

Making Complex Ontologies End User Accessible via Ontology Projections (Short Paper)

Ahmet Soylu^{1,2} and Evgeny Kharlamov^{3,4}

¹ Norwegian University of Science and Technology, Gjøvik, Norway

`ahmet.soylu@ntnu.no`

² SINTEF Digital, Oslo, Norway

³ University of Oxford, Oxford, the UK

`evgeny.kharlamov@cs.ox.ac.uk`

⁴ University of Oslo, Oslo, Norway

Abstract. Ontologies are a powerful mechanism to structure domains of interest. They have successfully been applied in medical domain, industry and other important areas. Despite the simplicity of ontological vocabularies that consist of classes and properties, ontologies can relate elements of the vocabulary with the help of axioms in a very non-trivial way. Thus, the relationship between classes and properties can become hardly accessible by end users thus affecting the practical value of ontologies. Indeed, it is essential for end users to be able to navigate or browse through an ontology, to get a big picture of what classes there are and what they have in common in terms of other related classes and properties. This helps end users in effectively performing various knowledge engineering tasks such as querying and domain exploration. To this end, in this short paper, we describe an approach to project OWL 2 ontologies into graphs and show how to leverage this approach in practical systems for visual query formulation and faceted search that we tested in various scenarios.

Keywords: OWL 2 · Ontologies · Graph navigation · Knowledge graphs.

1 Introduction

Ontologies are a powerful mechanism to structure domains of interest. They have successfully been applied in medical domain [12], industry [9,10] and other important areas. Particularly, OWL 2 ontologies [6] have been key for constructing semantic knowledge graphs. A knowledge graph describes real world entities and their interrelations [25]. They have been used both in academia, such as Yago [3] and DBpedia [13], and in industry such as Google’s Knowledge Graph, Facebook’s Graph Search, and Microsoft’s Satori. Semantic knowledge graphs are typically stored or exported as RDF datasets, which allow for storing sparse and diverse data in an extensible and adaptable way [23]. Semantics of such datasets are typically encoded in OWL 2 ontologies.

Despite the simplicity of ontological vocabularies that consist of classes and properties, ontologies can relate elements of the vocabulary with the help of axioms in a very non-trivial way. Thus, the relationship between classes and properties can become hardly accessible by end users thus affecting the practical value of ontologies. Indeed, it is essential for end users to be able to navigate or browse through an ontology, to get a big picture of what classes are there, what they have in common in terms of other related classes and properties [8,14]. This helps end users in effectively performing various knowledge engineering tasks such as querying and domain exploration, e.g., via query by navigation [22,18].

In order to help end users to query and explore ontologies, in this short paper, we describe an approach to project OWL 2 ontologies into graphs and show how to leverage this approach in practical systems for visual query formulation and faceted search that we tested in various scenarios.

In particular, we implemented this approach in two semantic tools, namely *OptiqueVQS*[21] for visual query formulation and *SemFacet* [1] for faceted search and evaluated under different use cases.

The rest of the paper is structured as follows. Section 2 presents our graph projection approach from ontologies, while Section 3 presents the tools using our approach. Finally, Section 5 concludes the paper.

2 Graph Projection

Our goal for graph projection is, given an ontology, to create a directed labelled graph, called navigation graph [1], whose nodes correspond to the named classes and datatypes in the ontology and edges between nodes to the object properties and datatype properties. Let C_1, C_2 , and C_3 be classes, r_1, r_2 , and r_3 object properties, d_1 a datatype property, i_1 and i_2 individuals, and dt_1 a data type. First, each class and datatype in the ontology is translated to a node in the navigation graph. Then we add edges of the form $r_1(C_1, C_2)$ and $d_1(C_1, dt_1)$ derived from the axioms of the ontology. The types of axioms resulting in an edge are presented with examples in the followings using description logic (DL) [2].

Ontologies have a propagative effect on the amount of information to be presented. This case is considered in two forms, namely the top-down and bottom-up propagation of property restrictions [5,18]. The first form emerges from the fact that, in an ontology, explicit restrictions attached to a class are inherited by its subclasses. The second form is rooted from the fact that the interpretation of an OWL class also includes the interpretations of all its subclasses. Therefore, for a given class, it may also make sense to derive edges from the (potential) object and datatype properties of its subclasses and superclasses.

2.1 Edges Through Object Properties

Domains and Ranges: Domain and range axioms using named classes are translated to an edge. For example, axioms given in Ex. 1 map to edge $r_1(C_1, C_2)$.

$$\exists r_1.\top \sqsubseteq C_1 \text{ and } \top \sqsubseteq \forall r_1.C_2 \quad (1)$$

$$\exists r_1.\top \sqsubseteq C_1 \text{ and } \top \sqsubseteq \forall r_1.(C_2 \sqcup C_3) \quad (2)$$

If a complex class expression, formed through intersection (\sqcap) or union (\sqcup), appears as a domain and/or range, then an edge is created for each pair of domain and range classes. For example, axioms given in Ex. 2 map to edges $r_1(C_1, C_2)$ and $r_1(C_1, C_3)$.

Object Property Restrictions: Object property restrictions used in class descriptions, formed through existential quantification (\exists), universal quantification (\forall), individual value restriction, max (\geq), min (\leq), and exactly ($=$), are mapped to edges. For example, axioms given in Ex. 3 to 5 map to $r_1(C_1, C_2)$. Note that in Ex. 5, there is a complex class expression on the left-hand-side.

$$C_1 \sqsubseteq \exists r_1.C_2 \quad (3)$$

$$C_1 \equiv \leq_n r_1.C_2 \quad (4)$$

$$\forall r_1.C_1 \sqsubseteq C_2 \quad (5)$$

Axioms given in Ex. 6 include an individual value restriction and an edge is created with the type of individual, that is $r_1(C_1, C_2)$.

$$C_1 \sqsubseteq \exists r_1.\{i_1\}, \text{ and } i_1 : C_2 \quad (6)$$

Axiom given in Ex. 7 includes a complex class expression. In this case, an edge is created for each named class, that is $r_1(C_1, C_2)$ and $r_1(C_1, C_3)$.

$$C_1 \sqsubseteq \exists r_1.(C_2 \sqcup C_3) \quad (7)$$

Given an enumeration of individuals, an edge is created for each individual's type. For example, axioms given in Ex. 8 map to two edges, that is $r_1(C_1, C_2)$ and $r_1(C_1, C_3)$.

$$C_1 \sqsubseteq \exists r_1.\{i_1\} \sqcup \{i_2\}, i_1 : C_2, \text{ and } i_2 : C_3 \quad (8)$$

Inverse Properties: Given an edge in the navigation graph such as $r_1(C_1, C_2)$ and an inverse property axiom for the corresponding object property such as given in Ex. 9, a new edge is created for the inverse property, that is $r_1^{-1}(C_2, C_1)$.

$$r_1 \equiv r_1^{-1} \quad (9)$$

Role Chains: Given two edges $r_1(C_1, C_2)$ and $r_2(C_2, C_3)$ in the navigation graph, and a role chain axiom between r_1, r_2, r_3 such as given in Ex. 10, a new edge is created for r_3 , that is $r_3(C_1, C_3)$.

$$r_1 \circ r_2 \sqsubseteq r_3 \quad (10)$$

Top-down Propagation: Given an edge $r_1(C_1, C_2)$ in the navigation graph and a subclass axiom such as as given in Ex. 11, a new edge is added to the graph, that is $r_1(C_3, C_2)$. Similar edges could be created for subproperties.

$$C_3 \sqsubseteq C_1 \quad (11)$$

Bottom-up Propagation: Given an edge $r_1(C_1, C_2)$ in the navigation graph and a subclass class axiom such as given in Ex. 12, a new edge is added to the graph, that is $r_1(C_3, C_2)$. Similar edges could be created for superproperties.

$$C_1 \sqsubseteq C_3 \quad (12)$$

2.2 Edges through Datatype Properties

Domains and Ranges: Domain and range axioms using datatype properties are translated to an edge. For example, axioms given in Ex. 13 map to an edge, that is $d_1(C_1, dt_1)$.

$$\exists d_1. \textit{DatatypeLiteral} \sqsubseteq C_1 \text{ and } \top \sqsubseteq \forall r_1. dt_1 \quad (13)$$

Datatype Property Restrictions: Datatype property restrictions, formed through existential quantification (\exists), universal quantification (\forall), max (\geq), min (\leq), exactly ($=$), and value are mapped to edges. For example, axiom given in Ex. 14 maps to $d_1(C_1, dt_1)$.

$$C_1 \sqsubseteq \exists d_1. dt_1 \quad (14)$$

Top-down Propagation: Given an edge $d_1(C_1, dt_1)$ in the navigation graph and a subclass axiom such as as given in Ex. 15, a new edge is added to the graph, that is $d_1(C_2, dt_1)$. Similar edges could be created for subproperties.

$$C_2 \sqsubseteq C_1 \quad (15)$$

Bottom-up Propagation: Given an edge $d_1(C_1, dt_1)$ in the navigation graph and a subclass class axiom such as given in Ex. 16, a new edge is added to the graph, that is $d_1(C_3, dt_1)$. Similar edges could be created for superproperties.

$$C_1 \sqsubseteq C_3 \quad (16)$$

3 Applications

Variants of this approach have been implemented and evaluated in OptiqueVQS [21], a visual query formulation tool, and SemFacet [1], a faceted search tool. Both interfaces support tree-shaped conjunctive queries.

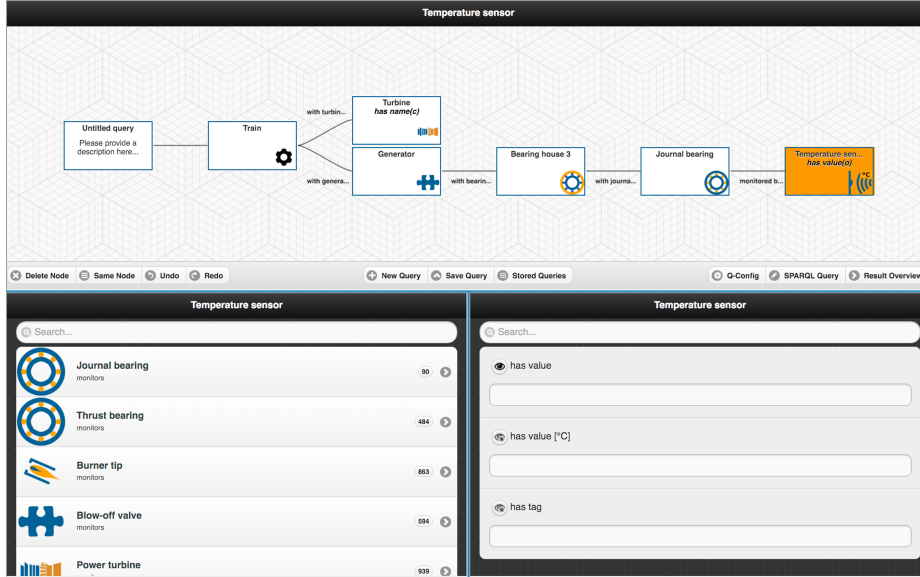


Fig. 1. OptiqueVQS over a use case provided by Siemens.

OptiqueVQS (see Figure 1) is a visual query system. It allows users to navigate the conceptual space and each traversal from a class to another adds a typed variable-node and object property connecting it to the query graph. OptiqueVQS was deployed and evaluated in different use cases, including Siemens’ case for sensor data [20], Statoil’s case for oil and gas [21], and on generic datasets [19]. In Figure 1, there is an example query asking for all trains with a turbine named “Bearing Assembly” and their journal bearing temperature readings in the associated generator.

SemFacet (see Figure 2) is full-fledged general-purpose faceted search interface. In typical faceted search, users are presented with facet-values organised in groups according to facet-names and it is often not allowed to navigate between classes. SemFacet allows end users to navigate between classes and browse data sets at the same time. The interface was deployed and evaluated over a slice of Yago database [1]. In Figure 2 there is an example search for US presidents who graduated from Harvard or Georgetown, and whose children graduated from Stanford. All this conditions are combined conjunctively and their constraints apply simultaneously. One can see that changing the focus of the query, one can

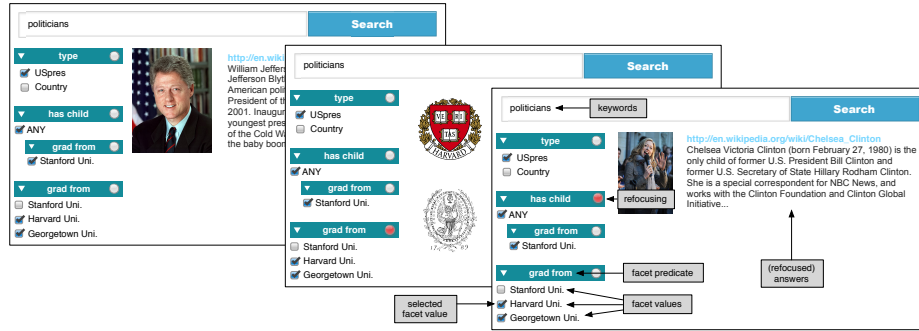


Fig. 2. SemFacet over Yago Knowledge Graph.

either see the presidents (left part of the figure), or their universities (centre part of the figure), or their children (right part of the figure).

4 Related Work

Visualisations for different aspects of the Semantic Web such as ontology visualisation, query formulation, and search are relevant for the work presented here, since they mainly require end users to examine and interact with the elements of a given ontology. However, to best of our knowledge, non of the existing works deal with projecting navigation graphs from ontologies, although the inverse exists such as for ontology axiomatization through diagramming [16].

Among others [8], graph paradigm is often used to depict the structure of ontological elements and relationships as they reflect the interconnected nature of ontology classes. There are various approaches using graphs for ontology visualisation and exploration such as GrOWL [11] and KC-Viz [15]. Similarly, tools for visual query formulation also often use graph paradigm to depict the information needs and domain exploration such as gFacet [7] and NITELIGHT [17]. In a graph based approach, classes are often represented as nodes and properties as edges.

Non-graph based approaches, such as form-based, still use a navigation approach for browsing through ontology classes. Examples include Rhizomer [4], a faceted search tool, and PepeSearch [24], a form-based query formulation tool. Typically, form-based approaches are meant to operate on a single class level; however, as in the case of Rhizomer and PepeSearch, navigation between classes is an essential instrument.

OptiqueVQS and SemFacet represent these two different paradigms, that is graph-based and form-based respectively. In OptiqueVQS, the navigation graph is used to explore domain, while a constrained tree-shaped representation is used for query visualisation instead of a graph for usability purposes, while SemFacet allows navigation between classes and employs form elements rather than graphical visualisations. We refer interested readers to related publications

[1,21] on these tools for end user and performance experiments, which present positive evidence for the usefulness of our approach.

5 Conclusions

In this paper, we presented an approach, together with two example applications, for navigating OWL 2 ontologies by projecting them into graphs through harvesting a set of axioms. We consider two major challenges to be addressed for the future work. First challenge is to enable users to navigate distant classes that are not directly connected but are multiple edges away. We call this non-local navigation, which could be useful for navigating large class networks. Second challenge, concerns the information overflow during the query formulation and ontology exploration due to large number of ontology elements; we aim to develop mechanism for adaptively filtering down ontology elements.

Acknowledgements This work is partially funded by EU H2020 TheyBuy-ForYou (780247) project.

References

1. Arenas, M., Grau, B.C., Kharlamov, E., Marciuska, S., Zheleznyakov, D.: Faceted search over RDF-based knowledge graphs. *Journal of Web Semantics* **37-38**, 55–74 (2016)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA (2003)
3. Biega, J., Kuzey, E., Suchanek, F.M.: Inside YAGO2s: A Transparent Information Extraction Architecture. In: *Proceedings of the 22nd International Conference on World Wide Web (WWW 2013)*. pp. 325–328. ACM (2013)
4. Brunetti, J.M., García, R., Auer, S.: From Overview to Facets and Pivoting for Interactive Exploration of Semantic Web Data. *International Journal on Semantic Web and Information Systems* **9**(1), 1–20 (2013)
5. Grau, B.C., Giese, M., Horrocks, I., Hubauer, T., Jiménez-Ruiz, E., Kharlamov, E., Schmidt, M., Soylu, A., Zheleznyakov, D.: Towards Query Formulation, Query-Driven Ontology Extensions in OBDA Systems. In: *Proceedings of the 10th International Workshop on OWL: Experiences and Directions (OWLED 2013)*. CEUR Workshop Proceedings, vol. 1080. CEUR-WS.org (2013)
6. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. *Journal of Web Semantics* **6**(4), 309–322 (2008)
7. Heim, P., Ertl, T., Ziegler, J.: Facet Graphs: Complex Semantic Querying Made Easy. In: *Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010)*. LNCS, vol. 6088, pp. 288–302. Springer (2010)
8. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.G.: Ontology visualization methods - a survey. *ACM Comput. Surv.* **39**(4) (2007)
9. Kharlamov, E., Hovland, D., Skjæveland, M.G., Bilidas, D., Jiménez-Ruiz, E., Xiao, G., Soylu, A., Lanti, D., Rezk, M., Zheleznyakov, D., Giese, M., Lie, H., Ioannidis, Y.E., Kotidis, Y., Koubarakis, M., Waaler, A.: Ontology Based Data Access in Statoil. *Journal of Web Semantics* **44**, 3–36 (2017)

10. Kharlamov, E., Mailis, T., Mehdi, G., Neuenstadt, C., Özçep, Ö.L., Roshchin, M., Solomakhina, N., Soylu, A., Svingos, C., Brandt, S., Giese, M., Ioannidis, Y.E., Lamparter, S., Möller, R., Kotidis, Y., Waaler, A.: Semantic access to streaming and static data at Siemens. *Journal of Web Semantics* **44**, 54–74 (2017)
11. Krivov, S., Williams, R., Villa, F.: GrOWL: A Tool for Visualization and Editing of OWL Ontologies. *Journal of Web Semantics* **5**(2), 54–57 (2007)
12. Lee, D., Cornet, R., Lau, F.Y., de Keizer, N.: A survey of SNOMED CT implementations. *Journal of Biomedical Informatics* **46**(1), 87–96 (2013)
13. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morse, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* **6**(2), 167–195 (2015)
14. Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with VOWL. *Semantic Web* **7**(4), 399–419 (2016)
15. Motta, E., Mulholland, P., Peroni, S., d’Aquin, M., Gomez-Perez, J.M., Mendez, V., Zablith, F.: A Novel Approach to Visualizing and Navigating Ontologies. In: *Proceedings of the 10th International Conference on The Semantic Web (ISWC 2011)*. LNCS, vol. 7031, pp. 470–486. Springer (2011)
16. Sarker, M.K., Krisnadhi, A.A., Hitzler, P.: OWLax: A Protege Plugin to Support Ontology Axiomatization through Diagramming. In: *Proceedings of the Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016)*. CEUR Workshop Proceedings, vol. 1690. CEUR-WS.org (2016)
17. Smart, P.R., Russell, A., Braines, D., Kalfoglou, Y., Bao, J., Shadbolt, N.R.: A Visual Approach to Semantic Query Design Using a Web-Based Graphical Query Designer. In: *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*. LNAI, vol. 5268, pp. 275–291. Springer (2008)
18. Soylu, A., Giese, M., Jiménez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I.: Ontology-based end-user visual query formulation: Why, what, who, how, and which? *Universal Access in the Information Society* **16**(2), 435–467 (2017)
19. Soylu, A., Giese, M., Jiménez-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* **15**(1), 129–152 (2016)
20. Soylu, A., Giese, M., Schlatte, R., Jiménez-Ruiz, E., Kharlamov, E., Özçep, Ö.L., Neuenstadt, C., Brandt, S.: Querying industrial stream-temporal data: An ontology-based visual approach. *Journal of Ambient Intelligence and Smart Environments* **9**(1), 77–95 (2017)
21. Soylu, A., Kharlamov, E., Zheleznyakov, D., Jimenez-Ruiz, E., Giese, M., Skjæveland, M.G., Hovland, D., Schlatte, R., Brandt, S., Lie, H., Horrocks, I.: OptiqueVQS: a Visual Query System over Ontologies for Industry. *Semantic Web* **9**(5), 627–660 (2018)
22. Soylu, A., Mödritscher, F., Causmaecker, P.D.: Ubiquitous web navigation through harvesting embedded semantic data: A mobile scenario. *Integrated Computer-Aided Engineering* **19**(1), 93–109 (2012)
23. Suchanek, F.M., Weikum, G.: Knowledge Bases in the Age of Big Data Analytics. *PVLDB* **7**(13), 1713–1714 (2014)
24. Vega-Gorgojo, G., Giese, M., Heggstøyl, S., Soylu, A., Waaler, A.: PepeSearch: Semantic Data for the Masses. *PLoS ONE* **11**(3) (2016)
25. Yan, J., Wang, C., Cheng, W., Gao, M., Zhou, A.: A Retrospective of Knowledge Graphs. *Frontiers of Computer Science* **12**(1), 55–74 (2018)