A Method and a Library for Visual Data Schemas

Lelde Lāce[0000-0001-7650-2355], Aiga Romāne[0009-0003-3609-1485], Jana Fedotova[0009-0005-5419-7798], Mikus Grasmanis[0000-0002-0668-0970] and Kārlis Čerāns[1][0000-0002-0154-5294]

Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia
[1] karlis.cerans@lumii.lv

Abstract. We describe a method for computing and visualizing data schemas for existing Linked data endpoints, combined with creation of visual query environments over these endpoints. We evaluate the method on a set of externally available small-to-medium size data sets and present the obtained visualization results in a form of interactive library.

Keywords: Linked data, Knowledge graph schema, Visual schema diagram.

1 Introduction

A data schema of a knowledge graph or a Linked data endpoint can be seen as a high-level presentation of the data set, involving the used class and property vocabularies and their connections. Visual presentation of a data schema can be expected to help a user to comprehend the graph/endpoint structure and, therefore, use more efficiently the data contained therein.

There are tools allowing visualization of existing Linked data endpoint schemas, such as LD-VOWL [1] and LODSight [2], allowing to obtain on-the-fly visualizations of the class-to-property relations present in the suitably sized data sets. We propose a schema visualization pipeline that separates the schema extraction, and the schema visualization steps since that allow working with schemas of larger size and heterogeneity, as well as allows for user interaction with the schema visualization process.

The RDF data shape languages SHACL [3] and ShEx [4] provide also rich means for knowledge graph schema description. The concepts used in the data schema can also be described by means of OWL ontologies [5], with a wealth of tools available for visual ontology structure presentation (cf. [7,8,9,10,11]). Our schemas differ both from the *a priori* built OWL ontology and ShEx/SHACL shape presentations in that it concerns the actual data structure, as it is present in the data endpoint, and it involves a focus on important nuances (as the relevance of a class as a property source or target) that are not present or are present partially in the existing visualization tools.

The idea of extracting the schema from the data set automatically has been explored already in [12], where the schema presentation in the form of a UML style diagram is considered. More recently, [13] demonstrates the possibility of extracting validating SHACL shapes from a given SPARQL endpoint (the shapes can be further

visualized in UML form). The uniqueness of our approach is in selecting means for practical diagram structuring (as e.g., subclassing, and characteristic property placement), and applying them in the context of numerous already existing data sets.

The described visualization method was presented earlier in [14], where the used data schema structure has been outlined. The work presented here involves new diagram structuring options (e.g., class contraction and property link splitting), and it analyzes the method applicability on more than 45 externally available data sets.

2 Visual Schema Diagram Principles

We propose creation of visual diagrams of the data schemas for existing data endpoints, based on the following principles and options:

a) Represent the classes as the schema graph nodes, and the properties as edges connecting their source and target class nodes, or as node attributes.
b) Ascribe the properties to their most characteristic places in the class hierarchy (e.g., to avoid the property ascription to both a subclass and a superclass).
c) Introduce anonymous super-classes to reduce edge and attribute repetitions in the diagram (no-loss and possibly lossy options are available).
d) Provide contraction of class nodes with equal or similar attribute and edge sets (no-loss and possibly lossy options available here, as well).
e) Possibility to split overloaded property links (by recording the other end information at both the edge source and target vertices).

The strength of (c), (d) and (e) parameters can be tuned within the user interface, where a no-loss mode can be applied for smaller-scale visualizations and a merging possibility can be increased gradually to obtain legible and informative diagrams for larger schemas at the cost of moving the attributes and link ends to nodes corresponding to a higher abstraction level over the data. The split of overloaded links (multiple links with the same property in the diagram) can be tuned independently.

Figure 1 contains a simple example of a schema for Nobel Prizes endpoint[1] with automated no-loss anonymous superclass *dbo:City or dbo:Country*. Note the single appearance of *dct:hasPart* and *dct:isPartOf* links in the diagram, in their "most characteristic" place in the class hierarchy, whereas a naïve way of connecting classes by properties would have extra edges for them starting or ending at *dbo:Award*, as well.

3 Schema Extraction and Visualization Pipeline

The SPARQL endpoint schema visualization process is available as a part of a larger schema extraction and visualization process that also creates an environment for visual creation of queries in the *ViziQuer* notation [6] over the endpoint. The pipeline for the process is the following:

[1] Data from the original https://data.nobelprize.org/sparql endpoint have been copied locally to http://85.254.199.72:8890/sparql, Named graph: http://nobelprizes.local for the analysis.

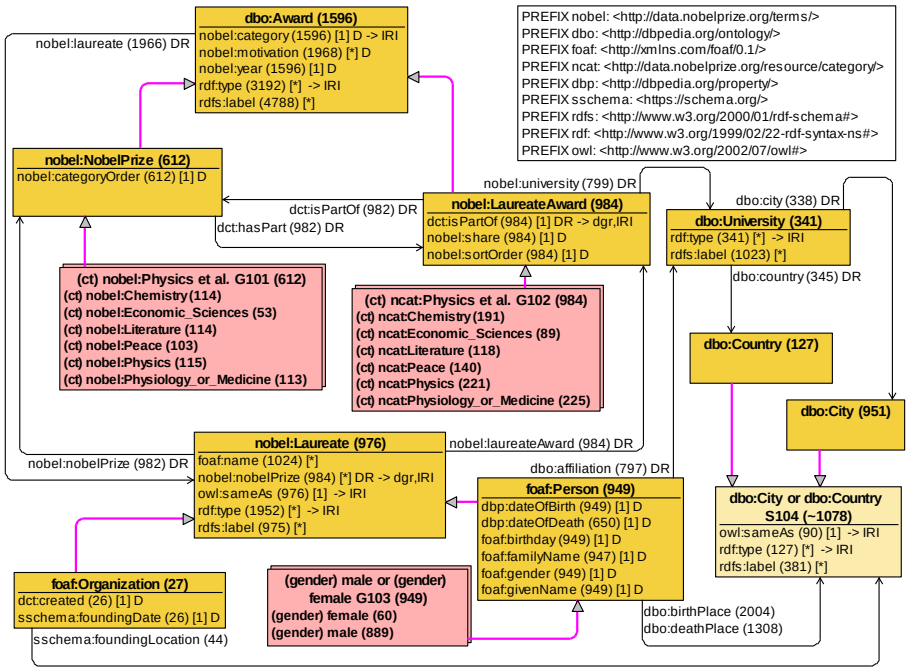

Fig. 1. Nobel Prizes Data Schema (with some classifiers)

1. Extract the schema of the SPARQL endpoint, using the *OBIS Schema Extractor* tool[2] (a schema file in a .JSON format is obtained, cf. [15] for the principles of the schema extraction description).
2. Store the schema in the database of *Data Shape server* (DSS)[3], ready to be served to *ViziQuer* visual query tool and to other potential clients (namespace prefix fine-tuning in the *ns* table of the data schema can be beneficial).
3. Create a visual query project with the respective schema in the *ViziQuer*[4] tool, enter the project and invoke the schema diagram creation, where the diagram creation parameters can be tuned, if desired. Export the schema diagram for use in the diagram visualization tool or visualize it directly within the *ViziQuer* environment (experimental functionality).
4. Open the created schema diagram in the diagram visualization tool (*DSS Schema explorer*[5]); one can manually fine-tune the positioning and the contents of the automatically created diagram. *DSS Schema Explorer* allows diagram export in .SVG format, as well.

Instructions for performing these steps are in the respective tool repositories.

[2] https://github.com/LUMII-Syslab/OBIS-SchemaExtractor
[3] https://github.com/LUMII-Syslab/data-shape-server
[4] https://github.com/LUMII-Syslab/viziquer, https://viziquer.lumii.lv/
[5] https://github.com/LUMII-Syslab/dss-schema-explorer, works on *Microsoft Windows* only.

4 Schema Library

We describe the schema extraction and visualization process and its result on a set of 58 small-to-medium schema size data sets (50 classes or less; 44 successful visualizations) from Linked Data repository, ISWC 2023 Proceedings and Inria Catalogue[6], as well as provide visualizations of select schemas including Nobel Prizes, Academy Sampo, War Sampo[7] and Inria Catalogue itself.

The experiment setup, its process, and results, the DSS database dump with the data schemas and obtained schema visualizations in .SVG format are made available on GitHub[8], where also a link to a running ViziQuer server instance is provided[9]. We also provide releases of DSS[10], ViziQuer[11] and Schema Explorer[12] tools for producing the visual query and schema environments from the provided DSS database dump.

The results of the experiment show that the proposed schema visualization pipeline can handle the visualization of schemas of the considered small-to-medium size and that it has a potential of handling larger schema visualizations, as well. All 14 missing data schemas were due to the problems in the schema extraction step (9 interface issues and 5 process complexity (e.g., timeout) issues). We note that only two schemas from the bulk schema data set required a considerable (above 15 minutes) time for manual schema positioning to obtain a reasonable schema presentation. The presentation of two schemas involved resorting to displayable names based on entity labels (currently to be done by a SQL script on the DSS schema level).

5 Discussion and Conclusions

The performed experiments show that the provided method offers a promising approach for obtaining an environment where both a visual schema presentation and visual data queries are available.

Although the visual query environments for large and heterogenous data endpoints as *DBPedia* and *Wikidata* are available, their schema visualizations are currently not within the scope of the proposed method. We expect that reasonable visualization solutions for schemas with up to 300 – 500 essential data classes can be provided (using the schema fragment/slice visualization options, where appropriate, and employing more powerful abstraction mechanisms), however, this would require further experiments. Important future work avenues would also be the schema creation pipeline simplification and providing the visual query option right from the visual data schema.

---

[6] http://prod-dekalog.inria.fr/sparql
[7] https://www.ldf.fi/datasets.html
[8] https://github.com/LUMII-Syslab/viziquer/tree/development/doc/demo/schemas24a
[9] Currently https://schemas24a.viziquer.app/
[10] https://doi.org/10.5281/zenodo.11069027
[11] https://doi.org/10.5281/zenodo.11072804
[12] https://doi.org/10.5281/zenodo.11072854

Acknowledgments. This work has been partially supported by a Latvian Science Council Grant lzp-2021/1-0389 “Visual Queries in Distributed Knowledge Graphs”.

References

1. Weise, M., Lohmann, S., & Haag, F. (2016). Ld-vowl: Extracting and visualizing schema information for linked data. In Voila!2016 (pp. 120-127).
2. Dudáš, M., Svátek, V., Mynarz, J.: Dataset summary visualization with LODSight. In: The 12th Extented Semantic Web Conference (ESWC2015). http://lod2-dev.vse.cz/lodsight/lodsight-eswc2015-demopaper.pdf
3. Shapes Constraint Language (SHACL), https://www.w3.org/TR/shacl/
4. ShEx - Shape Expressions, http://shex.io/
5. Web Ontology Language (OWL), https://www.w3.org/OWL/
6. Čerāns, K., Šostaks, A., Bojārs, U., Ovčiņņikova, J., Lāce, L., Grasmanis, M., Romāne, A., Sproģis, A., Bārzdiņš, J. (2018). ViziQuer: A Web-Based Tool for Visual Diagrammatic Queries Over RDF Data, in Gangemi, A., et al. (ed.), *Proceedings of The Semantic Web: ESWC 2018 Satellite Events.* ESWC 2018. Lecture Notes in Computer Science, Vol. 11155. Springer, Cham, pp. 158–163. https://doi.org/10.1007/978-3-319-98192-5_30
7. Lohmann, S., Negru, S., Haag F., Ertl, T.: Visualizing Ontologies with VOWL. In: Semantic Web 7(4), 399-419, (2016)
8. Bārzdiņš, J., Čerāns, K., Liepiņš, R., Sproģis, A.: UML Style Graphical Notation and Editor for OWL 2. In: Proc. of BIR’2010, LNBIP, Springer 2010, vol. 64, pp. 102-113, (2010)
9. Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M.: The simple, web-based tool for visualization and sharing of semantic data and ontologies. In: ISWC P&D 2015, CEUR, vol.1486, http://ceur-ws.org/Vol-1486/paper_77.pdf, (2015).
10. Dudáš, M., Lohmann, S., Svátek, V., Pavlov, D.: Ontology visualization methods and tools: a survey of the state of the art. In: The Knowledge Engineering Review, 33, (2018)
11. E. Labra Gayo, D. Fernández Álvarez and H. García González, RDFShape: An RDF Playground Based on Shapes, ISWC 2018 Posters & Demonstrations (2018).
12. Li H., Zhang X. Visualizing RDF data profile with UML diagram (2013) *Springer Proceedings in Complexity*, pp. 273 – 285. DOI: 10.1007/978-1-4614-6880-6_24.
13. Rabbani K., Lissandrini M., Hose K. Extraction of Validating Shapes from very large Knowledge Graphs (2023) *Proceedings of the VLDB Endowment*, 16 (5), pp. 1023 – 1032 DOI: 10.14778/3579075.3579078
14. Lāce L, Romāne A, Grasmanis M, Čerāns K. A Method of Visual Presentation of Data Schemas. In: *VOILA! 2023*. Vol 3508. CEUR Workshop Proceedings; 2023:57-62. https://ceur-ws.org/Vol-3508/paper6.pdf
15. Čerāns, K., Ovčiņņikova, J., Bojārs, U., Grasmanis, M., Lāce, L., Romāne, A. (2021b). Schema-Backed Visual Queries over Europeana and Other Linked Data Resources, *in* Verborgh, R., et al (ed.), *Proceedings of The Semantic Web: ESWC 2021 Satellite Events.* ESWC 2021. Lecture Notes in Computer Science, Vol. 12739. Springer, Cham, pp. 82–87. https://doi.org/10.1007/978-3-030-80418-3_15