Ontology Based Data Access: Where do the Ontologies and Mappings come from?

Juan F. Sequeda

Capsenta
juan@capsenta.com

1 Introduction

We are experiencing an increase of Ontology Based Data Access (OBDA) systems being deployed in industrial applications. In the OBDA paradigm, the ontology provides a logical abstraction, independent of how and where the data is physically stored. The ontology serves as a business view, using business terminology, which is then connected to data sources. Thus, providing a foundation for comfortable communication between business users and IT developers.

Even though OBDA has been widely researched theoretically, there is still need to understand how to effectively implement OBDA systems in practice. Our focus is in Business Intelligence (BI) reporting. The common definition of OBDA states that given a source relational database, a target ontology and a mapping from the relational database to the ontology, the goal is to answer queries over the target ontology using these three daysonents. From a practical point of view, this begs the question: where does the target ontology and the mappings come from?

Ontology Challenges Ontology engineering is a challenge by itself. In order to create the target ontology, users can follow traditional ontology engineering methodologies [2, 10], using competency questions [1, 5], test driven development [4], ontology design patterns [3], etc. Additionally, per standard practices, it is recommended to reuse and extend existing ontologies in domains of interest such as Good Relations for e-commerce [13], FIBO for finance [14], Gist for general business concepts [14], Schema.org [16], etc. In OBDA, the challenge increases because the source database schemas can be considered as additional inputs to the ontology engineering process. Common enterprise application's database schema commonly consist of thousands of tables and tens of thousands of attributes. A common approach is to bootstrap ontologies derived from the source database schemas, known also putative ontologies[6, 7]. The putative ontologies can gradually be transformed into target ontologies, using existing ontology engineering methodologies.

Mapping Challenges Once the Target ontology has been created, the source databases can be mapped. The W3C Direct Mapping standard can be used to bootstrap mappings [11]. The declarative nature of W3C R2RML mapping language[12] enables users to state which elements from the source database are connected to the target ontology, instead of writing procedural code. Given that source database schemas are very large, the OBDA mapping challenge is suggestive of an ontology matching problem: the putative ontology of the source database and the target ontology. In addition to 1-1 correspondences between classes and properties, mappings can be complex involving calculations

and rules that are part of business logic. For example, the notion of net sales of an order is defined as gross sales minus taxes, discounts given, etc. The discount can be different depending on the type of user. Therefore, a business user needs to provide these definitions before hand. That is why it is hard to automate this process. Another challenge is to create tools that can create and manage mappings [9].

Addressing these challenges is crucial for the success of OBDA in practice. To answer the main question of this paper: Where do the Ontologies and Mappings come from? Our answer: from the business questions.

2 Pay-as-you-go Methodology for OBDA

We recently introduced a methodology to create the target ontology and mappings for an OBDA system, driven by a prioritized list of business questions[8]. The objective is to create a target ontology and mappings, that enable answers to list of business questions, *in an incremental manner*. After a minimal set of business questions have been successfully modeled, mapped, answered and made into dashboards, then the set of business questions can be extended. The new questions, in turn, may extend the target ontology and new mappings incrementally added. With this methodology, the target ontology and mappings are developed in an iterative pay-as-you-go approach. Thus, providing an agile methodology for BI using the OBDA paradigm because the focus is to provide early and continuous delivery of answers to the business users. Figure 1 provides an overview of the methodology. We refer the reader to [8] for more details.

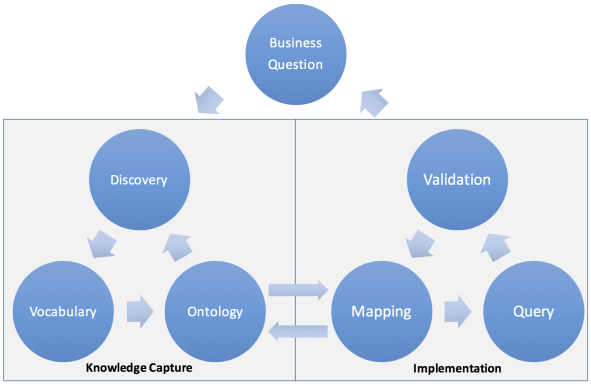

Fig. 1. The Pay-as-you-go Methodology for OBDA

However, this leads to another question: where do the business questions come from? We observe three common sources: 1) Questions coming directly from business users (e.g. What is the net sales of orders group by country and month) 2) Business application systems returning fixed reports which can not be customized or extended, 3) SQL queries that have been customized to answer specific questions. We focus on the latter: business questions coming from SQL queries used to generate BI reports.

3 Generating Ontologies and Mappings from SQL queries

A common scenario is the following: SQL queries are initially created by developers who are knowledgeable of the large database schema. Developers come and go within an organization. Queries get shared, altered, extended and combined. After time, users are executing SQL queries without any understanding of what the queries actually do. Users rely on a description of what the SQL query is suppose to be returning.

Our position is that we should be able to extract valuable information from a SQL query which is being used for BI. Specifically, we observe that it is possible to generate an Ontology and Mapping from a query. This Ontology and Mapping is the starting point to implement an OBDA system for BI. Consider the following SQL query which is used to return the net sales of all orders.

```
SELECT o.orderid, o.orderdate, o.ordertotal - ot.finaltax -
  CASE WHEN o.currencyid in ('USD', 'CAD') THEN o.shippingcost
       ELSE o.shippingcost - ot.shippingtax END AS netsales,
  o.currencyid
FROM order o, ordertax ot
WHERE o.orderid = ordertax.orderid AND o.statusid NOT IN (4, 5)
```

Just by analyzing the SQL query alone, without any other additional resource, we can extrapolate the following:

Relational Schema

order(orderid, orderdate, ordertotal, currencyid, shippingcost, statusid, ...)
ordertax(orderid,finaltax,shippingtax, ...)
order.orderid is a Primary Key
ordertax.orderid is a Primary Key
ordertax.orderid is a Foreign Key referencing order.orderid
order.statusid could be a Foreign Key referencing a table containing status codes

Instances

order.currencyid = 'USD', 'CAD'
order.statusid = 4, 5

Calculations

orderShippingCost = IF (currency = USD or CAD) THEN (RETURN shippingcost) ELSE (RETURN shippingcost - shippingtax)
netsales = ordertotal - final tax - orderShippingCost

We define the Relational Schema, Instances and Calculations derived from a query as the *components of a query*. The goal is to generate an ontology and mappings from the components of a query.

First we specify a conceptualization: Orders, Currency, Order Status, order date, order net sales, order total, order final tax, order shipping cost, order shipping tax. Furthermore, the database can be investigated to identify other valuable information. For

example, after further investigation and discussions we learn that order status 1-3 are active and 4-5 are inactive. We now have the enough elements to create an initial ontology. Focusing on the elements in the components of the query, we avoid the effort of creating a mapping from an unknown large database schema to the target ontology. The mapping problem has been reduced to a clear and well understood subset of the database schema. The relationship between the SQL query and the ontological element represents a mapping between the source database and the evolving target ontology at the most granular level. Figure 2 shows an example ontology and mapping derived from components of the example SQL query:

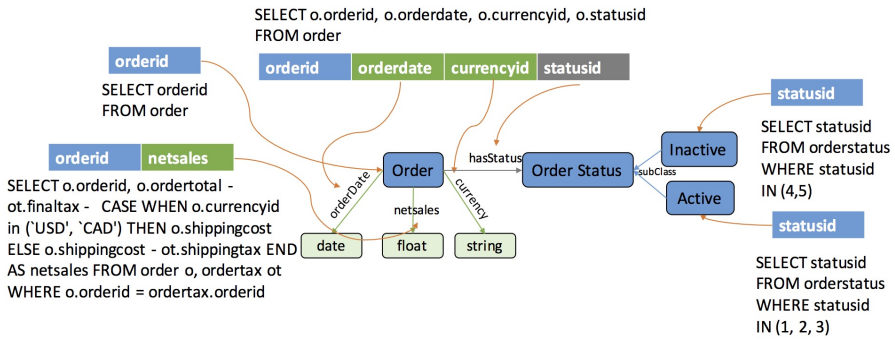

Fig. 2. Generate Ontology from the *AAA of a query*

4 Conclusion

Ontologies and mappings for OBDA can be generated from business questions. Our focus is on business questions coming from SQL queries used to generate existing BI reports. The first step is to extrapolate the components of a query: relational schema, instances and calculations. The next step is to generate an ontology and mappings from the components of a query.

This approach is currently being successfully deployed with Capsenta's customers. The results are early and continuous delivery of answers to the business users. This has not been achieved before with traditional BI methodologies.

This is just the beginning. To the best of our knowledge, the engineering of ontology and mappings for OBDA is still open grounds for research. In our current work, we are in the process of formalizing this approach in order to fully understand what should form the components of a query. There are several challenges going forward, such as: Automation: Given a SQL query, how can we automatically generate the components of a query? Given a components of a query, how can we automatically generate and OWL ontology and R2RML mappings? Iteration: Manage new business questions that extend the ontology and mappings. What happens if a new query contradicts the current ontology and/or mappings, hence it is non-monotonic? Tools: There is a need for tools that can manage large database schemas at scale.

References

1. Kamal Azzaoui et al. Scientific competency questions as the basis for semantically enriched open pharmacological space development. Drug Discovery Today 18(17-18): 843-852 (2013)
2. Oscar Corcho, Mariano Fernndez-Lpez, Asuncin Gmez-Prez. Methodologies, tools and languages for building ontologies: Where is their meeting point? Data Knowl. Eng. 46(1): 41-64 (2003)
3. Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, Valentina Presutti (eds.), Ontology Engineering with Ontology Design Patterns: Foundations and Applications. Studies on the Semantic Web 25, IOS Press/AKA, 2016.
4. C. Maria Keet, Agnieszka Lawrynowicz: Test-Driven Development of Ontologies. ESWC 2016
5. Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z. Pan, Kees van Deemter, Robert Stevens. Towards Competency Question-Driven Ontology Authoring. ESWC 2014
6. Juan Sequeda, Marcelo Arenas, Daniel P. Miranker. On directly mapping relational databases to RDF and OWL. WWW 2012
7. Juan F. Sequeda, Syed Hamid Tirmizi, Oscar Corcho, Daniel P. Miranker. Survey of directly mapping SQL databases to the Semantic Web. Knowledge Eng. Review 26(4): 445-486 (2011)
8. Juan F. Sequeda, Daniel P. Miranker.: A Pay-As-You-Go Methodology for Ontology-Based Data Access. IEEE Internet Computing 21(2): 92-96 (2017)
9. Juan F. Sequeda, Daniel P. Miranker. Ultrawrap Mapper: A Semi-Automatic Relational Database to RDF (RDB2RDF) Mapping Tool. ISWC Posters & Demos 2015
10. Mike Uschold, Michael Gruninger. Ontologies: principles, methods and applications. Knowledge Eng. Review 11(2): 93-136 (1996)
11. A Direct Mapping of Relational Data to RDF, `https://www.w3.org/TR/rdb-direct-mapping/`
12. R2RML: RDB to RDF Mapping Language `https://www.w3.org/TR/r2rml/`
13. Good Relations `http://www.heppnetz.de/projects/goodrelations/`
14. Finance Industry Business Ontology `http://www.edmcouncil.org/financialbusiness`
15. Gist `https://semanticarts.com/gist/`
16. Schema.org `http://schema.org/`