High Quality Schema and Data Transformations for Linked Data Generation

Ben De Meester[*]

IDLab, Department of Electronics and Information Systems,
Ghent University – imec, Ghent, Belgium
ben.demeester@ugent.be

Abstract. High quality Linked Data is an important factor for the success of the Semantic Web. However, the quality of generated Linked Data is typically assessed and refined after the dataset is generated, which is computationally intensive. Given Linked Data is typically generated from (semi-)structured data which highly influences the intrinsic dimensions of the resulting Linked Data quality, I investigate how a generation process can automatically be validated before RDF data is even generated. However, current generation processes are not easily validated: descriptions of the data transformations depend on the use case or are incomplete, and validation approaches would require manual (re-)definition of test cases aimed at the generated dataset. I propose (i) a generic approach to declaratively describe a generation process, and (ii) a validation approach for automatically assessing the quality of the generation process itself. By aligning declarative data and schema transformations, the generation process remains generic and independent of the implementation. The transformations can be automatically validated based on constraint rules that apply to the generated RDF data graph using custom entailment regimes. Preliminary results show the generation process of DBpedia can be described declaratively and (partially) validated.

Keywords: Generation, Linked Data, Data Transformation

1 Relevancy

High quality Linked Data is an important factor for the success of the envisaged Semantic Web. As machines are inherently intolerant at interpretation of unexpected input, low quality data produces low quality results. Quality assessment – specifically for the intrinsic dimensions, i.e., directly related to the RDF graph [22] – can be automated by checking constraint violations [13]. A Linked Data generation approach that eases validation lowers the threshold for Linked Data publishers to generate data of high quality. Having access to Linked Data of higher quality is beneficial for all Linked Data consumers.

[*] Co-Promotor prof. dr. ir. Ruben Verborgh and Promotor prof. dr. ir. Erik Mannens.

2 Problem Statement

Assessing an entire dataset is computation and memory intensive [7]. However, Linked Data is typically generated from (semi-)structured data [7], and this generation process highly influences the quality assessment's intrinsic dimensions of the resulting Linked Data. E.g., wrongly defined schema transformations result in violations such as *entities as members of disjoint classes*, and incorrect data transformations result in *inaccurate values* such as parsing *March 8, '17* into the date *08-03-0017* [22]. Violations can be resolved in the dataset, however, the generation process that causes these violations is not improved. A new iteration of the generation process can re-introduce the same errors, and data validation needs to be re-executed. This results in duplicate work, wasted computation, and wasted time.

Problem Statement Iteratively validating generated Linked Data is computationally intensive and makes it hard to determine the root causes of quality violations.

The earlier a dataset's quality is assessed, the better [7]. When validating an RDF data graph, it is not clear what the cause of a constraint violation is, e.g., whether the generation is badly modeled or the input data is inaccurate. When validating the RDF generation process instead, the assessment report points to the violating parts, which can be used to refine the transformations used in those parts. This allows identification of violations before they even occur and avoids propagation of flawed transformations, leading to many faulty triples [7]. Solving constraint violations on the generation level is thus more efficient and allows for iterative refinement.

3 Related Work

Linked Data is typically generated from (semi-)structured data, encompassing both schema and data transformations [18]. Schema transformations involve (re-)modeling the original data, describing how RDF terms are related, and deciding which vocabularies and ontologies to use [9]. Data transformations are needed to support any change in structure, representation, or content of data [18], e.g., performing string transformations or computations. However, the generation process' support for data transformations is currently uncombinable, restricted, part of a use-case specific system, or coupled [6]. This results in generation processes including preprocessing or custom implementations.

In this section, I first give an overview of Linked Data generation approaches, and how schema and data transformations are integrated. Then, I review existing validation approaches. Focus is given to declarative generation approaches, as existing work has shown that schema transformations can be validated and improved even before the generation execution, when making use of a declarative generation process [7]. The declarative schema transformations specify how the dataset will be formed, thus, the assessment of schema transformations and generated RDF graph are correlated [7]. Instead of directly validating the RDF data graph, a new shape that corresponds to the declarative schema transformations currently needs to be manually (re-)defined for automatic validation.

3.1 Generation Approaches

On high level, the following approaches for Linked Data generation are identified:

Hard-coded Custom tools and scripts were initially used to generate Linked Data from raw data. They incorporate directly in their implementation both the schema and data transformations, as in the case of, e.g., the DBpedia Extraction Framework (DBpedia EF) [2]. Updating the semantic annotations resulted in dedicated software development cycles to adjust the implementations [8].

Case-specific Solutions such as XSLT- or XPath-based approaches were established for generating Linked Data from data originally in XML format, e.g., [14]. These solutions are declarative: rules are detached from the implementation that executes them, thus, the implementation does not need to be updated when the rules are updated. However, only specific data sources are supported, and the range of possible schema or data transformations is limited by the respective language or syntax potential.

Generic The RDF Mapping Language (RML) [8] – based on W3C Recommendation R2RML [4] – is a declarative language, represented in RDF, that supports schema transformations for heterogeneous (semi-)structured data sources. This solution is no longer case-specific and the Linked Data generation process is machine-processable.

Generic solutions however only support schema transformations. Data transformations are either not supported, or the range of possible data transformations is determined by the range of transformations that can be defined when the data is retrieved from the data source (pre-processing), or after the Linked Data is generated (post-processing). More customization is enabled by solutions that allow embedded scripts inside declarative schema transformations, such as using FUN-UL [10], or using custom SPARQL binding functions, such as for SPARQL-Generate [16]. These solutions however depend on their implementation, and do not provide a declarative description.

Generic data transformations Besides the aforementioned solutions that partially integrate schema and data transformations, there are Linked Data generation processes which rely on distinct systems to perform the schema and data transformations. These types of transformations cannot always be distinguished, as data transformations may affect the original schema. Their support for data transformations range from a fixed predefined set of transformations (e.g., Linked Pipes [11]) to an embedded scripting environment (e.g., OpenRefine[1]).

Declarative data transformations Different approaches emerged that define data transformations and other functionalities declaratively, e.g., Hydra [15] for Web services, or VOLT [19] for SPARQL. However, these declarative approaches

[1] http://openrefine.org/

focus on specific implementations and can thus only be used within their context, i.e., Hydra can only be used for Web service implementations, and VOLT only for SPARQL endpoint implementations. No implementation-independent declarative solution is available.

3.2 Data Validation

Two approaches emerged for assessing constraint violations: (i) integrity constraints to detect violations, and (ii) query-based validation detection depending on the RDF graph's shape.

Integrity constraints Entailment regimes that are part of, e.g., RDFS and OWL, are used as integrity constraints to detect violations of an RDF graph [20]. However, these entailment regimes use the Open World Assumption, and assessing constraints assumes a Closed World. As such, these approaches need to redefine existing semantics. Using one standard to express both validation and reasoning is a strong point of this approach. However, this leads to ambiguity: The same formula having different meanings endangers the interoperability within the Semantic Web [1].

Query-based Query-based validation approaches depend on the RDF graph's shape to detect violations. Except for approaches that use SPARQL templates, e.g., RDFUnit [13], constraint description languages are proposed, of which SHACL [12] is a W3C Recommendation. Integration with entailment is either limited or requires a separate inferencing process.

However, the combination of inferencing with shape-based validation is needed [3]. This allows for integration of assessing shapes and ontological constructs. When no inferencing is provided, either too few or too many violations can be returned. On the one hand, a generated resource that is member of disjoint classes might not revealed without inferencing, i.e., too few violations. On the other hand, domain and range violations could be returned, whilst inferred domains and ranges solve the violation, i.e., too many violations.

4 Research Question

Given the related work, we can conclude that the generation process itself cannot easily be validated: current descriptions of the data transformations of the generation processes depend on the use case or are incomplete, and validation approaches would require manual (re-)definition of test cases aimed at the generated dataset, to apply to the generation process. I thus investigate the following two research questions:

Research Question 1 How can we provide a use-case independent declarative Linked Data generation description that includes both transformations?
  Subquestion 1 How can we declaratively define data transformations?
  Subquestion 2 How can we align these schema and data transformations?

Research Question 2 How can we automatically validate the generation description based on the constraint rules that apply to the RDF data graph without needing to manually (re-)define them?

5 Hypotheses

The first research question handles a declaratively described generation process. When the description is machine-processable, the validation can be automated. Existing solutions handle declarative schema transformations, however, when data transformations are supported they are either not declarative, or dependent on the implementation. Hypotheses related to Research Question 1 are:

Hypothesis 1 Declarative data transformations, independent of the implementation, are reusable across use cases and generation processes

Hypothesis 2 Aligned declarative schema and data transformations provide a generic framework for describing Linked Data generation processes.

The second research question handles a validation approach capable of validating the generation description itself, based on the constraint descriptions of the RDF datasets. Existing works do not support data transformations and require manual redefinition of the validation shape of the generation process [7]. Hypotheses related to Research Question 2 are:

Hypothesis 3 A declarative generation process containing both schema and data transformations can be automatically validated based on the RDF data graph constraint rules without needing manual redefinition.

Hypothesis 4 Validation of a declarative generation process is more computationally efficient using custom entailment regimes than using RDFUnit and SHACL. Root causes of both schema and data transformations can be found and refined before any RDF data is generated.

6 Approach

My approach consists of four steps, each related to one hypothesis.

1. Declarative description of functions Describing functions declaratively makes them independent of the implementation. The descriptions of these functions can be reused in other use cases and technologies, such as for describing declarative data transformations during Linked Data generation. As these functions are described in RDF, their descriptions can be validated using existing Linked Data validation approaches, e.g., when describing a birth date, it can be assessed whether the output type of the used function is in fact a date type.

2. Alignment of declarative schema and data transformations I create a declarative generation process by aligning declarative schema and data transformations, making them combinable. Thus, functions can be used as data transformations within the context of the schema transformations of the generation process. However, as the transformations are described separately, there are no interdependencies. The schema transformation descriptions do not necessarily depend on the data transformation descriptions and vice versa. The generation process can now be entirely described declaratively in RDF: the *generation graph.* This generation graph can thus be validated using existing Linked Data validation approaches.

3. Creation of a validation approach handling custom entailment regimes Custom entailment regimes can describe how to rewrite RDF data graph validation rules into RDF generation graph validation rules, without needing to manually (re-)define them. For example, when a validation rule defines that every resource of class schema:Person should have a schema:birthDate defined, it can be inferred that whenever a resource of class schema:Person is generated, also a predicate schema:birthDate and object with a valid date type should be generated. Not only does my approach allow for integration of assessing shapes and ontological constructs [3], it remains independent of the language that describes the constraint rules of RDF data graphs. Existing constraint languages can be reused, and constraint rules of RDF data graphs can be automatically interpreted for the generation process.

4. Evaluating the validation approach to the generation approach I apply this RDF generation graph containing both schema and data transformations, to real-world use cases. Validating this RDF generation graph, using constraint rules of the RDF data graph, can then be computationally compared with validating the generated dataset. Given the example of previous step: instead of validating every resource of class schema:Person of the data graph, only one part of the generation graph needs to be validated, namely, the part handling the generation of resources of class schema:Person. The latter is thus assumed more computationally efficient.

7 Evaluation Plan

For evaluating my approach, separate from comparing to a gold standard, I investigate the generation process of a real-world use case: the DBpedia EF. This is a non-trivial generation process, taking into account both schema and data transformations. Furthermore, the DBpedia dataset is a valuable resource and quality issues have persisted over a long period of time [17]. Improving the generation process and the quality of the DBpedia dataset is beneficial for the Semantic Web community.

To evaluate the first hypothesis, declarative functions are functionally evaluated by the means of real-world use cases to (a) be able to apply the same descriptions to multiple implementations, and (b) describe existing implementations declaratively.

The second hypothesis – alignment of declarative schema and data transformations – is evaluated by providing complete executable declarative descriptions of existing Linked Data generation processes, namely, the DBpedia generation process. Completeness and correctness of the generation description with respect to the original generation process is measured by comparing the generated triples, and performance of the declarative generation process is compared with the DBpedia EF in terms of processing speed.

For the third hypothesis – applying the validation approach to the generation approach – I compare my approach to RDFUnit [13] and SHACL processors [12]. I compare functionalities, namely, with the inclusion of custom validation regimes, whilst performance should be at least comparable for small datasets. Then, I create a golden standard, describing generation processes that are capable of generating the Linked Data as used by the SHACL test suite[2]. By successfully applying the SHACL test suite to this golden standard, I functionally prove the third hypothesis.

For the final hypothesis I evaluate my approach by applying it to the DBpedia EF to describe, use, and validate a generation process. Comparison metrics will be completeness and correctness of the validation result, and processing speed of the validation assessment, comparing the quality assessment of the declarative description of the generation process with the quality assessment of the generated dataset.

8 Preliminary Results

For the first hypothesis, I proposed the Function Ontology (FnO) [5], a way to declaratively describe functions, without restricting to programming language-dependent implementations. The ontology allows for extensions, and is proposed as a possible solution for semantic applications in various domains. As evaluation, FnO has been successfully applied to describe the actions of a Dockerfile [21], and the extracted parsing functions that existed in the original DBpedia EF as a separate, reusable module [6].

For the second hypothesis, I aligned FnO with RML [6]: a use-case independent declarative generation process supporting both schema and data transformations where the extraction, transformation and mapping rules execution are decoupled [17]. As evaluation, I successfully applied it to the DBpedia EF [17], covering 98% completeness with comparable performance. Any part of the Linked Data generation can be reused to generate other datasets, such as the mapping and transformation rules; or the previously extracted parsing functions [17].

Validating my third hypothesis is ongoing. I already showed that rule logic can cover both validation and custom entailment regimes if it is expressive enough [1]. Practical feasibility has been shown by providing a proof-of-concept in N3Logic which supports all RDFUnit constraint types.

9 Reflections

Validating declarative generation processes provides traceability of the root causes of the violations, which allows improving the generation process instead of the generated RDF graph. A more scalable and efficient approach to generate high quality Linked Data is achieved. Preliminary results have shown it is possible to declaratively describe generation processes such as the DBpedia EF including both schema and data transformations.

[2] https://w3c.github.io/data-shapes/data-shapes-test-suite/

Validation of the schema transformations can already be performed by manually redefining constraint rules. By enabling validation approaches to validate the generation process based on constraint rules of the RDF data graph without needing manual adjustments, we can enable a more qualitative generation process without additional effort.

My approach allows declaration and validation of the entire generation process. Each transformation can be validated, and each module is use-case and implementation independent. More granular control is given to the data modeler. Linked Data generation is made more precise, and can be validated better, resulting in Linked Data of higher quality.

References

1. Arndt, D., De Meester, B., Dimou, A., Verborgh, R., Mannens, E.: Using rule based reasoning for RDF validation. In: RuleML+RR (2017)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a Web of Open Data. In: The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings. vol. 4825, pp. 722–735. Busan, Korea (2007)
3. Bosch, T., Acar, E., Nolle, A., Eckert, K.: The role of reasoning for RDF validation. In: Proceedings of the 11th International Conference on Semantic Systems. pp. 33–40 (2015)
4. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. Working group recommendation, World Wide Web Consortium (W3C) (Sep 2012), `http://www.w3.org/TR/r2rml/`
5. 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. vol. 9989, pp. 46–49 (Oct 2016)
6. 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. pp. 33–48 (May 2017)
7. Dimou, A., Kontokostas, D., Freudenberg, M., Verborgh, R., Lehmann, J., Mannens, E., Hellmann, S., Van de Walle, R.: Assessing and refining mappings to RDF to improve dataset quality. In: The Semantic Web – ISWC 2015. vol. 9367, pp. 133–149. Bethlehem, PA, USA (Oct 2015)
8. 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: Proceedings of the 7th Workshop on Linked Data on the Web. vol. 1184 (2014)
9. Hyland, B., Atemezing, G., Villazón-Terrazas, B.: Best Practices for Publishing Linked Data. WG Note, W3C (Jan 2014), `https://www.w3.org/TR/ld-bp/`
10. Junior, A.C., Debruyne, C., Brennan, R., O'Sullivan, D.: An evaluation of uplift mapping languages. International Journal of Web Information Systems 13(4), 405–424 (2017)
11. Klímek, J., Škoda, P., Nečaskỳ, M.: LinkedPipes ETL: Evolved Linked Data preparation. In: The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete,

Greece, May 29 – June 2, 2016, Revised Selected Papers. vol. 9989 LNCS, pp. 95–100 (2016)

12. Knublauch, H., Kontokostas, D.: Shapes Constraint Language (SHACL). W3C recommendation, W3C (Jul 2017), https://www.w3.org/TR/shacl/
13. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A.: Test-driven evaluation of linked data quality. In: Proceedings of the 23$^{\text{rd}}$ international conference on World Wide Web. pp. 747–757 (Mar 2014)
14. Lange, C.: Krextor -An Extensible Framework for Contributing Content Math to the Web of Data. In: Intelligent Computer Mathematics: 18$^{\text{th}}$ Symposium, Calculemus 2011, and 10$^{\text{th}}$ International Conference, MKM 2011, Bertinoro, Italy, July 18-23, 2011. Proceedings. pp. 304–306 (2011)
15. Lanthaler, M.: Hydra Core Vocabulary. Unofficial Draft, Google (Mar 2018), http://www.hydra-cg.com/spec/latest/core/
16. Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL extension for generating RDF from heterogeneous formats. In: The Semantic Web 14th International Conference, ESWC 2017, Portoro, Slovenia, May 28 June 1, 2017, Proceedings. pp. 35–50. Portoroz, Slovenia (May 2017)
17. Maroy, W., Dimou, A., Kontokostas, D., De Meester, B., Verborgh, R., Lehmann, J., Mannens, E., Hellmann, S.: Sustainable linked data generation: The case of DBpedia. In: Proceedings of the 16$^{\text{th}}$ International Semantic Web Conference: In-Use Track. Vienna, Austria (Oct 2017)
18. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin 23(4), 3–13 (2000)
19. 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: Proceedings of the 13$^{\text{th}}$ International Conference on The Semantic Web. Latest Advances and New Domains. pp. 523–538 (2016)
20. Tao, J., Sirin, E., Bao, J., McGuinness, D.L.: Integrity constraints in owl. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010). Atlanta, Georgia, USA (Jul 2010)
21. Tommasini, R., De Meester, B., Heyvaert, P., Verborgh, R., Mannens, E., Della Valle, E.: Representing dockerfiles in RDF. In: ISWC 2017 Posters & Demonstrations and Industry Tracks. vol. 1963. Vienna, Austria (Oct 2017)
22. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for linked data: A survey. Semantic Web Journal 7(1), 63–93 (Mar 2015)