Towards Practical OBDA with Temporal Ontologies

(Position Paper)

Diego Calvanese, Elem Güzel Kalaycı$^{(\boxtimes)}$, Vladislav Ryzhikov,
and Guohui Xiao

KRDB Research Centre for Knowledge and Data, Free University
of Bozen-Bolzano, Bolzano, Italy
{calvanese,kalayci,ryzhikov,xiao}@inf.unibz.it

Abstract. The temporal dimension of data, which contains such important information as duration or sequence of events and is present in many applications of ontology-based data access (OBDA) concerned with logs or streams, is getting growing attention in the community. To give a proper treatment to the events occurring in the data from the ontological perspective, we assume in our approach that every concept is temporalized, i.e., has temporal validity time, and the ontology language expresses the constraints between validity times of concepts. In this paper we outline the state of art and the future challenges of our research. On the theoretical side, we are interested in enriching the ontology languages with the operators for constructing the temporal concepts that are expressive enough to capture the patterns required by industrial use-cases. On the practical side, we are interested in implementing the ontology-mediated query answering with temporalized concepts in the OBDA system Ontop and performing extensive evaluations using large amounts of real-world data.

Keywords: Ontology-based data access · Temporal logic · Description logic

1 Introduction and Motivation

Ontology-based data access (OBDA) [9,21], one of the most promising applications of Knowledge Representation in the Semantic Web area, exposes a high level conceptual layer in the form of an ontology on top of (potentially very large and heterogeneous) data sources. The conceptual view of the data, represented by an (OWL) ontology, models the domain of interest and hides the complex structure of the underlying data sources. In the ontology, classes and properties are mapped through a declarative specification into views over data expressed in terms of SQL queries. In the virtual approach, the OBDA system first rewrites end-user queries with respect to the ontology, then translates them into SQL queries, and finally delegates the query execution to the SQL engine over relational data sources. OBDA has a strong impact in both scientific and industrial communities.

M. Ortiz and S. Schlobach (Eds.): RR 2016, LNCS 9898, pp. 18–24, 2016.
DOI: 10.1007/978-3-319-45276-0_2

Research in OBDA has grown to maturity and OBDA has become a prominent direction in the development of the Semantic Web. The OWL 2 QL profile of the Web Ontology Language (OWL 2) based on *DL-Lite* [9], a lightweight DL family that enjoys a low complexity of reasoning, has been introduced by the W3C as a standard for OBDA. *Ontop*[1] is a state-of-the-art OBDA engine developed at the Free University of Bozen-Bolzano. Ontop is currently adopted as the core OBDA engine of the EU FP7 Optique project whose goal is to overcome the problem of end-user access to big data [11]. More recently, Ontop has also been integrated in the commercial graph database system *Stardog*[2] to provide support for SPARQL end-user queries.

In many applications data has a temporal dimension, which is important to consider (see, e.g., [13]). In such scenarios, it is reasonable to assume that the concepts of the conceptual OBDA layer have an associated temporal validity periods. If the data, for example, is the stream of wind speed measurements at weather stations, the concept $\mathsf{HurricaneForceWind}(x, t)$ can be associated to the data by means of a mapping that extracts the stations and time stamps, where the wind speed exceeded 118 km/h. An ontology designer can then use classical (atemporal) ontology constructors to define new concepts, e.g., “hurricane force wind is a wind”. Many studies develop this approach (see [4,7,8,12,14,19] and references therein) and extend the query language of conjunctive or SPARQL queries with the constructors to retrieve temporal information; e.g., “extract stations and timestamps, where hurricane force wind occurred and it also occurred one hour ago”. The latter pattern represents the definition of a hurricane (hurricane force wind lasting one hour or longer). The above mentioned approach, in spite of allowing to query for hurricanes, does not allow for defining a concept hurricane that would be very natural in the paradigm of OBDA.

To overcome the limitation of a temporal approach, other studies (see [1,3,5,15] and references therein) focused on using ontology languages with temporal constructors [2,17] in the setting of OBDA. One can define a hurricane as a new concept by means of temporal operators (e.g., as a conjunction $\mathsf{HurricaneForceWind} \wedge \mathbf{X}^- \ \mathsf{HurricaneForceWind}$), where $\mathbf{X}^-$ is a temporal operator “previous time”). As another example, the concept $\mathsf{Blizzard}$ can be defined as an occurrence of $\mathsf{Blizzard\ Condition}$ lasting for more than 3 hours, whereas $\mathsf{Blizzard\ Condition}$ is defined as simultaneous occurrences of $\mathsf{Strong\ Wind}$, $\mathsf{Low\ Visibility}$, and Snow (i.e., $\mathsf{BlizzardCondition} = \mathsf{Strong\ Wind} \wedge \mathsf{Low\ Visibility} \wedge \mathsf{Snow}$) [18].

The approach that considers atemporal ontologies only is less expressive. It has, however, the advantage that the complexity of the temporal query answering mostly coincides with the complexity of answering usual queries. Therefore, implementations for this setting can be with a reasonable effort reduced to atemporal query answering. With some notable exceptions [1], the complexity of reasoning grows significantly in the approach with temporal ontologies (as compared to reasoning in the underlying ontology languages) [17]. Therefore, it is more

---

[1] http://ontop.inf.unibz.it/.

[2] http://stardog.com/.

challenging to develop a practical query answering system for that setting. As we move towards this goal, we are aware that using more temporal constructors results in higher complexity. Thus, we attempt to allow only those that are necessary for practical use-cases.

The objectives of this ongoing research are: *(a)* to enrich the ontology languages with the operators for building the temporalized concepts that are expressive enough to capture the patterns required by industrial use-cases, *(b)* to implement ontology-mediated query answering with temporalized concepts in the OBDA system Ontop, and *(c)* to perform extensive evaluations using large amounts of real-world data.

As mentioned above, the direction *(a)* has been sufficiently studied. However, more investigation is needed there continuously with respect to new use-cases of temporal OBDA that are being discovered. Regarding the directions *(b)* and *(c)*, to the best of our knowledge, none of the available OBDA implementations take temporal ontologies into account. In this study our aim is to extend the OBDA techniques to support temporal reasoning and implement these techniques in the state-of-the-art framework Ontop.

In the following sections, we identify the research problems and challenges of this study and we propose our methodology. In Sect. 2 we discuss potential applications, in Sect. 3 we explain the main challenges in defining new languages for ontology, mapping, and querying by taking the trade-off between expressivity and efficiency into consideration. In Sect. 4 we analyze the challenges in implementation side of extending the existing system Ontop.

2 Applications and Use Cases

We have already discussed in Sect. 1 how weather concepts such as hurricane and blizzard can be defined using temporal ontologies. Those concepts hold for weather stations (assuming that the data is recorded at them) and time instants. We can then use a role (which can be mapped to an appropriate database) that connects a station with a town, a county, or a state it is located in. Then, one can define, e.g., a (temporal) concept for counties affected by hurricane as “counties which have some station located in them that recorded a hurricane”. More interestingly, we can define a concept for cyclone as “states which have four stations located in them such that one of them records southern wind, one northern, one western, and one eastern”. (Note that if we have data describing relative position of a station w.r.t. other stations, we can define a cyclone even more precisely.) Another interesting example is a concept for showery counties defined as “counties that have a station that records no precipitation and a station that records precipitation but recorded no precipitation 20 min ago”. Use of such and other similar concepts makes sense to detect development of weather in historical or streaming data. A large database of records of weather stations across the US is available through National Weather Service’s Mesonet program[3]. It can conveniently be used as a data set to evaluate the performance of our approach.

---

[3] http://mesowest.org.

Another important application of temporal ontologies is analysis of log data of mechanical or electronic devices. For example, if a speed (measured in RpM) of a working engine is continuously recorded in a database, we can extract by means of the mappings such temporal concepts as idle speed, intermediate speed, and running speed. A concept smooth shutdown can then be defined as "idle speed preceded for 15 min by intermediate speed, which is, in its turn, preceded by running speed". On the other hand, rapid shutdown can be defined as "idle speed preceded by occurrence of running speed within 5 min". Another interesting example is a concept consistent vibration defined as "high vibration occurring every 10 s for 1 min". Clearly, using temporal ontologies to conceptually define abnormal situations in performance of devices is a novel and relevant approach to monitoring. We are collaborating with a major industrial company to obtain such data. This company runs several data centers for monitoring thousands of devices related to power generation, including gas and steam turbines, compressors, and generators. Each device is monitored by many sensors of different kinds. These sensors have generated terabytes of data so far. We aim to observe the performance and the scalability of our system over these large amount of real data in collaboration with the researchers in this company.

3 Methodological and Theoretical Challenges

Initially, the most important question to answer is what are the appropriate temporal languages for expressing/formulating ontologies, mappings, and queries in terms of expressivity and efficiency. For both the ontology and the query language level, we have to investigate to which degree the recently proposed temporal ontology languages and query languages satisfy our needs. Below we consider potential challenges in these three areas.

Temporal Extension of the Ontology Language. The first candidate for the role of an ontology language is a Linear Temporal Logic-based Description Logic (DL) proposed in [1], which was shown to have low data complexity (AC^0) for some important fragments. However there are two main reasons that make this logic not perfectly well suited to capture our requirements. On one hand, this logic uses ABoxes with concept assertions of the form $A(a, n)$, where a is an object name and n is a natural number representing a time point. It is often hard to adapt the real-world scenarios to this setting, as neither the timestamps of data records feature fixed periodicity, nor a reasonable atomicity of time in a data source is known a priori. On the other hand, this logic does not provide an explicit way to express metric constraints for temporal concepts (e.g., "hurricane is a strong wind continuing for at least 1 h"). We can overcome the first drawback by using a Halpen-Shoham Interval Logic-based DL proposed in [3,15], where the ABox concept assertions are assumed to be of the form $A(a, n_1, n_2)$ with n_1, n_2 real or natural numbers indicating a validity interval. This logic was shown be tractable in data complexity too, however, it is even less expressive in terms of the metric constraints, and does not overcome the second drawback. We believe that using ontology languages based on Metric Temporal Logics (MTL) [16] will be needed in our approach. Nothing is known yet neither about the complexity of MTL fragments underlying our temporal constraints, nor about the complexity of reasoning in MTL-based ontology languages.

Temporal Mapping Language. The mapping languages for temporal concepts over log or stream databases is a novel problem that is fundamental to our OBDA approach. In general, if one considers ABoxes with concept assertions of the shape $A(a, n)$, the solution is seemingly easy: a mapping should be an SQL query returning pairs of object names and time stamps. In real-world situations, however, the data may be noisy and, e.g., to detect whether high temperature occurred at a moment of time n, one needs to look at the value of the temperature at several surrounding time moments and take the average. Other approximation functions, such as exponential average, should be considered too, as they are known to be more appropriate for processing certain types of signals. Our aim is to consider both the approximations computable in SQL, as well as other languages for data access.

As mentioned above, in our approach it is more advantageous to consider concept statement of the form $A(a, n_1, n_2)$. Therefore, an SQL query of a mapping should return a pair of time moments, between which, e.g., high temperature occurred. In simple scenarios, where data records are complete for the time stamps, one can use the `LEAD` function of SQL to compute the n_2 to be "paired" with n_1. In the case when the database is missing values in some fields for some time stamps, computing the temporal concepts may require more elaborate SQL queries ignoring or taking into account (depending on assumptions about a data source) time moments with missing signal values.

Temporal Query Language. Query languages for ontologies over temporal data have been widely considered [4,7,12,14,19], in particular, with SPARQL-inspired syntax [1,20]. In our approach we intend to keep the end-user query language simple by moving the temporal patterns into the ontology level. One important feature that we plan to enable in the queries is the direct use of temporal constants of various granularity, such as 2016, May 2016, afternoon May 5 2016, May 5 2016 11:24, etc. An end-user then can formulate in a natural way queries such as "locations where a blizzard occurred in May 2016", "counties and days when it rained in May 2016", or "engines and minutes where/when consistent vibration occurred in the past hour". We plan to investigate the languages that allow to express such queries.

4 Implementation Challenges

Implementing new forms of mappings and temporal operators of ontologies in Ontop framework will require substantial work. The most reasonable method to store and process the information on validity times of a concept seems to be using the tables (possibly, virtual) representing intervals. In terms of cost efficiency, one of the most challenging tasks in translating temporal operators into SQL queries is computing *coalescing* [6], i.e., the largest time intervals where a concept holds.

For example, in order to compute intervals where hurricane holds, we need to consider a coalescing of the time intervals where hurricane force wind holds. There are various approaches to computing coalescings (e.g., through transitive closure), we are going to investigate what algorithm suits best to our setting.

The other challenging task in translating into SQL is to provide a cost efficient way of performing temporal joins [10]. In the case of the concept for blizzard, in order to get intervals where it holds, one should compute intersections of the intervals where strong wind, low visibility, and snow hold. Given a pair of intervals, one has to consider various relative positions of the first interval w.r.t. the second, in order to find a pair of numbers that represents the intersection. Therefore, a straightforward implementation of the intersection in SQL will result in multiple unions. On the other hand, the `CASE` operator can help handle those conditions and prevent from making unions, which reduce the performance when a number of joined tables is large. We will also investigate other methods to decrease the temporal join cost by employing SQL cursors. The idea behind using cursors is to perform the temporal join in a merge-sort fashion over the tables that are ordered by starting point of intervals. This approach enables one to apply the temporal join by doing just one iteration of scan over each table that is joined. The drawback of it, however, is that it requires an additional sort step before applying the temporal join.

Acknowledgements. This paper is supported by the EU under the large-scale integrating project (IP) Optique (*Scalable End-user Access to Big Data*), grant agreement n. FP7-318338.

References

1. Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: First-order rewritability of temporal ontology-mediated queries. In: Proceedings of IJCAI 2015, pp. 2706–2712. AAAI Press (2015)
2. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: A cookbook for temporal conceptual data modelling with description logics. ACM Trans. Comput. Log. **15**(3), 25:1–25:50 (2014)
3. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Tractable interval temporal propositional and description logics. In: Proceedings of AAAI 2015, pp. 1417–1423. AAAI Press (2015)
4. Baader, F., Borgwardt, S., Lippmann, M.: Temporalizing ontology-based data access. In: Bonacina, M.P. (ed.) CADE 2013. LNCS, vol. 7898, pp. 330–344. Springer, Heidelberg (2013)
5. Basulto, V.G., Jung, J., Kontchakov, R.: Temporalized EL ontologies for accessing temporal data: complexity of atomic queries. In: Proceedings of 25th International Joint Conference on Artificial Intelligence (IJCAI-2016). AAAI Press (2016)
6. Böhlen, M.H., Snodgrass, R.T., Soo, M.D.: Coalescing in temporal databases. In: Proceedings of 22th International Conference on Very Large Data Bases (VLDB 1996), pp. 180–191 (1996)
7. Borgwardt, S., Lippmann, M., Thost, V.: Temporal query answering in the description logic DL-Lite. In: Proceedings of FroCoS 2013, pp. 165–180 (2013)

8. Borgwardt, S., Lippmann, M., Thost, V.: Temporalizing rewritable query languages over knowledge bases. J. Web Semant. **33**, 50–70 (2015). Ontology-Based Data Access
9. Calvanese, D., Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. J. Autom. Reason. **39**(3), 385–429 (2007)
10. Gao, D., Jensen, C.S., Snodgrass, R.T., Soo, M.D.: Join operations in temporal databases. VLDB J. **14**(1), 2–29 (2005)
11. Giese, M., Soylu, A., Vega-Gorgojo, G., Waaler, A., Haase, P., Jiménez-Ruiz, E., Lanti, D., Rezk, M., Xiao, G., Özçep, Ö.L., Rosati, R.: Optique - zooming in on big data access. IEEE Comput. **48**(3), 60–67 (2015)
12. Gutiérrez-Basulto, V., Klarman, S.: Towards a unifying approach to representing and querying temporal data in description logics. In: Krötzsch, M., Straccia, U. (eds.) RR 2012. LNCS, vol. 7497, pp. 90–105. Springer, Heidelberg (2012)
13. Kharlamov, E., et al.: How semantic technologies can enhance data access at siemens energy. In: Mika, P., et al. (eds.) ISWC 2014, Part I. LNCS, vol. 8796, pp. 601–619. Springer, Heidelberg (2014)
14. Klarman, S., Meyer, T.: Querying temporal databases via OWL 2 QL. In: Kontchakov, R., Mugnier, M.-L. (eds.) RR 2014. LNCS, vol. 8741, pp. 92–107. Springer, Heidelberg (2014)
15. Kontchakov, R., Pandolfo, L., Pulina, L., Ryzhikov, V., Zakharyaschev, M.: Temporal and spatial OBDA with many-dimensional Halpern-Shoham logic. In: Proceedings of 25th International Joint Conference on Artificial Intelligence (IJCAI-2016). AAAI Press (2016)
16. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Syst. **2**(4), 255–299 (1990)
17. Lutz, C., Wolter, F., Zakharyaschev, M.: Temporal description logics: a survey. In: Proceedings of TIME 2008, pp. 3–14. IEEE Computer Society (2008)
18. National Oceanic and Atmospheric Administration: National weather service glossary (2009). http://w1.weather.gov/glossary/index.php?letter=b. Accessed 03 Jun 2016
19. Özçep, Ö., Möller, R., Neuenstadt, C., Zheleznyakov, C., Kharlamov, E.: OBDA with temporal and stream-oriented queries: optimization techniques, D5.2. Technical report, FP7-318338, EU (2014)
20. Özçep, Ö.L., Möller, R., Neuenstadt, C.: A stream-temporal query language for ontology based data access. In: Lutz, C., Thielscher, M. (eds.) KI 2014. LNCS, vol. 8736, pp. 183–194. Springer, Heidelberg (2014)
21. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 133–173. Springer, Heidelberg (2008)