OptiqueVQS: Visual Query Formulation for OBDA (Abstract)

Ahmet Soylu[1], Evgeny Kharlamov[2], Dmitriy Zheleznyakov[2], Ernesto Jimenez-Ruiz[2], Martin Giese[1], and Ian Horrocks[2]

[1] University of Oslo; [2] University of Oxford

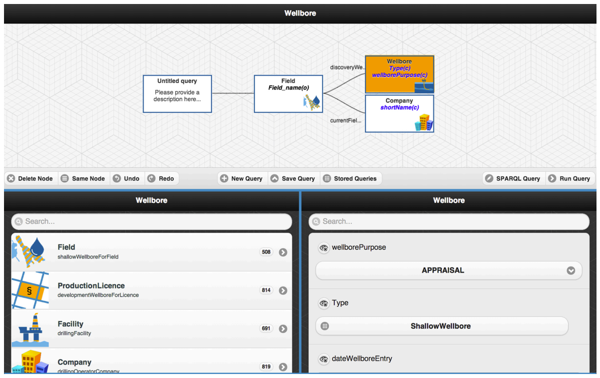
Motivation *Ontology Based Data Access* (OBDA) [16] is a recently proposed prominent approach that aims at providing domain experts with a *direct* access to available enterprise data sources without IT-experts being involved. OBDA is an alternative to centralised approaches, where an IT-expert translates the requirements of domain experts into Extract-Transform-Load (ETL) processes to first integrate the data and then to apply predefined analytical reporting tools. Currently, centralised approaches are commonly used in enterprises; they, however, can become too heavy-weight and inflexible in some cases [12], that can be addressed the OBDA approach.

The key idea behind OBDA is to use *ontologies* to mediate between users and data. Ontologies describe the domain of interest on a higher level of abstraction in terms that are clear for domain experts, and introduce modeling concepts such as inheritance and relationships between classes of objects, thus allowing to describe the intended meaning of the ontological vocabulary. Ontologies have become a common and successful mechanism to describe application domains in, e.g., biology, medicine, the (Semantic) Web [13]. This success is partially due to a number of available formal languages for describing ontologies, including RDF(S) [7] and OWL 2 [5] standardised by W3C.

In OBDA, users formulate their information needs as queries using terms defined in the ontology, and ontological queries are translated into SQL and executed over the data automatically, without an IT-expert's intervention. To this end a set of *mappings* is maintained that describe the relationship between the ontological vocabulary and the elements of the schema of the underlying data.

The standard query language for ontologies is SPARQL [8]. Writing queries using SPARQL, however, is not easy for domain experts and thus intuitive *visual query formulation* support is required for OBDA systems. Existing OBDA systems, e.g., [1, 2, 9, 10, 18–21] typically offer limited or no visual query formulation support. Our goal is to provide a solution for visual query formulation over ontologies that is specifically tailored for OBDA systems. The solution should rely on solid theory, be efficient, support interactive data exploration, and should follow the best Human-Computer Interaction practices to guarantee good usability. In the following we give a short overview of our ideas that were partially implemented in our OptiqueVQS system [22].

OptiqueVQS We first describe functionality of OptiqueVQS' components and then give their formal description. OptiqueVQS is a system for visual query formulation support that allows the user to construct a query over an ontology step by step where at each step the system provides the user with relevant information to continue the query construction. OptiqueVQS has a widget-based architecture and exploits multiple representation and interaction paradigms, see Fig. 1 for a screenshot where a sample query




```
SELECT DISTINCT ?c1 ?a1 ?c2 ?c3 WHERE{
  ?c1 ns1:type ns2:Field.
  ?c2 ns1:type ns2:Company.
  ?c3 ns1:type ns2:Wellbore.
  ?c3 ns1:type ns2:ShallowWellbore.
  ?c1 ns2:currentFieldOperator ?c2.
  ?c1 ns2:discoveryWellbore ?c3.
  ?c1 ns2:name ?a1.
  ?c2 ns2:name "Statoil Petroleum AS".
  ?c3 ns2:wellborePurpose "APPRAISAL".
}
```

Fig. 1. Interface of OptiqueVQS

over an ontology for the Oil and Gas domain is composed[1] together with its SPARQL counterpart. The query asks for oil fields, wellbores operated on these fields, and companies currently exploiting the fields. OptiqueVQS has three widgets: W1 employs the graph metaphor, gives an overview of the constructed query, and allows further manipulation of it, W2 employs the menu-based representation paradigm to visualise suggestions that users can use to extend the query, W3 employs the form-based representation paradigm to visualise possible constraints (projection and selection) that users can set on different parts of the queries.

Query construction process in OptiqueVQS works as follows [3]. The user starts with selecting in W2 a 'starting' suggestion, i.e., a class, from the list of available ones and the selected suggestion appears in W1 and becomes 'active'. Then, the user can extend the query either by selecting in W2 one of the offered suggestions, i.e., a class reachable from the active suggestion via some object property, or by setting constraints, i.e., by restricting in W3 the data properties of the objects belonging to the class of the active suggestion. W1 displays all selected suggestions and organise them in a tree. The user can change the active suggestion by clicking on the ones in W1, or by adding a new one through W2. For each active suggestion OptiqueVQS automatically generates relevant further suggestions in W2 and constraints in W3. The generation is done via reasoning (e.g., extraction of classification, inferred domain and ranges) over the ontology underlying the system and to this end we exploit the HermiT reasoner [17]. Moreover, users have partial control on output variables, can delete fragments of constructed queries, access query catalogue, save/load queries, and undo/redo actions.

Queries of OptiqueVQS. The queries follow the following grammar:

$$\begin{aligned}
\texttt{query} &::= A(x)(\wedge\texttt{constr}(x))^*(\wedge\texttt{expr}(x))^*, \text{ where } A \text{ is an atomic class},\\
\texttt{expr}(x) &::= \texttt{sug}(x,y)(\wedge\texttt{constr}(x))^*(\wedge\texttt{expr}(y))^*,\\
\texttt{constr}(x) &::= \exists y\, R(x,y) \mid R(x,y) \mid R(x,c), \text{ where } R \text{ is an atomic data property},\\
\texttt{sug}(x,y) &::= Q(x,y) \wedge A(y), \text{ where } A \text{ and } Q \text{ are atomic class and object property},
\end{aligned}$$

where variables y in different expressions $\texttt{expr}(x)$ of a structure $\texttt{str}$ are different. An OptiqueVQS $\texttt{query}$ is constructed using suggestions $\texttt{sug}$ and constraints $\texttt{constr}$,

[1] This ontology was designed for Statoil [4] as a part of the Optique project [15].

that are combined in expressions expr. Such queries are conjunctive and tree shaped: the graph corresponding to the query where nodes are variables and edges are properties is a tree. All the variables that occur in classes and object properties are output variables and some variables occurring in data properties can also be output variables.

When users interacts with OptiqueVQS, then for every $\mathtt{sug}(x, y)$ that an $\mathtt{expr}(x)$ starts with, that is, for every active suggestion, the system offers a list of relevant $\mathtt{constr}(x)$ via W3 and relevant $\mathtt{sug}(y, z)$ via W2 that can be used to construct further $\mathtt{expr}(y)$. We explore several notions of relevance, including *local* where offered constraints and suggestions depend on $\mathtt{sug}(x, y)$ only, and *global* where they depend on the entire query. We currently investigate complexity of suggestion generation for different ontologies and notions of relevance.

Treatment of Data Properties. An important feature of OptiqueVQS is a special treatment of data properties in W3: 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. Throughout empirical evaluations we determined that this treatment of data properties is of high importance for end-users. To support different intuitive representations for data properties, we encode relevant information in the ontology underlying the system and generate the representations on the fly.

Query Construction vs Rewriting Ontology. We use OptiqueVQS for query formulation in the Optique OBDA system, and thus we convert queries constructed via OptiqueVQS in SPARQL and then they are processed by the Optique query processing component [20] that rewrites them with the system's ontology and unfolds it with mappings in SQL. We use OWL 2 QL ontologies for query rewriting, while the query construction is based on much richer OWL 2 ontologies that, in particular, make use of nominals. There are both theoretical and practical reasons for having two ontologies: conjunctive query rewriting for OBDA is well studied for OWL 2 QL ontologies [6], while for effective and efficient query support of conjunctive query construction the expressive power of OWL 2 QL ontologies is not sufficient. The query construction ontology that we use in the system extends the query rewriting ontology.

Feedback in Query Construction. To improve query construction experience and to allow data exploration OptiqueVQS provides users with feedback at each step of query construction: the users can see answers relevant to the constructed query. Since computation of answers in OBDA systems is expensive, we investigate several possibilities for the feedback: it can be, for example, a set of sample query answers, or a summary of query answers, or some statistics on query answers. We currently investigate complexity of different types of feedback. Moreover, we investigate influence of different types of feedback on the usability of the system.

To Sum Up. We developed OptiqueVQS in cooperation with Statoil and did preliminary user evaluation with Statoil geologists that gave us encouraging results. We also presented the system at several venues [11, 14, 22, 23]. Currently we investigate theoretical properties of our techniques. We also work on improvements of the system in several directions, e.g., we develop ranking functions for suggestions and constraints.

References

1. http://virtuoso.openlinksw.com/
2. http://www.revelytix.com/content/spyder
3. OptiqueVQS. https://www.youtube.com/watch?v=ks5tcPZVHp0
4. Statoil. http://www.statoil.com/en/Pages/default.aspx
5. W3C: OWL 2 Web Ontology Language. http://www.w3.org/TR/owl2-overview/
6. W3C: OWL 2 Web Ontology Language Profiles. http://www.w3.org/TR/owl-profiles/
7. W3C: Resource Description Framework (RDF). http://www.w3.org/RDF/
8. W3C: SPARQL 1.1 Query Language. www.w3.org/TR/sparql11-query/
9. Bizer, C., Seaborne, A.: D2RQ—Treating non-RDF Databases as Virtual RDF Graphs. In: ISWC (2004)
10. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO System for Ontology-Based Data Access. Semantic Web 2(1), 43–53 (2011)
11. Cuenca Grau, B., Giese, M., Horrocks, I., Hubauer, T., Jimenez-Ruiz, E., Kharlamov, E., Schmidt, M., Soylu, A., Zheleznyakov, D.: Towards Query Formulation, Query-Driven Ontology Extensions in OBDA Systems. In: OWLED (2013)
12. Doan, A., Halevy, A.Y., Ives, Z.G.: Principles of Data Integration. Morgan Kaufmann (2012)
13. Horrocks, I.: What are Ontologies Good for? In: Evolution of Semantic Systems, pp. 175–188. Springer (2013), download/2013/Horr13a.pdf
14. Kharlamov, E., et al.: Optique 1.0: Semantic Access to Big Data: The Case of Norwegian Petroleum Directorate's FactPages. In: ISWC (Posters & Demos) (2013), https://www.youtube.com/watch?v=PToyue4BFXA
15. Kharlamov, E., Jiménez-Ruiz, E., Zheleznyakov, D., Bilidas, D., Giese, M., Haase, P., Horrocks, I., Kllapi, H., Koubarakis, M., Özçep, Ö.L., Rodriguez-Muro, M., Rosati, R., Schmidt, M., Schlatte, R., Soylu, A., Waaler, A.: Optique: Towards OBDA Systems for Industry. In: ESWC (SE). pp. 125–140 (2013)
16. Kogalovsky, M.R.: Ontology-Based Data Access Systems. Programming and Computer Software 38(4), 167–182 (2012)
17. Motik, B., Shearer, R., Horrocks, I.: Hypertableau Reasoning for Description Logics. Journal of Artificial Intelligence Research 36, 165–228 (2009)
18. Munir, K., Odeh, M., McClatchey, R.: Ontology-Driven Relational Query Formulation Using the Semantic and Assertional Capabilities of OWL-DL. Knowledge-Based Systems 35, 144–159 (2012)
19. Priyatna, F., Corcho, O., Sequeda, J.: Formalisation and Experiences of R2RML-Based SPARQL to SQL Query Translation Using Morph. In: WWW (2014)
20. Rodriguez-Muro, M., Kontchakov, R., Zakharyaschev, M.: Ontology-Based Data Access: Ontop of Databases. In: ISWC. pp. 558–573 (2013)
21. Sequeda, J.F., Miranker, D.P.: Ultrawrap: SPARQL Execution on Relational Data. Journal of Web Semantics 22(0), 19 – 39 (2013), http://www.sciencedirect.com/science/article/pii/S1570826813000383
22. Soylu, A., Giese, M., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I.: OptiqueVQS: Towards an Ontology-Based Visual Query System for Big Data. In: MEDES (2013)
23. Soylu, A., Skjæveland, M., Giese, M., Horrocks, I., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D.: A Preliminary Approach on Ontology-Based Visual Query Formulation for Big Data. In: MTSR. pp. 201–212 (2013)