

How to Stay Ontop of Your Data: Databases, Ontologies and More

Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Davide Lanti, Martin Rezk, and
Guohui Xiao

Faculty of Computer Science, Free University of Bozen-Bolzano, Italy

Abstract. *Ontop* is an Ontology Based Data Access system allowing users to access a relational database through a conceptual layer provided by an ontology. In this demo, we use the recently developed NPD benchmark (+4 billion triples) to demonstrate the features of *Ontop*. First we use *Ontop* as a SPARQL end-point to load the ontology and mappings, and answer SPARQL queries. Then, we will show how to use *Ontop* to check inconsistencies and exploit SWRL ontologies.

Keywords: *Ontop* framework, OBDA, SPARQL, R2RML, OWL, OWL 2 QL

1 Introduction

As databases grow in volume and complexity, finding and interpreting the information stored in them becomes a challenging task. As a result, there has been a growing trend to use a conceptual layer containing a vocabulary that is familiar to the users, to capture the domain knowledge and hide the complexity of data storage. This gives rise to the setting of *Ontology-Based Data Access* (OBDA) [2], illustrated in Figure 1. In an nutshell, in an OBDA setting users access data through a conceptual layer, which provides a convenient query vocabulary. Usually the conceptual layer is expressed in the form of an RDF(S) [7], SWRL [10], or OWL 2 [9] ontology, and the data is stored in relational databases. Terms in the conceptual layer (classes and properties in the ontology) are populated with elements from the data sources using mappings. Mappings define triples (subject, property, object) out of SQL queries. In the virtual approach these triples do not need to be materialized and are accessible at query time using SPARQL. Alternatively these triples can be materialized and loaded into a triplestore, but we will not explore this approach here.

In this demo, we illustrate *Ontop* [5], which is a platform that implements the OBDA paradigm, by allowing users to query relational databases as *Virtual RDF Graphs* using SPARQL [4], through a domain ontology expressed in the OWL 2 QL fragment [?] of the OWL 2 ontology language.

This demo, unlike its predecessors [8,1], will challenge *Ontop* with a real world scenario: the NPD Benchmark [6]¹. This benchmark contains thousands of axioms and mappings, and its dataset exposes several billion triples. In addition we will show the

¹ The benchmark is based on the commercially used dataset Norwegian Petroleum Directorate FactPages, available at <http://factpages.npd.no/factpages/>.

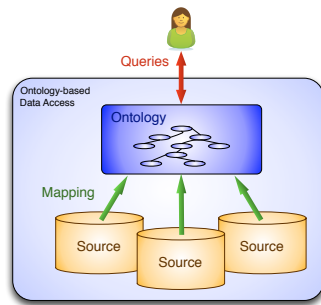
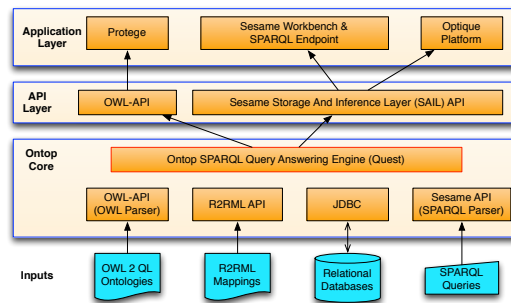


Fig. 1: OBDA Paradigm

Fig. 2: Architecture of *Ontop* System

new extensions of the system: mapping language (new native syntax, and support for the W3C standard R2RML), query language (SPARQL 1.1), ontology language (SWRL), optimization techniques, accessibility (SPARQL end-point), consistency check, among others. Attendants will learn how to build an OBDA system from scratch using *Ontop*, and which are the benefits and the limitations of this approach. The files used for the demonstration are available online ².

2 *Ontop*

Ontop is an open-source³ project released under the Apache license, and developed at the Free University of Bozen-Bolzano⁴. The *Ontop* system answers SPARQL queries by rewriting them into SQL queries over the underlying database. Its architecture is depicted in Figure 2. To the best of our knowledge, *Ontop* is currently the only OBDA system supporting all the relevant W3C standards (OWL 2 QL, SWRL, R2RML, SPARQL) and all major relational databases (PostgreSQL, MySQL, H2, DB2, ORACLE, HSQL, and MS SQL Server). *Ontop* can be used as: (i) a *plugin for Protégé*, which provides a graphical interface for mapping editing, SPARQL query execution, and inconsistency checking, (ii) a *SPARQL end-point* through Sesame Workbench, and (iii) a *Java library*, which implements both the OWL API and the Sesame API interfaces, available as Maven dependencies. In addition, *Ontop* acts as the core query transformation component of the Optique platform [3].

In [8] we showed how to use *Ontop* in Protégé over a small dataset. In this demo we will focus on how to use *Ontop* as a SPARQL end-point over a real world dataset.

3 A Demo of *Ontop* over NPD

NPD⁵ is a governmental organization whose main objective is to contribute to maximize the value that society can obtain from the oil and gas activities. NPD FactPages contains

² <https://github.com/ontop/ontop-examples/tree/master/eswc-2015>

³ <http://github.com/ontop/ontop>

⁴ <http://ontop.inf.unibz.it/>

⁵ <http://www.npd.no/en/>

information regarding the petroleum activities on the Norwegian continental shelf, and is actively used by oil companies. The **schema** of the NPD dataset consists of 70 tables with 276 distinct columns (about 1000 columns in total), and 94 foreign keys. We use a MySQL database to store the NPD dataset.

To ease the access of end-users to this dataset, we will show how to set up an OBDA framework using *Ontop* as a SPARQL endpoint. The user can access the dataset using the vocabulary specified in the NPD **ontology**. The NPD ontology contains OWL axioms specifying comprehensive information about the underlying concepts in the dataset; specifically hierarchies of classes (e.g., *Wellbore*, *ProductionWellbore*), and properties (e.g., *wellboreAgeHc*, *wellboreAgeHcLevel*). In this demo, we use an OWL 2 QL fragment of the NPD ontology that contains 343 classes, 142 object properties, 238 data properties, and 1451 axioms. The NPD **mapping** set consists of 1190 assertions, mapping a total of 464 classes, objects properties, and data properties, exposing +4 billion triples⁶.

To illustrate these concepts, we will take a simple query that retrieves all the wellbores⁷:

```
SELECT * WHERE {?w rdf:type epds:Wellbore}
```

and show how this simple query over the ontology gets translated into a complex SQL query over the database, with five unions and accessing three different tables. Intuitively, the generated SQL query has the following form:

```
SELECT uri(wellb-at1) FROM wellbore_exploration_all WHERE filter... UNION
SELECT uri(wellb-at1) FROM wellbore_development_all WHERE filter... UNION
SELECT uri(wellb-at2) FROM wellbore_exploration_all WHERE filter... UNION
SELECT uri(wellb-at2) FROM wellbore_development_all WHERE filter... UNION
SELECT uri(wellb) FROM wellbore_npdid_overview WHERE filter...
```

where *uri(attr)* builds the URIs specified by the mappings. The average execution time for the fastest executing queries in this benchmark (with +4B triples) is 6.7 seconds. There are queries, however, that take much longer to execute. For instance, query 11 in the benchmark already takes 10 seconds in a dataset with only 25 million triples [6]. One of the main tasks of an OBDA system is to optimize redundant joins and unions that often appear during the SPARQL to SQL translation. We will show how we have been tackling these issues in *Ontop* to allow for scalability of OBDA over real world data.

3.1 Ontop as a SPARQL Endpoint

Ontop extends the Sesame Workbench to create and manage *Ontop* repositories, which can then be used as standard SPARQL HTTP endpoints. A SPARQL endpoint is a (referenceable) entity over which SPARQL queries can be posed and that is compliant with the SPARQL Protocol for RDF specification. Sesame OpenRDF Workbench is a web application for administrating Sesame repositories.

In this demo we demonstrate how to use *Ontop* with the Sesame Workbench over NPD. We will use a pre-configured Jetty web server including the Sesame workbench

⁶ In NPD₁₅₀₀, see [6].

⁷ A wellbore is a hole drilled for the purpose of exploration or extraction of natural resources.

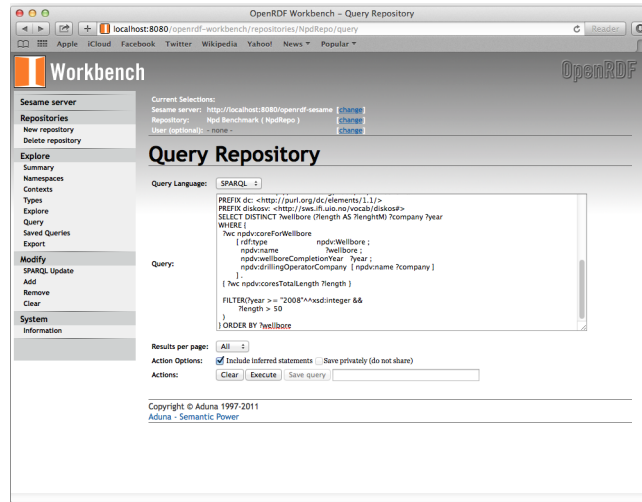


Fig. 3: Query a repository in Sesame Workbench

with the *Ontop* extension. We will show how to create a Virtual RDF Store using the NPD ontology and mappings, how to explore it, and query it (depicted in Figure 3). We will also show how the SPARQL queries are translated into SQL, what are the semantic and structural optimizations applied to the SQL query, and what are the main challenges in this approach.

3.2 New Features in Ontop

We will present two new features of the system:

Consistency Check. Ontology languages, such as OWL 2, allow for the specification of constraints on the data. If the data exposed by the database through the mappings does not satisfy such constraints, then we say that the ontology is *inconsistent w.r.t. the mappings and the data*. The OWL 2 QL profile of OWL 2, which is specifically designed to be used in the context of OBDA, allows for expressing *disjointness constraints* between classes and properties. Two disjoint classes cannot have individuals in common. For example, in NPD the class *OneDimensionalRegion* is declared to be disjoint from the class *TwoDimensionalRegion*. The notion of disjointness is extended in the obvious way to properties as well. The original NPD ontology contained inconsistencies, and it is possible to detect those using the *Ontop* Protégé plugin.

SWRL. The Semantic Web Rule Language (SWRL) is a widely used Semantic Web language combining a DL ontology component with rules (allowing only unary and binary predicates). SWRL is implemented in many systems, such as, Pellet, Stardog, and HermiT. We show how fragments of the NPD ontology can be expressed in SWRL, and how *Ontop* can handle it, including a weak form of linear recursion.

4 Conclusions

In this demonstration we present the *Ontop* framework over the NPD Benchmark, which offers a complex, authentic and challenging scenario for OBDA systems. We show how to set up an OBDA system from scratch using *Ontop*, and which are the benefits and the limitations of this approach. In addition, we explore several new interesting features of the system such as *Ontop* as a SPARQL end-point, support for additional relevant standards (SWRL, R2RML) and consistency checks.

Acknowledgement. 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. T. Bagosi, D. Calvanese, J. Hardi, S. Komla-Ebri, D. Lanti, M. Rezk, M. Rodríguez-Muro, M. Slusnys, and G. Xiao. The ontop framework for ontology based data access. In *Proc. of the 8th Chinese Semantic Web Symposium & Web Science Conference (Posters and Demos)*, Wuhan, China, August 2014.
2. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, and R. Rosati. Ontologies and databases: The *DL-Lite* approach. In *5th Int. Reasoning Web Summer School Tutorial Lectures (RW 2009)*, volume 5689, pages 255–356. Springer, 2009.
3. 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.
4. S. Harris and A. Seaborne. SPARQL 1.1 Query Language. W3C Recommendation, World Wide Web Consortium, Mar. 2013. Available at <http://www.w3.org/TR/sparql11-query>.
5. R. Kontchakov, M. Rezk, M. Rodríguez-Muro, G. Xiao, and M. Zakharyashev. Answering SPARQL queries over databases under OWL 2 QL entailment regime. In *Proc. of the 13th Int. Semantic Web Conf. (ISWC 2014), Part I*, volume 8796, pages 552–567. Springer, 2014.
6. D. Lanti, M. Rezk, G. Xiao, and D. Calvanese. The NPD benchmark: Reality check for OBDA systems. In *Proc. 18th International Conference on Extending Database Technology (EDBT)*, 2015.
7. F. Manola and E. Mille. RDF primer. W3C Recommendation, World Wide Web Consortium, Feb. 2004. Available at <http://www.w3.org/TR/rdf-primer-20040210/>.
8. M. Rodríguez-Muro, J. Hardi, and D. Calvanese. Quest: Efficient SPARQL-to-SQL for RDF and OWL. In *Proceedings of the ISWC 2012 Posters & Demonstrations Track, Boston, USA, November 11-15, 2012*, 2012.
9. W3C OWL Working Group. OWL 2 web ontology language document overview (second edition). W3C Recommendation, World Wide Web Consortium, 2012. Available at <http://www.w3.org/TR/owl2-overview/>.
10. G. Xiao, M. Rezk, M. Rodríguez-Muro, and D. Calvanese. Rules and ontology based data access. In *Proc. of the 8th Int. Conf. on Web Reasoning and Rule Systems (RR)*, Lecture Notes in Computer Science. Springer, 2014.