RML and FnO: Shaping DBpedia Declaratively

Ben De Meester, Wouter Maroy, Anastasia Dimou,
Ruben Verborgh, and Erik Mannens

Ghent University – imec – IDLab, Department of Electronics and Information Systems, Belgium
{firstname.lastname}@ugent.be

Abstract. DBpedia data is largely generated from extracting and parsing the wikitext from the infoboxes of Wikipedia. This generation process is handled by the DBpedia Extraction Framework (DBpedia EF). This framework currently consists of *data transformations*, a series of custom hard-coded steps which parse the wikitext, and *schema transformations*, which model the resulting RDF data. Therefore, applying changes to the resulting RDF data needs both Semantic Web expertise and development within the DBpedia EF. As such, the current DBpedia data is being shaped by a small amount of core developers. However, by describing both schema and data transformations declaratively, we shape and generate DBpedia data using solely declarations, splitting the concerns between implementation and modeling. The parsing functions development is decoupled from the DBpedia EF, and other data transformation functions can easily be integrated during DBpedia data generation. This demo showcases an interactive Web application that allows non-technical users to (re-)shape the DBpedia data and use external data transformation functions, solely by editing a mapping document via HTML controls.

Keywords: DBpedia, Data Transformations, FnO, Linked Data Generation, RML

1 Introduction

One of the most widely known Linked Datasets is DBpedia, a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web [1]. Data from DBpedia is generated using the DBpedia Extraction Framework (DBpedia EF) in two parts: the first part directly maps relationships from the relational database of the underlying application on which Wikipedia is built (WikiMedia), and the second part extracts and parses data from the article texts and infobox templates within the articles [2].

The successive steps of extracting the Wikipedia articles, selecting the right content, parsing the values, creating the resources and adding the relationships to generate RDF data are currently performed by the DBpedia EF in custom, hard-coded steps. For example, the founding date is extracted from the infobox (e.g., `19-4-1839` for Belgium), parsed into the correct date format (a *data transformation* function generating `1839-04-19`), and linked with the resource for Belgium

using the correct DBpedia predicate (a *schema transformation* generating the triple `dbr:Belgium dbo:foundingDate "1839-04-19"^^xsd:date`).

Limited changes in the schema (e.g., adding a predicate-object pair) are currently possible using the DBpedia mapping wiki, but more extensive changes, both in the schema (e.g., using a different ontology) or in the data (e.g., using a different parsing function) involve changing the DBpedia EF source code. Thus, desired changes in the resulting RDF data currently needs both Semantic Web expertise and development within the DBpedia EF. These combined requirements are currently met by only a small amount of core developers to shape DBpedia data. As adding schema or data transformations is complex, the DBpedia EF inhibits problems that are currently not easily solved. For instance, Blake et. al. [8] unveiled quality issues in DBpedia as the current extraction framework does not support basic geographic calculations, e.g., calculating the population density. Being a custom, hard-coded framework, the DBpedia EF does not easily allow generating alternative RDF data solving these issues.

A fully declarative solution would no longer require development effort when apply changes to the DBpedia data. Instead, only editing the mapping document that shapes DBpedia is needed. This involves decoupling both the schema and data transformations from the DBpedia EF implementation. Previous work[1] has already extracted the schema transformations as RML mapping documents [5], and our recent work – which this demo accompanies – provides an approach to integrate data and schema transformations declaratively [4]. Before, changing the data transformation functions would require developers to improve the implementation of the DBpedia EF. Now, data modelers without technical background can change and replace data transformation functions or schema transformations by editing the mapping documents. Thus, the concerns between developers and modelers is decoupled. Moreover, the data transformation functions can be reused for different use cases, not only for DBpedia data.

This demo, which is available at `https://fnoio.github.io/dbpedia-demo/`, shows how declarative data and schema transformations make it easier to apply changes in DBpedia data. Users can alter among different data transformation functions – even functions not yet supported in the current DBpedia EF – by solely adjusting single fields within the mapping document. Using exemplary Wikipedia articles, the users can immediately verify their changes, as they are reflected at the generated Linked Data for each Wikipedia article. This Web application shows that technical knowledge about the DBpedia EF is no longer needed to make significant changes when shaping the DBpedia data.

2 Background: Integrated Schema and Data Transformations using RML and FnO

Generating Linked Data involves making changes to both the schema and transforming the data values of the data sources [7]. The same is the case for DBpedia.

[1] `http://www.mail-archive.com/dbpedia-discussion@lists.sourceforge.net/msg07837.html`

On the one hand, schema transformations are needed to make sure the right ontologies and vocabularies are used, and that the values are related as intended, with the right data type (e.g., modeling the founding date of Belgium using the correct predicate of the DBpedia ontology and using a date as data type results in `dbr:Belgium dbo:foundingDate "1839-04-19"^^xsd:date`). On the other hand, very specific data transformations are required for DBpedia to parse the manually entered data in the Wikipedia infoboxes, as the input data can be inserted using *different formats* for the same data type (e.g., `04-10-1830`, and `October 4th 1830` denote the same date), using *different units* (e.g., entering degrees in Fahrenheit in a Celsius-valued field), or having *typos and misspellings*. In the current DBpedia EF these functions are hard-coded, thus changing these specific functions (or using different ones) entails a significant development effort.

We aligned the following technologies:

- RML [5] – a mapping language to define *schema transformations* to generate Linked Data derived from heterogeneous data, wikitext in our case; and
- FnO [3] – an ontology to describe *data transformations*, independently of their implementation and the data to which they are applied.

This way, schema transformations may be aligned with data transformations. Moreover, the aforementioned alignment does not restrict a mapping processor to support a specific set of data transformations.

This integration was kept minimal by using a single class and predicate[2]. The resulting implementation depends on the RMLProcessor[3] and a generic Function Processor[4]. We have uncoupled the DBpedia parsing functions from the DBpedia EF and re-published them as a stand-alone library[5], and allowed describing more advanced schema and data transformations declaratively [4].

3 Easily Shaping DBpedia

By decoupling the declaration from the implementation, as we explain in details at De Meester et al. [4], users without technical expertise can shape the generated DBpedia data by directly editing the DBpedia mapping document. The current DBpedia EF allows limited changes in the schema transformations without development effort using the DBpedia mapping wiki. The fully declarative solution allows more editing options: users can apply changes to both schema transformations (e.g., adding/removing types to resources and changing predicates), and data transformations (e.g., changing the parameters of the parsing functions, using different parsing functions, or even using externally defined functions). The data transformation functions are no longer restricted by the DBpedia EF.

`https://fnoio.github.io/dbpedia-demo/` shows an interactive Web application that allows users to easily apply changes, both for schema and data

[2] `http://semweb.datasciencelab.be/ns/fnml#`
[3] `https://github.com/RMLio/RML-Mapper/tree/extension-fno`
[4] `https://github.com/FnOio/function-processor-java`
[5] `https://github.com/FnOio/dbpedia-parsing-functions-scala`

transformations, to the DBpedia mapping documents (Figure 1). The Web application does not require users to learn a new syntax, instead, HTML form elements are used to make changes to the underlying mapping document. After the extended RMLProcessor executes the updated transformations, users can review the applied changes to the newly generated RDF data.

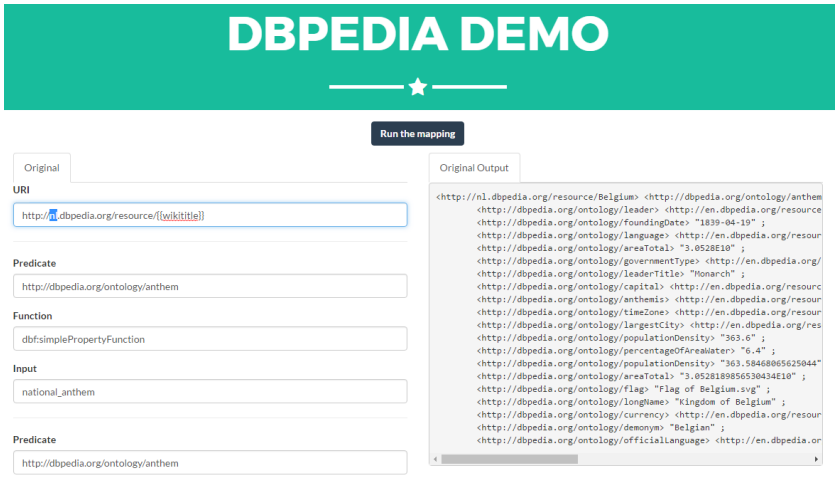

Fig. 1: When DBpedia is fully shaped declaratively, simple form controls can be used to edit the DBpedia mapping document and technical expertise about the DBpedia EF is no longer needed.

Users can generate RDF data for different types of Wikipedia infoboxes (e.g., infoboxes denoting persons or countries), using different Wikipedia articles (e.g., Belgium or The United States of America). Relying on HTML form elements, such as dropdowns and radio buttons, users can apply changes to the mapping document. Generic data transformations as defined by the popular data cleansing tool OpenRefine[6] are also selectable. This showcases that the restriction on which data transformation functions you can use is lifted. Via the *Generate* button, the updated mapping document is executed server-side. The resulting RDF data is returned to the users for inspection, together with the mapping document. The mapping documents with aligned RML and FnO statements can then be used in the updated DBpedia EF to generate the RDF data.

The implementation[7] shows that the Web application entirely depends on the mapping document, which contains the aligned declarative schema and data transformations. The mapping document is changed based on user interactions and saved as JSON-LD, instead of Turtle, for easier JavaScript manipulation. The server implementation is also provided, and as can be inspected, this is merely

[6] `https://github.com/OpenRefine/OpenRefine/wiki/GREL-Functions`
[7] `https://github.com/FnOio/dbpedia-demo`

a wrapper around the extended RMLProcessor, no case specific development was needed.

4 Conclusions

Integrating data and schema transformations in mapping documents which contain aligned declarative schema and data transformations gives the opportunity to fully decouple the implementation of a Linked Data generation system without limiting its capabilities, as we show in more details at De Meester et al. [4]. The Web application presented in this demo showcases this potential to shape the generation of DBpedia merely using HTML form elements. Lowering the required skills (i.e., development skills, learning a new syntax) to make changes in theDBpedia mapping document can thus increase community involvement. To make full advantage of the possibilities of the alignment of RML and FnO, a full-fledged editor is advised. The RMLEditor [6] allows full manipulation of RML mapping documents and is extended with data transformation capabilities.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives, Z.: DBpedia: A nucleus for a Web of Open Data. In: 6th International Semantic Web Conference. pp. 11–15. Springer, Busan, Korea (2007)
2. Auer, S., Lehmann, J.: What have Innsbruck and Leipzig in common? Extracting semantics from wiki content. In: European Semantic Web Conference. Lecture Notes in Computer Science, vol. 4519, pp. 503–517. Springer (2007)
3. De Meester, B., Dimou, A., Verborgh, R., Mannens, E., Van de Walle, R.: An Ontology to Semantically Declare and Describe Functions. In: The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 – June 2, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9989, pp. 46–49. Springer (2016)
4. De Meester, B., Maroy, W., Dimou, A., Verborgh, R., Mannens, E.: Declarative data transformations for Linked Data generation: the case of DBpedia. In: Proceedings of the 14th ESWC (May 2017)
5. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: a generic language for integrated RDF mappings of heterogeneous data. In: Bizer, C., Heath, T., Auer, S., Berners-Lee, T. (eds.) Proceedings of the 7th Workshop on Linked Data on the Web. CEUR Workshop Proceedings, vol. 1184 (Apr 2014)
6. Heyvaert, P., Dimou, A., Herregodts, A.L., Verborgh, R., Schuurman, D., Mannens, E., Van de Walle, R.: RMLEditor: A Graph-based Mapping Editor for Linked Data Mappings. In: The Semantic Web – Latest Advances and New Domains (ESWC 2016). Lecture Notes in Computer Science, vol. 9678, pp. 709–723. Springer (2016)
7. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin 23(4), 3–13 (2000)
8. Regalia, B., Janowicz, K., Gao, S.: VOLT: A provenance-producing, transparent sparql proxy for the on-demand computation of linked data and its application to spatiotemporally dependent data. In: The Semantic Web. Latest Advances and New Domains (2016)