

SPARQLING: painlessly drawing SPARQL queries over GRAPHOL ontologies

Sara Di Bartolomeo ^b, Gianluca Pepe ^b,
Valerio Santarelli ^{a,b}, Domenico Fabio Savo ^{a,b}

(a) Sapienza Università di Roma
<lastname>@diag.uniroma1.it

(b) OBDA Systems
<lastname>@obdasystems.com

Abstract. There is a reason why the proverb says “A picture is worth a thousand words”: diagrams or sketches of concepts and of the relationships that exist between them help understand their meaning. In the Semantic Web, the concepts and relationships of the world are modeled through ontologies, but current standards for ontology languages such as OWL 2 don’t buy into this idea, recurring to formula-based representations, that work well with machines and not so well with humans. Similarly, the languages used to query ontologies, in particular SPARQL, are not immediately intuitive for people who lack specific training or background, so providing graphical support for query formulation can make the life of the end user easier. In this work we present the SPARQLING system for SPARQL query building based on the GRAPHOL visual language for ontologies. The characterizing feature of SPARQLING is the idea to preserve and take advantage of the native diagrammatic representation of GRAPHOL ontologies, allowing the user to navigate it, and to construct a graph-based representation of the query over it through a simple point-and-click mechanism. The system then automatically transforms the graphical query into the SPARQL syntax.

1 Introduction

Ontologies provide a conceptualization of a domain of interest, meant to foster knowledge sharing and integration among different information systems, and to support human understanding and communication through a high level of abstraction that is clear for domain experts. Recent years have seen a sharp increase in the popularity of ontologies in modern applications, underlined by wide-spread use in biomedicine, e-commerce, Semantic Web, and Data Integration [22].

Ontologies can be formalized in different languages, from Description Logics [4] (DLs) to RDF(S)¹ to OWL 2², the latter two being W3C standards, and the use of these languages is propaedeutic to enabling automated reasoning techniques over ontologies in information systems. However, people operating in organizational settings typically do not possess the necessary skills to interpret the formulas that are used as

¹ <http://www.w3.org/TR/rdf-primer-20040210/>

² <http://www.w3.org/TR/owl2-primer>

means of expression in such languages. This introduces a bottleneck in the ontology design phase, where ontology engineers usually work together with domain experts. To alleviate this issue, there have been various efforts in recent years to devise graphical ontology syntaxes based on standard conceptual modeling languages, e.g., UML class diagrams or ER diagrams [5, 9, 10]. These languages however commonly suffer from a lack of expressiveness, being unable to represent OWL 2 ontologies in a completely graphical way. In some cases, a solution has been to present a language that introduces formulaic elements, which is a choice that hinders the diagrammatic representation of the ontology and its intuitive understanding. Similar difficulties arise, particularly in the enterprise context, with other visual ontology notations that are not based on languages such as UML and ER [14]. The latest contribution to this line of research is the GRAPHOL language [8, 15]. GRAPHOL resembles ER and UML class diagrams, but has an inherent formal semantics based on DLs, presents an entirely graphical syntax, and is able to fully capture OWL 2. The effectiveness of GRAPHOL as the language for representing ontologies has been proven in various research and industrial projects in recent years [2, 3]. GRAPHOL comes equipped with its own custom editor, Eddy [16], which allows to draw ontologies in GRAPHOL, and export them into OWL 2.

Aside from acting as a standardized reference model, ontologies can also be used as a tool for accessing and managing data. In this scenario, the use of triple stores for managing (Linked) enterprise data is quickly spreading [23], and semantic technologies, particularly ontologies, are adopted as a means to manage legacy data sources [17]. The standard language for querying both triples stores and ontologies is SPARQL³. This language, while somewhat simpler in structure than traditional structured query languages such as SQL, is still however not immediately intuitive for people who lack specific training or background, and thus formulating SPARQL queries can be quite challenging for end users. Once again, graphical support could be an effective solution for query formulation. Indeed, such solutions have been the subject of research for many years now [7], going back to the development of visual query systems and languages (VQSs and VQLs) for structured and semi-structured data, which utilize techniques such as direct manipulation of visual objects that represent the domain and query language. These early approaches [6, 7] established the research field, but their success remains somewhat limited in practice. More recently, various solutions for SPARQL query building systems have been proposed, most commonly recurring to some sort of graphical representation or interface to construct the query and to exploiting the axioms in the ontology to suggest relevant parts of it. These works can be divided into two broad categories. The first are approaches that attempt to exploit the alphabet of the ontology to formulate queries in natural language, or to iteratively build them through context-sensitive completion mechanisms, and then produce the corresponding SPARQL query [11, 13, 18]. The second, instead, adopt a graphical notation, most typically graph-based, for the query, and provide GUI-support for its construction [1, 12, 20, 19, 21].

In this work we present the SPARQLING system for SPARQL query building based on the GRAPHOL visual language for ontologies. The characterizing feature of SPARQLING, which differentiates it from these previous works, is the idea to preserve and take advantage of the native diagrammatic representation of GRAPHOL ontologies, al-

³ <https://www.w3.org/TR/rdf-sparql-query/>

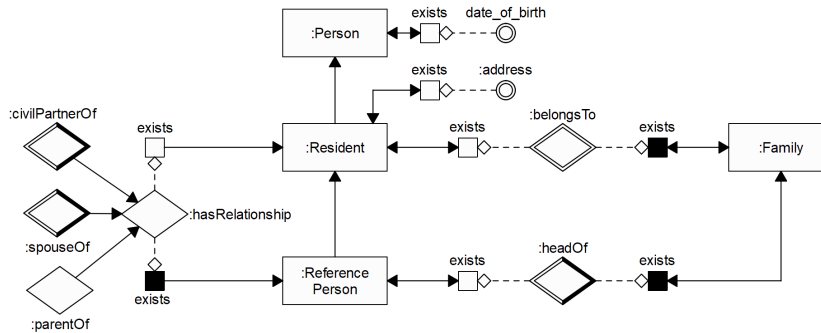


Fig. 1. GRAPHOL representation of a portion of the Base Registry of Individual, Families, and Cohabitation ontology, developed for the Italian National Institute of Statistics.

lowing the user to navigate it, and to construct a graph-based representation of the query over it through a simple point-and-click mechanism. The system then automatically transforms the graphical query into the SPARQL syntax. SPARQLING is an open-source project, available on Github⁴, and is supported by OBDA Systems⁵.

In the following sections we first provide some background on the GRAPHOL language and on SPARQL, and then we introduce the SPARQLING system. Finally, we conclude the work with a brief discussion on future developments of the system.

2 Background

In this section, we briefly introduce the GRAPHOL [8, 15] visual language for OWL 2 ontologies and the SPARQL query language.

GRAPHOL is a completely graphical language for representing OWL 2 ontologies which has an inherent formal semantics based on DLs and resembles ER diagrams, but which is in fact able to fully capture OWL 2 without recurring to formulas or annotations. The basic idea behind GRAPHOL is to represent the ontology as a set of inclusions, as in DLs or OWL 2, while preserving a graphical representation of it, given in terms of a directed graph. The nodes of the graph are atomic predicates or graphical operators, whereas the edges are solid or dashed arrows. The latter denote input to operators, whereas the former denote inclusions. In GRAPHOL, rectangles are used for classes (called entities in ER), diamonds are used for object properties (corresponding to binary relationships in ER), circles are used for data properties, and rounded rectangles are used for data types (e.g., *string* and *integer*). GRAPHOL expressions are either nodes representing predicates of the ontology (e.g., named classes and properties), or a combination of these nodes with other nodes that represent operators, shaped as blank or solid boxes, or hexagons. For instance, labeled blank and solid boxes are used for constructing object or data property restrictions on the domain and range, respectively, and solid hexagons represent disjoint unions. Figure 1 depicts a portion of

⁴ https://github.com/picorana/painless_sparql, where a demo is available.

⁵ <https://www.obdasystems.com/>

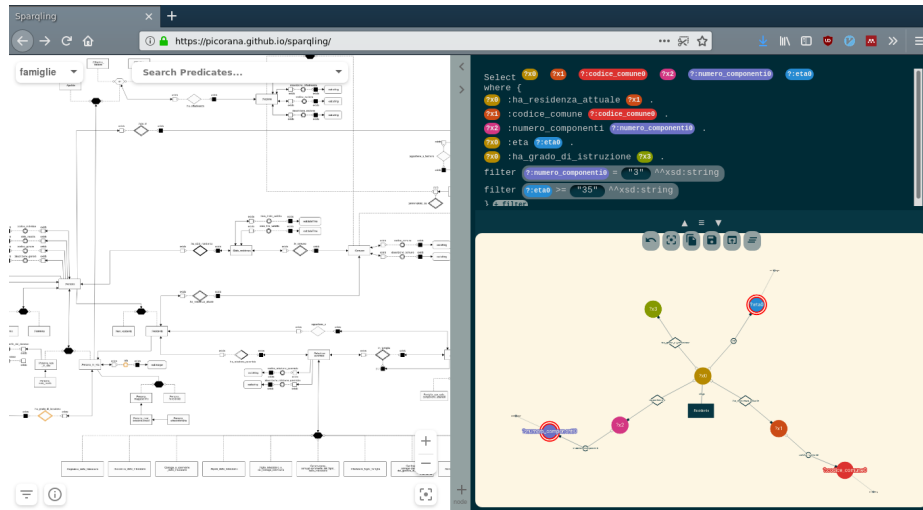


Fig. 2. The user interface of the SPARQLING system.

the Base Registry of Individual, Families, and Cohabitation ontology, developed for the Italian National Institute of Statistics [3] in GRAPHOL.

SPARQL is the W3C recommendation query language for RDF data. The structure of the language is based on graph patterns and query evaluation consists in subgraph matching. The basic building block of a SPARQL query is a basic graph pattern (BGP), which consists in a set of triple patterns, i.e., RDF triples that may contain query variables at the subject, predicate, and object position. More complex SPARQL query patterns are built over BGPs.

3 The System

SPARQLING is a web-based application for constructing SPARQL queries by exploiting the GRAPHOL representation of the ontology. The user interface is split in three different sections, as shown in Figure 2. On the left, the user can see the GRAPHOL representation of the ontology, while on the bottom right the query canvas displays the visual representation of the query under construction, and on the top right the system provides the SPARQL translation of the query. The current implementation of SPARQLING is tailored towards queries for Ontology-based Data Management systems [17], for which the support of the conjunctive query fragment of SPARQL is sufficient.

The distinguishing feature of SPARQLING is that the system keeps the GRAPHOL representation of the ontology always accessible to the user. The intuition is that having a clear understanding of the structure and content of the ontology, in its GRAPHOL form, is helpful in designing queries. Indeed, the mental process of writing a query is natural when looking at a GRAPHOL diagram because it recalls the act of tracing a path on it. SPARQLING attempts to capture these query paths on the diagram through the visual representation of the query provided in the query canvas.

This representation, which we refer to as *query graph*, indeed has the structure of a graph, to naturally recall SPARQL's BGPs. Generally speaking, a triple pattern in the query corresponds to an edge connecting two nodes, each representing a variable or a constant. To provide a visual correspondence with GRAPHOL, the graphical representation of the ontology predicates in the query graph is the same as in GRAPHOL.

To add a triple to the basic graph pattern of the query, the user double clicks on a predicate symbol in the GRAPHOL diagram. According to the specific predicate symbol, the corresponding edge and nodes are automatically added to the query graph. If a node of the query graph is selected prior to choosing the symbol in the GRAPHOL diagram, the new edge will be linked to it. Otherwise, a new component, disconnected from the rest of the query graph, will be added. Actions on specific nodes of the graph are performed through contextual menus that appear by right clicking on a node, allowing for renaming, deleting, switching between variable and constants, adding to the select clause, and specifying filter expressions.

The system maintains at all times a SPARQL representation of the query which the user can inspect and modify. Each modification of the SPARQL query is immediately mirrored in the query graph, and viceversa. By hovering onto the name of a variable in the SPARQL query, the corresponding node in the query graph is highlighted in order to make its position stand out. The user can actually use SPARQLING by interacting exclusively with the query graph or with the SPARQL representation, according to the user's preference. The application is developed in JavaScript, using the Cytoscape.js library⁶ for the rendering of the GRAPHOL diagram and the query graph, and the cola.js library⁷ for the force-directed layout of the query graph.

4 Conclusions

We have presented the SPARQLING system for drawing SPARQL queries over GRAPHOL ontologies. The development of this system aims to allow users of SPARQL-based query systems to produce queries in a more intuitive fashion. Although the current version is fully functional, the SPARQLING project is still under development and will be extended in different directions. We aim to extend the portion of SPARQL which is currently supported by SPARQLING; we will work to further improve the synergy between the GRAPHOL diagram and the query graph that the user is constructing; and we will allow importing SPARQL queries over the ontology into the system. Finally, testing of the system has so far been carried out through its use in the field; we would like to conduct a more structured and thorough user evaluation to identify potential areas of improvement in the system.

References

1. O. Ambrus, K. Möller, and S. Handschuh. Konduit VQB: a Visual Query Builder for SPARQL on the Social Semantic Desktop. In *Proc. of VISSW2010*, 2010.

⁶ <http://js.cytoscape.org/>

⁷ <http://ialab.it.monash.edu/webcola/>

2. N. Antonioli, F. Castanò, S. Coletta, S. Grossi, D. Lembo, M. Lenzerini, A. Poggi, E. Virardi, and P. Castracane. Ontology-based data management for the Italian public debt. In *Proc. of FOIS 2014*, pages 372–385, 2014.
3. R. M. Aracri, A. M. Bianco, R. Radini, M. Scannapieco, L. Tosco, F. Croce, D. F. Savo, and M. Lenzerini. On the Experimental Usage of Ontology-based Data Management for the Italian Integrated System of Statistical Registers: Quality Issues. In *Proc. of Q2018*, 2018. (to appear).
4. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, 2007.
5. S. Brockmans, R. Volz, A. Eberhart, and P. Löffler. Visual modeling of OWL DL ontologies using UML. In *Proc. of ISWC 2004*, volume 3298 of *LNCS*, pages 198–213. Springer, 2004.
6. T. Catarci. What happened when database researchers met usability. *Information Systems*, 25(3):177–212, 2000.
7. T. Catarci, M. F. Costabile, S. Levialdi, and C. Batini. Visual query systems for databases: A survey. *Journal of Visual Languages & Computing*, 8(2):215–260, 1997.
8. M. Console, D. Lembo, V. Santarelli, and D. F. Savo. Graphol: Ontology representation through diagrams. In *Proc. of DL 2014*, volume 1193 of *CEUR*, pages 483–495, 2014.
9. D. Djuric, D. Gasevic, V. Devedzic, and V. Damjanovic. A UML profile for OWL ontologies. In *Proc. of MDFA2004*, volume 3599 of *LNCS*, pages 204–219, 2004.
10. R. Falco, A. Gangemi, S. Peroni, D. Shotton, and F. Vitali. Modelling OWL ontologies with Graffoo. In *Proc. of ESWC 2014 (Satellite Events)*, volume 8798 of *LNCS*, pages 320–325, 2014.
11. E. Franconi, P. Guagliardo, S. Tessaris, and M. Trevisan. Quelo: an ontology-driven query interface. In *Proc. of DL 2011*, volume 745, pages 488–498, 2011.
12. F. Haag, S. Lohmann, S. Siek, and T. Ertl. QueryVOWL: Visual composition of SPARQL queries. In *Proc. of ISWC 2015*, pages 62–66. Springer, 2015.
13. E. Kaufmann, A. Bernstein, and R. Zumstein. Querix: A natural language interface to query ontologies based on clarification dialogs. In *Proc. of ISWC 2006*, pages 980–981. Springer, 2006.
14. S. Krivov, R. Williams, and F. Villa. GrOWL: A tool for visualization and editing of OWL ontologies. *J. of Web Semantics*, 5(2):54–57, 2007.
15. D. Lembo, D. Pantaleone, V. Santarelli, and D. F. Savo. Easy OWL Drawing with the Graphol Visual Ontology Language. In *Proc. of KR 2016*, pages 573–576, 2016.
16. D. Lembo, D. Pantaleone, V. Santarelli, and D. F. Savo. Drawing OWL 2 ontologies with Eddy the editor. *AI Commun.*, 31(1):97–113, 2018.
17. M. Lenzerini. Ontology-based Data Management. In *Proc. of CIKM 2011*, pages 5–6, 2011.
18. L. McCarthy, B. Vandervalk, and M. Wilkinson. SPARQL Assist language-neutral query composer. *BMC Bioinformatics*, 13(1):S2, 2012.
19. D. Schweiger, Z. Trajanoski, and S. Pabinger. SPARQLGraph: a web-based platform for graphically querying biological Semantic Web databases. *BMC bioinformatics*, 15(1):279, 2014.
20. P. R. Smart, A. Russell, D. Braines, Y. Kalfoglou, J. Bao, and N. R. Shadbolt. A Visual Approach to Semantic Query Design Using a Web-Based Graphical Query Designer. In *Proc. of EKAW 2008*.
21. A. Soyulu, M. Giese, E. Jimenez-Ruiz, E. Kharlamov, D. Zheleznyakov, and I. Horrocks. OptiqueVQS: towards an ontology-based visual query system for big data. In *Proc. of MEDES 2013*, pages 119–126. ACM, 2013.
22. S. Staab and R. Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.
23. D. Wood. *Linking enterprise data*. Springer Science & Business Media, 2010.