Vrije Universiteit Brussel

Open Information Systems 2017 - 2018

Final report - Group 8

ROMAIN Maximilien - 0543411

ROJAS Felipe - ???

PERALE Thomas - ???

LEHAL Sherik - ???

December 20, 2017

1 Section 1

2 Rules description

First rule : If a book belongs to a category and that category is also a subcategory to another category, it implies that the book belongs to both the main category and its subcategory.

Ebook(?x), Category(?y), Category(?z), subCategoryOf(?y, ?z), hasCategory(?x, ?y), differentFrom(?y, ?z) → hasCategory(?x, ?z)

This rule is relevant because an ebook could have more than one category and one of those categories could be a subcategory of one of these, which means that the ebook will have a category and a subcategory. Example : if a book belongs to *"war"* category and the *"war"* category belongs to *"action"* category, it is implied that the book belongs to both "war" and "action" categories.

Second rule : if a user makes a purchase and an ebook is part of the purchase, then we can imply that the user *"HasPurchased"* an ebook.

User(?x), Purchase(?y), Ebook(?z), isMaking(?x, ?y), isPartOf(?z, ?y) → hasPurchased(?x, ?z)

This rule is relevant because a purchase needs at least one ebook purchased by the user, meaning that a user purchased an eBook.

Third rule : if a user is rating an eBook or an eBook has been rated implies that the user hasPurchased an ebook. The rule is relevant because users who did not purchase books cannot rate them.

User(?x), Rating(?y), Ebook(?z), hasRated(?x, ?y), hasRating(?z, ?y) → hasPurchased(?x, ?z)

This rule is relevant since only users that has purchased an eBook would be able to rate that eBook.

The last rule implies that if a user has the role "admin", they can manage the ebooks database on the system.

User(?admin), Ebook(?book), AdminRole(?role), hasAdminRole(?admin,

?role) → manage(?admin, ?book)

This rule is relevant because an admin should be the only type of user that can manage eBooks.

3 Sparql endpoint & Mapping

To be able to implement the sparql endpoint in the project, we used the tool *D2RQ*[1]. D2RQ is Open Source software proposes a language of association between ontologies and databases, and an endpoint *Sparql* allowing query the database through *Sparql* queries. We used for the project, one of the specifications of the tool, allowing to map the ontology to the database directly by using the database, creating a mapping file. "This file called the default mapping, maps each table to a new RDFS class that is based on the table's name, and maps each column to a property based on the column's name"[2]

This mapping can be done by using the following command in the D2RQ repertory :

```
./generate-mapping -u user -p password -o mapping.ttl
        jdbc:mysql://localhost:PORT/databaseName
```

It is then possible to run the server to be able to use the provided Sparql endpoint and to make queries with the Sparql syntax providing by the mapping.

```
./d2r-server mapping.ttl
```

[1] http://d2rq.org

[2] http://d2rq.org/generate-mapping