DKA-robo: dynamically updating time-invalid knowledge bases using robots

Ilaria Tiddi, Emanuele Bastianelli, Enrico Daga, Mathieu d'Aquin

Knowledge Media Institute, The Open University, United Kingdom
name.surname@open.ac.uk

Abstract. In this paper we present the DKA-robo framework, where a mobile agent is used to update those statements of a knowledge base that have lost validity in time. Managing the dynamic information of knowledge bases constitutes a key issue in many real-world scenarios, because constantly reevaluating data requires efforts in terms of knowledge acquisition and representation. Our solution to such a problem is to use RDF and SPARQL to represent and manage the time-validity of information, combined with an agent acting as a mobile sensor which updates the outdated statements in the knowledge base, therefore always guaranteeing time-valid results against user queries. This demo shows the implementation of our approach in the working environment of our research lab, where a robot is used to sense temperature, humidity, wifi-signal and number of people on demand, updating the lab knowledge base with time-valid information.

1 Introduction

Managing dynamic data in knowledge bases, i.e. statements that are only valid for a certain period of time, is a well-known problem for knowledge acquisition and representation, because data need to be constantly re-evaluated to allow reasoning. Our current work focuses on representing the validity in time of statements in a knowledge base, and on using an autonomous agent (such as a robot) as a mobile sensor that updates the outdated statements, therefore constantly guaranteeing the "freshness" of the requested information.

Let us imagine a knowledge base representing the working environment of the Knowledge Media Institute (KMi) research department, where information about locations is static (e.g. the coordinates of a room) but information such as temperature, humidity, wi-fi signal or number of people in a room varies more often and needs to be re-evaluated. Common solutions to this problem, such as providing time-stamped versions of the knowledge base [3, 4, 7] or using sensors to constantly stream the information [1, 2, 5], are more costly in terms of data collection because much of the information they provide is likely not to be required. Besides, they are less flexible as they might only allow queries at specific locations.

The alternative solution of moving an autonomous agent upon request to re-collect the expired information (i.e. that has lost time-validity) and update the knowledge base has the advantage of guaranteeing that, when queried, the knowledge base will always return time-valid information. In this context, there is a number of challenges to be faced at knowledge representation and management level, namely: how to represent time-validity in the knowledge base; how to establish that statements have expired; and how to instruct a robot to perform a set of actions by favouring the time-validity of the information that is collected.

Our solution is to use RDF and SPARQL as a framework for knowledge representation, combined with an autonomous mobile agent which updates the time-invalid information of the knowledge base on demand. More specifically, our tool uses statements in the knowledge base with a time-stamp representing their time-validity (i.e. their expiry date). When the knowledge base is queried through SPARQL, we decide how long a piece of information necessary to answer this query will be considered valid. On this basis, the tool creates a plan for the robot to collect such information, in a way that guarantees the time-validity of the information returned.

This demo, that we call DKA-robo (Dynamic Knowledge Acquisition with a Robot), presents the implementation of our approach based on the simulated working environment of our research lab. Users will be shown how the robot can be instructed to move and sense information, and how the evaluated plan is executed to preserve the time-validity of our knowledge base.

2 Overview

DKA-robo is implemented as a process in which a user submits a query to a knowledge base and receives a set of results that are valid in time, provided that there exists a plan that enables the agent to collect the required information in the limited time before it becomes invalid. The process, described more in detail in [6], is articulated as reported below.

1. **Query.** The user expresses the query to the knowledge base. The knowledge base is represented as a set of RDF quads $q = (t, g)$, where t is the $\langle$subject, predicate, object$\rangle$ triple and g is the named graph to which t belongs, representing the time at which t will expire.
2. **Invalid information collection**. The query is first executed onto the current knowledge base, and the execution process is monitored to retrieve the graphs from which triples are obtained. From these, triples which have expired and which have not are identified.
3. **Planning.** The planner receives the time-invalid quads and asks the mobile agent for its location. Based on these, it calculates the plan to send to the agent, i.e. the right sequence of actions to perform so that none of the statements in the answer set remains time-invalid. The plan is evaluated using a best-first strategy designed to minimise the time-to-invalidity of the considered quads[1].
4. **Knowledge base update.** The robot receives the plan and performs it. When a new piece of information is collected, this is sent to the knowledge

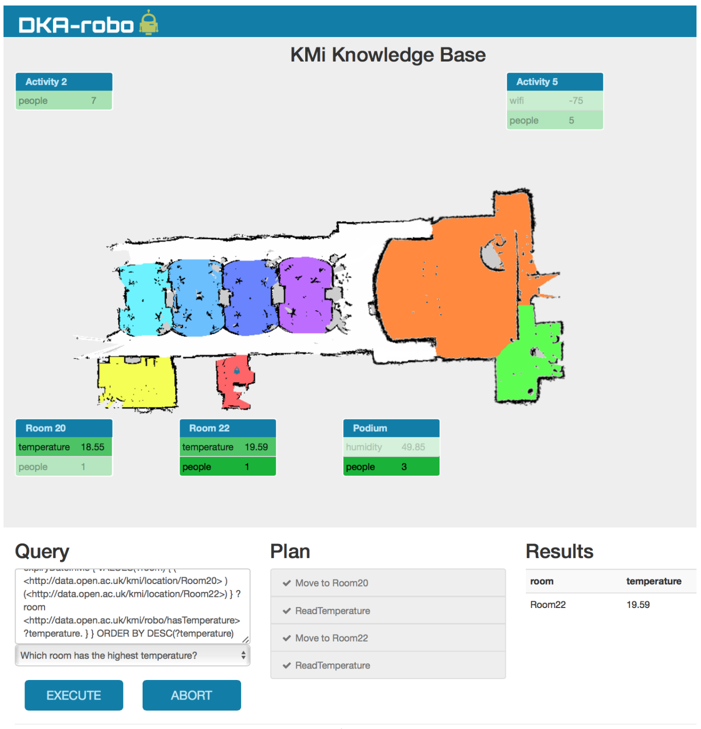
[1] Since we focus on the representation of time-validity in the knowledge base, we employ a naïve implementation of the planner without claiming for its efficiency.

base to be updated. The time-to-invalidity of the new information is evaluated using a set of time-validity rules expressed as a triple pattern p associated to a duration d. The rule for which the triple matches the pattern p that has the shortest duration is selected, and a new quad $q = (t, \text{current\_time} + d)$ is written in the knowledge base.

5. **Query results.** Once all the actions have been executed, the user is shown the answer to its query. If no answer is received, this means that there is no plan that can be executed such that all the statements in the answer are time-valid.

3 DKA-robo demonstration

During the demonstration, we will present scenarios in a simulated environment from a real-world example, where a robot moves in KMi on demand, updating the outdated information of the knowledge base. The audience will be able to interact with DKA-robo through the user interface shown in Figure 1 and described below.



```
<http://data.open.ac.uk/kmi/location/Room20> )
(<http://data.open.ac.uk/kmi/location/Room22>) } ?
room
<http://data.open.ac.uk/kmi/robo/hasTemperature>
?temperature. } } ORDER BY DESC(?temperature)
```

✓ Move to Room20
✓ ReadTemperature
✓ Move to Room22
✓ ReadTemperature

room	temperature
Room22	19.59

Fig. 1. DKA-robo demo.

In the first part, shown at the top of Figure 1, the user is shown the KMi knowledge base and its status as a map. The dynamic information of each room (temperature, humidity, wi-fi strength in dB and number of people) fades out the more the information is approaching its expiry date. For instance, Figure 1 shows that information about the Podium and Activity 5 are about to expire.

These will reappear when the robot will send the new sensed information. The user can also see the location of the robot on the map while executing the plan (in Room 22 in our example).

The second part, shown at the bottom in Figure 1, is dedicated to the user's interaction with both the knowledge base and the robot. In the "Query" panel, users can freely query the knowledge base, either by choosing one of the predefined queries expressed in natural language, or by inserting their own SPARQL query. In Figure 1, a user asked for the room with the highest temperature. Given a query, three scenarios are possible:

1. no outdated statement is used to compute the result set. No plan is sent to the robot and the result set is shown directly in the "Results" panel.
2. some of the statements used to compute the result set are outdated, but the robot receives no plan. This means that there is no possible plan to be found by the planner, which guarantees the time-validity of all the necessary statements. The user is therefore alerted that the query has to be simplified.
3. some of the statements used to compute the result set are outdated, and the robot receives a plan, which is shown in the "Plan" panel. In our example, the plan consists in moving and measuring the temperature of Room 20 and then of Room 22. The user can see the execution of the plan in real-time, i.e. the progress of the operations performed/to be performed, the robot moving in the space, and the new data showing in the knowledge base when sensed. Once all the operations are performed, the answer is shown to the user in the "Results" panel. In our example, Room 20 is the one with the highest temperature.

Additional features of the demo include the possibility to stop the robot and abort the execution of the plan, and to instruct it to randomly move and sense outdated information. If circumstances allow it, we will attempt to reproduce the KMi scenario within one of the locations of the EKAW conference.

References

1. Balduini, Marco, et al. "Social listening of city scale events using the streaming Linked Data framework." International Semantic Web Conference, 2013.
2. Calbimonte, Jean-Paul, et al. "Enabling query technologies for the Semantic Sensor Web." International Journal on Semantic Web and Information Systems, 2012.
3. Fernández, Javier D., et al. "The DBpedia wayback machine." The 11th International Conference on Semantic Systems. ACM, 2015.
4. Halpin, Harry, and James Cheney. "Dynamic provenance for SPARQL updates using named graphs." 23rd International Conference on World Wide Web, 2014.
5. Le-Phuoc, Danh, et al. "Linked stream data processing engines: Facts and figures." International Semantic Web Conference. Springer Berlin Heidelberg, 2012.
6. Tiddi, Ilaria, et al. "Update of time-invalid information in knowledge bases through mobile agents." Integrating Multiple Knowledge Representation and Reasoning Techniques in Robotics, 2016.
7. Van de Sompel, Herbert, et al. "An HTTP-based versioning mechanism for Linked Data." arXiv preprint arXiv:1003.3661, 2010.