Improving the Cost of Updates in Virtual Knowledge Graphs

Romuald Esdras Wandji

Department of Computing Science, Umeå Universitet, Umeå, Sweden

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 source are kept virtual and available to the user at query time. The main reasoning service

Proceedings of the Doctoral Consortium at ISWC 2024, co-located with the 23rd International Semantic Web Conference (ISWC 2024)

romuald.esdras.wandji@umu.se (R. E. Wandji)
https://www.umu.se/en/staff/romuald-esdras-wandji (R. E. Wandji)
0009-0008-5036-2452 (R. E. Wandji)

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. My PhD 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:

- Insert new objects into a class of the ontology and populate the corresponding relations that are mapped to this class;
- Add a new data property instance to an object in a class and populate the corresponding attribute(s) that are mapped to this class;
- 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;
- 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;
- Perform a combination of multiple operations of the types above.

Overall, my 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. Related Literature

Besides reasoning and querying in VKGs, it is of interest to update the systems through the ontology and, more specifically, to update its extensional data by translating the requested ontology-based update into a suitable update over the source data. This form of update is called instance-level update and is the focus of this research. It is also important to mention that for instance-level updates; we enforce the condition that the knowledge graph resulting from the application of an update operation preserves the same ontology as the original one; hence, we allow changes only in the ABox.

The task of instance-level update in VKGs, as described so far, has not been studied yet in the literature and comes with some important challenges. First, we have to address the complex logic-based inferences in the ontologies that can lead to a non-deterministic approach to updating knowledge bases (KBs) [10, 11] and in that context, several update semantics addressing the challenge of ABox updates of KBs have been proposed [11]. These semantics can be classified into two groups: *formula-based* and *model-based*. In formula-based, updates are represented as a set of logical formulas, representing a semantic change that should be reflected in the ontology, and the consistency in the updated ABox is gained by removing the minimal number of facts that contradict the ontology, and the model-based considers updating all models of the KB. [12, 13] proposed formula-based semantic under which the ABox that maximally accomplishes (based on the notion of repair) a given update is unique given the property of *DL-Lite*. [14, 15] considers a model-based approach, where [14] proposed an algorithm called *ComputeUpdate* for computing ABox update for $\mathit{DL\text{-}Lite}_F$ KB that extends the work proposed in [15]. [16, 17] considers more expressive DLs and shows update is not expressible for all DL languages. However, these proposed solutions are limited to the KB and, therefore, do not cover mappings and, thus, updates over actual source data from which the KB is formed.

Updating the source data remains a desirable way to practically reflect the change requested over the ontology. [18] proposed an alternative approach to this problem by maintaining the independence of the data sources. However, in practical cases, managing additional structures like mappings and auxiliary tables can lead to increased storage and processing overhead and affect the system's performance and scalability. In our research, we are more concerned with translating the update to source data, and this leads to the second main challenge we are facing and, which comes from the presence of VKG mappings that define ontology elements in terms of queries over the source data.

When it comes to translating a KB update into an equivalent data source update, the problem we face comes from the ambiguity of the VKG mapping, which means that there may be more than one possible translation over the source data, and this has been studied in the context of *view-update* [19, 20]. Such ambiguity poses important challenges in our work, and that has a long story in relational database research [21, 22, 23, 19, 20]. In order to address such ambiguity, some existing approaches consist of either syntactically restricting the view function [19, 24, 25] or propose a dialog-based solution to interact with the user in order to choose the correct translation manually [26, 23]. Unfortunately, advanced commercial systems like PostgreSQL provide limited support for view updates (for instance, UCQ views cannot be updated) [27]. Recent approaches to deal with such ambiguities focused on the bidirectional approach [28, 29] where a default program is used to map view update into source data update uniquely [30, 31].

3. Research Questions and Hypotheses

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 [32, 33, 34].

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?
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?

4. Preliminary Result and Next Steps

In an effort to provide an answer to the research questions related to this study, our preliminary result so far is focused on **RQ1** [35]. In this study, we addressed the challenge of updates in VKG, providing conditions of realisability of a given ontology-based updates in VKGs and studying possible techniques for updating a VKG system. As we explained earlier in this paper, our research combines two main issues that have already been studied independently in the literature: the first one is related to the repair of an inconsistent KB, and the second is related to the view update problem, and in order to provide a sound solution to the issue of updates in VKGs, we have based our research on the following assumptions, common for the VKG setting.

The Ontology is specified in *DL-Lite*$_R$. In order to avoid non-determinism of repair over the KB when an update is requested, we adopt *DL-Lite*$_R$ as a language to specify ontologies, since it does not allow for qualified existential restriction over the left-hand side of its assertions, and all its axioms are binary in the sense that they involve exactly one atom in each side of an inclusion assertion. This feature allows *DL-Lite*$_R$ KB to have a unique repair in the presence of an update causing an inconsistency [36], which is important in our framework when trying to repair a KB with the updates provided by the user. Moreover, we rely here on the non-recursive Datalog program presented in [13], which derives the set of insertions and deletions needed to realize updates over the KB.

Mappings are specified in R2RML. Mappings in VKGs make use of so-called *IRI-templates* to specify how values retrieved from a data source should be used to construct the IRIs (i.e., identifiers) of the objects in the ontology. Such IRIs formally correspond to Skolem functions, and it is commonly assumed that such functions are injective, i.e., different database values never result in two objects that have the same IRI (neither when the objects are generated using the same IRI-template, nor using different IRI-templates). As a consequence, there is a unique way to "invert" an IRI-template, so as to obtain what we call *inverse templates*. These can be applied to the IRI of an object obtained via a mapping, to obtain in a unique way the DB value(s) used to construct that IRI. When performing insertions of new objects over the ontology, we want to avoid introducing ambiguities due to the fact that the IRIs of such objects could be constructed in different ways from database values, hence giving rise to different combinations of database values to be inserted in the database. To obtain this, we make the assumption that also for the objects appearing in insertions requested by the user, there is a unique way to reverse their IRIs so as to obtain the database values to be used to propagate the insertion to the underlying data source. For instance, assume that A is a class name in the ontology and there is a single IRI template $t_A(x, y)$ used in mappings to generate IRIs of instances of A from DB values for x and y. Then, we assume that there are two inverse templates t_A^x and t_A^y that can be applied to an IRI to produce in a unique way a pair of values. Consider now an atom $A(c)$, where c is an IRI, inserted by the user in the ABox. When we apply t_A^x and t_A^y to c, we obtain a pair of values $a_1 = t_A^x(c)$ and $a_2 = t_A^y(c)$ such that $c = t_A(a_1, a_2)$, and moreover such pair is unique.

Considering the above assumptions, the key challenge in our work becomes that of translating the set of deletions/insertions derived from the repair of the KB w.r.t. a given ontology-based update, into source updates. This can be broken down into two distinct tasks, one for *deletion* and another one for *insertions*:

Deletion. Deleting an assertion (or a set of assertions) from an ABox in VKGs will naturally require first finding the set of source tuples from which that assertion (or set of assertions) has been generated using the VKG mapping. One of the first results we have achieved in that context is to introduce in VKGs the notion of *lineage*, which describes the set of source tuples from which a given assertion (or set of assertions) in the ABox is generated. We further this notion by also providing a definition for the *exclusive lineage*, which describes the set of source tuples that do not contribute to any other assertion than the one given in the ABox. We propose to compute such (exclusive) lineage by using the *maximum recovery*, a notion introduced in the data exchange setting in databases [37]. Such maximum recovery is a reverse mapping of the given VKG mapping, which is essentially a rewriting of each target atom in the ontology in terms of a query over the source schema. We provide a deletion algorithm that extracts from the lineage the set of source tuples whose deletion will entail the deletion of the given assertion (or set of given assertions in the ABox) with zero or minimum side effects (which, in this case, is the deletion of additional ABox facts). However, the complexity of such an algorithm is worst-case exponential in the size of the lineage, and we are interested in studying how it can be improved in meaningful cases.

Insertion. Unlike deletion, the insertion procedure for an assertion in the ABox is challenging because the assertion being inserted along with its lineage does not currently exist, and since VKG mappings are usually non-injective, rewriting the ABox insertion request (by means of maximum recovery) can result in multiple potential translations into insertions in the data source. Another challenge is related to the existence of existential quantifiers in the rewritten insertion requests, resulting from the variables that are projected out in the VKG mapping query. The choice of the right assignment of a DB value to such variables that will lead to the minimum possible side effects (which, in this case, is the insertion of additional ABox facts) remains a challenge. Towards a solution to this problem, we have proposed an algorithm that rewrites an ABox insertion request into a source update request over the underlying source data and finds a possible combination of assignments to the existential variables that minimizes the side effects in the ABox. Also this algorithm runs in worst-case exponential time in the size of the update request, hence we need to investigate possible optimizations. Moreover, we are still investigating whether it actually produces minimal side effects on the ABox.

In addition to addressing the challenges above, we are interested in studying how constraints in the source schema can influence the translation of KB updates into source updates. Our plan to provide a solution to **RQ2** is to develop a dialog-based system that requests from the user a minimal amount of missing information in order to not only disambiguate the requested update but also to ensure that our procedure causes a minimal side effect in the ABox. Finally, our end goal in this project is to make all the techniques we plan to develop ready-implementable so they can be adopted by state-of-the-art VKG tools like Ontop [7] and Mastro [38], which is the answer to **RQ3**.

5. Conclusions

In this research proposal, we have explored the foundational and practical problem of instance-level updates in VKGs, which accounts for changing the extensional data of the system by propagating to the source data a requested update over the ontology. The solution to this problem represents an important achievement in the context of ontology-based data access and integration and will enrich VKG systems with greater flexibility and consistency in managing large and complex datasets. The challenges related to this extension have been briefly described along with the limitation of the methods we have proposed so far in terms of complexity. In our subsequent results, we hope to provide solutions to the limitations we are currently facing and to the remaining research questions.

Acknowledgments

This work is under the supervision of Prof. Diego Calvanese and is supported by the Wallenberg AI, Autonomous Systems, and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

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] H. Katsuno, A. Mendelzon, On the difference between updating a knowledge base and revising it, in: Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR), 1991, pp. 387–394.

[11] T. Eiter, G. Gottlob, On the complexity of propositional knowledge base revision, updates and counterfactuals, Artificial Intelligence 57 (1992) 227–270.

[12] D. Calvanese, E. Kharlamov, W. Nutt, D. Zheleznyakov, Updating ABoxes in *DL-Lite*, in: Proc. of the 4th Alberto Mendelzon Int. Workshop on Foundations of Data Management (AMW), volume 619 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2010, pp. 3.1–3.12.

[13] G. De Giacomo, X. Oriol, R. Rosati, D. F. Savo, Instance-level update in DL-Lite ontologies through first-order rewriting, J. of Artificial Intelligence Research 70 (2021) 1335–1371. doi:10.1613/jair.1.12414.

[14] G. De Giacomo, M. Lenzerini, A. Poggi, R. Rosati, On instance-level update and erasure in description logic ontologies, J. of Logic and Computation 19 (2009) 745–770.

[15] M. Winslett, A model-based approach to updating databases with incomplete information, ACM Trans. on Database Systems 13 (1988) 167–196.

[16] H. Liu, C. Lutz, M. Miličić, F. Wolter, Foundations of instance level updates in expressive description logics, Artificial Intelligence 175 (2011) 2170–2197.

[17] H. Liu, C. Lutz, M. Milicic, F. Wolter, Updating description logic ABoxes, in: Proc. of the

10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR), 2006, pp. 46–56.
[18] G. De Giacomo, D. Lembo, X. Oriol, D. F. Savo, E. Teniente, Practical update management in ontology-based data access, in: The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16, Springer, 2017, pp. 225–242.
[19] U. Dayal, P. A. Bernstein, On the correct translation of update operations on relational views, ACM Transactions on Database Systems (TODS) 7 (1982) 381–416.
[20] U. Dayal, P. A. Bernstein, On the updatability of relational views., in: VLDB, volume 78, 1978, pp. 368–377.
[21] Y. Kotidis, D. Srivastava, Y. Velegrakis, Updates through views: A new hope, in: 22nd International Conference on Data Engineering (ICDE'06), IEEE, 2006, pp. 2–2.
[22] R. Langerak, View updates in relational databases with an independent scheme, ACM Transactions on Database Systems (TODS) 15 (1990) 40–66.
[23] J. A. Larson, A. P. Sheth, Updating relational views using knowledge at view definition and view update time, Information Systems 16 (1991) 145–168.
[24] Y. Masunaga, An intention-based approach to the updatability of views in relational databases, in: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, 2017, pp. 1–8.
[25] Y. Masunaga, Y. Nagata, T. Ishii, Extending the view updatability of relational databases from set semantics to bag semantics and its implementation on postgresql, in: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, 2018, pp. 1–8.
[26] A. M. Keller, Choosing a view update translator by dialog at view definition time, in: Proceedings of the 12th International Conference on Very Large Data Bases, 1986, pp. 467–474.
[27] postgresql, PostgreSQL, ???? URL: https://www.postgresql.org.
[28] K. Czarnecki, J. N. Foster, Z. Hu, R. Lämmel, A. Schürr, J. F. Terwilliger, Bidirectional transformations: A cross-discipline perspective: Grace meeting notes, state of the art, and outlook, in: Theory and Practice of Model Transformations: Second International Conference, ICMT 2009, Zurich, Switzerland, June 29-30, 2009. Proceedings 2, Springer, 2009, pp. 260–283.
[29] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, A. Schmitt, Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem, ACM Transactions on Programming Languages and Systems (TOPLAS) 29 (2007) 17–es.
[30] V.-D. Tran, H. Kato, Z. Hu, Programmable view update strategies on relations, arXiv preprint arXiv:1911.05921 (2019).
[31] A. Bohannon, B. C. Pierce, J. A. Vaughan, Relational lenses: a language for updatable views, in: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2006, pp. 338–347.
[32] F. Bancilhon, N. Spyratos, Update semantics of relational views, ACM Trans. on Database Systems 6 (1981) 557–575.
[33] S. S. Cosmadakis, C. H. Papadimitriou, Updates of relational views, J. of the ACM 31 (1984) 742–760. doi:10.1145/1634.1887.

[34] 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.

[35] R. E. Wandji, M. Šimkus, D. Calvanese, Towards techniques for updating virtual knowledge graphs, in: 17th International Rule Challenge and 7th Doctoral Consortium@ RuleM+ RR, RuleML+ RR-Companion 2023, Oslo, September 18-20, 2023, CEUR-WS, 2023.

[36] D. Calvanese, E. Kharlamov, W. Nutt, D. Zheleznyakov, Evolution of *DL-Lite* knowledge bases, in: Proc. of the 9th Int. Semantic Web Conf. (ISWC), volume 6496 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 112–128. doi:10.1007/978-3-642-17746-0_8.

[37] M. Arenas, J. Pérez, J. Reutter, C. Riveros, The language of plain SO-tgds: Composition, inversion and structural properties, Journal of Computer and System Sciences 79 (2013) 763–784. doi:10.1016/j.jcss.2013.01.002.

[38] C. Civili, M. Console, G. De Giacomo, D. Lembo, M. Lenzerini, L. Lepore, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, et al., Mastro studio: managing ontology-based data access applications, Proceedings of the VLDB Endowment 6 (2013) 1314–1317.