

# Towards Visual Federated SPARQL Queries

Kārlis Čerāns, Uldis Bojārs, Jūlija Ovčinnikova, Lelde Lāce, Mikus Grasmanis, and Artūrs Sproģis

*Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia*

## Abstract

We demonstrate a method for visual creation of schema-backed federated queries that features schema summary visualizations and context-aware auto-completion of queries, based on schemas of multiple data sets. The method is implemented in the context of the *ViziQuer* query tool, based on the collection of multiple stored data schemas, including schemas for *DBpedia* and *Wikidata*. The environment for schema visualization and for the creation of visual federated queries is available as an online playground and as an open-source software for local installation.

## Keywords

Data schema, Schema visualization, SPARQL, Federated visual queries

## 1. Introduction

Writing a SPARQL query requires knowledge of both the SPARQL query language and the schema of the data to be queried. Writing a SPARQL query over a federation of endpoints requires knowledge of the schemas of all endpoints in the federation, which can be even more difficult.

We address this difficulty by providing a visual-centered system for (i) creating a visual presentation of SPARQL endpoint data schemas in the style of UML class diagrams and (ii) providing multi-endpoint schema support for creating visual federated SPARQL queries.

The use of the visual paradigm is generally acknowledged to ease the comprehension of the data model or the data set structure since logically connected model entities can be shown together. There is a multitude of visual tools for data schema/structure presentation using either OWL ontology notation (cf. [1], [2]) or RDF data shapes expressed in SHACL or ShEx (cf. [3]).

The visual support for data queries allows to invoke the user's visual perception capabilities in the query building process and can be shown to be helpful in query creation at least for a range of users and data queries (cf. [4]).

The main novel point of this paper is to demonstrate a single visual environment that supports the visualization of multiple data endpoint schemas and provides a context-aware visual query auto-completion over data set federations.

Both the data structure visualization and context-aware query auto-completion facilities rely on the data schemas that need to be made available within the visual tool. Conceptually, the data schema describes the data classes and properties, as well as their connections, preferably with the size/frequency characteristics. Ontological domain/range information, cardinalities and data types can be included, as well. We implement schema extraction directly from a SPARQL endpoint (cf. [5]) to obtain the schema that exactly matches the data to be queried. The creation of schemas from data dumps or importing them from (enriched) SHACL shapes can be envisaged, as well.

The obtained schemas or their fragments can then be visualized in detailed and/or summary form. These schemas can be used to support the construction of visual query diagrams, both by visually presenting options for query expansion (available in the context of the visual query built so far) and by auto-completing textual query fragments.

We develop the federated SPARQL query eco-system in the context of the *ViziQuer* notation and tool environment (cf. [6]) that provides the capability for the visual creation of rich data

---

*Proceedings Acronym: Proceedings Name, Month XX-XX, YYYY, City, Country*

✉ karlis.cerans@lumii.lv (K. Čerāns); uldis.bojars@lumii.lv (U. Bojārs)

id 0000-0002-0154-5294 (K. Čerāns); 0000-0001-7444-565X (U. Bojārs)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

queries. We substantially simplify the data schema visualization process, introduced in [7], by introducing novel web-based in-tool data schema visualizations.

To support the development and visualization of federated queries, the capability for adding the GRAPH/SERVICE descriptions was integrated into the *ViziQuer* syntax. The query auto-completion structures were enriched to enable the use of multiple data schemas, and the SPARQL query generation and SPARQL query visualization algorithms of [8] were expanded accordingly.

We illustrate the introduced concepts and the user interaction process in visual query creation on a simple example of the *StarWars* data set (cf. [9]) connected to *Wikidata*.

The resources supporting the paper are available online<sup>2</sup>. The project can also be accessed from the *ViziQuer* playground at <https://viziquer.app> (choose the *StarWars* example project).

## 2. Related Work

The visualization of the data structure and the support for visual query creation are based on the concept of data set schema (cf. [6]) that can be extracted from a SPARQL endpoint. Since the data set schema is based on the class and property inter-relations, most of its aspects can be encoded in an RDF data shape language such as SHACL or ShEx. In this work, we use an abstract data set schema concept whose information is easily enriched, for instance, with the information on the frequency of entities and their relations, or other custom attributes. The functionality for import/export between the schemas and ShEx/SHACL data shapes can be created, although obtaining the additional information (e.g., the entity frequencies) from the data set (or from some custom attributes, as made available in [10]) would be highly desired.

There is a multitude of tools for diagrammatic presentation of OWL ontologies (cf. [1,2,11] or [12] for a survey). Some of the known UML style presentations of ShEx / SHACL data shapes are [3] and [13]. The summarization of data schemas for visual presentation in a more compact form relates to the landmark work on RDF data summarization [14], where legible query diagrams can still be created only for rather moderate-size data schemas. The visualization of data schemas used to support visual queries has been provided for the *ViziQuer* tool (cf. [7]); however, previously, the visualization required a platform-specific (MS Windows) external component, restricting and complicating the data visualization pipeline. In this paper, we demonstrate an integrated web-based and platform-independent data schema visualization.

The assistants for visual SPARQL query creation include *RDF Explorer* [15], *GRUFF* [16], as well as the schema-based *OptiqueVQs* [4], *LinDA* [17] and *ViziQuer* [6]. These tools, in their existing versions, support queries over stand-alone SPARQL endpoints. This paper extends the concept of schema-backed visual queries to be available over data endpoint federations, as well.

## 3. Data Schema Visualizations

In the context of (federated) SPARQL query creation, a visually presented data schema allows the user to visually comprehend the data set entities and their connections to understand what data are available for querying and to identify the entities and relations to be used in the query. The visual schema compacting methods (cf. [7]) allow to increase the size of schemas that can be legibly presented visually. Still, for larger and heterogeneous data sets such as *DBpedia* and *Wikidata* only certain meaningful schema fragments can be expected to be reasonably visualized.

Figure 1 contains a condensed visualization of the *StarWars* data schema [9] used further in this paper for demonstrating visual federated queries. The visualization employs class grouping (e.g., 39 classes are grouped together in the *Character et al.* node) and the shortening of node class lists (in the visual tool, the lists can be seen in full; the *StarWars* schema contains 51 classes). A finer-grained version of the schema (also using the concept of abstract superclass) is available in the *StarWars* example project in the *ViziQuer* playground and the project support page.

---

<sup>2</sup> [https://github.com/LUMII-Syslab/viziquer/tree/development/doc/demo/fed\\_queries](https://github.com/LUMII-Syslab/viziquer/tree/development/doc/demo/fed_queries)

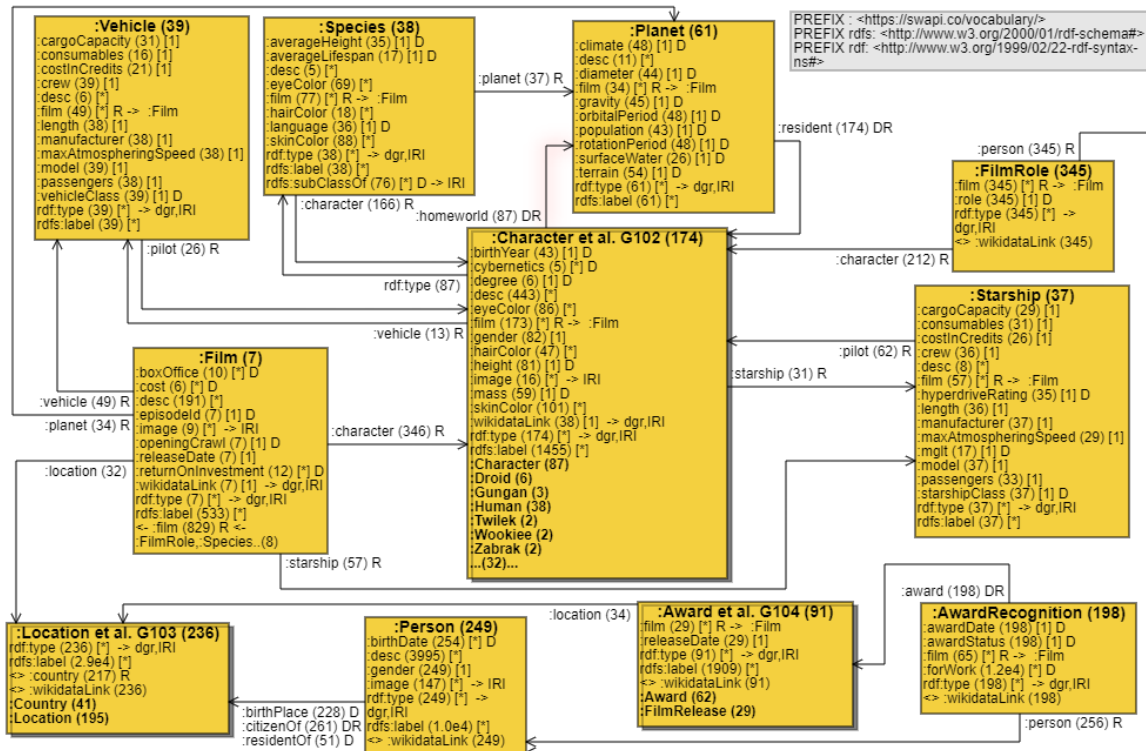


Figure 1: StarWars Data Schema Visualization

## 4. Federated Visual Queries

The visual query concept in *ViziQuer* is based on an (extended) tree of data nodes and control nodes. A data node corresponds to a variable in the query pattern and can be optionally assigned a class name from the schema class vocabulary (or a class name variable) and an instance name (to be translated into a SPARQL variable name). A control node (a unit node or a union node) can be used for further query structuring. There can be attribute expressions at the nodes building up the query selection list, as well as the conditions. The data edges correspond to property-based links among the node variables (property paths are allowed, as well), or they can be marked by edge property variables. There can also be structural edges, labelled by ‘++’ (no data connection specified by the edge) or ‘==’ (both edge ends correspond to the same instance).

The reader may consult [18] for further explanation and examples of the *ViziQuer* notation.

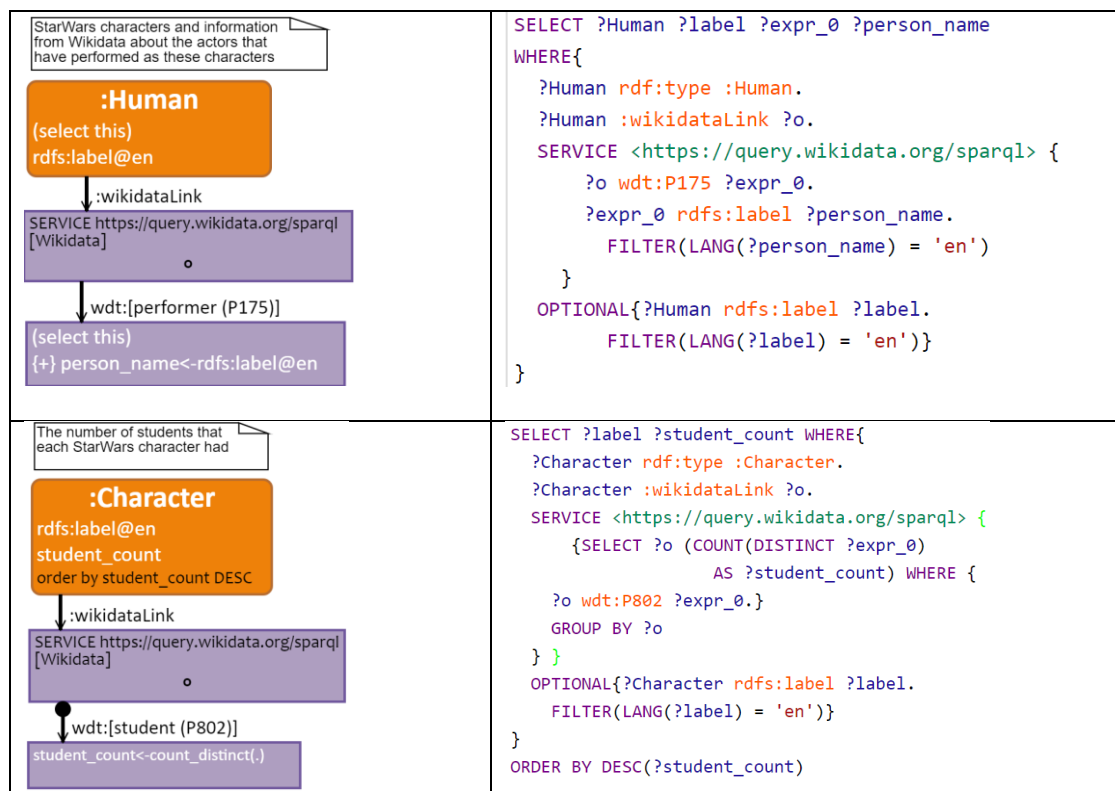
The root node in the query tree is depicted as an orange rounded rectangle (see Figure 2) that determines the scoping of fragment-based query constructs such as OPTIONAL and subquery, as well as the newly introduced GRAPH and SERVICE labels.

A federated query (as any other query) is created in the context of a certain data schema that serves the class and property vocabulary (including the entity labels that can be used in the visual query) and provides the resources for query auto-completion. To create a federated query, a SERVICE specification can be introduced either at the node level or at the edge level to attribute the node/edge and the entire query fragment behind it to another data schema and to include its SPARQL code in a SERVICE block that is to be executed over the specified SPARQL endpoint.

Each running *ViziQuer* tool instance provides a list of available data schemas. These schemas can be extracted from SPARQL endpoints by the OBIS Schema Extractor tool<sup>3</sup> and they are stored in the tool instance database by its administrator. The schemas for *DBpedia* and *Wikidata*, created by custom extraction processes, are available, as well. Should a query involve a SPARQL endpoint without the schema available in the visual query tool, the visual queries over it can still be created, relying on the common and/or explicitly defined namespace prefix declarations.

Figure 2 contains examples of visual federated queries over an instance of the *StarWars* data set [9], federated with remote data from *Wikidata*, and their translation to SPARQL.

<sup>3</sup> <https://github.com/LUMII-Syslab/OBIS-SchemaExtractor>



**Figure 2:** Example federated queries and their translations to SPARQL (standard prefix declarations omitted for presentation purposes)

Both queries in Figure 2 are initiated in the context of the *StarWars* data schema. They use the *:wikidataLink* property (still within the *StarWars* data set) to find the stored *Wikidata* URIs corresponding to the selected *StarWars* resources. These URIs are then used in the context of the *Wikidata* schema and SPARQL endpoint to find related information – either the list of performers for the *StarWars* characters or the count of students for each of them. In the second example, a subquery within the query service fragment is created. We note that the auto-completion of *Wikidata* properties *wdt:[performer (P175)]* and *wdt:[student (P802)]* was available for name auto-completion within the query link-building dialogue.

Although the benefits of the visual notation are most apparent for simpler queries, the *ViziQuer* tool allows users to create queries with a more complex structure, as well. These features also apply to federated queries. The *ViziQuer* tool also supports the visualization of existing textual SPARQL queries, with a rich set of full SPARQL constructs supported (cf. [8]). This functionality has been extended to include the federated query scenarios.

## 5. Conclusions and Future Work

In this work we have demonstrated how the *ViziQuer* visual query environment can be used for the visual creation of federated SPARQL queries backed by the data schemas of the SPARQL endpoints involved in these queries and offering context-aware query element auto-completion from the data structure described in multiple data schemas.

The visual presentation of the data schemas can help the user to comprehend the structure of the data sets to be queried and to identify the entities to be used in the query.

To further enhance support for creating federated queries, the auto-completion mechanism of the visual tool can be extended to include additional information about the possible cross-schema class and property connections (e.g., by comparing the namespace parts of class instance URIs). We plan to explore the options for this kind of functionality in the future.

Although the data schemas can be added to the query environment, an important future work would be to expand the library of schemas (cf. [7]) ready to be used for federated query support.

## Acknowledgements

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

## References

- [1] Lohmann, S., Negru, S., Haag F., Ertl, T. (2016). Visualizing Ontologies with VOWL. In: *Semantic Web 7(4)*, 399-419.
- [2] Bārzdīņš, J., Čerāns, K., Liepiņš, R., Sproģis, A. (2010). UML Style Graphical Notation and Editor for OWL 2. In: *Proc. of BIR'2010, LNBIP, Springer 2010*, vol. 64, pp. 102-113.
- [3] Fernandez-Álvarez, D., Labra-Gayo, J. E., & Gayo-Avello, D. (2022). Automatic extraction of shapes using sheXer. *Knowledge-Based Systems*, 238, 107975.
- [4] Soylyu A., Kharlamov, E., Zheleznyakov, D., Jimenez Ruiz, E., Giese M., Skjaeveland, M.G., Hovland, D., Schlatte, R., Brandt, S., Lie, H., Horrocks, I. (2018). OptiqueVQS: a Visual Query System over Ontologies for Industry, *Semantic Web 9(5)*, 627-660, IOS Press.
- [5] Čerāns, K., Ovčinnikova, J., Bojārs, U., Grasmanis, M., Lāce, L., Romāne, A. (2021). Schema-Backed Visual Queries over Europeana and Other Linked Data Resources, in Verborgh, R., et al. (ed.), *ESWC 2021 Satellite Events*. Springer LNCS, vol. 12739, 82-87. [https://doi.org/10.1007/978-3-030-80418-3\\_15](https://doi.org/10.1007/978-3-030-80418-3_15)
- [6] Čerāns, K., Šostaks, A., Bojārs, U., et al. (2018). ViziQuer: A Web-Based Tool for Visual Diagrammatic Queries Over RDF Data, in Gangemi, A., et al. (ed.), *ESWC 2018 Satellite Events*. LNCS, Vol. 11155. Springer, pp. 158-163. [https://doi.org/10.1007/978-3-319-98192-5\\_30](https://doi.org/10.1007/978-3-319-98192-5_30)
- [7] Lāce, L., Romāne, A., Fedotova, J., Grasmanis, M., Čerāns, K. (2024). A Method and Library for Visual Data Schemas. To appear in *Proc. of ESWC'2024 Satellite Events*, Springer LNCS.
- [8] Čerāns K, Ovčinnikova J, Grasmanis M, Lāce L, Romāne A. (2021). Visual presentation of SPARQL queries in ViziQuer. In: *Visualization and Interaction for Ontologies and Linked Data 2021*. Vol 3023. CEUR Workshop Proceedings, 29-40. <http://ceur-ws.org/Vol-3023/paper12.pdf>
- [9] Star Wars, Example Dataset. Last accessed on 2024-07-05. Available at <https://platform.ontotext.com/semantic-objects/datasets/star-wars.html>
- [10] Rabbani, K., Lissandrini, M., & Hose, K. (2023). Extraction of validating shapes from very large knowledge graphs. In *Proceedings of the Very Large Databases 2023*, 16(5), pp. 1023-1032.
- [11] Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M. (2015). 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](http://ceur-ws.org/Vol-1486/paper_77.pdf)
- [12] Dudáš, M., Lohmann, S., Svátek, V., Pavlov, D. (2018). Ontology visualization methods and tools: a survey of the state of the art. In: *The Knowledge Engineering Review*, 33.
- [13] Labra Gayo, J. E., Fernández-Álvarez, D., & García-González, H. (2018). RDFShape: An RDF playground based on Shapes. *CEUR Workshop Proceedings*, 2180.
- [14] Goasdoué, F., Guzewicz, P., & Manolescu, I. (2020). RDF graph summarization for first-sight structure discovery. *The VLDB journal*, 29(5), pp. 1191-1218.
- [15] Vargas, H., Buil-Aranda, C., Hogan, A., López, C. (2019). RDF Explorer: A Visual SPARQL Query Builder. In: Ghidini, C., et al. *The Semantic Web – ISWC 2019. Lecture Notes in Computer Science*, vol. 11778. Springer, Cham.
- [16] Aasman, J., & Cheetham, K. (2011). RDF browser for data discovery and visual query building. In *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2011)*, Co-located with ACM IUI (p. 53).
- [17] Thellmann, K., Orlandi, F., & Auer, S. (2014). LinDA - Visualising and Exploring Linked Data. In *SEMANTiCS 2014 (Posters & Demos)*, pp. 39-42.
- [18] Ovčinnikova J., Šostaks A., Čerāns K. (2023). Visual Diagrammatic Queries in ViziQuer: Overview and Implementation. *Baltic Journal of Modern Computing*, 11(2):317-350. doi:10.22364/bjmc.2023.11.2.07