. As it knows that in general drinking glasses contain drinks, in case of a yellowish color the glass may either be filled with mango or orange juice. These were the only yellow drinks recorded in the knowledge base. As a next step, the system reasoned to identify which actions might have influenced the content of the glass. This can only be achieved by a filling process acting on a container containing mango or orange juice. The last filling process performed on the glass instance was a *FillingProcess* featuring *TetraPak-1* as a source, which as a general Tetra Pak instance may contain one of the juices. Thus, the robot correctly informs the user that to its knowledge the content of the glass is the same as the content of *TetraPak-1*. This hypothesis could not have been generated by mere visual perception of the filled glass. Now in case the robot is asked to serve a glass of orange juice, this hypothesis may be utilized to identify the contents of the glass by inspecting the label of *TetraPak-1*.

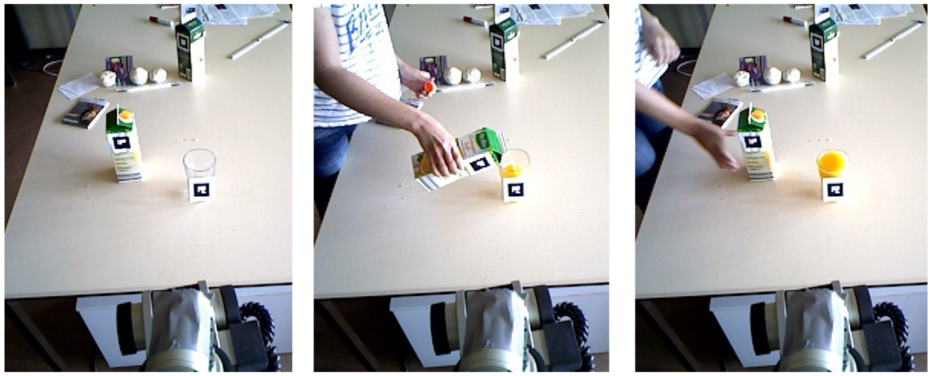

Fig. 3. Illustration of the different steps of a filling process as observed from our robot's camera.

Note that if presented several filling processes, the robot only regarded the last one as relevant for the contents of the glass. If the last filling process involved a Tetra Pak known to contain cranberry juice, but the glass contained a yellow liquid, the filling process was correctly estimated as outdated. Furthermore, with slight changes it is possible to apply the system for the reverse reasoning process. If the specific contents of the glass is known and a filling process from a general Tetra Pak instance has been observed, the system can conclude the contents of the Tetra Pak to be the same as the contents of the glass. Afterwards, in case there are other glasses on the table recorded to have been filled from the same Tetra Pak, the type of their contents is known, too.

Our code, a readme with installation and running instructions, as well as rosbags of this experiment are available online[3].

4 Related Work

Obviously the introductory delivery problem could be solved in many different ways. First of all we could handcraft a probabilistic model that predicts the occupancy of a persons office [10], based on factors such as the door status and day time. This might be even more accurate sometimes, yet a model would be necessary not only for each person but for every other scenario as well. Our method generalizes to all actions and effects recorded in the underlying knowledge base.

Other approaches could include default reasoning [6] and non-monotonic reasoning techniques such as preferential reasoning [1]. However, so far none of these have been utilized in a practical approach comparable with our work, to our knowledge. In the robotic community similar contributions are rare, too. Mason [2, 3] utilizes environment change, in particular the appearance or disappearance of groups of features, to classify these as objects rather than stationary

[3] `http://kos.informatik.uos.de/infer_hidden_params`

background. Nitti et al. [4, 5] use Statistical Relational Learning to infer hidden parameters such as magnetism like we do, but do not reason about past actions.

Especially when performing abductive reasoning online, new challenges arise. The main issue is how to limit abduction, so that our robot is still able to operate effectively, rather than busy generating hypotheses about its surroundings, most of which are irrelevant. We tackle this by applying our abductive reasoning methods only in accordance with the robot's current task. The robot monitors changes in the environment continuously via its sensors. However, only if we need to deliver a package to Martin, we analyze the cause of the status change of his office door. Similarly, only if we are interested in the content of a specific drinking glass, we abduce its source. This way, we keep our knowledge base compact and prevent the robot from being occupied with permanently abducing new hypotheses.

5 Conclusion and Future Work

We enabled a robot to perform abductive reasoning about past events to explain and generate hypotheses about hidden features of the current world state. The robot was able to perform this reasoning on-line on the basis of the available KnowRob Knowledge Processing Framework. The hypothesis generation shown in the examples in this paper is mundane and essentially simple; yet, they were beyond the reasoning capabilities of state-of-the-art robots. Our reasoning approach is added to an existing knowledge representation framework in use for robots, KnowRob, which gives us the flexibility to apply this approach to a variety of situations. Additionally we propose to apply abductive reasoning only if relevant for the currently pursued task, rather than generating all possible conclusions directly, which would be resource-consuming.

In future work we aim at applying our reasoning methods in further and more complex online scenarios. One of these scenarios could include tracking the ownership of certain objects. Imagine a breakfast table at which all persons have their own drinking glasses, yet these are not distinguishable by their looks. The people at this table keep their drinking glasses throughout the day, but they might change their positions, for instance, from the table to the lounge and take their glasses with them. Now if Martin's glass has disappeared from the table and a new glass appeared in the lounge, we can hypothesize that this belongs to Martin. Further we intend to use the presented reasoning methods to facilitate object recognition via preselection. In the drink scenario, based on the coloring of the drinking glass contents, the robot hypothesizes that a certain Tetra Pak instance contains either mango or orange juice. Thus, it is able to take this information into account upon identifying the exact label of the Tetra Pak.

References

1. Lehmann, D., Magidor, M.: Preferential logics: the predicate calculus case. Proc. 3rd. Theoretical aspects of reasoning about knowledge pp. 57–72 (1980)

2. Mason, J., Marthi, B.: An object-based semantic world model for long-term change detection and semantic querying. Proc. IROS pp. 3851–3858 (2012)
3. Mason, J., Marthi, B., Parr, R.: Object disappearance for object discovery. Proc. IROS pp. 2836–2843 (2012)
4. Nitti, D., Laet, T.D., Raedt, L.D.: A particle filter for hybrid relational domains. Proc. IROS pp. 2764–2771 (2013)
5. Nitti, D., Laet, T.D., Raedt, L.D.: Relational object tracking and learning. Proc. IROS (2014)
6. Reiter, R.: A logic for default reasoning. Artificial intelligence 13(1), 81–132 (1980)
7. Sterling, L., Shapiro, E.: The Art of Prolog: Advanced Programming Techniques. MIT Press (1994)
8. Tenorth, M., Beetz, M.: A Unified Representation for Reasoning about Robot Actions, Processes, and their Effects on Objects. Proc. IROS (2012)
9. Tenorth, M., Beetz, M.: KnowRob: A knowledge processing infrastructure for cognition-enabled robots. Int.J. of Robotics Research 32 (2013)
10. Wang, D., Federspiel, C.C., Rubinstein, F.: Modeling occupancy in single person offices. Energy and Buildings 37(2), 121–126 (2005)