Knowledge Representation as Linked Data

Joachim Van Herwegen
IDLab, Dep. of Electronics and Information Systems, imec – Ghent University
Joachim.VanHerwegen@ugent.be

Pieter Heyvaert
IDLab, Dep. of Electronics and Information Systems, imec – Ghent University
pheyvaer.Heyvaert@UGent.be

Ruben Taelman
IDLab, Dep. of Electronics and Information Systems, imec – Ghent University
Ruben.Taelman@UGent.be

Ben De Meester
IDLab, Dep. of Electronics and Information Systems, imec – Ghent University
ben.demeester@ugent.be

Anastasia Dimou
IDLab, Dep. of Electronics and Information Systems, imec – Ghent University
Anastasia.Dimou@ugent.be

ABSTRACT

The process of extracting, structuring, and organizing knowledge requires processing large and originally heterogeneous data sources. Offering existing data as Linked Data increases its shareability, extensibility, and reusability. However, using Linking Data as a means to represent knowledge can be easier said than done. In this tutorial, we elaborate on how to semantically annotate data, and generate and publish Linked Data. We introduce [R2]RML languages to generate Linked Data. We also show how to easily publish Linked Data on the Web as Triple Pattern Fragments. As a result, participants, independently of their knowledge background, can model, annotate and publish Linked Data on their own.

KEYWORDS

Linked Data Generation, Linked Data Publishing, [R2]RML, Linked Data Fragments

ACM Reference format:
Joachim Van Herwegen, Pieter Heyvaert, Ruben Taelman, Ben De Meester, and Anastasia Dimou. 2018. Knowledge Representation as Linked Data. In *Proceedings of The 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, October 22–26, 2018 (CIKM '18),* 2 pages. ACM, New York, NY, USA.
https://doi.org/10.1145/3269206.3274275

1 INTRODUCTION

Semantic Web technologies and Linked Data gains traction as a prominent solution for machine-interpretable knowledge representation. However, only a limited amount of data is available as Linked Data, as acquiring semantically enriched representations remains complicated, in addition to scalability issues that emerge once Linked Data is published for consumption.

This tutorial shows how to perform the different steps to make data available as Linked Data, forming a *Linked Data publishing workflow*. By the end of this tutorial, data owners should know how to profit of modeling the knowledge that appears in their data (Section 2), semantically annotating them to generate corresponding Linked Data (Section 3), and publishing them (Section 4).

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10.
https://doi.org/10.1145/3269206.3274275

2 LINKED DATA MODELING

Modeling is the first step of a Linked Data publishing workflow. It involves defining how to make knowledge available as Linked Data. In this step, raw data is modeled and semantically annotated using vocabularies. Data owners indicate (i) the **entities** that appear in the dataset, by assigning *IRIs*; (ii) how **attributes** are related to the entities, using *predicates*; (iii) what **(data)types** are of these entities, by using *classes*, and of these attributes, by using `xsd`[1] or custom *datatypes*; and (iv) **relationships** between entities which might originally be in different data sources, by using *predicates*.

Mapping languages allow to declaratively specify the rules which are defined during the modeling step. However, directly editing them using these languages is difficult for data owners who are not Semantic Web experts. Therefore, graphical user interface tools, such as the RMLEditor [7], are developed to ease modeling and support in defining rules to semantically annotate raw data and generate Linked Data by hiding the underlying mapping language.

3 LINKED DATA GENERATION

The next step in a Linked Data publishing workflow is the *generation* step. Mapping languages detach the rules from the implementation that executes them, specifying in a declarative way how Linked Data is generated from raw data. Dedicated tools *validate* and *execute* these to generate the desired Linked Data and assure their quality.

Generation. The R2RML mapping language [2] was recommended by W3C in 2012 to define rules for generating Linked Data, but only from data which is derived from relational databases. In 2014, the RDF Mapping language (RML) [5] was proposed as a superset of R2RML, extending its applicability and broadening its scope. RML is a generic mapping language defined to specify customized rules that generate Linked Data derived from different heterogeneous data formats –e.g., DBs, XML, or JSON– and from different interfaces –e.g., files or Web APIs. RML is also considered to automatically

[1] https://www.w3.org/TR/swbp-xsch-datatypes/

generate related metadata information to assert provenance and determine ownership and trust [3].

Validation. Linked Data validation aids data owners acquire high quality Linked Data. Most frequent violations are related to the dataset's schema [9], which derives from the classes and properties specified in mapping rules. Applying mapping rules to raw data results in same violations being repeatedly observed even within the same Linked Data set. In this tutorial, we follow a methodology [4] that incorporates systematically the Linked Data validation in the Linked Data workflow, uniformly validating both the mapping rules and the resulting Linked Data. We consider following tools enabling high quality Linked Data generation: (i) the **RMLMapper**[2] executes rules expressed in RML to generate Linked Data, and (ii) **Validatrr**[3] validates rules expressed in RML or Linked Data [1].

4 LINKED DATA PUBLISHING

The next step in a Linked Data publishing workflow is *publication*. In this section, we discuss how Linked Data can be published.

4.1 Licensing, Announcement & Maintenance

Other tasks in the Linked Data publishing process include licensing and announcement, following the best practices for publishing Linked Data [8]. Concerning *licensing*, in most cases, the goal of Linked Data publishing is to reach Linked *Open* Data. By default, non-licensed data is not open, because regular copyright rules apply. The Open Knowledge Foundation[4] argues that openness is defined by (i) the availability and access of data; (ii) the possibility for anyone to reuse and redistribute the data; and (iii) universal participation, namely that no one should be excluded from these rights. A popular open license is the CC0 license[5], which is in line with the aforementioned definition of openness, and can be mentioned using for example the Creative Commons vocabulary[6]. For non-open data licenses, the publication strategy should then support this license by adding an authentication and authorization layer to the data-access interface for confidential and private information.

Generated Linked Data must be *announced* to the public. Communication channels include mailing lists, blogs, newsletters. A feedback channel must be in place to receive issues and questions about the Linked Data set. Furthermore, the dataset can also be published on registries, such as https://datahub.io/.

4.2 Linked Data Interfaces

One of the Linked Data publication main goals is for data to be *retrieved* and *discovered* by machines through HTTP interfaces. *Linked Data Fragments* (LDF) [11] was introduced as a framework for comparing different Linked Data publication interfaces. Next, we discuss 4 types of interfaces using the LDF framework.

Data Dump. A file containing a serialized Linked Data set. Data dumps do not provide any querying functionality themselves. Hence, significant effort is required from the client to query such Linked Data sets, although little effort from the data owner is required.

Linked Data Document. Returning the provided information when a URI is *dereferenced*. Such Linked Data require more effort from the publisher when compared to data dumps. Publication involves low server cost, browsing is easy, and limited querying is possible by traversing links.

SPARQL Query Result. SPARQL endpoints expose Linked Data through an interface that supports queries in the SPARQL query language [6], but suffers from significant availability issues.The server performs the entire query evaluation process, making SPARQL query engines a costly approach for publishing Linked Data. However, the required querying effort for clients is low.

Triple Pattern Fragments (TPF). TPF [11] was introduced as a trade-off between server and client effort for querying. The approach consists of a low-cost server interface that accepts triple pattern queries, while clients evaluate more complex SPARQL queries. TPF requires less effort from the server when compared to SPARQL endpoints [11], at the cost of slower query execution times and increased bandwidth. This approach allows publishing Linked Data at a low cost while still enabling efficient querying.

4.3 Querying

Once the data is published, there are multiple ways users can query the data, depending on which publishing interface was chosen. Comunica [10] is a modular SPARQL query engine which supports many of these interfaces, even at the same time. In this tutorial we will also cover how Comunica can be used to query your data.

REFERENCES

[1] Dörthe Arndt, Ben De Meester, Anastasia Dimou, Ruben Verborgh, and Erik Mannens. 2017. Using Rule Based Reasoning for RDF Validation. In *RuleML+RR*.

[2] Souripriya Das, Seema Sundara, and Richard Cyganiak. 2012. *R2RML: RDB to RDF Mapping Language*. W3C Recommendation. W3C. http://www.w3.org/TR/r2rml/

[3] Anastasia Dimou, Tom De Nies, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. 2016. Automated Metadata Generation for Linked Data Generation and Publishing Workflows. In *Proc. of the 9th Workshop on Linked Data on the Web*.

[4] Anastasia Dimou, Dimitris Kontokostas, Markus Freudenberg, Ruben Verborgh, Jens Lehmann, Erik Mannens, Sebastian Hellmann, and Rik Van de Walle. 2015. Assessing and Refining Mappings to RDF to Improve Dataset Quality. In *The Semantic Web – ISWC 2015*, Vol. 9367. 133–149.

[5] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. 2014. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Proc. of the 7th Workshop on Linked Data on the Web*, Vol. 1184.

[6] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. 2013. *SPARQL 1.1 Query Language*. Recommendation. W3C. http://www.w3.org/TR/sparql11-query/

[7] Pieter Heyvaert, Anastasia Dimou, Ben De Meester, Tom Seymoens, Aron-Levi Herregodts, Ruben Verborgh, Dimitrie Schuurman, and Erik Mannens. 2018. Specification and implementation of mapping rule visualization and editing: MapVOWL and the RMLEditor. *Web Semantics: Science, Services and Agents on the World Wide Web* 49 (2018), 31–50.

[8] Bernadette Hyland, Ghislain Atemezing, and Boris Villazón-Terrazas. 2014. *Best Practices for Publishing Linked Data*. Working Group Note. W3C. http://www.w3.org/TR/ld-bp/

[9] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. 2014. Test-driven Evaluation of Linked Data Quality. In *Proc. of the 23rd International Conference on World Wide Web*.

[10] Ruben Taelman, Joachim Van Herwegen, Miel Vander Sande, and Ruben Verborgh. 2018. Comunica: a Modular SPARQL Query Engine for the Web. In *Proceedings of the 17th International Semantic Web Conference*.

[11] Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. 2016. Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web. *Journal of Web Semantics* 37–38 (2016), 184–206.

[2] https://github.com/RMLio/rmlmapper-java
[3] https://github.com/IDLabResearch/validation-reasoning-framework
[4] https://okfn.org/opendata/
[5] https://creativecommons.org/publicdomain/zero/1.0/
[6] https://creativecommons.org/ns