Towards a Pay-as-you-go Methodology for Ontology and Mapping Engineering in Ontology Based Data Access

Juan F. Sequeda and Daniel P. Miranker

Capsenta Inc.
{juan,miranker}@capsenta.com

1 Introduction

The Ontology Based Data Access (OBDA) paradigm enables (1) a clean separation between a conceptual business view from heterogeneous data sources and (2) ability to ask questions in terms of the business view, independent of how and where the data is physically stored. Two key components are ontologies and mappings. Capsenta is applying the OBDA paradigm for Business Intelligence applications. Even though OBDA has been widely researched theoretically, there is still need to understand how to effectively implement OBDA systems in the real world. From an practical point of view, this begs the question: where does the ontology and the mapping come from? We present our ongoing work of a pay-as-you-go methodology for ontology and mapping engineering focused on the Business Intelligence (BI) questions that need to be answered.

Consider the following real-world Business Intelligence example: Executives of a large e-commerce company need to know how many orders were placed in a given month and the corresponding net sales. Depending on whom they ask they get different answers. The IT department managing the website records an order when a customer has checked out. The fulfillment department records an order when it has shipped. Yet the accounting department records an order when the funds charged against the credit card are actually transferred to the company's bank account, regardless of the shipping status. Unaware of the source of the problem, the executives have inconsistencies across their business reports.

This is precisely where the use of ontologies to be the bridge between IT developers and business users is valuable. Ontologies serve as a uniform conceptual federated model describing the domain of interest. 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.

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 components. 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 methodolo-

gies [2, 8], 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[1] for e-commerce, FIBO[2] for finance, Gist[3] for general business concepts, Schema.org [4], 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 as 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[5] standard can be used to bootstrap mappings . The declarative nature of W3C R2RML[6] mapping language 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.

Addressing these challenges is crucial for the success of OBDA in practice.

2 Pay-as-you-go Methodology for OBDA

We present our on-going work of a methodology to create the target ontology and mappings for an OBDA system, driven by a prioritized list of business questions. 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. The result is 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.

We identify three actors involved throughout the process: 1) Business user: subject matter expert who has knowledge of the business and can identify the list of prioritized business questions, 2) IT developer: has knowledge of databases and knows how the

[1] http://www.heppnetz.de/projects/goodrelations/
[2] https://spec.edmcouncil.org/fibo/
[3] https://semanticarts.com/gist/
[4] http://schema.org/
[5] https://www.w3.org/TR/rdb-direct-mapping/
[6] https://www.w3.org/TR/r2rml/

data are interconnected and 3) Knowledge engineer: communication bridge between business users and IT developers, and has expertise in modeling data using ontologies.

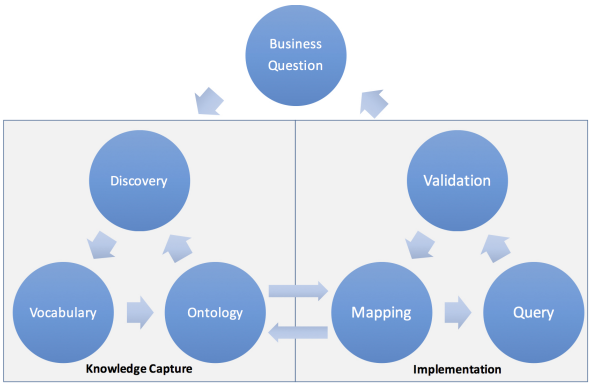

Fig. 1. The Pay-as-you-go Methodology for OBDA

Our methodology is divided into two phases: knowledge capture and implementation. Figure 1 provides an overview of the methodology.

Knowledge Capture: Discovery-Vocabulary-Ontology The goal of the knowledge capture phase is first to extract key concepts and relationships from the set of prioritized business questions and second to identify which source database(s) contains data relating to the extracted concepts and relationships. The steps are: 1) Discovery: knowledge engineer works with business users to discover the concepts and relationships from the input set of prioritized business questions in order to eliminate ambiguity. Furthermore, the knowledge engineer takes what has been extracted with the business users and works with IT developers to identify which tables and attributes from the database(s) are required. 2) Vocabulary: knowledge engineer works with business users to identify the business terminology such as preferred labels, alternative labels, and natural language definitions for the concepts and relationships. 3) Ontology: knowledge engineer formalizes the ontology in OWL such that it covers the business questions.

Implementation: Mapping-Query-Validation The goal of the implementation phase is to enable answering the business questions by connecting the ontology with the data. The steps are: 1) Mapping: knowledge engineer takes what was learned from the Discovery and Ontology steps and implements the mapping in R2RML. The mapping is then used to setup the OBDA system. 2) Query: knowledge engineer implements the business questions as SPARQL queries. 3) Validation: knowledge engineer confirms with the business users that the SPARQL queries return the correct answers.

However, this leads to another question: where do the business questions come from? In practice, we observe that is is often the case that business questions are currently being answered by a small set of expert users by running multiple SQL queries to manually generate BI reports. The problem is that the answers to these questions usually take a long time to be generated and they are not always trusted by the executives. Consider the following common scenario: Business users asks IT developers to answer

a business question. SQL queries are initially created by IT 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, business 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 supposed to be returning.

Our hypothesis is that we should be able to extract valuable information from SQL queries which are being used by small amount of expert users to manually create BI reports. Specifically, by valuable information we mean, the possibility to generate an Ontology and Mapping from a query. This Ontology and Mapping is the starting point to implement an OBDA system for BI.

3 Conclusion

Based on Capsenta's real world experiences of deploying ODBA systems, the main challenges that we encounter is the engineering of ontologies and mappings. In this poster we present our ongoing work towards tackling this challenge. Our hypothesis is that ontologies and mappings for OBDA can be generated from business questions through a pay-as-you-go methodology. Our focus is on business questions coming from existing SQL queries used to manually generate existing BI reports.

To the best of our knowledge, the engineering of ontology and mappings for OBDA is still open grounds for research. There are several challenges going forward, such as: Automation: Given a SQL query, how can we automatically generate an 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) (2013)
2. Oscar Corcho, Mariano Fernndez-Lopez, Asuncin Gmez-Prez. Methodologies, tools and languages for building ontologies: Where is their meeting point? Data Knowl. Eng. 46(1) (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. Mike Uschold, Michael Gruninger. Ontologies: principles, methods and applications. Knowledge Eng. Review 11(2): 93-136 (1996)