The Virtual Knowledge Graph System Ontop (Extended Abstract) ⋆

Guohui Xiao[1,2], Davide Lanti[1], Roman Kontchakov[3], Sarah Komla-Ebri[2], Elem Güzel-Kalaycı[4], Linfang Ding[1], Julien Corman[1], Benjamin Cogrel[2], Diego Calvanese[1,2,5], and Elena Botoeva[6]

[1] Free University of Bozen-Bolzano, Italy *lastname*@inf.unibz.it
[2] Ontopic s.r.l., Bolzano, Italy *firstname.lastname*@ontopic.biz
[3] Birkbeck, University of London, UK roman@dcs.bbk.ac.uk
[4] Virtual Vehicle Research GmbH, Graz, Austria elem.guezelkalayci@v2c2.at
[5] Umeå University, Sweden diego.calvanese@umu.se
[6] Imperial College London, UK e.botoeva@imperial.ac.uk

The full paper is published at the Resource Track of ISWC 2020 [14]

VKG. The Virtual Knowledge Graph (VKG) approach, also known in the literature as Ontology-Based Data Access (OBDA) [8,12], has become a popular paradigm for accessing and integrating data sources [13]. In such approach, the data sources, which are normally relational databases, are *virtualized* through a mapping and an ontology, and presented as a unified *knowledge graph*, which can be queried by end-users through a vocabulary they are familiar with. At query time, a VKG system translates user queries over the ontology to SQL queries over the database. This approach frees end-users from the low-level details of data organization, so that they can concentrate on their high-level tasks. As it is gaining more importance, the VKG paradigm has been implemented in several systems [1,2,9,11] and adopted in a wide range of use cases. Here, we present the latest major release, *Ontop* v4, of a popular VKG system.

Ontop v1. The development of *Ontop* has spanned the past decade. Developing such a system is highly non-trivial and requires both a theoretical investigation of the semantics and strong engineering efforts to implement all the required features. *Ontop* started in 2009, only one year after the first version of SPARQL had been standardized, while OWL 2 QL [6] and R2RML [4] appeared 3 years later, in 2012. At that time, the VKG research focused on union of conjunctive queries (UCQs) as a query language. With this target, *Ontop* v1 relied on non-recursive Datalog as its core data structure [10] because it perfectly fit the UCQ-based setting. The development of *Ontop* was boosted by the EU FP7 project Optique (2013–2016), during which the compliance with the relevant W3C recommendations became a priority, and significant progress was made in this direction. The last release of *Ontop* v1 was v1.18 in 2016 [1].

New challenges. A natural requirement that emerged during the Optique project were aggregates introduced in SPARQL 1.1 [5]. The *Ontop* development team spent a major effort, internally called *Ontop* v2, on implementing this query language feature. However, it became exceedingly clear that the Datalog representation was not well suited

for this implementation. Some prototypes of *Ontop* v2 were used in the Optique project for internal purposes, but never reached the level of a public release. During this development, as *Ontop* moved towards supporting the W3C recommendations for SPARQL and R2RML, we have identified the following new challenges:

- In contrast to the usual DL encoding with unary and binary predicates for classes and properties, in SPARQL triple pattern variables can occur in positions of class and property names, which means that there are effectively only two 'predicates': `triple` for triples in the RDF dataset default graph, and `quad` for named graphs.
- More importantly, SPARQL is based on a rich algebra, which goes beyond the expressivity of CQs. Non-monotonic features like OPTIONAL and MINUS, cardinality-sensitive query modifiers (DISTINCT) and aggregation (GROUP BY with functions such as SUM, AVG, COUNT) are difficult to model even in extensions of Datalog.
- Even without SPARQL aggregation, cardinalities have to be treated carefully: the SQL queries in a mapping produce *bags* (multisets) of tuples, but their induced RDF graphs contain no duplicates and thus are *sets* of triples; however, when a SPARQL query is evaluated, it results in a bag of solutions mappings.

These challenges turned out to be difficult to tackle in the Datalog setting.

Ontop v4. To address the challenges posed by aggregation, and others that had emerged in the meantime, we started to investigate an alternative core data structure. The outcome has been what we call *intermediate query* (IQ), an algebra-based data structure that unifies both SPARQL and relational algebra. Using IQ, we reimplemented most of the *Ontop* code base. After two beta releases in 2017 and 2018, we released the stable version of *Ontop* v3 in 2019. Following *Ontop* v3, the development focussed on improving compliance and adding several major features. In particular, aggregates have now been supported since *Ontop* v4-beta-1, released in late 2019. The stable version of *Ontop* v4 was released in July 2020. Currently, anyone wishing to test *Ontop* v4 can obtain the latest source code and precompiled binaries from Github[1]. The documentation, including tutorials, is provided at the official website[2].

Evaluation. *Ontop* v4 has greatly improved its compliance with relevant W3C recommendations and provides good performance in query answering. It supports almost all the features of SPARQL 1.1, R2RML, OWL 2 QL, and SPARQL entailment regime, and the SPARQL 1.1 HTTP Protocol. In particular, in Table 1, we present a summary of *Ontop* v4 compliance with SPARQL 1.1, where rows correspond to sections of the WC3 recommendation. Most of the features are supported, but some are unsupported or only partially supported. Note that most of the missing SPARQL functions (Section 17.4) are not so challenging to implement but require a considerable engineering effort to carefully define their translations into SQL. We will continue the process of implementing them gradually and track the progress in a dedicated issue[3].

Recently, two independent evaluations [3,7] of VKG systems have confirmed the robust performance of *Ontop*. When considering all the perspectives, like usability, completeness, and soundness, *Ontop* clearly stands out among the open-source systems.

[1] `https://github.com/ontop/ontop`

[2] `https://ontop-vkg.org/`

[3] `https://github.com/ontop/ontop/issues/346`

Table 1. SPARQL Compliance: unsupported features are ~~crossed out~~.

Section in SPARQL 1.1 [5]	Features	Coverage
5–7. Graph Patterns, etc.	`BGP`, `FILTER`, `OPTIONAL`, `UNION`	4/4
8. Negation	`MINUS`, ~~`FILTER [NOT] EXISTS`~~	1/2
9. Property Paths	~~PredicatePath~~, ~~InversePath~~, ~~ZeroOrMorePath~~, ...	0
10. Assignment	`BIND`, `VALUES`	2/2
11. Aggregates	`COUNT`, `SUM`, `MIN`, `MAX`, `AVG`, `GROUP_CONCAT`, `SAMPLE`	6/6
12. Subqueries	Subqueries	1/1
13. RDF Dataset	`GRAPH`, ~~`FROM [NAMED]`~~	1/2
14. Basic Federated Query	~~`SERVICE`~~	0
15. Solution Seqs. & Mods.	`ORDER BY`, `SELECT`, `DISTINCT`, `REDUCED`, `OFFSET`, `LIMIT`	6/6
16. Query Forms	`SELECT`, `CONSTRUCT`, `ASK`, `DESCRIBE`	4/4
17.4.1. Functional Forms	`BOUND`, ~~`IF`~~, `COALESCE`, ~~`EXISTS`~~, ~~`NOT EXISTS`~~, `||`, `&&`, `=`, `sameTerm`, ~~`IN`~~, ~~`NOT IN`~~	6/11
17.4.2. Fns. on RDF Terms	`isIRI`, `isBlank`, `isLiteral`, `isNumeric`, `str`, `lang`, `datatype`, ~~`IRI`~~, ~~`BNODE`~~, ~~`STRDT`~~, ~~`STRLANG`~~, `UUID`, `STRUUID`	9/13
17.4.3. Fns. on Strings	`STRLEN`, `SUBSTR`, `UCASE`, `LCASE`, `STRSTARTS`, `STRENDS`, `CONTAINS`, `STRBEFORE`, `STRAFTER`, `ENCODE_FOR_URI`, `CONCAT`, `langMatches`, `REGEX`, `REPLACE`	14/14
17.4.4. Fns. on Numerics	`abs`, `round`, `ceil`, `floor`, `RAND`	5/5
17.4.5. Fns. on Dates&Times	`now`, `year`, `month`, `day`, `hours`, `minutes`, `seconds`, ~~`timezone`~~, `tz`	8/9
17.4.6. Hash Functions	`MD5`, `SHA1`, `SHA256`, `SHA384`, `SHA512`	5/5
17.5. XPath Constructor Fns.	~~casting~~	0
17.6. Extensible Value Testing	~~user defined functions~~	0

Community and Adoption. *Ontop* v4 is the result of an active developer community. It has been downloaded more than 30K times from Sourceforge. In addition to the research groups, *Ontop* is also backed by a commercial company, Ontopic s.r.l., born in April 2019. *Ontop* has been adopted in many academic and industrial use cases. However, due to its liberal Apache 2 license, it is essentially impossible to obtain a complete picture of all use cases and adoptions. Nevertheless, a few significant use cases have been summarized in a recent survey paper [13]. Finally, we mention two recent commercial deployments of *Ontop*: *UNiCS* (`http://unics.cloud/`) is an open data platform for research and innovation, and *ODH-VKG* (`https://sparql.opendatahub.bz.it/`) is a project publishing South Tyrolean tourism data as a Knowledge Graph.

References

1. D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. Ontop: answering SPARQL queries over relational databases. *SWJ*, 8(3):471–487, 2017.
2. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, and D. F. Savo. The MASTRO system for ontology-based data access. *SWJ*, 2(1):43–53, 2011.
3. M. Chaloupka and M. Necasky. Using Berlin SPARQL benchmark to evaluate relational database virtual SPARQL endpoints. Submitted to SWJ, 2020.
4. S. Das, S. Sundara, and R. Cyganiak. R2RML: RDB to RDF mapping language. W3C recommendation, W3C, 2012.
5. S. Harris, A. Seaborne, and E. Prud'hommeaux. SPARQL 1.1 query language. W3C recommendation, W3C, 2013.
6. B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language: Profiles. W3C Recommendation, W3C, 2012.
7. M. Namici and G. De Giacomo. Comparing query answering in OBDA tools over W3C-compliant specifications. In *Proc. DL*, volume 2211. CEUR-WS.org, 2018.
8. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Sem.*, 10:133–173, 2008.
9. F. Priyatna, Ó. Corcho, and J. F. Sequeda. Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph. In *WWW*, pages 479–490, 2014.
10. M. Rodriguez-Muro and M. Rezk. Efficient SPARQL-to-SQL with R2RML mappings. *J. Web Sem.*, 33:141–169, 2015.
11. J. F. Sequeda, M. Arenas, and D. P. Miranker. Ontology-based data access using views. In *Proc. RR*, volume 7497 of *LNCS*, pages 262–265. Springer, 2012.
12. G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyaschev. Ontology-based data access: A survey. In *Proc. IJCAI*, pages 5511–5519, 2018.
13. G. Xiao, L. Ding, B. Cogrel, and D. Calvanese. Virtual knowledge graphs: An overview of systems and use cases. *Data Intelligence*, 1:201–223, 2019.
14. G. Xiao, D. Lanti, R. Kontchakov, S. Komla-Ebri, E. Güzel-Kalayci, L. Ding, J. Corman, B. Cogrel, D. Calvanese, and E. Botoeva. The virtual knowledge graph system ontop. In *ISWC*, 2020.