Query by Combination in the Internet of Things

Sven Huysmans, Peter Rigole, Yolande Berbers

Query by Combination in the Internet of Things

Sven Huysmans Peter Rigole Yolande Berbers

Department of Computer Science, K.U.Leuven
Celestijnenlaan 200A,
3001 Leuven, Belgium
Sven.Huysmans@gmail.com {Peter.Rigole,Yolande.Berbers}@cs.kuleuven.be

Abstract. In the Internet of Things, physical objects, or Things, are integrated into a virtual world that harbors capabilities to augment the physical appearance of those Things with additional information. In this paper, we present a software solution that enables users to deduce additional information from Things by querying the relations between them. These relations are represented by a joint knowledge base that is constructed on the fly from ontologies that are associated with the Things. Based on the information found in the acquired knowledge, suitable applications are instantiated to process and represent the knowledge about the Things to the user. The practical use of the presented solution is illustrated by a concrete scenario.
Keywords: RFID, Internet of Things, ontologies

1 Introduction

The *Internet of Things* (IoT) represents a research domain that supports the connection between the physical world and the Internet. It gives the Internet a new dimension: from anytime, any place connectivity for anyone, we move to connectivity for anything [5]. To achieve this, every Thing in the real world should have an additional virtual dimension. In this paper, we focus on the combination of Things to derive information from their combined virtual representation. The same way a combination of physical Things can result in practical uses that are absent when the Things are separated (e.g. a key and a lock), the combination of their virtual counterpart could also render new applications.
Radio Frequency Identification (RFID) is a technical keystone of the emerging *Internet of Things*. The rise of small RFID tags makes it possible to extend the physical appearance of object and places. These tags are the extension of the existing bar code technology [1,2]. Every tag has it's own unique code. The majority of applications that use this RFID technology can be found in transport and logistics, access control, etc.. In spite of the fact that RFID is not a new technology (its first patent was issued in 1973), not every Thing is equipped with a tag yet. In the near future, however, we expect new RFID applications to arise quickly as the price of tags is decreasing steadily. This will ensure the widespread tagging of Things so that every Thing can be connected to a virtual counterpart by means of its unique ID.

We chose to define the virtual representation of Things in an ontology because of its expressiveness. This is a data model that represents a set of concepts, the attributes of concepts and the relations between them [4]. Such a representation of knowledge provides a means to reason about it. Ontologies are often used in Artificial Intelligence, and the Web Ontology Language (OWL) [3] is considered as the foundation of the Semantic Web and its applications.
The challenge we undertook in this research area is to create a solution that combines a number of real-world Things in a logical manner by exploiting different applications and web services. For example, an application could tell a user which meals can be prepared from a set of ingredients and a cookbook.
This paper is further organized as follows. The representation of Things in the digital world is presented in the next section, followed by our solution to augment a set of Things and the architectural description of our solution. A usage scenario using photos and a map is given in section 4. We briefly discuss related work in section 5, and we conclude and give indications for future work in section 6.

2 Representation of Things

An important part of our current solution is the representation of the information associated with the Things. In our OWL description we make a distinction between three main classes: a Thing class, a data class and a tool class. The first represents physical Things, the second represents flat text-based information about the Thing, and the last class represents the tools, which are applications or web services. For example, a map is a Thing and has a relation with Google Maps[1] as a tool. Obviously, these three main classes can be extended with extra classes whenever needed. The example OWL code shown in Fig. 1 describes a picture of the Atomium. It describes a Thing individual that has a relation *hasPosition* with a position individual and a data individual.
As mentioned before, the connection between the physical state of a Thing and its virtual information is created by tags that identify the Things and their ontology location. The ID of the tag is tightly coupled with the name of the Thing and a URL that points to where the description and information of the Thing can be found. This approach allows applications to look up the information in several distributed ontologies and combine this information.

3 Solution and Architecture

The work presented in this paper enables users to merge the digital representation of Things that appear in their world. These Things are tagged with an RFID tag that provides a unique ID, allowing our application to lookup the corresponding OWL data. Rather than introducing

[1] http://maps.google.com/

```
<Photo rdf:ID="Atomium">
 <title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Atomium</title
      >
 <hasPositon>
  <Position rdf:ID="Position_Atomium">
   <Longitude rdf:datatype="http://www.w3.org/2001/XMLSchema#float">4.3413887
        </Longitude>
   <Latitude rdf:datatype="http://www.w3.org/2001/XMLSchema#float">50.895</
        Latitude>
  </Position>
 </hasPositon>
 <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The
      Atomium monument represents a unit cell of an iron crystal</description>
</Photo>
```

Fig. 1. OWL example of a photo

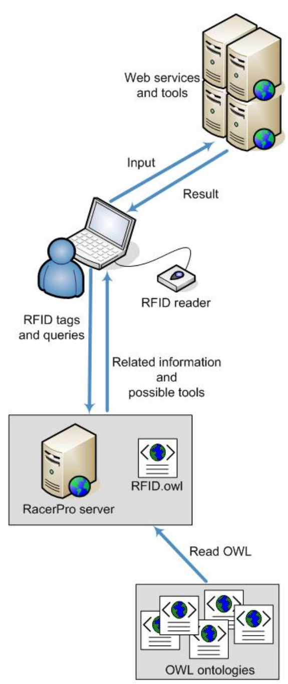

Fig. 2. Architecture of the current solution

an indirection to find the OWL data via the ID of the tag, we chose to embed the location of the OWL data onto the tag itself. This location is stored as a Persistent URL (PURL) [8]. This PURL address points to a resolution server that returns the URL that points to the actual OWL data. This way, the OWL data can be relocated to another host without invalidating existing tags.
Our solution is built as a .NET application running on a PC or a PDA. We used a USB RFID reader from ACS[2] attached to the PC and a SDiD 1020[3] reader attached to the PDA. Both readers support anti-collision and can read up to several Mifare tags simultaneously. However, we experienced a practical limitation of four simultaneous reads, meaning that multiple scans are needed to read more than four collocated Things.

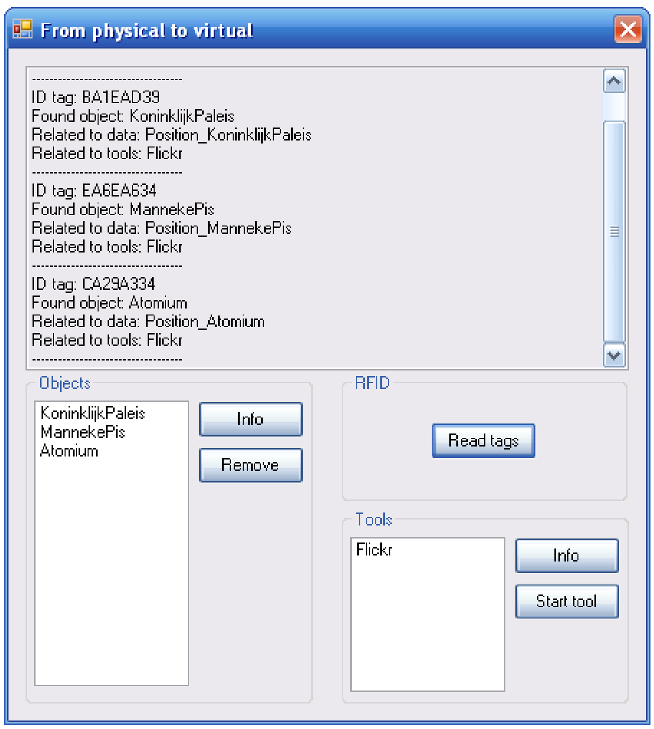

Fig. 3. Interfacing with the application

Fig. 2 gives an overview of our architecture. Our application on the PDA or PC does not use the OWL files associated with Things directly. After lookup, it feeds the OWL data into an external reasoner and then sends queries to that reasoner to fetch the desired information. In our solution

[2] http://www.acs.com.hk
[3] http://www.sdid.com

we chose to use a RacerPro server [9] that serves as an OWL reasoner and information repository for the Semantic Web. Our choice here was mainly driven by the completeness of this reasoner and the fact that it can be instantiated as a separate server application, so that the PDA could be saved from the computation efforts needed by the reasoning algorithm. This way we try to ensure a certain level of Quality of Service (QoS) in our application.

In our solution, we also kept an eye on Quality of Service at the communication level between the PDA and the external services. The application scenarios we envision should still work using mobile technology such as GPRS (56kbps) or UMTS (128kbps). Since the OWL data can be fetched directly by the external reasoner based on the PURL that is embedded onto the RFID tag, this data does not need to pass through the PDA's communication infrastructure. The PDA only has to send its queries directly to the reasoner that returns only those facts the PDA is interested in. From the experience with our scenario, we know this data rate does not exceed the bandwidth available on mobile networks. Obviously, the use of external tools (through web services) is bound to the bandwidth needs of the tool at hand, and cannot be predicted in general. Currently, the tools we have used are web-based services like Google Maps[4] and Flickr[5].

In the most general use case, a user presents several tagged Things to the PDA to initiate an application, if there is one available. Using the embedded PURL, the system requests the reasoner to load the OWL data associated with the Things. Depending on the nature of this data, several tools can be found that are able to process or represent the combined data. Using these tools, new information can be deduced from the relations between the Things. For example, two paintings in a museum can lead to information about the relationship between their painters. In the application that runs on the PDA (see Fig. 3), the user can select Things to combine, he can delete those Things he is not interested in, or the application can try to combine Things randomly and return information that may be of interest to the user. This way it is possible that the user gets information he had never thought of. The program also suggests the tools that are available and that are able to process the data.

4 Scenario: culture with a touch of adventure

Today, cultural tourism is a popular motivation for traveling, although many of the authentic cultural aspects of a specific region are often difficult to convey to the average traveler who wants to visit as much as possible in a typically small time frame. To make cultural tourism more appealing and easier to grasp, we designed an application that adds some adventure to the user experience in the form of value-adding activities.

Our scenario is built in the setting of a national park that contains several points of interest such as sculptures, pieces of art, protected trees, etc..

[4] http://maps.google.com

[5] http://www.flickr.com/

The goal of our setup is to find a secret agent who disappeared somewhere in the park, based on several belongings (Things) of the agent that were recovered (see Fig. 4). The Things at hand are: a pair of pliers, some pictures, a map and a PDA equipped with an RFID reader and our application. Obviously, many approaches can be taken to find leads to the secret agent. In the following paragraphs, we give an example of how this could be done.

Fig. 4. The cultural tourism scenario

The first lead toward the secret agent's whereabouts is the picture of a chapel that stands in the park. Inside the chapel, the visitor explores the Things he finds, looking for more directions. At a painting of the last supper, the PDA notifies the visitor about a relation that may be relevant. The map, a picture of the bell tower in Firenze and the data that is associated with the painting combines to a person named Leonardo da Vinci. Based on this finding, the digital information associated with the layout map of the park leads to a replica of the statue of Leonardo da Vinci, which is also part of the park's pieces of art. Based on the info from the digital counterpart of the map, the layout of the park is presented on the PDA and a route-planner tool is made available to the user (see Fig. 5). When the user activates this tool, the path from the chapel to the statue is automatically calculated and displayed on the PDA screen.

At the statue, the user may try to find a link between the statue and the remaining Things. Although there is still no match found between

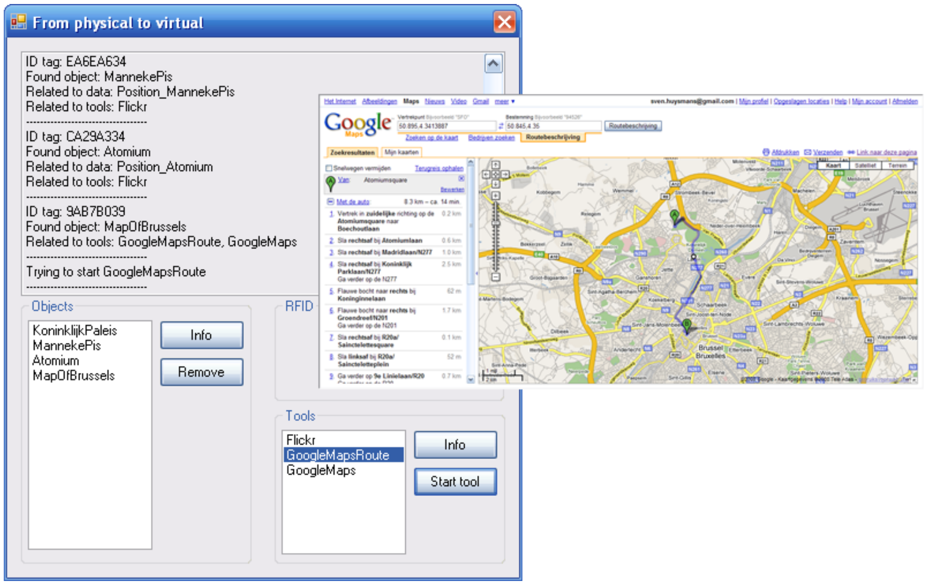

Fig. 5. The map is presented to the reader and map service is used

the pliers and any other Thing found so far, a clue is detected in the environment of the statue. Our PDA correlates an iron wire statue and the pliers to new data. It seems that the artist of the iron wire statue created several statues made by using pliers only. The statue the user stands before seems to be a statue of Adam, leading to the logical quest to find Eve, which appears to be the hiding place of the secret agent.
Our approach involves the user into playing a game while discovering information about the park and its pieces of art. The algorithms we used are based upon queries that are sent to a reasoner after it is provided with data that is associated with several RFID tags. Cross matches between ontology individuals result in new data elements that are presented to the user and stored for later use in even more combinations.

5 Related work

Invisible media [6] is a software/hardware platform that uses RFID to provide the user with relevant augmented information on Things. The system spontaneously suggests relevant knowledge about Things that are in the focus of the user's visual attention. One of the presented scenarios is Engine-Info, a training application that explains the purpose of the components of a combustion engine.
From the same research group comes the idea of TaPuMa [7], which is a scenario that is closely related to our work. It is a digital public map

that allows people to use their own personal Things to display pertinent information on the map. An extension is suggested in which this can be used as a query engine for all kind of tagged Things.

6 Conclusion and future work

This work is only a small step in the direction of an *Internet of Things*. Combining Things to get related information and a closer virtual look to these Things brings a new insight to this paradigm. The RFID technology brings the physical and the virtual world a whole lot closer to each other. A distinction between regular Things and tools result in a visualization when different objects of both classes are placed near an RFID reader. Ontology languages are a very natural way for presenting (related) information based on the real world and were therefore very useful for the development of this idea.
Our solution can still be improved and extended with functionalities that support a better user experience. It might be useful that the user can enter extra information or even add some tags into the system to extend the existing knowledge base. The usage scenario suggested by Time2Trace[6] will be further examined and tested. Another interesting viewpoint is the integration of the Electronic Product Code (EPC) EPCglobal[7] of the RFID tags.

References

1. RFID and the Internet of Things. Digital ID World, 66-69, Nov. 2003.
2. Jürgen Bohn and Friedemann Mattern, Super-Distributed RFID Tag Infrastructures. Proceedings of the 2nd European Symposium on Ambient Intelligence (EUSAI 2004), LNCS 3295, Eindhoven, The Netherlands, Nov. 2004.
3. W3C. OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/, Feb. 2004.
4. Wikipedia. Ontology (information science). Wikipedia the free encyclopedia, http://en.wikipedia.org/wiki/Ontology, Jan. 2008.
5. ITU Internet Reports 2005: The Internet of Things Executive Summary. International Telecommunication Union (ITU), Geneva, Nov. 2005.
6. D. Merrill and P. Maes. Invisible Media: Attention-sensitive informational augmentation for physical objects. 2005.
7. Pranav Mistry. TaPuMa. Ambient Intelligence Group MIT, http://ambient.media.mit.edu/, Feb. 2008.
8. Online Computer Library Center (OCLC), http://purl.oclc.org
9. Racer Systems, Racer, http://www.racer-systems.com

[6] http://www.time2trace.com/
[7] http://www.epcglobalinc.org/