Improving the Cost of Updates in Virtual Knowledge Graphs

Romuald Esdras Wandji[1,*], Diego Calvanese[1,2]

[1]Department of Computing Science, Umeå Universitet, Umeå, Sweden
[2]Research Centre for Knowledge and Data, Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy

Abstract
Virtual Knowledge Graph (VKG) is known as a data integration paradigm used to efficiently manage the heterogeneity of richly structured data that is common inside several organizations, in inter-organizational settings, and more openly on the Web. Although such a paradigm continues to gain importance in both foundational and applied research, updates in VKG systems remain an open challenge that has received less attention. Yet, a solution to such a problem would be of great importance, as it would allow VKG systems to be full-fledged, thus allowing end-users to fully manage source data through the lens of the ontology they are exposed to. This research aims to propose a comprehensive framework for instance-level updates in VKGs, where updates posed over the ontology have to be translated into source-level updates and, more importantly, how the side effects related to the propagation of ontology-based updates to the underlying data source can be minimized.

Keywords
Knowledge Representation, Virtual Knowledge Graph (VKG), Ontology-based Data Access, View Updates

1. Introduction

As a rapidly growing field, VKGs are robust tools for integrating heterogeneous data source systems with the help of ontologies. VKG systems are virtual approaches that allow users to issue high-level ontological queries, which are automatically translated into equivalent low-level queries (like SQL in a relational setting) that the underlying database engine can execute. Formerly known as ontology-based data Access (OBDA), a VKG system consists of three main components: an ontology, a set of data sources, and the mapping between the two. The ontology is a unified and abstract view of multiple local data sources, thus allowing for more expressive data querying and improving data integration [1, 2, 3, 4], and is typically represented using a formal language such as OWL 2 (the ontology language standardized by the W3C) or one of its profiles (i.e., sub-languages) [5]. The data sources to be integrated, which are typically relational, contain information concerning the domain of interest and are accessed and managed by (possibly) different organizations. Finally, the mapping is a set of declarative assertions expressed in the R2RML language [6] that describe how to populate the ontology from data sources. In the VKG approach, the facts generated by the mapping from the underlying data

Woodstock'22: Symposium on the irreproducible science, June 07–11, 2022, Woodstock, NY
*Corresponding author.
romuald.esdras.wandji@umu.se (R. E. Wandji); diego.calvanese@unibz.it (D. Calvanese)
https://www.umu.se/en/staff/romuald-esdras-wandji (R. E. Wandji)
0009-0008-5036-2452 (R. E. Wandji); 0000-0001-5174-9693 (D. Calvanese)

source are kept virtual and available to the user at query time. The main reasoning service provided by the VKG systems so far is *query answering*, which is carried out through *query rewriting* and *query unfolding* [1, 7].

Problem Statement and Contribution. One of the greatest achievements made in VKGs is the ability to query information stored in source data using Semantic Web technologies, such as Resource Description Framework (RDF) [8] and Web Ontology Language (OWL) [9], which allows the user to leverage the open-world assumption provided by the Semantic Web while maintaining the data in sources that traditionally operate under the close-world assumption. However, by taking advantage of the Knowledge Graph's capacity to handle incomplete information, it would be desirable to also provide support for update operations over the source data through the lens of the ontology. Such a feature will allow data and content owners to detach from low-level details of the underlying source structure and organization. Unfortunately, the issue of updates in VKGs, which accounts for the translation of a set of (complete/incomplete) insertion or deletion operations over the ontology into equivalent operations over the data source, has received little attention so far and yet remains a challenging task. A solution to this problem would be of great practical relevance since it would allow the management of the key operations that are of interest in an information system (i.e., queries *and updates*) through the lens of an ontology. This research aims at introducing in VKGs the notion of *ontology-based update* and *evolution* and to study foundational and applied issues related to this extension. In particular, it would be possible to: (a) Insert new objects into a class of the ontology and populate the corresponding relations that are mapped to this class; (b) Add a new data property instance to an object in a class and populate the corresponding attribute(s) that are mapped to this class; (c) Connect two objects in two classes of the ontology through an object property instance and populate the corresponding attributes and relations that are mapped to these classes; (c) Remove an object, an instance of a data property, or an instance of an object property by deleting the corresponding data from the underlying mapped relations; (d) Perform a combination of multiple operations of the types above.

Overall, this research is aimed at extending the capabilities of the VKG framework from "*read-only*" to "*write-also*" so it can dynamically manage and evolve data through ontology-based operations.

Importance. Enriching VKGs with update and evolution capabilities while maintaining consistency represents an important step toward the practical usefulness of the VKG paradigm, as it will impact how modern information systems handle data, making them more flexible and responsive to changes. Using low-level languages like SQL to manage complex and large data can be challenging and time consuming as it requires domain experts for maintenance. However, by leveraging ontologies specified in user-friendly languages, organizations could simplify data management, reducing reliance on domain experts, lowering operational costs, and increasing organizational agility.

2. Research Questions

We observe that in the typical context of incomplete information provided by an ontology, each of the insertion operations and their combination may generate an inconsistency in the data with respect to the axioms contained in the ontology. In order to characterize the semantics of such a system it becomes therefore necessary to rely on a suitably-defined consistency-tolerant semantics, e.g., based on the notion of repair. A second challenge in VKG systems comes from the presence of mappings, due to the inherent ambiguity that such mappings introduce when there is the need to propagate an ontology-based update (even one that does not generate an inconsistency, such as a delete operation) to the source data. Indeed, a VKG mapping is essentially a view that defines an ontology element (class or property) in terms of a query over the data source. Hence, each update over the ontology element translates into an update over the view that combines all queries that correspond to mappings for that ontology element, and thus faces the *view-update* problem that has a long history in relational database management [10, 11, 12].

This scenario poses a set of challenges and research questions that I aim to investigate:
RQ1: Under which conditions can update operations over the VKG defined by an ontology be rewritten into update operations performed directly over the objects that constitute the VKG (without the need to take into account the ontology axioms)?
RQ2: Which additional information is it necessary to maintain in order to find an effective solution to the view-update problem for VKG mappings?
RQ3: How can ontology-based updates be implemented effectively in a state-of-the art VKG system that supports query rewriting?

References

[1] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, J. on Data Semantics 10 (2008) 133–173. doi:10.1007/978-3-540-77688-8_5.

[2] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, Ontologies and databases: The *DL-Lite* approach, in: S. Tessaris, E. Franconi (Eds.), Reasoning Web: Semantic Technologies for Informations Systems – 5th Int. Summer School Tutorial Lectures (RW), volume 5689 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 255–356.

[3] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: A survey, in: Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI), IJCAI Org., 2018, pp. 5511–5519. doi:10.24963/ijcai.2018/777.

[4] G. Xiao, L. Ding, B. Cogrel, D. Calvanese, Virtual Knowledge Graphs: An overview of systems and use cases, Data Intelligence 1 (2019) 201–223. doi:10.1162/dint_a_00011.

[5] M. Krötzsch, OWL 2 profiles: An introduction to lightweight ontology languages, in: Reasoning Web: Semantic Technologies for Advanced Query Answering – 8th Int. Summer School Tutorial Lectures (RW), volume 7487 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 112–183. doi:10.1007/978-3-642-33158-9_4.

[6] S. Das, R2rml: Rdb to rdf mapping language, http://www. w3. org/TR/r2rml/ (2011).

[7] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, G. Xiao, Ontop: Answering sparql queries over relational databases, Semantic Web 8 (2017) 471–487.

[8] J. Z. Pan, Resource description framework, in: Handbook on ontologies, Springer, 2009, pp. 71–90.

[9] D. L. McGuinness, F. Van Harmelen, et al., Owl web ontology language overview, W3C recommendation 10 (2004) 2004.

[10] F. Bancilhon, N. Spyratos, Update semantics of relational views, ACM Trans. on Database Systems 6 (1981) 557–575.

[11] S. S. Cosmadakis, C. H. Papadimitriou, Updates of relational views, J. of the ACM 31 (1984) 742–760. doi:10.1145/1634.1887.

[12] A. C. Kakas, P. Mancarella, Database updates through abduction, in: Proc. of the 16th Int. Conf. on Very Large Data Bases (VLDB), volume 90, 1990, pp. 650–661.