Towards an Interface for User-Friendly Linked Data Generation Administration

Anastasia Dimou, Pieter Heyvaert, Wouter Maroy, Laurens De Graeve, Ruben Verborgh, and Erik Mannens

In: Proceedings of the 15th International Semantic Web Conference: Posters and Demos. , 2016.

Towards an Interface for User-Friendly Linked Data Generation Administration*

Anastasia Dimou, Pieter Heyvaert, Wouter Maroy, Laurens De Graeve, Ruben Verborgh, Erik Mannens, and Rik Van de Walle

Ghent University - iMinds, Belgium {firstname.lastname}@ugent.be

Abstract. Linked Data generation and publication remain challenging and complicated, in particular for data owners who are not Semantic Web experts or tech-savvy. The situation deteriorates when data from multiple heterogeneous sources, accessed via different interfaces, is integrated, and the Linked Data generation is a long-lasting activity repeated periodically, often adjusted and incrementally enriched with new data. Therefore, we propose the RMLWorkbench, a graphical user interface to support data owners administrating their Linked Data generation and publication workflow. The RMLWorkbench's underlying language is RML, since it allows to declaratively describe the complete Linked Data generation workflow. Thus, any Linked Data generation workflow specified by a user can be exported and reused by other tools interpreting RML.

Keywords: Linked Data Generation, Linked Data Workbench, [R2]RML

1 Introduction

Administrating the integration of the ever-increasing amounts of data from multiple sources in different formats into a common knowledge domain remains challenging and complicated, in particular for data owners who are not Semantic Web experts or tech-savvy [4]. Generating Linked Data requires dealing with data that can originally (i) reside on diverse, distributed locations, (ii) be approached using different access interfaces, and (iii) be expressed in heterogeneous structures and formats [3]. As the Linked Data generation becomes a long-lasting activity, which is repeated periodically and is incrementally adjusted with new data, administrating the different components becomes difficult.

To minimize the effort and knowledge that data owners need to administrate their data and the overall Linked Data generation and publication workflow, we developed a multi-user browser application, the RMLWorkbench. This demo shows how data owners can use the RMLWorkbench. Depending on their assigned roles, data owners can view and manage different sources for retrieving raw data and the corresponding mappings to generate Linked Data, in contrast to our earlier work on the RMLEditor [4] which only focuses on editing mapping rules. A screencast of the RMLWorkbench, is available at https://youtu.be/8UkI01nQNxc.

* This paper's research activities described in this paper were funded by Ghent University, iMinds, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders) and the European Union.

2 State of the Art

The FluidOps Information Workbench[1], Ultrwrap[2] and LinDa workbench[3] are GUI tools supporting data owners to generate Linked Data. However, the latter two only support tabular data, while the former, even though it supports more data sources, does not allow specifying different access interfaces. Most importantly though, none of them allows users to export their specified Linked Data generation workflow in a declarative, complete and interoperable way that allows to replicate the same Linked Data generation by other tools.

Linked Pipes, and its predecessor Unified Views[4], are general-purpose tools that allow users to administrate, execute, debug, monitor and share Linked Data processing tasks, for smooth and efficient management. However, they are not focused on Linked Data generation. They perform direct mappings which are afterwards processed via SPARQL construct queries. Moreover, they only allow to export the different processes descriptions, using their own custom descriptions.

The Silk Workbench[5] follows a similar approach as the RMLWorkbench. Even though it is a GUI supporting users to administrate RDF dataset linking, it also requires corresponding aspects to be specified. Its function relies on projects which consist of linkage rules associated with data sources (data dumps or SPARQL endpoints) constituting altogether linkage tasks, as mapping rules are associated with data sources constituting generation tasks in the case of the RMLWorkbench.

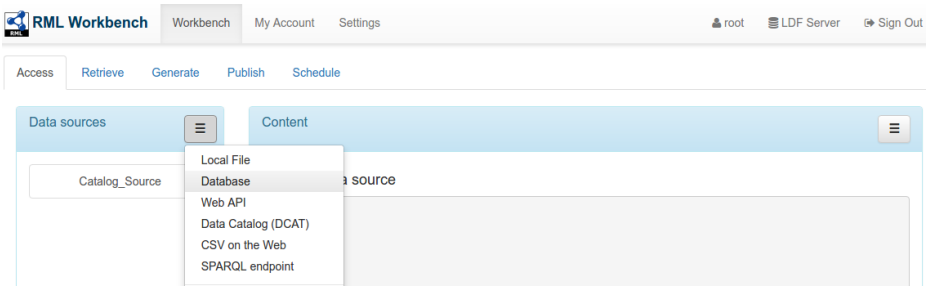
3 The RML Workbench Interface

The RMLWorkbench design principles are generic, following the classical multi-tier client-server architecture. Its underlying language to declaratively define the Linked Data generation workflow specified by the user is RML. RML [2] is a generalization of the W3C recommended R2RML mapping language [1], which is defined to specify rules to generate Linked Data from data in relational databases. RML extends R2RML to also specify rules from data in any semi-structured format, e.g., CSV, XML, or JSON. RML was furthermore aligned with different vocabularies, e.g., DCAT[6], CSVW[7], or Hydra[8], to specify how to access data used to generate the desired Linked Data. The RMLWorkbench considers RML as its underlying language, since it is the only one able to declaratively describe the complete Linked Data generation workflow, independently of data sources and formats [3]. Thus, all mapping rules, including the aligned data sources description may be exported and re-used by other tools beyond the RMLWorkbench to replicate the generation of same Linked Data.

[1] https://www.fluidops.com/en/portfolio/information_workbench/
[2] https://capsenta.com/
[3] https://github.com/LinDA-tools/LindaWorkbench
[4] http://unifiedviews.eu/
[5] https://github.com/silk-framework/silk/blob/master/doc/Workbench.md
[6] https://www.w3.org/TR/vocab-dcat/
[7] https://www.w3.org/TR/tabular-metadata/
[8] https://www.hydra-cg.com/spec/latest/core/

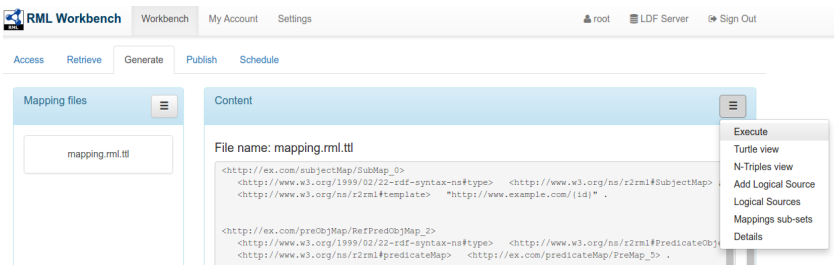
The RMLWorkbench consists of five panels: *Access*, *Retrieve*, *Generate*, *Publish* and *Schedule*. In the remaining of this section, each panel is briefly presented.

Access panel. Users can manage their own sources, which can be accessed through interfaces for local files, databases or Web sources. The descriptions are annotated using different vocabularies, e.g., DCAT, CSVW or Hydra. For instance, a user specifies a database accessed via a certain JDBC, and labels it "DB Source" and a dataset published on a DCAT catalog, which he labels "Catalog Source".



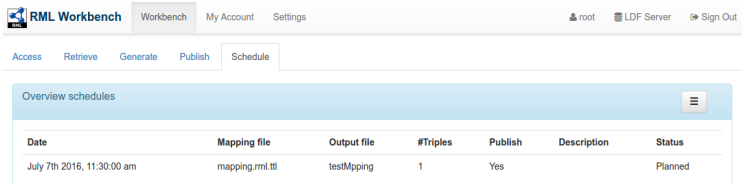
Retrieve panel. Not all data which appear in a data source are required to generate the desired Linked Data. Distinct subsets may be considered separately for generating different Linked Data sets. The *Retrieve* panel allows users to specify which exact data is retrieved for each selection. For instance, via the *Retrieve panel*, the aforementioned user specifies the exact tables, which are eventually considered to generate certain Linked Data. The user specifies the "Singers" and "Albums" tables of the "DB source" and labels them as "Singer data" and "Album data" respectively. Moreover, the same user specifies and labels as "Performance data", among the different datasets of the "Catalog source", the dataset about performances, and precisely its XML distribution.

Generate panel. To generate the desired Linked Data, the users need to specify sets of mapping rules. The RMLWorkbench allows users to (i) upload a mapping document, (ii) specify a Web source with mapping rules, or (iii) directly edit them via its interface. Different sets of mapping rules may be associated with the same data, generating thus different Linked Data *views*. Once the set of mapping rules is associated with some raw data, the users can execute the mapping and generate the desired Linked Data ("Execute" button, as shown in the following figure). The dataset is then listed among the datasets available for publishing, or the users are notified if the generation was not successful. The users can specify mapping rules, for instance the sets of "Singer mappings" and "Performance mappings". Once the mapping rules are listed among the available sets, the users can associate them with the corresponding data ("Add Logical Source" button), in our example the "Singer data" and "Performance data". Furthermore, the users may desire to generate another Linked Data set with the same data. In that case, another set of mapping rules is added, e.g., the "Person mappings", and the user associates it with the "Singer data" as well.



Publish panel. A frequent activity after generating Linked Data is its publication. The RMLWorkbench supports users to easily accomplish this activity. In our example, the Linked Data is published via an LDF server[9]. Nevertheless, the administrator can easily configure other interfaces, for instance a SPARQL endpoint. The users can then choose one or more of them to publish their Linked Data.

Schedule panel. In most cases, the Linked Data generation and publication is a recurring activity. Data owners periodically regenerate their Linked Data set to keep it up-to-date with the original data. The RMLWorkbench allows data owners to schedule the Linked Data generation and publication activities.



To summarize, the RMLWorkbench allows data owners to specify their complete Linked Data generation workflow, without being restricted by the tool. In the future, the RMLEditor [4] will be integrated with the RMLWorkbench and users can then directly use the RMLEditor to edit their mapping rules.

References

1. S. Das, S. Sundara, and R. Cyganiak. R2RML: RDB to RDF Mapping Language. Working Group Recommendation, W3C, Sept. 2012. `http://www.w3.org/TR/r2rml/` .
2. A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Workshop on Linked Data on the Web*, 2014.
3. A. Dimou, R. Verborgh, M. Vander Sande, E. Mannens, and R. Van de Walle. Machine-Interpretable Dataset and Service Descriptions for Heterogeneous Data Access and Retrieval. In *SEMANTiCS 2015*, 2015.
4. P. Heyvaert, A. Dimou, A.-L. Herregodts, V. Ruben, S. Dimitri, M. Erik, and V. de Walle Rik. RMLEditor: A Graph-Based Mapping Editor for Linked Data Mappings. In *The Semantic Web: ESWC 2016*. Springer, 2016.

[9] `https://github.com/LinkedDataFragments/Server.js`