Visual Schemas and Queries for Knowledge Graph Exploration

Kārlis Čerāns[1,*], Jūlija Ovčiņņikova[1,*], Mikus Grasmanis[1] and Lelde Lāce[1]

[1]Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulvāris 29, Rīga, LV-1459, Latvia

Abstract

We present a method for integrating visual schema diagrams and visual queries within a single interactive environment for knowledge graph exploration. This approach addresses the gap between schema visualization tools and visual query interfaces, thereby enabling users, including domain experts, to seamlessly perform schema-based queries without the need for switching between different tools. We implement the method in the ViziQuer tool, which provides also means for full visual queries alongside a schema-based querying. The option for a transition from schema-based queries to the full visual query environment is provided as well.

Keywords

Knowledge graphs, Visual schema diagrams, Visual queries

1. Introduction

The visual presentation of data schemas is commonly believed to help to engage the end users in the work with the data. There are tools for visually presenting schemas of knowledge graphs (KG), involving visualizers of KG schema descriptions created in SHACL/ShEx (cf. [1, 2]) or OWL (cf. [3, 4]). These tools can be used if the schema of the data set has been made available in the respective notation. There are tools for visualizing the actual data schema, as well, working either on-the-fly, as [5, 6], or via the intermediate schema storage (cf. [7], for schema visualization inside *ViziQuer*[8, 9]). SHACL data shape visualization tools can also be used in combination with data shape retrieval from the data set (cf. [10]) to obtain a pipeline of visualization of the actual data set structure.

The schema visualization provides a high-level perspective of the data set from a schema-centric viewpoint, with the possibility to visualize either the full schema or just the relevant schema fragments (to obtain a schema presentation of legible size). Beyond naive methods for the computation of data schema fragments, some more advanced approaches are described in [11]. Collections of schema visualizations for Linked Open Data datasets are described in [12, 7].

A limitation of the schema visual presentation approach is that it allows getting insight just into the schema aspect of the data, and the information about the actual data behind the schema classes and properties is very limited (e.g., the instance or triple counts can be provided on the schema level). Still, it might be important for the data set users to access this information as well.

Writing textual SPARQL queries for the data access is generally believed to be too hard for a typical data set user. So, various approaches for query creation assistance have been developed, including facet-based environments such as *PepeSearch* [13] and *Sampo UI* [14] (cf. also [15]), visual query tools such as *OptiqueVQS* [16], *RDF Explorer* [17] and *ViziQuer*[8], as well as recent natural language-based approaches involving neural network and/or LLM usage (cf., e.g. [18, 19]).

Each of the considered approaches, however, involves setting up and using an external environment besides the schema diagram, to create the data queries.

Semantics 2025: Posters and Demos, September 03–05, 2025, Vienna, Austria

*Corresponding author.

karlis.cerans@lumii.lv (K. Čerāns); julija.ovcinnikova@lumii.lv (J. Ovčiņņikova); mikus.grasmanis@lumii.lv (M. Grasmanis); lelde.lace@lumii.lv (L. Lāce)

0000-0002-0154-5294 (K. Čerāns); 0000-0002-5884-763X (J. Ovčiņņikova); 0000-0002-0668-0970 (M. Grasmanis); 0000-0001-7650-2355 (L. Lāce)

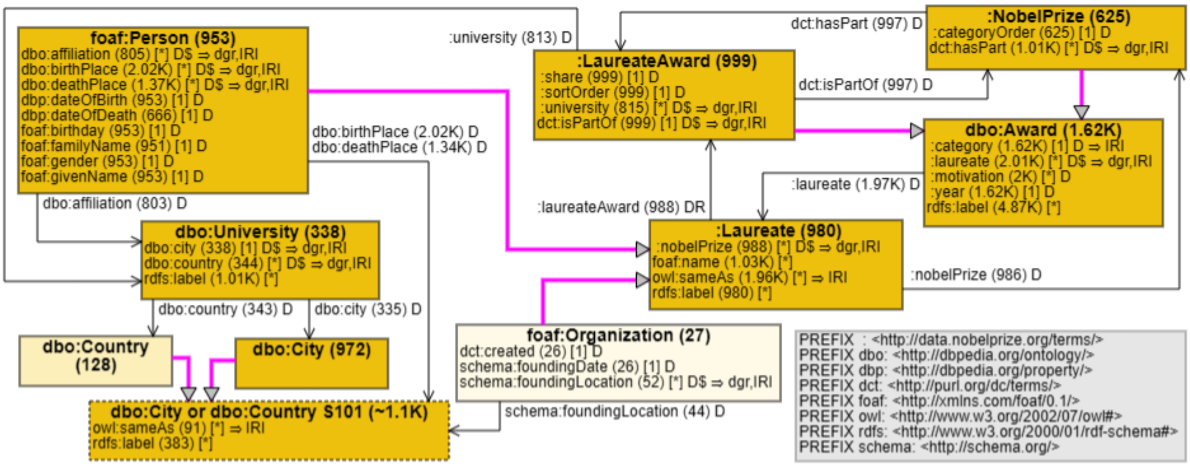

Figure 1: Data schema diagram of the Nobel Prize dataset [7]

In this paper, we demonstrate the options for the creation of simple data queries *directly from the visual schema diagram environment* (expected to be easy for the end user) involving just a few clicks. There is a possibility of simultaneous gradual growing of the expressive power of the queries and the user interface complexity.

We demonstrate the options of diagram-based queries within the context of open source *ViziQuer* tool[1] that provides means for both visual schema diagrams and visual schema-backed query creation. For a professional or advanced casual user, there are means for transiting from diagram-based queries into the full visual query environment [8], where more advanced queries (coming close to the full expressive power of SPARQL) can be created, as well.

ViziQuer has both the playground and local setup options described on its website[2]. The pipeline for local running *ViziQuer* over the user's data uses tools for schema extraction[3] and storing[4], as well as a (docker) container system[5] for local running of the visual tool itself.

2. Schema Diagrams

Intuitively, a *KG schema* (cf. [7]) is an abstract structure involving the vocabularies of KG classes and properties, as well as the class and property connections (including the subclass relation and the class-to-property and property-to-class connections, as well as class-property-class connections, where possible, together with their appearance statistics). The schema can also involve additional information such as cardinalities, property domain and range classes, and connections among properties[6].

We note that most of the core schema aspects can be serialized into an RDF data shape language such as SHACL, however, within this demonstration, the abstract schema concept is sufficient.

Following [7], the *schema diagram* is a visual presentation of the schema where the data classes are generally depicted as graph nodes and the properties are shown in attribute and/or link form, as appropriate. This diagram can represent either the complete schema or its targeted fragment, enabling also a clear visual presentation of schemas that would be incomprehensible if displayed in their entirety (cf. [11]). Figure 1 shows an illustration of a simple schema diagram for the Nobel Prize data set[7].

The *ViziQuer* implementation [7] of the schema diagram presentation also takes into account the class importance as a source or target class for a property (not to ascribe the property to both a subclass

[1]Code on GitHub: https://github.com/LUMII-Syslab/viziquer
[2]ViziQuer website: https://viziquer.lumii.lv
[3]OBIS Schema Extractor: https://github.com/LUMII-Syslab/OBIS-SchemaExtractor
[4]https://github.com/LUMII-Syslab/data-shape-server/tree/main/import-generic
[5]ViziQuer Tools: https://github.com/LUMII-Syslab/viziquer-tools
[6]one property following the other, two properties sharing the same subject, or the same object
[7]Nobel Prize data set: https://data.nobelprize.org/sparql

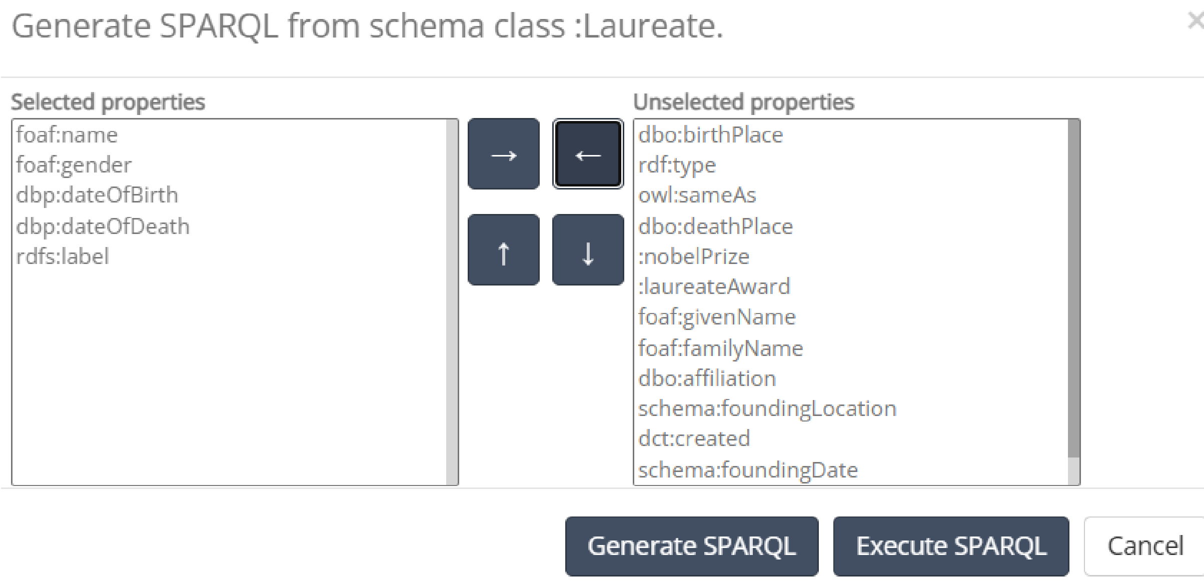

Figure 2: Property selection window

Only 50 rows shown. Total number of rows: 1024.

#	Laureate	name	gender	dateOfBirth	dateOfDeath	label
1	http://data.nobelprize.org/resource/laureate/18	Heike Kamerlingh Onnes	male	1853-09-21	1926-02-21	Heike Kamerlingh Onnes
2	http://data.nobelprize.org/resource/laureate/1	Wilhelm Conrad Röntgen	male	1845-03-27	1923-02-10	Wilhelm Conrad Röntgen
3	http://data.nobelprize.org/resource/laureate/10	J.J. Thomson	male	1856-12-18	1940-08-30	Joseph John Thomson
4	http://data.nobelprize.org/resource/laureate/100	Martin Ryle	male	1918-09-27	1984-10-14	Sir Martin Ryle
5	http://data.nobelprize.org/resource/laureate/101	Antony Hewish	male	1924-05-11	2021-09-13	Antony Hewish
6	http://data.nobelprize.org/resource/laureate/102	Aage N. Bohr	male	1922-06-19	2009-09-08	Aage Niels Bohr
7	http://data.nobelprize.org/resource/laureate/103	Ben R. Mottelson	male	1926-07-09	2022-05-13	Ben Roy Mottelson

Figure 3: The example result of custom query from the schema

and a superclass[8], thus reducing the schema overload with property links). A compact presentation of the schema in the diagram is obtained by grouping together similar classes into single graph nodes, as well as grouping together the links connecting the same nodes [7].

3. Queries from Schema Diagrams

Since a node in a data schema diagram represents a data class (or a set of classes in the case of compacting), a SPARQL query retrieving the URIs of the class instances, as well as a certain amount of presumably typical data properties, can be naturally generated from the context of the diagram node ('Generate SPARQL'), as well as executed ('Show Data'), presenting the results to the user.

We propose by default to show, along with the class instance IRI, up to 7 most populated data properties for the entities of the class. To change the properties that are included in the result set for a class, the 'Custom Data' command can be activated to open a form where the properties to be shown in the query result can be adjusted. Figure 2 provides a screenshot of a sample window of the selection of properties to be displayed in the context of the Nobel Prize data set (Fig 1) Laureate class, and Figure 3 - the corresponding data result presentation.

[8]For instance, the property dct:isPartOf in Fig. 1 is ascribed to :LaureateAward only, and not to :Award since all dct:isPartOf subjects fall into the subclass :LaureateAward, see [7] for details

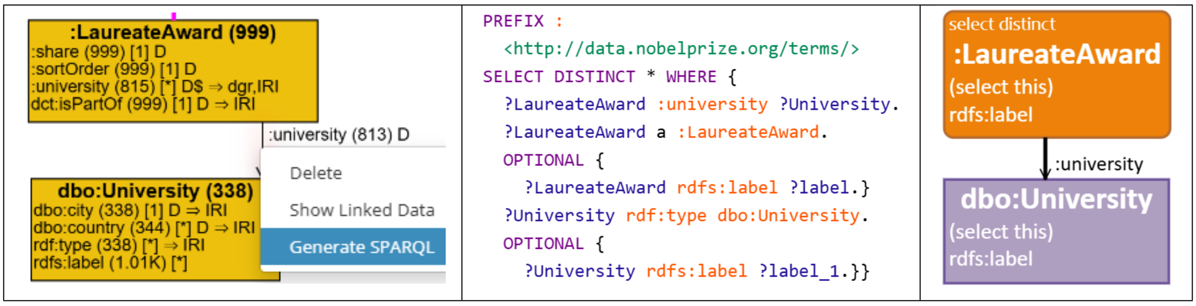

Figure 4: A schema-based query example: select a link; generated SPARQL query; the query in the full visual query environment

We also offer a possibility to create and execute queries based on links between two nodes and from larger selected schema fragments (both the nodes and their connecting links need to be selected).

Further modifications to the generated SPARQL queries are available in a dedicated SPARQL query window, which offers schema-based autocompletion for relevant entity names[9].

Since the visual query creation options over the schema diagram backbone are limited, the generated SPARQL query can be transferred to the dedicated visual query diagram, available in *ViziQuer* (cf. [8]), where it can be further edited and enriched with data filters and more advanced constructs such as aggregations and subqueries, as well as custom data field expressions.

Figure 4 provides an illustration of a link-based query creation in the schema diagram, the corresponding SPARQL expression, and the visual query in the query diagram.

We provide an online demo[10] that walks through the steps of the entire process of creating and using queries from schema diagrams in *ViziQuer*.

4. Conclusions

We have demonstrated the feasibility of constructing queries over KG data directly from the visual presentation of the KG schema, implemented in *ViziQuer* tool. We expect this to help the users that can *view the schema diagrams* to seamlessly access also the *underlying KG data*.

We note that our approach of providing data access from schema diagrams generalizes easily to other eventual diagrammatic environments, where the individual nodes can be selected and have connection anchors in the data (e.g., a node has the URI of a data class); these can also be visual SHACL shape diagrams or even visual OWL ontology presentations from which the data connections can be made.

Further work on schema diagram queries shall involve enrichment of options and user experience of the schema diagram-based query creation, automation of migrating a query to the full visual query environment and performing an evaluation of the approach.

Acknowledgments

This work has been partially supported by a Latvian Science Council Grant lzp-2024/1-0665 "What is in Your Knowledge Graph?".

Declaration on Generative AI

The authors used ChatGPT 4 for Citation management. After using the tool the authors reviewed and edited the content as needed and take full responsibility for the publication content.

[9] The domain experts are expected to be able to work with the automatic data queries. SPARQL editing is expected to be appreciated by data analysts, although simple edits can be mastered by a wider user base as well

[10] Demo: https://github.com/LUMII-Syslab/viziquer/tree/development/doc/demo/schema-diagrams

References

[1] J. E. L. Gayo, D. Fernández-Álvarez, H. García-González, RDFShape: An RDF playground based on shapes, in: CEUR WS, volume 2180, 2018. URL: https://ceur-ws.org/Vol-2180/paper-35.pdf.

[2] Sparna, Shacl play!, 2025. URL: https://shacl-play.sparna.fr/play/, accessed: 2025-07-04.

[3] S. Lohmann, S. Negru, F. Haag, T. Ertl, Visualizing ontologies with VOWL, Semantic Web 7 (2016) 399–419.

[4] K. Čerāns, J. Ovčiņņikova, R. Liepiņš, M. Grasmanis, Extensible visualizations of ontologies in OWLGrEd, in: The Semantic Web: ESWC 2019 Satellite Events, volume 11762 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 191–196. doi:10.1007/978-3-030-32327-1_38.

[5] M. Weise, S. Lohmann, F. Haag, Ld-VOWL: Extracting and visualizing schema information for linked data, in: Voila!2016, 2016, pp. 120–127. URL: https://ceur-ws.org/Vol-1704/paper11.pdf.

[6] M. Dudáš, V. Svátek, J. Mynarz, Dataset summary visualization with LODSight, in: The 12th Extended Semantic Web Conference (ESWC2015), 2015. URL: http://lod2-dev.vse.cz/lodsight/lodsight-eswc2015-demopaper.pdf.

[7] L. Lāce, A. Romāne-Ritmane, M. Grasmanis, A. Sproģis, J. Ovčiņņikova, U. Bojārs, K. Čerāns, Visual presentation and summarization of Linked Data schemas, in: Proc. of KGSWC 2024, volume 15459 of *Lecture Notes in Computer Science*, 2024, pp. 290–305. doi:10.1007/978-3-031-81221-7_20.

[8] K. Čerāns, A. Šostaks, U. Bojārs, J. Ovčiņņikova, L. Lāce, M. Grasmanis, A. Romāne, A. Sproģis, J. Bārzdiņš, ViziQuer: A web-based tool for visual diagrammatic queries over RDF data, in: ESWC 2018 Satellite Events, volume 11155 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 158–163. doi:10.1007/978-3-319-98192-5_30.

[9] K. Čerāns, U. Bojārs, J. Ovčiņņikova, L. Lāce, M. Grasmanis, A. Sprogis, Towards visual federated sparql queries, in: SEMANTiCS-PDWT 2024, CEUR Workshop Proceedings, volume 3759, 2024. URL: https://ceur-ws.org/Vol-3759/paper24.pdf.

[10] K. Rabbani, M. Lissandrini, K. Hose, Extraction of validating shapes from very large knowledge graphs, Proc. of the Very Large Databases 16 (2023) 1023–1032.

[11] S. Siliņa, L. Lāce, M. Grasmanis, K. Čerāns, Schema fragment visualization to view knowledge graph entities in context, in: Poster and Demo Proceedings of ESWC 2025, 2025. To appear.

[12] L. Lāce, A. Romāne, J. Fedotova, M. Grasmanis, K. Čerāns, A method and a library for visual data schemas, in: ESWC 2024 Satellite Events, volume 15344 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 254–258. doi:10.1007/978-3-031-78952-6_37.

[13] G. Vega-Gorgojo, M. Giese, S. Heggestoyl, A. Soylu, A. Waaler, PepeSearch: Semantic data for the masses, PLoS ONE 11 (2016) e0151573. doi:10.1371/journal.pone.0151573.

[14] H. Rantala, A. Ahola, E. Ikkala, E. Hyvönen, How to create easily a data analytic semantic portal on top of a SPARQL endpoint: introducing the configurable Sampo-UI framework, in: VOILA! 2023 Visualization and Interaction for Ontologies, Linked Data and Knowledge Graphs 2023, volume 3508, CEUR Workshop Proceedinbgs, 2023. URL: https://ceur-ws.org/Vol-3508/paper3.pdf.

[15] N. K. Grislis, K. Čerāns, M. Grasmanis, H. Rantala, E. Hyvönen, How to add a user interface on top of an external SPARQL endpoint: Case Nobel Prize Sampo, in: Semantics 2025: Posters and Demos, CEUR Workshop Proceedings, Vienna, Austria, 2025. This Volume.

[16] A. Soylu, E. Kharlamov, D. Zheleznyakov, E. J. Ruiz, M. Giese, M. Skjaeveland, D. Hovland, R. Schlatte, S. Brandt, H. Lie, I. Horrocks, OptiqueVQS: a visual query system over ontologies for industry, Semantic Web 9 (2018) 627–660.

[17] H. Vargas, C. Buil-Aranda, A. Hogan, C. López, RDF Explorer: A visual SPARQL query builder, in: The Semantic Web – ISWC 2019, volume 11778 of *Lecture Notes in Computer Science*, Springer, 2019. doi:10.1007/978-3-030-30793-6_37.

[18] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying Large Language Models and Knowledge Graphs: A roadmap, arXiv preprint arXiv:2306.08302 (2023). URL: https://doi.org/10.48550/arXiv.2306.08302. doi:10.48550/arXiv.2306.08302.

[19] Anthropic, Introducing the Model Context Protocol, 2024. URL: https://www.anthropic.com/news/model-context-protocol.