Ultrawrap Mapper: A Semi-Automatic Relational Database to RDF (RDB2RDF) Mapping Tool

Juan F. Sequeda, Daniel P. Miranker

Capsenta Inc.
{juan,miranker}@capsenta.com

1 Introduction

In this demo, we will show the operation of Ultrawrap Mapper, a semi-automatic software for creating mappings from Relational Databases to RDF in the R2RML language.

In 2012, the W3C ratified two related standards for mapping relational database contents to RDF: the Direct Mapping [1] and R2RML [2]. The Direct Mapping is a default mapping of relational data to RDF. The organization and content of the resulting RDF triples is based on the schema and content of the database. Programs that implement the Direct Mapping do not need user input. R2RML is a language where users can express the mapping of relational data elements to RDF. A user must detail the mapping of relational data and metadata, (table and column names) to each field of a triple. Components of the language include, TripleMap, SubjectMap and PredicateObjectMap, etc. Depending on the size of the database, the R2RML mapping files can be very large.

Recent R2RML editors have been developed which provide users a GUI to create the mappings [3,4,5]. These tools are a first step to eliminating the burden of learning the R2RML syntax. However, we observe that these tools are still R2RML and RDF driven: the user needs to understand how R2RML conceptually works.

Ultrawrap Mapper is distinguished in several ways; It embodies a workflow, integrates semi-automatic ontology mapping and hides the complexity of R2RML. Given a JDBC connection to the source database, an R2RML file is (internally) automatically created representative of an enhanced Direct Mapping, bootstrapping the mapping process. The Direct Mapping is enhanced by making it consistent with an ontological model of the source database, the Putative Ontology.

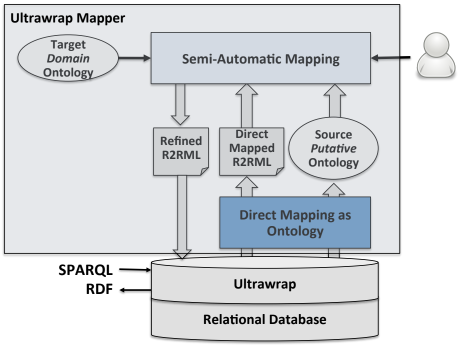


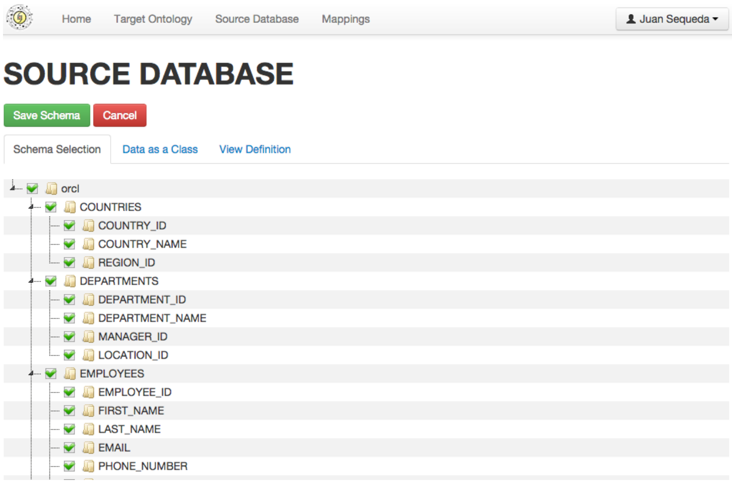
Fig. 1. Ultrawrap Mapper Architecture

The Ultrawrap Compiler product is embedded to produce the Putative Ontology. The Ultrawrap Compiler reads the metadata of the database and, using rule-based inference, identifies additional semantics beyond the standard direct mapping. Users then upload a target ontology, (the domain ontology). The third and final step is to create the mappings. Ultrawrap Mapper incorporates automatic ontology mapping methods to align the putative and target ontologies [6]. In an interactive panel, a user reviews the suggested mappings and may mark it correct, or manually select the correct ontological concept in the target. The internal R2RML statements are updated accordingly.

The output of Ultrawrap Mapper is the mapping between the source relational database and target ontology represented in R2RML. Once the R2RML mapping has been created, any R2RML processor may be used, such as Ultrawrap. Figure 1 depicts the architecture.

2 Three Step Process

Step 1: Source Database The developer may choose precisely which tables and attributes of the relation database are to be exposed in the mapping. The tables and attributes that have been checked are the ones that will be part of the mapping process. See Fig 2. Ultrawrap Mapper will read the databases metadata and automatically build an ontological model of the data source; the source putative ontology.

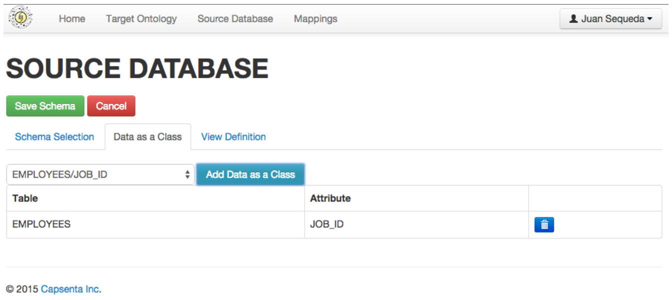


- ☑ orcl
  - ☑ COUNTRIES
    - ☑ COUNTRY_ID
    - ☑ COUNTRY_NAME
    - ☑ REGION_ID
  - ☑ DEPARTMENTS
    - ☑ DEPARTMENT_ID
    - ☑ DEPARTMENT_NAME
    - ☑ MANAGER_ID
    - ☑ LOCATION_ID
  - ☑ EMPLOYEES
    - ☑ EMPLOYEE_ID
    - ☑ FIRST_NAME
    - ☑ LAST_NAME
    - ☑ EMAIL
    - ☑ PHONE_NUMBER

Fig. 2. Schema Selection

The user may improve the source ontology by incorporating data into the ontology. It is common that data values under an attribute are metadata. For example, in an EMPLOYEE table, an attribute named JOB will have values that may appear in a Job role taxonomy. The user can choose which attributes have these characteristics. Ultrawrap Mapper will make the data values explicit as ontology classes, hence the name Data as a Class. See Fig 3.

Complex data transformations and mappings can be introduced using SQL syntax in the form of view definitions. Ultrawrap Mapper allows users to input SQL queries. For example a view can represent a new concept of Executive Employees which are those who have salary greater than $15000. The result of a query can be previewed in order to verify the result. See Fig 4.

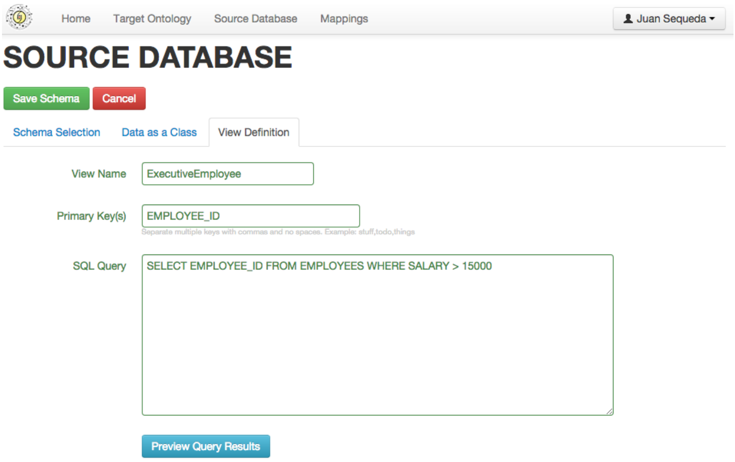


EMPLOYEES/JOB_ID

Table	Attribute	
EMPLOYEES	JOB_ID	

© 2015 Capsenta Inc.

Fig. 3. Data as a Class



View Name: ExecutiveEmployee

Primary Key(s): EMPLOYEE_ID

Separate multiple keys with commas and no spaces. Example: stuff,todo,things

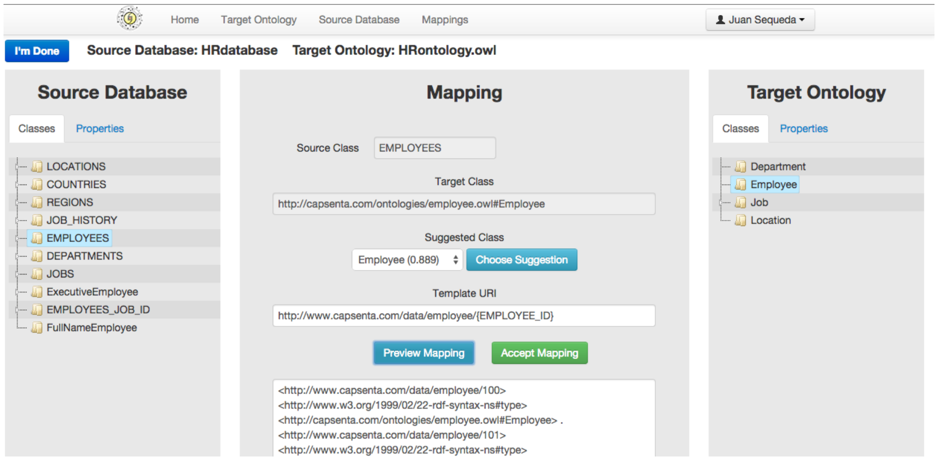
SQL Query:

```
SELECT EMPLOYEE_ID FROM EMPLOYEES WHERE SALARY > 15000
```

Fig. 4. View defining a concept of Executive Employees

Step 2: Target Ontology Any valid OWL ontology or SKOS taxonomy can be uploaded to Ultrawrap Mapper from a file or from the web. If a Target Ontology doesn't exist, the user can use the OWL representation of the source database as a way to bootstrap the creation of the Target Ontology.

Step 3: Mapping Ultrawrap Mapper bootstraps the editing process by automatically generating an R2RML file consistent with the enriched version of the direct mapping. This enriched version reflects Ultrawrap's inference process that deduces additional ontological semantics. For each field of the source model the user chooses the corresponding field in the target model. i.e. this defines a mapping. The actual action is to replace the default label with a label from the target ontology. Given that Ultrawrap forms an ontological model of the source relational database, the source putative ontology, the mapper is able to make mapping recommendations base on aligning the source and target ontology by using existing ontology matching techniques [6]. The user can choose from a list of suggestions.



Source Database: HRdatabase Target Ontology: HRontology.owl

Source Class: EMPLOYEES

Target Class: http://capsenta.com/ontologies/employee.owl#Employee

Suggested Class: Employee (0.889)

Template URI: http://www.capsenta.com/data/employee/{EMPLOYEE_ID}

```
<http://www.capsenta.com/data/employee/100>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://capsenta.com/ontologies/employee.owl#Employee> .
<http://www.capsenta.com/data/employee/101>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
```

Fig. 5. Mapping panel of Ultrawrap Mapper

Each row needs to be uniquely identified by a URI. The Template URI field provides a default URI identifier based on the primary key. Users can edit the template URI, if desired. The Preview Mapping button will return the resulting RDF per the chosen mapping.

3 Conclusions

Ultrawrap Mapper is a tool that enables users to create mappings between a relational database and a target ontology without having the conceptual knowledge of R2RML. Additionally, it incorporates semi-automatic ontology matching techniques in order to reduce the burden of creating the mappings. Ultrawrap Mapper is currently being used commercially for relational database integration.

References

1. M. Arenas, A. Bertails, E. Prud'hommeaux, and J. Sequeda. Direct mapping of relational data to RDF. W3C Recomendation 27 September 2012, `http://www.w3.org/TR/rdb-direct-mapping/`.
2. S. Das, S. Sundara, and R. Cyganiak. R2RML: RDB to RDF mapping language. W3C Recomendation 27 September 2012, `http://www.w3.org/TR/r2rml/`.
3. L. E. T. Neto, V. M. P. Vidal, M. A. Casanova, and J. M. Monteiro. R2RML by assertion: A semi-automatic tool for generating customised R2RML mappings. In *ESWC Poster*, 2013.
4. C. Pinkel, C. Binnig, P. Haase, C. Martin, K. Sengupta, and J. Trame. How to best find a partner? an evaluation of editing approaches to construct R2RML mappings. In *ESWC*, 2014.
5. K. Sengupta, P. Haase, M. Schmidt, and P. Hitzler. Editing R2RML mappings made easy. In *ISWC Posters & Demonstrations Track*, 2013.
6. A. Tian, J. F. Sequeda, and D. P. Miranker. QODI: query as context in automatic data integration. In *ISWC*, 2013.