SQuaRE: A Visual Tool for Creating R2RML Mappings

Jarosław Bąk and Michał Blinkiewicz

Institute of Control and Information Engineering,
Poznan University of Technology,
Piotrowo 3a, 60-965 Poznan, Poland
firstname.lastname@put.poznan.pl

Abstract. We present the recent progress of SQuaRE, the SPARQL Queries and R2RML mappings Environment that provides a graphical interface for creating R2RML mappings which can be immediately tested by executing SPARQL queries. SQuaRE is a web-based tool with easy to use interface that can be applied in the ontology-based data access applications. We describe SQuaRE's main features, its architecture as well as technical details.

1 Introduction and Motivation

OWL/OWL2 ontologies require data to be in a format of RDF triples. Then, using an appropriate reasoner we can obtain new data in the same format. In such a way we are adding the semantics to data and as a consequence, may obtain semantic results by formulating and executing SPARQL queries. Nevertheless, ontologies and data need to follow the RDF-based representation to be queried by SPARQL. Since most of data are stored in different formats, any application of an OWL ontology rises the integration problem. In this case we can transfer our current data format into RDF-based representation and change our software and architecture environment or we can create mappings between ontology and our data and use an appropriate tool that handles such a solution. The first option is very cost-expensive and needs a lot of changes in the current software architecture. The second approach is easier and cheaper since minor changes in the software architecture are required. We need to create mappings and then query non-RDF data with SPARQL using ontology, mappings and a tool that enables on-the-fly transfer from non-RDF into RDF data. In this method the most important part is to create appropriate mappings. Currently, a very popular standard for expressing mappings from relational databases to RDF data is W3C's R2RML[1]. The standard allows to use existing relational data in the RDF data model, and then use SPARQL to query such data.

In this paper, we provide a detailed description of SQuaRE, the SPARQL Queries and R2RML mappings Environment, which provides a visual editor for creating and managing R2RML mappings and for creating and executing

[1] https://www.w3.org/TR/r2rml/

SPARQL queries. SQuaRE is a web-based application that simplifies the creation of mappings between a relational database and an ontology. It also enables to test created mappings by defining and executing SPARQL queries.

2 The SQuaRE Framework

Features. The SQuaRE environment is aimed at providing easy-to-use functions that will support creation and execution of SPARQL queries as well as creation of R2RML mappings. Thus, SQuaRE provides the following features:

1. Browsing a relational database – a user can choose a data source and browse its schema. In this view the user sees table names, column names as well as data types stored in each column. The current implementation supports JDBC-based interfaces for relational databases. We already tested PostgreSQL and MySQL servers.
2. Browsing an OWL ontology – a user sees hierarchies of classes, object properties and datatype properties.
3. Browsing mappings – a list of all created mappings is shown to a user. The user can choose a mapping and then edit it in the mapping creation view.
4. Graphical creation of R2RML mappings – in a mapping creation view a user can create R2RML mappings. The user needs to choose tables that are going to be mapped. Then, she/he needs to search for appropriate classes and properties to create mappings using a graphical interface. An example mapping of a table that contains data about companies to the NPD-benchmark ontology[2] is shown in Figure 1.
5. Management of R2RML mappings – each created mapping is saved for future reference. A user can delete mappings, correct them or generate an R2RML file that contains all or selected mappings.
6. Textual creation of SPARQL queries – current version of SQuaRE provides an option to create SPARQL queries using a text-based interface. A user can write and execute a query.
7. Management of SPARQL queries – each constructed SPARQL query can be saved and used in future. A user can execute, delete or export: (i) a query or (ii) a set of selected queries.
8. Execution of SPARQL queries – created SPARQL queries can be executed and results are shown as (i) a table, (ii) a graph or (iii) 'yes'/'no' answer (according to the type of a query: (i) SELECT, (ii) DESCRIBE, CONSTRUCT and (iii) ASK, respectively).

The aforementioned main list of features provides an intuitive ontology-based access to relational data. Moreover, by importing/exporting functionality a user can use SQuaRE to create mappings and test them by creating SPARQL queries, and then save everything into external files. As a result we are able to import queries and mappings into another tool that supports SPARQL and R2RML.

[2] `https://github.com/ontop/npd-benchmark`

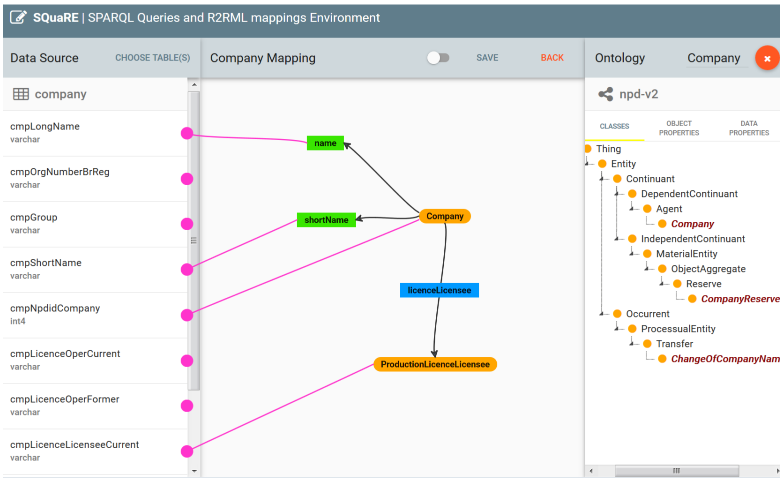

Fig. 1: Creating a mapping.

Architecture and Applied Tools. SQuaRE is developed in Java as a web application. The architecture of SQuaRE is presented in Figure 2. The tool consists of modules that provide different functionality.

The SQuaRE architecture is divided into two separate sides – client and server. The client-side main module, from the user's point of view, is Visual R2RML Mapper which provides tools for visual graph-based mappings of relational database metadata, such as table columns, and user provided ontology entities. Moreover, there are modules responsible for data source configuration and ontology management.

The server-side modules consist of: (i) Data Source and DBMS Manager which manages the user defined data sources and provides JDBC-based access, (ii) Ontology Handler for OWL ontology processing, and (iii) SPARQL Query Executor which utilizes already defined mappings and allows to execute SPARQL queries in the context of a relational database.

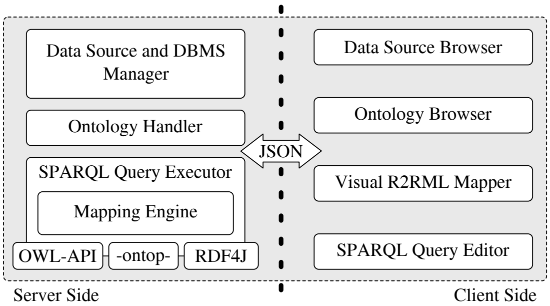

Fig. 2: The architecture of SQuaRE.

SQuaRE applies well-known tools to handle OWL ontologies, relational data and SPARQL queries. The main tools that SQuaRE uses are the following:

- OWL-API– this is a very often used Java library to handle OWL/OWL2 ontologies. It contains a lot of features that are useful when manipulating ontology elements, using reasoner or serialising ontologies.
- The Spring Framework – it is an application framework for the Java platform. Among others, it allows for an easy creation of RESTful web services and building backend API.
- -ontop-[3] – it is a platform to query relational databases as virtual RDF graphs using SPARQL. The tool accepts mappings in R2RML and its own OBDA mapping language. SQuaRE uses -ontop- to query a relational database using mappings and SPARQL.
- RDF4J – it is a framework for processing RDF data. It enables to parse, store, infer and query of/over such data, it supports SPARQL 1.1. SQuaRE uses it to save all data connected with created mappings and queries.
- Javascript libraries – we use a set of popular Javascript tools like: AngularJS, jQuery, Cytoscape.js, jsPlumb and jsTree.

3 Summary and Future Work

In this paper, we presented the SQuaRE tool which is a web-based environment that provides: (i) creation of R2RML mappings between relational databases and OWL ontologies, and (ii) creation and execution of SPARQL queries.

Currently, we are implementing a graph-based method for creating and executing SPARQL queries. In this case we will fully support a graphical environment for handling R2RML and SPARQL. We also plan to include RuQAR [1] to extend reasoning capabilities and provide support for SWRL rules. Moreover, the long term plans are to support other mapping languages, like RML [2]. As a result we will be able to map different data sources like CSV, JSON and others. We intend to make SQuaRE open source as soon as we finish the graphical method of creating SPARQL queries.

Acknowledgments. The work presented in this paper was supported by 04/45/DSMK/0158 project.

References

1. Jarosław Bąk. RuQAR : Reasoning with OWL 2 RL using forward chaining engines. In *Informal Proceedings of the 4th International Workshop on OWL Reasoner Evaluation (ORE-2015) co-located with the 28th International Workshop on Description Logics (DL 2015), Athens, Greece, June 6, 2015.*, pages 31–37, 2015.
2. Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. RML: a generic language for integrated RDF mappings of heterogeneous data. In *Proceedings of the 7th Workshop on Linked Data on the Web*, April 2014.

[3] http://ontop.inf.unibz.it/