Duplicate reduction using daml:sameIndividualAs

Marta Sabou

August 23, 2002

1 Problem

By merging several data sources the phenomena of having multiple resources for the same physical entity (person, article) emerged. This resulted in the fact that the created portals displayed all these resources, which in fact were referring to the same object. Therefore duplicates appeared. Example: Frank van Harmelen was described by different identifiers in each data source. Therefore, while being different, the resources: `http://www.cs.vu.nl/~frankh/spool/sw.bib#f_van_harmelen` and `http://www.cs.vu.nl/~marta/michel.xml#frank_van_harmelen` referred in fact to the same person.

2 Question

How can we get rid of these duplicates?

3 Solution

Use daml:sameIndividualAs to specify which resources are the same. We conducted a little experiment on this, described in the next paragraph.

4 Implementation

Step1) First we provided the semantic data that described the identity of certain resources. Here is the specification of all resources that describe Frank:

```
<rdf:Description
rdf:about="http://www.cs.vu.nl/~marta/marta.xml#frank_van_harmelen">
    <daml:sameIndividualAs
        rdf:resource="http://www.cs.vu.nl/~frankh/spool/sw.bib#f_van_harmelen"/>
    <daml:sameIndividualAs
        rdf:resource="http://www.cs.vu.nl/~marta/michel.xml#frank_van_harmelen"/>
    <daml:sameIndividualAs
        rdf:resource="http://www.cs.vu.nl/~marta/marta.xml#frank_van_harmelen"/>
    <daml:sameIndividualAs
        rdf:resource="http://www.cs.vu.nl/~frankh/spool/sw.bib#frank_van_harmelen"/>
    <daml:sameIndividualAs
        rdf:resource="http://www.cs.vu.nl/~marta/heiner.xml#f_van_harmelen"/>
    </rdf:Description>
```

The file containing the identity statements (http://www.cs.vu.nl/ marta/sameas.xml) was uploaded to Sesame in a repository that contained all the publications provided by Frank, Michel, Heiner and Marta.

Observation: It was quite a difficult task to filter these equivalencies and to add them manually to the repository. Support in acquiring similarity data is very important.

Step2) We tried to formulate queries in which we take advantage of having this new semantic data. We hoped that the present reasoning support of Sesame would suffice to eliminate duplicates.

Here is a query that retrieves all publications of Frank, i.e. all publications that have as author one of the references to Frank:

```
select  Y, Z from {X}
http://www.daml.org/2001/03/daml+oil#sameIndividualAs {Y}, {Z}
http://www.ontoweb.org/ontology/1#author {Y} where
X=http://www.cs.vu.nl/~marta/marta.xml#frank_van_harmelen
```

There (at least) two problems with this approach:

- It assumes that a "normaliser" exists, i.e. a resources that is known to be equal with all the other resources pointing to Frank. In our case this is `http://www.cs.vu.nl/~marta/marta.xml#frank_van_harmelen`. One would expect that when the above mentioned query is run with any of the resources for Frank as a value of X, the same answer would be given. This assumption is based on the simetry of the relation. However that is not specified for sesame. Therefore the answer will be null.

- The answer to this query is a set of items that should also be filtered based on the sameIndividualAs relations. I did not find a way to encode it in a Sesame query.

Step3) Clearly the reasoning support of Sesame is too limited to solve this issue. Therefore we decided to implement this missing reasoning part in the code that translates Sesame repositories into portals. We had the following experience:

- It is very difficult to implement in code such reasoning because one has to solve reflexivity and transitivity issues, which are getting very complex if several declarations exist for a single physical object. It is possible that more people add such identity statements, therefore one first has to combine them to decide on a normaliser. For this experiment we assumed that the equivalencies are given in the form of linking all representations to a normaliser.

- For our specific problem, we used the identity data only to determine a normaliser that would be the only published element. Therefore we assumed that all items are declared according to the same ontology. We did not deal with matching the properties/property values of the same items. However, normaly one would want that the normaliser contains all data that is specified in the different representations. Note that in this case inconsistancies can appear very easily.

Step 4) We created the portals with no duplicate elements. To see how important this change is compare:

- the portal without duplicate reduction (http://zpad.cs.vu.nl/spectacle/channel/swvu_allsm) and
- with duplicate reduction (http://zpad.cs.vu.nl/spectacle/channel/swvu_siall).

5 Conclusion

- 1) Duplicate reduction is definitely needed.
- 2) Sesame does not provide sufficient reasoning even if the identity statements are provided using daml:sameIndividualAs.
- 3) Manually providing identity data is difficult.
- 4) Implementing the missing reasoning part is quite difficult and optimising the code will be even harder.
- 5) The assumptions made to build this example do not convey to real life situations.
- 6) The present code does not exploit the given semantics to the maximum, i.e. to derive new knowledge.
- 7) Therefore: help in providing identity semantics as well as built-in reasoning are two important issues to work on.