QueryVOWL: Visual Composition of SPARQL Queries

Florian Haag, Steffen Lohmann, Stephan Siek, and Thomas Ertl

Institute for Visualization and Interactive Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart, Germany
{florian.haag,steffen.lohmann,thomas.ertl}@vis.uni-stuttgart.de

Abstract. In order to make SPARQL queries more accessible to users, we have developed the visual query language QueryVOWL. It defines SPARQL mappings for graphical elements of the ontology visualization VOWL. In this demo, we present a web-based prototype that supports the creation, modification, and evaluation of QueryVOWL graphs. Based on the selected SPARQL endpoint, it provides suggestions for extending the query, and retrieves IRIs and literals according to the selections in the QueryVOWL graph. In contrast to related work, SPARQL queries can be created entirely with visual elements.

Keywords: Visual Querying, QueryVOWL, VOWL, SPARQL, RDF, OWL, Visualization, Linked Data, Semantic Web.

1 Introduction

As an increasing amount of Linked Data is becoming available, various visual concepts for specifying search queries on that data have been proposed. While some of them focus on visualizing the Boolean connections between filter criteria [4,7], others represent the structure of the object graph [3,6,11,12]. Examples from the latter group reflect the basic idea of queries specified in SPARQL, and the visualizations are often syntactically close to textual SPARQL queries. For instance, they explicitly show variable names or textual filter expressions.

We have developed QueryVOWL, a graph-based visual query language for SPARQL endpoints. QueryVOWL reuses graphical elements of VOWL, the Visual Notation for OWL Ontologies [9], and defines SPARQL mappings for them. We strive for expressing query restrictions in a way that does not require any knowledge of SPARQL and aim to reduce the learning effort by reusing elements that users of ontologies might already know from VOWL. Furthermore, we decided to reuse VOWL, as empirical results indicate that it is comparatively intuitive and understandable, also and especially to lay users [9].

In this demo, we present a web-based prototype of a visual query system that allows for the composition of queries in the QueryVOWL notation and retrieves results from a SPARQL endpoint. At ESWC 2015, we will demonstrate the idea and functionality of QueryVOWL by creating several SPARQL queries with the prototype. A more in-depth description of the QueryVOWL visual language and its SPARQL mappings has been presented at a workshop [5].

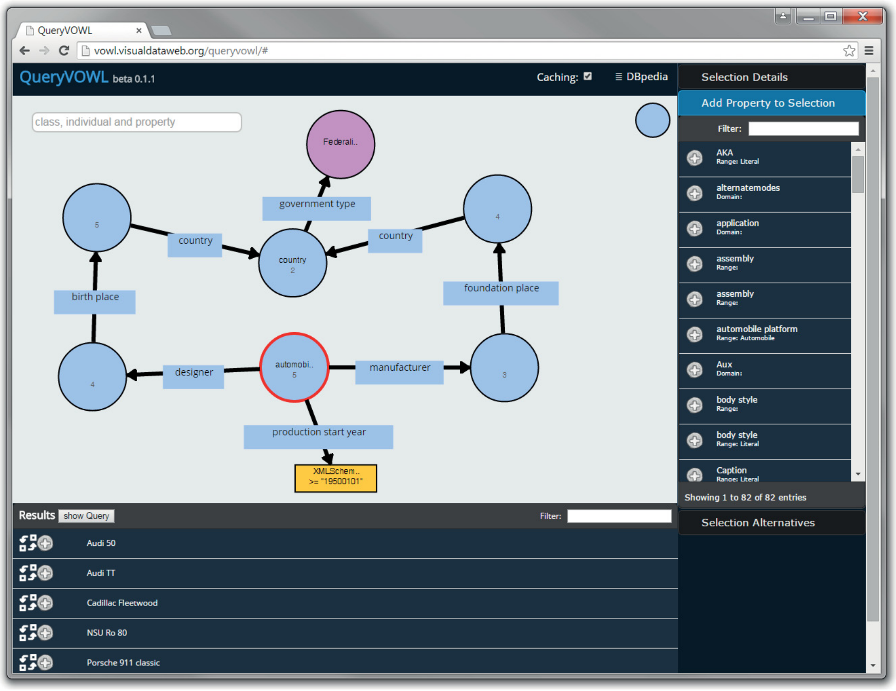
2 QueryVOWL

VOWL, the Visual Notation for OWL Ontologies, provides a set of visual elements that represent concepts and components defined in OWL ontologies [9]. The meaning of the shapes, colors, labels, and combinations thereof is defined in a specification document [10], which maps graphical features to OWL constructs.

For QueryVOWL, we have redefined the graphical elements used in VOWL by mapping them to SPARQL fragments, while still conceptually adhering to the original definitions of the elements with respect to OWL. Moreover, we have slightly extended some of the visual elements to introduce interactive functionality that assists in the creation of SPARQL queries.

We developed a web-based prototype of QueryVOWL that implements the main elements of the visual query language. It is based on open web standards (HTML, JavaScript, CSS, SVG) and integrates some JavaScript libraries, most importantly D3 [2] for the visualization of the QueryVOWL graph. Figure 1 depicts a screenshot of the prototype, showing an exemplary QueryVOWL graph created on the DBpedia dataset [1].

The QueryVOWL concept focuses on a selected element—a class node, a literal node, or a property label—chosen by the user for retrieving results. In the exemplary query of Figure 1, a user is looking for specific cars restricted by sev-



AKA
Range: Literal

alternatemodes
Domain:

application
Domain:

assembly
Range:

assembly
Range:

automobile platform
Range: Automobile

Aux
Domain:

body style
Range:

body style
Range: Literal

Caption
Range: Literal

Showing 1 to 82 of 82 entries

Audi 50

Audi TT

Cadillac Fleetwood

NSU Ro 80

Porsche 911 classic

Fig. 1. Web-based implementation of QueryVOWL visualizing an exemplary query.

eral attributes. The queries are automatically generated from the QueryVOWL graphs and sent to a SPARQL endpoint of choice in order to retrieve results. Listing 1 shows the SPARQL query generated by the prototype for the class *Automobile* selected in Figure 1.

Listing 1. SPARQL query resulting from the QueryVOWL graph shown in Figure 1.

```
SELECT ?Node1
WHERE {
  ?Node1 a <http://dbpedia.org/ontology/Automobile>.
  ?Node1 <http://dbpedia.org/property/manufacturer> ?Node2.
  ?Node2 <http://dbpedia.org/ontology/foundationPlace> ?Node4.
  ?Node4 <http://dbpedia.org/ontology/country> ?Node6.
  ?Node3 <http://dbpedia.org/property/birthPlace> ?Node5.
  ?Node5 <http://dbpedia.org/ontology/country> ?Node6.
  ?Node6 a <http://dbpedia.org/ontology/Country>.
  ?Node6 <http://dbpedia.org/ontology/governmentType>
    <http://dbpedia.org/resource/Federalism>.
  ?Node1 <http://dbpedia.org/property/designer> ?Node3.
  ?Node1 <http://dbpedia.org/ontology/productionStartYear> ?Node7.
  FILTER(?Node7 >= "19500101"^^<http://www.w3.org/2001/XMLSchema#gYear>).
  FILTER(datatype(?Node7) = <http://www.w3.org/2001/XMLSchema#gYear>).
}
```

The prototype can be used to create various kinds of SPARQL queries, and it can be applied to any RDF dataset that provides a SPARQL endpoint. The user interface of the prototype consists of three views—the main view, a sidebar, and a result list—which are described in the following.[1]

2.1 Main View

The main view contains the drag-and-drop enabled QueryVOWL visualization, using SVG graphics similar to the WebVOWL implementation [8]. Like in WebVOWL, long class labels are abbreviated, but the full label is always available as a tooltip. If no label is set for an element, the last part of its IRI is used. Interactive spots on the visual elements react to hovering, clicking, and dragging. For each class node, the number of individuals that match this class is displayed. Upon any change to the graph, the numbers of affected classes are re-requested from the SPARQL endpoint and updated.

Apart from the QueryVOWL visualization, the main view features icons for directly inserting graph elements (currently, only for class nodes), as well as a search box equipped with auto-completion. The search box serves for finding specific entities in the dataset, such as classes or individuals, by their name, and inserting them into the QueryVOWL graph. It can also be used to directly input IRIs and add the corresponding element, should a user wish to copy and paste an IRI from another source.

[1] The prototype and a video are available at `http://queryvowl.visualdataweb.org`.

2.2 Sidebar

The sidebar is divided into three lists organized in an accordion widget. All the information shown in the lists is retrieved by means of SPARQL queries, which are processed in the background once the selection in the QueryVOWL graph changes.

The first list provides details about the selected element, which always includes a hyperlink to its IRI and some additional literal values if the selected element is an individual. The second list suggests properties that might be added to the selected class or individual in order to extend the query (cf. Figure 1). The properties are all linked to the selected element in the accessed RDF data, so that the suggestions assist the users in defining restrictions that make it less likely that no results are returned. The third list suggests elements that might be used as a replacement for the currently selected element. For classes and individuals, other classes are listed that might be appropriate replacements; other properties are listed when a property is selected. These suggestions are again retrieved based on the modeled QueryVOWL graph and on how the elements are used in the RDF data. For the convenience of the user, the suggestions are also available in dropdown lists accessible directly on the QueryVOWL elements of the main view.

2.3 Result List

The result list shows the labels of all individuals that are valid replacements for the selected class node, along with hyperlinks to their IRIs. In other words, all individuals that match the focused property restrictions (i.e., the corresponding RDF subgraph) are displayed in the result list. Similar to the functionality of the sidebar, classes in the main view can be replaced by individuals from the result list to further restrict the QueryVOWL graph.

Moreover, the textual SPARQL query used to retrieve the individuals of the result list can be displayed and copied. This enables expert users to first create a SPARQL query visually with QueryVOWL and then edit it textually according to their needs.

2.4 Data Retrieval

The QueryVOWL prototype generates SPARQL queries like the one shown in Listing 1. We integrated an optional cache module, as the response time for complex queries may be noticeable on some endpoints. In addition, the cache module helps reduce the workload and resource use of those endpoints. Recently sent SPARQL queries and their results are stored in the cache, and the remote SPARQL endpoint is only accessed for non-cached queries.

In order to recognize equivalent but differently written SPARQL queries, the queries are normalized in the cache module. For this purpose, the graph patterns are brought into a specified order and names of variables are substituted based on a fixed scheme. This normalization is invisible to the caller of the module, as the original variable names are replaced in the returned result.

3 Conclusion

QueryVOWL supports the construction of SPARQL queries without the need to input any structured text. Once set up on a SPARQL endpoint, no particular RDF knowledge is required to use the approach. Since QueryVOWL reuses visual elements of the ontology notation VOWL, we expect it to be especially comprehensible to users who have previously come in touch with VOWL. However, it may also be easily understandable to people who never used VOWL before. This is at least indicated by the results of a preliminary user study we conducted to evaluate QueryVOWL [5].

The presented prototype demonstrates how the QueryVOWL concept can be used in practice for the visual composition of SPARQL queries. Currently, it does not support all envisioned graphical elements of QueryVOWL, but we plan to advance it in the future and to provide more options and features. Moreover, the QueryVOWL concept may be extended by adding enhanced capabilities for comparing literals, and for improved support of logical combinations of filter criteria to visually form conjunctions and disjunctions, among others.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia—a crystallization point for the web of data. Web Semantics 7(3), 154–165 (2009)
2. Bostock, M., Ogievetsky, V., Heer, J.: D3 data-driven documents. IEEE Transactions on Visualization and Computer Graphics 17(12), 2301–2309 (2011)
3. Groppe, J., Groppe, S., Schleifer, A.: Visual query system for analyzing social semantic web. In: WWW '11. pp. 217–220. ACM (2011)
4. Haag, F., Lohmann, S., Ertl, T.: SparqlFilterFlow: SPARQL query composition for everyone. In: ESWC 14 Satellite Events, LNCS, vol. 8798, pp. 362–367. Springer (2014)
5. Haag, F., Lohmann, S., Siek, S., Ertl, T.: Visual querying of linked data with QueryVOWL. In: HSWI '15. CEUR-WS (2015), to appear
6. Heim, P., Ziegler, J., Lohmann, S.: gFacet: A browser for the web of data. In: IMC-SSW '08. CEUR-WS, vol. 417, pp. 49–58 (2008)
7. Jarrar, M., Dikaiakos, M.D.: MashQL: A query-by-diagram topping SPARQL. In: ONISW '08. pp. 89–96. ACM (2008)
8. Lohmann, S., Link, V., Marbach, E., Negru, S.: WebVOWL: Web-based visualization of ontologies. In: EKAW 14 Satellite Events, LNAI, vol. 8982. Springer (2015), to appear
9. Lohmann, S., Negru, S., Haag, F., Ertl, T.: VOWL 2: User-oriented visualization of ontologies. In: EKAW '14, LNCS, vol. 8876, pp. 266–281. Springer (2014)
10. Negru, S., Lohmann, S., Haag, F.: VOWL: Visual notation for OWL ontologies. `http://purl.org/vowl/` (2014)
11. OpenLink: iSPARQL. `http://oat.openlinksw.com/isparql/`
12. Russell, A., Smart, P., Braines, D., Shadbolt, N.: NITELIGHT: A graphical tool for semantic query construction. In: SWUI '08. CEUR-WS, vol. 543 (2008)