# OPTIQUE: Ontology-Based Data Access Platform[⋆]

E. Kharlamov[1] E. Jiménez-Ruiz[1] C. Pinkel[2] M. Rezk[3] M. G. Skjæveland[4]
A. Soylu[4,5] G. Xiao[3] D. Zheleznyakov[1] M. Giese[4] I. Horrocks[1] A. Waaler[4]

[1] University of Oxford; [2] fluid Operations AG; [3] Free University of Bozen-Bolzano;
[4] University of Oslo; [5] Gjøvik University College

**Abstract.** Ontology-Based Data Access (OBDA) is an approach to query relational data via a unified semantic access point powered by an ontology that is 'connected' to the underlying databases via mappings. OPTIQUE is an end-to-end OBDA platform. It offers support for semi-automatic bootstrapping of ontologies and mappings from relational databases thus facilitating system deployment, an intuitive interface to pose queries over a deployed system, and a query processing and optimisation module that allows to efficiently answer user queries. In this demonstration attendees will be able to experience OPTIQUE with data from the oil and gas industry and data from the music domain.

## 1 Introduction

In enterprises the ability of domain experts to quickly understand and analyze data is at the core of making accurate business decisions. In many cases this requires an interactive data exploration: domain experts need to access and analyze available data sources *directly* without involving IT-experts [7]. Challenges in providing such direct data access include the *complexity* of database schemata that can contain hundreds and thousands of tables, and the *conceptual mismatch* between the language and structures that the domain experts use to describe the data, and the way the data is described and structured by database schema languages [2, 7, 8].

*Ontology-Based Data Access* (OBDA) [12] is a prominent approach to end-user oriented direct data access. OBDA provides semantic access to databases via an *ontology* while leaving the data in its original stores. A virtue of an ontology is that it allows domain experts to express information needs in their own terms without considering the way data is organized in the source, which makes the query formulation task independent from IT-expert involvement. OBDA *mappings* describe the relationships between the ontological vocabulary and the schema of the underlying data. In OBDA user queries formulated over ontologies are processed in two stages: first, the query is *enriched* using logical reasoning by compiling relevant parts of the ontology into the query, second, the resulting query is *unfolded*, i.e., translated into a SQL query using mappings. The resulting SQL query is executed over the underlying data and the obtained answers are returned to the user.

OBDA has recently attracted a lot of attention, e.g., [1, 13], however, to the best of our knowledge no system supports the full OBDA life cycle from system deployment to end user query formulation. In this demo we present OPTIQUE [3, 6], an end-to-end integrated

[⋆] This research has been partially supported by the EU project Optique (FP7-IP-318338), the Royal Society, the EPSRC grants Score!, DBonto, and MaSI$^3$.

OBDA platform for enterprises that comes with a suit of novel components covering needs of both IT-experts and end users to deploy an OBDA system in an enterprise from scratch, effectively maintain and use it for data access tasks. The demonstration will focus on three features from OPTIQUE platform:

*(i)* System deployment using semi-automatic bootstrapping of ontologies and mappings from relational databases and aligning them with existing ontologies,
*(ii)* Query processing and optimization for efficient query answering,
*(iii)* Visual query formulation for enabling end-user formulation of queries without prior knowledge of the SPARQL language.

During the demo we will allow attendees to experience the above mentioned OPTIQUE components and the platform as a whole on two datasets: the Northwind database, and public data from our work with Statoil [7].

## 2 The OPTIQUE Platform

The three-layer architecture of OPTIQUE is depicted in Figure 1, where double arrows represent a query or data flow, and solid arrows represent a dependency between components: if A points to B, then A can call B. The OPTIQUE implementation is based on the Information Workbench [4], a generic and extensible platform for semantic data management that provides many base components for OPTIQUE and APIs for managing metadata assets.

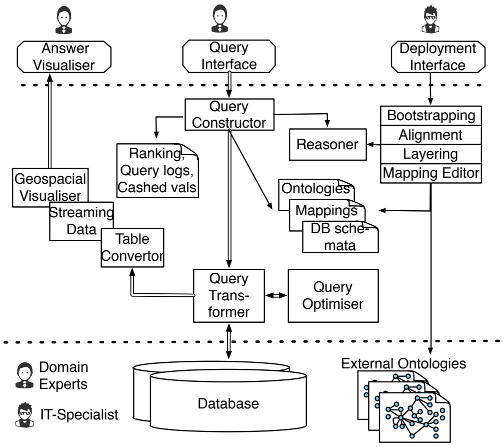

**Fig. 1.** Architecture of the OPTIQUE platform

*Deployment.* OPTIQUE's deployment support allows IT-specialists to author (write and edit), bootstrap, and import OWL ontologies and mappings from the underlying relational DBs (RDB) using an OPTIQUE module based on our BOOTOX [5] system. More precisely, an OBDA instance $(\mathcal{O}, \mathcal{M}, \mathcal{D})$ is a triple where $\mathcal{O}$ is an OWL ontology, $\mathcal{D}$ is an RDB, and $\mathcal{M}$ is a set of mappings between $\mathcal{O}$ and $\mathcal{D}$ consisting of assertions of the form: $C(x) \leftarrow \mathtt{SQL}(x)$ or $P(x, y) \leftarrow \mathtt{SQL}(x, y)$ where $C$ and $P$ are class and property names; and $\mathtt{SQL}(x)$ and $\mathtt{SQL}(x, y)$ are unary and binary SQL queries. In these terms OPTIQUE support three deployment scenarios [5, 11]:

*(i)* *bootstrapping*: for a given DB $\mathcal{D}$ we compute a set of mappings $\mathcal{M}$ relating $\mathcal{D}$ to a new ontological vocabulary and an ontology $\mathcal{O}$ over this vocabulary,
*(ii)* *alignment*: for a given OBDA instance $(\mathcal{O}_1, \mathcal{M}, \mathcal{D})$ and some ontology $\mathcal{O}_2$, we compute a new OBDA instance $(\mathcal{O}, \mathcal{M}, \mathcal{D})$ where $\mathcal{O}$ is a 'merger' of $\mathcal{O}_1$ and $\mathcal{O}_2$,
*(iii)* *layering*: for a given ontology $\mathcal{O}$ and a given database $\mathcal{D}$ we compute a set of mappings $\mathcal{M}$ relating $\mathcal{O}$ and $\mathcal{D}$ such that $(\mathcal{O}, \mathcal{M}, \mathcal{D})$ is an OBDA instance.

Intuitively, the main difference between bootstrapping and layering is that the former extracts an ontology and mappings from a DB and it requires only the DB as the input, while the latter relates a giver ontology to a DB by computing necessary mappings and it requires both the DB and the ontology as the input. OPTIQUE supports query answering over OWL 2 QL ontologies only, thus, if the bootstrapped or imported OWL ontology is not in QL, then OPTIQUE approximates it to QL.

*Query Answering.* OPTIQUE's query processing module is based on our ONTOP [9] system. The naive implementation of the two stage approach for answer computation in OBDA performs poorly in practice and optimizations are required [14]. Thus, we developed a number of techniques to optimize both stages and implemented them in the OPTIQUE query processing module. Enrichment is optimized by addressing both the *redundancy* in the enriched queries and the *inefficiency* of enrichment computation. In the former case we minimize the mappings and the enriched queries with respect to query containment. For the latter, we use a variant of a graph reachability algorithm, we improve computation of class hierarchies entailed by the ontology, which the enrichment heavily relies on. Additionally, we move part of online reasoning offline: for all atomic queries we perform expensive enrichment offline and compile the results of this computation into the existing mappings, thus, enriching mappings. Unfolding is optimized by turning large and highly redundant SQL queries returned after the second stage of query processing into compact and efficiently executable SQL queries. Optimizations are achieved both *structurally*, by pushing joins inside the unions and special functions (such as URI construction) as high as possible in the query tree, and *semantically*, by detecting and removing inefficient joins between sub-queries. Experiments show that these optimization techniques allow us to dramatically outperform existing OBDA query processing engines [7, 10].

*Query Formulation.* The query formulation module is based on our OPTIQUEVQS [16] system. Visually formulated queries are automatically translated into SPARQL which can be sent to the query transformation module. Users can also write queries in SPARQL directly. The query formulation module has a widget-based architecture and exploits multiple representation and interaction paradigms for query composition. In particular, it uses a graph metaphor for navigation between classes via object properties, and faceted search for query refinement via data properties. At each step of the query formulation process ranked suggestions are automatically generated to guide users in constructing the query. The suggestions are generated by reasoning over the ontology and query logs. An important feature of the system is a special treatment of data properties: it automatically generates different end-user oriented representations of data values, including sliders restricting possible ranges of numerical values, such as age, depths, etc., and drop boxes with precomputed lists for categorical data, such as names of companies, geographical locations, etc. The current version of the system allows the construction of tree-shaped conjunctive queries enhanced with simple aggregate functions.

## 3 Demonstration overview

Figure 2 contains different screenshots from the OPTIQUE platform applied to our demonstration scenarios. The central screenshot has the main menu of the platform for administering data sources, mappings, ontologies and queries, and performing actions on these: bootstrapping, query transformation setup, general system configuration including query optimization, and visual queries construction. The bottom-left screenshot shows a visual query in the OptiqueVQS, and images on the right show the answers to this query in a table and a map view. The top-left screenshot visualizes the integration of a bootstrapped and an imported ontology. During the demonstration we will present

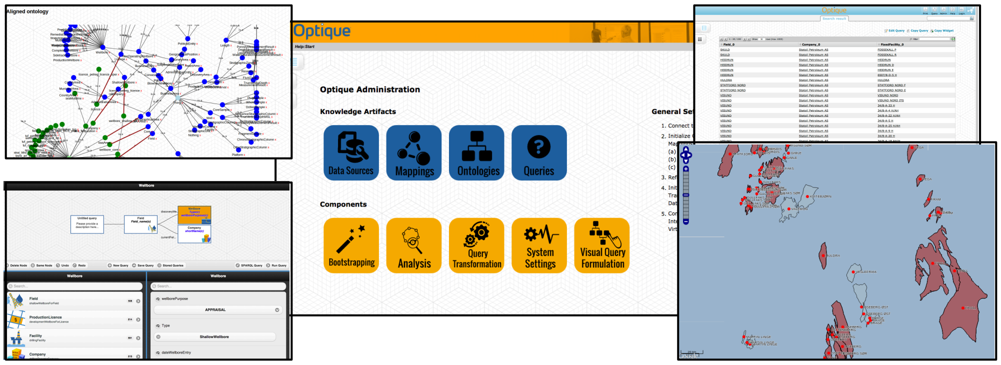

**Fig. 2.** Screenshots of the OPTIQUE platform

OPTIQUE end-to-end, with the tools and techniques behind these screenshots and more, over two datasets:

**Northwind DB** (`northwinddatabase.codeplex.com`) is a demo database with easy-to-understand business data comprising customers, products, orders, employees, etc. It contains a total of 14 tables and 12 referential constraints.

**NPD FactPages [15]** is a public fragment from our Statoil deployment. This data is heavily used in the oil and gas industry, it consists of 70 tables, 276 different attributes, 96 foreign keys, and about 50 MB of mostly aggregated data and metadata. Exploration of this scenario requires from demo attendees some basic knowledge of the oil and gas domain.

## 4 References

[1] D. Calvanese et al. The MASTRO System for Ontology-Based Data Access. In: *Semantic Web* 2.1 (2011).

[2] J. Crompton. *Keynote talk at the W3C Workshop on Sem. Web in Oil & Gas Industry*. `http://www.w3.org/2008/12/ogws-slides/Crompton.pdf`. 2008.

[3] M. Giese et al. Optique — Zooming In on Big Data Access. In: *Computer* 48.3 (2015).

[4] P. Haase et al. The Information Workbench as a Self-Service Platform for Linked Data Applications. In: *WWW*. 2012.

[5] E. Jiménez-Ruiz et al. BootOX: Practical Mapping of RDBs to OWL 2. In: *ISWC*. 2015.

[6] E. Kharlamov et al. Optique: Towards OBDA Systems for Industry. In: *ESWC (SE)*. 2013.

[7] E. Kharlamov et al. Enabling Ontology Based Access at Statoil. In: *ISWC*. 2015.

[8] E. Kharlamov et al. How Semantic Technologies Can Enhance Data Access at Siemens Energy. In: *ISWC*. 2014.

[9] R. Kontchakov et al. Answering SPARQL Queries over Databases under OWL 2 QL Entailment Regime. In: *ISWC*. 2014.

[10] D. Lanti et al. The NPD Benchmark: Reality Check for OBDA Systems. In: *EDBT*. 2015.

[11] C. Pinkel et al. IncMap: Pay as You Go Matching of Relational Schemata to OWL Ontologies. In: *OM*. 2013.

[12] A. Poggi et al. Linking Data to Ontologies. In: *J. Data Sem.* 10 (2008).

[13] F. Priyatna et al. Formalisation and Experiences of R2RML-based SPARQL to SQL query translation using Morph. In: *WWW*. 2014.

[14] M. Rodriguez-Muro et al. Efficient SPARQL-to-SQL with R2RML mappings. In: *To appear, Journal of Web Semantics* (2015).

[15] M. G. Skjæveland et al. Publishing the NPD FactPages as Semantic Web Data. In: *ISWC*. 2013.

[16] A. Soylu et al. OptiqueVQS: Towards an Ontology-Based Visual Query System for Big Data. In: *MEDES*. 2013.