CEUR-WS.org/Vol-2456/paper77.pdf

Visual Queries over Scholarly Data and other Linked Data Endpoints[1]

Kārlis Čerāns*, Lelde Lāce, Aiga Romāne, Jūlija Ovčiņņikova, Sergejs Kozlovičs[2],
Mikus Grasmanis, Jūlija Hodakovska, Artūrs Sproģis, Agris Šostaks

Institute of Mathematics and Computer Science, University of Latvia
*karlis.cerans@lumii.lv

Abstract. We demonstrate the option to use the schema-based visual query tool ViziQuer over realistic Linked Data endpoints, with examples over the Semantic Web conference-related Scholarly Data. We present the pipeline of enabling visual query creation over a SPARQL endpoint and ready-to-use data schemas over existing public Linked data endpoints, available in the ViziQuer Schema store.

Keywords: Visual query tool, RDF data, SPARQL, Linked Data

1 Introduction

Visual query composition (cf. [1], [2], [3], [4], [5]) along with facet-based ([6], [7]) and controlled natural language-based ([8]) approaches offers a promising avenue to enable end-user involvement in query composition over RDF/SPARQL data (cf. [1]).

The recent version of *ViziQuer* notation ([5], [9]), implemented in a web-based tool[3] [10], allows for visual presentation of rich instance level and aggregate queries, involving data expressions, as well as query nesting, with expressive power approaching that of the full SPARQL 1.1 [11]. Meanwhile, the currently available examples of the *ViziQuer* notation and tool usage are largely related to in-house RDF data stores, not the publicly available Linked data sets that would be one of its primary usage targets.

We shall report in this paper and present in the demonstration:

- Visual query notation and tool usage examples over a public Linked Data endpoint of Scholarly data[4] [12];
- The pipeline of enabling visual query creation over a SPARQL endpoint, including a novel easy-to-use data schema extractor implementation;
- Ready-to-use data schemas over existing public Linked data endpoints, available in the *ViziQuer* Schema store (within the *ViziQuer* tool page).

[1] Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[2] Supported by ERDF project 1.1.1.2/VIAA/1/16/214

[3] http://viziquer.lumii.lv/

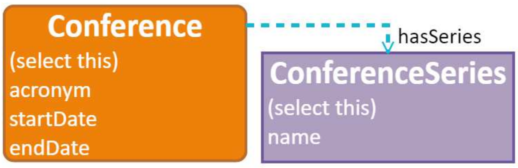
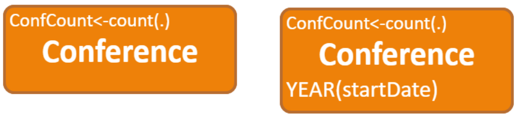
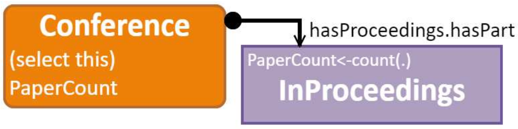
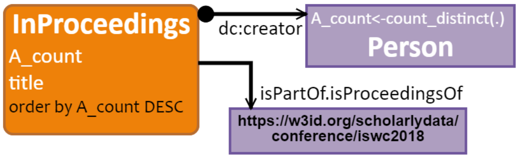
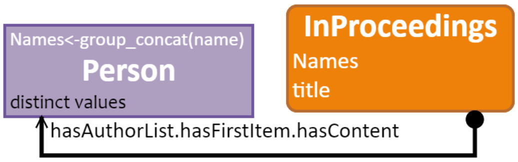
[4] http://www.scholarlydata.org/

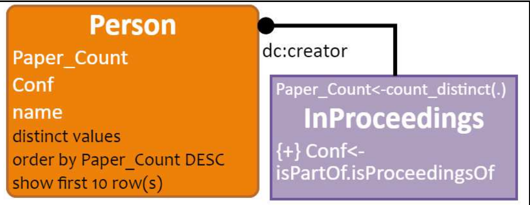
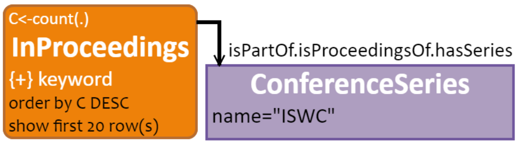
2 Scholarly Query Examples

The visual queries are defined and constructed in the context of a data schema listing the available classes and properties and their connectivity, as well as matching the entity short names to their full IRIs. We present a few examples here based on the data schema of Scholarly Data [12] (the schema can be downloaded from the *ViziQuer* Schema store).

Each query is a connected graph with one main query node (orange round rectangle). The linked nodes correspond either to joined classes or to nested queries (if the link starts with a black bullet). The *ViziQuer* notation is explained in detail in [5], [9], and the *ViziQuer* tool page.

Table 1. Visual Query examples over Scholarly Data

1. *List all conferences (IRI, acronym, start and end dates), together with optional conference series information (IRI, name).* Each query node is a pattern for a data instance, listing the class name and conditions, as well as the selection items; the links show the instance connections.	 Conference (select this) acronym startDate endDate hasSeries ConferenceSeries (select this) name
2. *Find the number of conferences and the number of conferences by the year of their start date (two queries).* The grouping is automatic by all non-aggregated selection fields.	 ConfCount<-count(.) Conference ConfCount<-count(.) Conference YEAR(startDate)
3. *List all conferences together with their Proceedings paper count.* A nested query construct is used for counting the papers in the context of each conference separately.	 Conference (select this) PaperCount hasProceedings.hasPart PaperCount<-count(.) InProceedings
4. *List titles of all papers from the ISWC 2018 conference, together with their author count (sort descending).* A query node can also hold an instance IRI information.	 InProceedings A_count title order by A_count DESC dc:creator A_count<-count_distinct(.) Person isPartOf.isProceedingsOf https://w3id.org/scholarlydata/conference/iswc2018
5. *List titles of all papers together with the papers' first author name(s).* The data may contain duplicate relation information. So, the distinct person name values are selected for each paper, then concatenated to include all name forms in the output.	 Names<-group_concat(name) Person distinct values InProceedings Names title hasAuthorList.hasFirstItem.hasContent

6. *Find the top 10 situations with persons publishing most papers at a single conference.* Nested queries can return more than single-item result sets projected onto the host query.	 Person Paper_Count Conf name distinct values order by Paper_Count DESC show first 10 row(s) dc:creator Paper_Count<-count_distinct(.) InProceedings {+} Conf<-isPartOf.isProceedingsOf
7 *List the top 20 keywords in proceeding papers of ISWC series conferences, together with using paper counts.* The keyword presence is mandatory, denoted by {+}.	 C<-count(.) InProceedings {+} keyword order by C DESC show first 20 row(s) isPartOf.isProceedingsOf.hasSeries ConferenceSeries name="ISWC"

3 Enabling Visual Queries over Linked Data Endpoints

The visual queries over a data endpoint are created within a context of a *ViziQuer* project that requires a pre-loaded data schema and a specified link to the SPARQL endpoint itself (if the created queries are to be executed from the tool environment).

The *ViziQuer* data schema can be either generated from an OWL ontology or extracted directly from the SPARQL endpoint (via schema-level SPARQL queries) using an open-source schema extractor service, accompanying the *ViziQuer* tool. The links to the service are maintained from the *ViziQuer* tool home page. The novel schema extractor implementation is as a JAVA web service, allowing its seamless usage directly from end-user computers.

There are public data endpoint schemas, extracted and collected at the *ViziQuer* Schema Store, including Scholarly Data, UNESCO[5] (SKOS), Social Semantic Web Thesaurus[6] and WikiPathways[7]. The schemas of resources like DBPedia and WikiData cannot be currently extracted in their full form due to space considerations.

It can also be envisaged that the SPARQL endpoint holders may store and maintain the visual query data schemas, or even the visual project examples along with the endpoints themselves to enable faster users starting up the work with the endpoint.

The visual query environment, after the data schema loading, displays the schema to the end user in the form of a class tree, arranged by the namespaces of the top-level classes and by the subclass relation; this form has been found suitable for moderately-sized data schemas (as the ones, listed in the schema store). A double click on the class tree node adds the main query node with this class into the diagrammatic query pane; further query elements then can be added in the context of the created query element.

There are also options for the data schema export in the form of OWL ontology from the *ViziQuer* tool, to enable its further analysis or eventual graphic visualization, e.g., in the OWLGrEd ontology editor[8].

[5] http://vocabularies.unesco.org/sparql

[6] http://vocabulary.semantic-web.at/PoolParty/sparql/semweb

[7] http://sparql.wikipathways.org/

[8] http://owlgred.lumii.lv/

4 Conclusions

There is an option to compose visual queries over Linked data endpoints, such as Scholarly Data, in the *ViziQuer* tool, so allowing the rich visual query experience over Linked Data.

The data schema retrieval as the necessary pre-processing step can be performed by an open-source schema extractor that can be used either from a public server or run on an end-user computer. Further development of the schema extraction service to be able to handle the peculiarities (query execution limits, supported protocols, and SPARQL subsets) of various SPARQL endpoints, is work in progress.

Another avenue of further work is creating an interactive graphical visualization of the data schema within the query tool to ease the end user query creation experience further.

References

1. Soylu, A., Giese, M., Jimenez-Ruiz, E., Vega-Gorgojo, G.., Horrocks, I.: Experiencing OptiqueVQS: A Multi-paradigm and Ontology-based Visual Query System for End Users. Universal Access in the Information Society, March 2016, Volume 15, Issue 1, pp 129–152.
2. Zviedris, M., Barzdins, G.: ViziQuer: A Tool to Explore and Query SPARQL Endpoints. In: The Semantic Web: Research and Applications, LNCS, Volume 6644, pp. 441-445, (2011)
3. Kapourani, B., Fotopoulou, E., Papaspyros, D., Zafeiropoulos, A., Mouzakitis, S., Koussouris, S.: Propelling SMEs Business Intelligence Through Linked Data Production and Consumption, In OTM 2015 Workshops pp 107-116, 2015.
4. Haag, F., Lohmann, S., Siek, S., Ertl, T.: QueryVOWL: Visual Composition of SPARQL Queries. In: The Semantic Web: ESWC 2015 Satellite Events. LNCS, Vol.9341, pp. 62-66. Springer, (2015), http://vowl.visualdataweb.org/queryvowl/
5. Čerāns, K., Bārzdiņš, J., Šostaks, A., Ovčiņņikova, J., Lāce, L., Grasmanis, M. and Sproģis, A.: Extended UML Class Diagram Constructs for Visual SPARQL Queries in ViziQuer/web In Voila!2017, CEUR Workshop Proceedings, Vol.1947, (2017) pp.87-98.
6. Vega-Gorgojo, G., Giese, M., Heggestoyl, S., Soylu, A., Waaler, A.: PepeSearch: Semantic Data for the Masses. In: PLoS ONE 11(3): e0151573. doi: 10.1371/journal.pone.0151573, 2016. http://dx.doi.org/10.1371/journal.pone.0151573
7. Khalili, A., Meroño-Peñuela, A.: WYSIWYQ --- What You See Is What You Query. In Voila!2017, CEUR, Vol.1947, (2017) pp.123-130. http://ceur-ws.org/Vol-1947/paper11.pdf
8. Ferré, S.: Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language, Semantic Web, 2017, Vol 8, pp 405-418
9. Čerāns, K., Šostaks, A., Bojārs, U., Bārzdiņš, J., Ovčiņņikova, J., Lāce, L., Grasmanis, M. and Sproģis, A., ViziQuer: A Visual Notation for RDF Data Analysis Queries. In *Research Conference on Metadata and Semantics Research*. Springer CCIS, Vol.846, pp.50-62, 2018
10. Čerāns, K., Šostaks, A., Bojārs, U., Ovčiņņikova, J., Lāce, L., Grasmanis, M. Romāne, A., Sproģis, A., Bārzdiņš, J. ViziQuer: A Web-Based Tool for Visual Diagrammatic Queries Over RDF Data. In: ESWC 2018 Satellite Events. LNCS, vol 11155, pp. 158-163, 2018.
11. SPARQL 1.1 Query Language. W3C Recommendation 21 March 2013, http://www.w3.org/TR/2013/REC-sparql11-query-20130321/
12. A. L. Gentile and A. G. Nuzzolese. cLODg - Conference Linked Open Data Generator. In *ISWC 2015 Posters & Demonstrations Track,* CEUR-WS.org, Vol. 1486, 2015.