Monolith: an OBDM and Knowledge Graph Management Platform

Lorenzo Lepore[a,b], Manuel Namici[a,b], Giacomo Ronconi[a,b], Marco Ruzzi[a,b], Valerio Santarelli[a,b], Domenico Fabio Savo[c,b]

(a) Sapienza Università di Roma
⟨lastname⟩@diag.uniroma1.it

(b) OBDA Systems
⟨lastname⟩@obdasystems.com

(c) Università degli Studi di Bergamo
domenicofabio.savo@unibg.it

Abstract. In this demo we present MONOLITH, a novel application that combines Ontology-based Data Management (OBDM) capabilities with a Knowledge Graph platform. The idea of the system is to provide an integrated environment offering OBDM services such as ontology inspection, query answering, data quality checking, and to couple these services with a Knowledge Graph IDE. Users can access and explore enterprise data through the ontology and build Knowledge Graphs from them, by exploiting the underlying OBDM system. MONOLITH allows to query the Knowledge Graphs through a SPARQL endpoint, to integrate external resources such as Linked Open Datasets, and export the Knowledge Graphs for third-party analytics.[1]

1 Introduction

Ontology Based Data Management (OBDM) [10] is a paradigm for data integration and governance, which enables accessing existing enterprise data sources by means of a comprehensible and semantically rich representation of the application domain, expressed by an ontology. OBDM relies on a three-tier architecture: the ontology, the data sources, and the mappings, which declaratively link the ontology predicates to the data in the sources. In recent years, the enterprise and industry world have shown a constantly growing interest in OBDM and its potential deep impacts on ITC scenarios are widely recognized [1, 6, 8].

Another rising trend in enterprise data management are Knowledge Graphs (KGs) [7]. A KG is a map of the data that is available across an organization, that uses a graph structure (nodes and edges) to highlight the relationships that exist between the data. These links can be defined in terms of the domain ontology, providing semantics and meaning to the relationships. The graph model is extremely flexible, as it can be applied easily to a wide range of use cases, and expandable, making it easy to integrate new data sources. Also, it abstracts from the underlying application data stores, meaning that is provides a potential means to abandon the dreaded data silo approach.

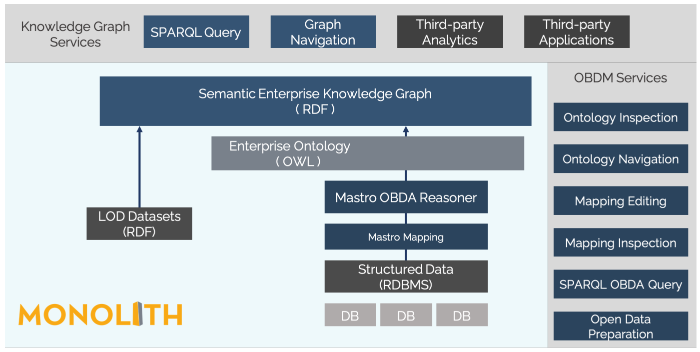

Fig. 1. The MONOLITH system.

In this demo we present MONOLITH, a novel application developed by OBDA Systems, a spinoff of Sapienza University of Rome, that combines OBDM capabilities with a semantic KG platform. The idea is to provide a system which offers OBDM services such as ontology and mapping inspection, query answering, data quality controls with respect to the ontology rules, and couples them with a KG IDE. Through semantic data access provided by the MASTRO OBDM reasoner [3], MONOLITH allows to construct RDF datasets and build KGs from them. Then, it allows to interrogate the KGs through a SPARQL endpoint, integrate external resources such as Linked Open Data from the semantic web, and export the KGs for third-party analytics.

2 The Monolith System: architecture and features

In this section we highlight the architecture and main features of MONOLITH. As shown in Figure 1, the functionalities provided by MONOLITH can be logically split into two different macro-areas: one dedicated to managing OWL ontologies and providing OBDM services through MASTRO, exploiting the mappings between ontology and database; the other to managing KGs and providing services over them. These two sections are indeed linked together, allowing to build the KGs through semantic data access, from the results of the ontology queries computed by MASTRO.

In the *Ontology* section of the system, users can upload ontologies in OWL 2 and GRAPHOL [9] format. GRAPHOL is a visual ontology language for OWL 2, whose distinguishing features are that it allows for drawing ontologies completely diagrammatically, and that it is equivalent to OWL 2. These ontologies can be navigated by inspecting the documentation for each entity, in both a wiki-like textual form or through a custom GRAPHOL viewer, both of which show the entity metadata, and the OWL axioms they are involved in. Furthermore, users can create, inspect and edit the mappings, providing for each mapping a datasource connection, and run queries over the ontologies through an OBDM SPARQL endpoint, producing tabular or RDF datasets. A Query Catalog allows to store the more significant ontology queries for future re-use. In Figure 2 we show a screenshot of MONOLITH's OBDM SPARQL endpoint.

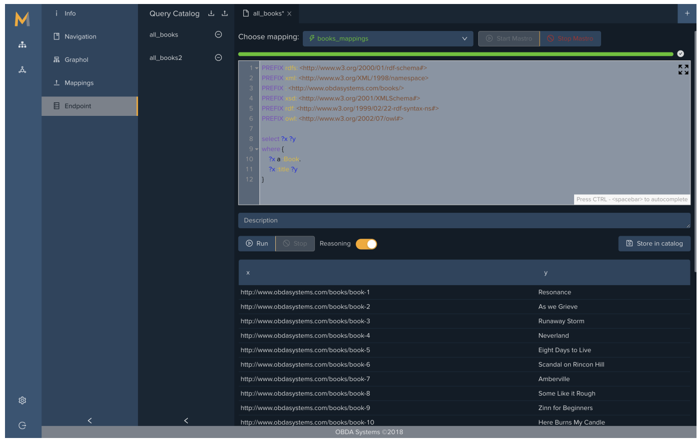

Fig. 2. The ontology SPARQL endpoint in the MONOLITH system.

Such OBDM query answering services are provided by the MASTRO ontology reasoner, which supports data access through ontologies specified in logics of the *DL-Lite* family [4]. In MASTRO, the data source layer is seen as a relational database, and in those cases in which several, possibly non-relational, sources need to be accessed, MASTRO can be coupled with a relational data federation tool, which wraps sources and represents them as a single (virtual) federated relational database. Finally, the mapping layer is constituted by a set views over the database and mapping assertions [10] which associate ontology elements with such views. By virtue of these design choices, OBDM services, such as query answering, are realized in MASTRO through a very efficient technique that reduces them, via query rewriting [5], to standard SQL query evaluation. In essence, the user query is reformulated with respect to the ontology and to the mappings, in such a way that a new query, which encodes this reasoning and that can be directly executed on the relational data sources, is produced.

In the *Knowledge Graph* section, users can create, import data into, and inspect their semantic KGs. The RDF datasets that can be imported into a KG can be either the results of MASTRO SPARQL queries over the ontology, as previously mentioned, or external datasets, thus allowing the possibility to integrate the data from the enterprise data sources with Linked Open Data from other organizations or from the Semantic Web community. For each dataset imported into the KG, the user can choose the named graph, or context, in the KG into which the RDF triples will be imported. Exploration of the data in each KG is provided by both a SPARQL endpoint over the KG, or by a catalog of all the classes in each KG. For each instance of such classes, the system provides a description, the list of RDF triples it is involved in, and the possibility to download such triples in RDF format, choosing the desired RDF syntax.

From a technical standpoint, MONOLITH is developed through the ReactJS Javascript library, and communicates with the MASTRO Java server for all OBDM services through a RESTful web service interface. Management of the KGs is delegated to the Apache Jena framework for Semantic Web and Linked Data applications, and, specifically, RDF file storage and query is done through the TDB component of Jena.

3 Application scenarios and Demo Session Overview

We demonstrate MONOLITH through two different specifications. Firstly, the ACI ontology and RDF datasets, developed in a joint project by Sapienza University of Rome, the Automobile Club d'Italia (ACI), OBDA Systems, and OKKAM, a spinoff of the University of Trento. The project goals were the definition of an ontology of ACI's Public Vehicle Register (PRA) and car tax domains, the development of an OBDM system to access the data through such an ontology, and the creation of a web portal[2] for the publication of ACI's car parc data in *Linked Open* format. The second specification is the Movie Ontology [2], which provides a vocabulary to semantically describe movie related concepts. For this ontology, we have developed a set of mappings for the MASTRO system to allow for semantic data access through OBDM.

During the demo, attendees will interact with the MONOLITH system in the above scenarios. They will exploit the OBDM services to inspect the ontology and mapping specifications and to define queries through the ontology, and will be able to build and then interrogate semantic KGs from the data extracted from such queries.

References

1. N. Antonioli, F. Castanò, C. Civili, S. Coletta, S. Grossi, D. Lembo, M. Lenzerini, A. Poggi, D. F. Savo, and E. Virardi. Ontology-based data access: The experience at the italian department of treasury. In *Proc. of CAISE 2013*, volume 1017 of *CEUR Workshop Proceedings*, pages 9–16, 2013.
2. A. Bouza. Mo - the movie ontology, 2010. [Online; 26. Jan. 2010].
3. 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. *Semantic Web J.*, 2(1):43–53, 2011.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
5. F. Di Pinto, D. Lembo, M. Lenzerini, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, and D. F. Savo. Optimizing query rewriting in ontology-based data access. In *Proc. of EDBT 2013*, pages 561–572. ACM Press, 2013.
6. M. Giese, A. Soylu, G. Vega-Gorgojo, A. Waaler, P. Haase, E. Jiménez-Ruiz, D. Lanti, M. Rezk, G. Xiao, Ö. L. Özçep, and R. Rosati. Optique: Zooming in on big data. *IEEE Computer*, 48(3):60–67, 2015.
7. J. M. Gómez-Pérez, J. Z. Pan, G. Vetere, and H. Wu. Enterprise knowledge graph: An introduction. In *Exploiting Linked Data and Knowledge Graphs in Large Organisations*, pages 1–14. Springer, 2017.
8. E. Kharlamov, D. Hovland, M. G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, D. Lanti, M. Rezk, D. Zheleznyakov, M. Giese, H. Lie, Y. E. Ioannidis, Y. Kotidis, M. Koubarakis, and A. Waaler. Ontology based data access in statoil. *J. Web Semant.*, 44:3–36, 2017.
9. D. Lembo, D. Pantaleone, V. Santarelli, and D. F. Savo. Easy OWL drawing with the graphol visual ontology language. In *Proc. of KR*, pages 573–576. AAAI Press, 2016.
10. M. Lenzerini. Managing data through the lens of an ontology. *AI Magazine*, 39(2):65–74, 2018.

[2] `http://lod.aci.it`