Ontology-Based Data Access: The experience at the Italian Department of Treasury

Natalia Antonioli[1], Francesco Castanò[3], Cristina Civili[2], Spartaco Coletta[1], Stefano Grossi[1], Domenico Lembo[2], Maurizio Lenzerini[2], Antonella Poggi[2], Domenico Fabio Savo[2], and Emanuela Virardi[3]

[1] CONSIP s.p.a.
`firstname.lastname@consip.it`

[2] DIAG – Sapienza Università di Roma
`lastname@dis.uniroma1.it`

[3] Dip. del Tesoro – Ministero dell'Economia e delle Finanze
`firstname.lastname@tesoro.it`

Abstract. Ontology-based data access (OBDA) is a new paradigm for managing complex information systems through semantic technologies. In this paper we present what we believe is the first industrial experience of a comprehensive OBDA project, developed jointly by Sapienza University of Rome and the Department of Treasury of the Italian Ministry of Econonomy and Finance. After illustrating the state of the art of ODBA research at the beginning of the project, we describe the various steps we have carried out in order to apply the research results in a real-world settings, and the new techniques and tools we have devised to make the overall approach effective.

1 Introduction

Ontology-based data access [9] (OBDA) is a new paradigm for accessing and integrating data, whose key idea is to resort to a three-level architecture, constituted by the ontology, the data sources, and the mapping between the two. The ontology is a formal description of the domain of interest, given in terms of concepts, roles, i.e., binary relations between them, and attributes, and is the heart of the whole system. The data sources are the repositories used in the organization by the various processes and the various applications. The mapping layer explicitly specifies the relationships between the domain concepts on the one hand and the data sources on the other hand.

In this paper we present what we believe is the first experience of a comprehensive OBDA project, developed jointly by Sapienza University of Rome and the Department of Treasury of the Italian Ministry of Economy and Finance (MEF), with the support of Consip S.pA., an in-house IT company owned by MEF. When we started the project in 2011, the OBDA paradigm was in its very early days. The study of OBDA had essentially focused on the case where data is stored in a so-called *ABox*, a specialized structure representing a set of membership assertions on concepts and roles. The outcome of this study had been twofold. On the one side, it had allowed to outline the ontology language expressibility boundaries for achieving query answering tractability [2, 10, 7]. In particular, it had been shown that in order for query answering to be performed with reasonable computational complexity with respect to the size of the data, and to be implemented using current DataBase Management Systems (DBMSs) technology, the ontology has to be expressed in a *lightweight* ontology language that is *first-order rewritable*, i.e. for which query answering over the ontology can be reduced to the evaluation of a suitable first-order query (i.e., an SQL query) expressed over the ABox. This basically restricts the spectrum of possible ontology languages to the *DL-Lite* family [2], whose basic members are tractable fragments of the OWL standard[1]. In a nutshell, *DL-Lite* allows to capture the basics of ontology languages and conceptual modeling formalisms used in software engineering, such as Entity-Relationship (ER) and UML class diagrams.

On the other side, assuming that data is stored in an ABox had quickly turned out to be unrealistic in practice. Indeed, organizations actual data reside in their information systems, and are typically managed by commercial DBMSs. Hence, assuming data to be stored in an ABox would require the organization either to restructure its information system in order to make it compliant with the ontology, or to devise an *Extract-Transform-Load* process. Clearly, both these solutions would be extremely expensive in terms of initial investment, as well as of the overall information system efficiency.

Some exceptions among works on OBDA, before 2011, are [11] and [4], which both address the problem of accessing through an ontology pre-existing data actually serving specific applications, based on a set of mappings from the data sources to the ontology. These works are at the origin of MASTRO, the OBDA system whose experimentation we are reporting here. Still, they are preliminary and present various limitations that required to be overcome, as we will discuss in the next section.

As for OBDA experimentation, in 2011, only few projects had been carried out, which are reported in [1] and in [13], and both used MASTRO for OBDA. However, these projects had the objective of providing a proof of concept of the OBDA approach, rather than experimenting the underlying OBDA system in a real-world scenario.

As we said before, the aim of this paper is to report on a comprehensive experimentation of the OBDA approach. In particular, the focus is on describing the various steps we have out in order to apply the research results in a real-world settings, and the new techniques and tools we had to devise according to the peculiar needs of such setting.

2 The Scenario

The Second Directorate of the Department of Treasury, *a.k.a.* the *Public Dept Directorate*, is responsible for the following matters: issuance and management of the public debt, liquidity management, management of the government securities amortization fund, analysis of the problems inherent to the management of the public debt at both national and international level and to the functioning of the financial markets, coordination and supervision of the access to the financial markets by public entities.

The share of greater significance is however the Central Administration debt, which consists, for the most part, of securities issued on the domestic market. The Public Dept Directorate is in turn organized into offices that deal with specific components, each of which has solved its information problems with dedicated applications, none of which can represent the whole issue of the public debt. Furthermore, the euro area debt crisis

[1] `http://www.w3.org/TR/owl2-overview/`

that hit Italy in last years has led the Public Dept Directorate to introduce frequent innovations in the market of government securities, both in terms of new securities offered to expand the demand for risk diversification and greater penetration of the market, which required a greater monitoring of the secondary market. These frequent innovations put under stress the information systems, unable to respond quickly to contingent needs.

Within the above scenario, the adoption of OBDA has several motivations:

- Although each sub-unit of the department has a clear understanding of a particular portion of the public debt domain, a shared and formalized description of the relevant concepts and relations in the whole domain is missing. This lack means that, when different offices manage the same information according to different purposes, is not uncommon a misunderstanding about the nature and theoretical definition of data.

- Data within the various sources are managed by different systems, and their structure has been heavily modified and updated during the years, often to serve specific application needs. The result is that the original modeling of the data is hidden in the data structures currently used by applications and processes. Consequently, only few experts of the systems have the skill to access data, while domain users access the information system only by means of pre-defined queries. Hence, when a new information need arises, managers of the Department of Treasury have to resort to complex processes, typically requiring several weeks and a considerable investment to be accomplished.

- Integrity constraints on the data are often not enforced in the running systems, mainly because of performance reasons. Business rules are therefore managed within software processes. The result is that the data quality is hampered, or difficult to assess.

3 The Ontology Design

The ontology we realized formalizes the overall scenario we sketched in the previous section. It is specified over an alphabet containing around 350 concept names, 150 role names, 200 attribute names, and contains around 3000 *DL-Lite* axioms.

Our experience proved that (i) on the one hand, the basic logics of the *DL-Lite* family [2, 11], i.e., tractable fragments of the OWL language, allowed us to model the main aspects of the domain, and (ii) on the other, axioms not allowed in OWL were indeed necessary to fully capture some specific characteristics of our project. As for the non-OWL constructs we used, we notice that *identification assertions* were particularly useful for both imposing important domain constraints and modeling role attributes or n-ary relations through the well-known reification technique. As for the first aspect, we needed, for example, to impose that a bond is uniquely identified by its ISIN (International Securities Identification Number). As for the second aspect, we recall that OWL and *DL-Lite* do not allow for the specification of role attributes or n-ary relations, which are instead very common in the practice. Reification is a means to overcome this limitation, since it allows to use an object to denote a tuple of objects. As an example, consider shared loans, which are special loans associated with several institutions that are the loan borrowers, each one holding a particular percentage of the loan. This situation could be modeled using a role attribute, but in languages without such constructs, it must be represented by the reified concept entitlement, to which the percentage attribute

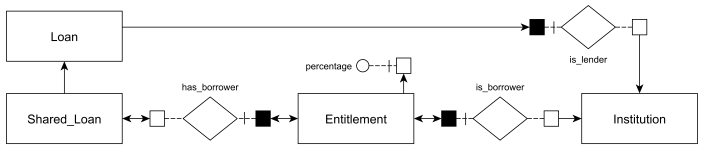


Fig. 1. Graphical representation of an excerpt of the ontology

can be associated. The resulting modeling is graphically represented in Figure 1[2], which is a simplified excerpt, translated in English, of the overall ontology. However, in order for this reified representation to be correct, we must also impose that no two distinct instances of Entitlement exist that are associated through the roles has_borrower and is_borrower with the same pair of instances for Shared_Loan and Institution, respectively. This can be indeed specified through an identification assertion.

As shown by the theoretical studies in [3], identification assertions can be smoothly added to languages of the *DL-Lite* family, still guaranteeing first-order rewritability of query answering. Then, our experience at the Department of Treasury confirmed how crucial is to use such assertions in the practice.

Continuing the above example, consider that institutions may act as both lenders and borrowers of loans (cf. role is_lender in Figure 1). For example, a region can be beneficiary of a loan guaranteed by a credit institution, but also lender of loans whose borrowers are the municipalities located in the region. Of course, we do not want an institution to be at the same time lender and borrower of the same loan. This can be specified through a *denial assertion*, which imposes that the boolean query

$$\mathsf{has\_borrower}(x, y), \mathsf{is\_borrower}(y, z), \mathsf{is\_lender}(z, x)$$

has to be evaluated to false on every ontology instance. Again, this feature is not allowed by ontology languages such as OWL, but important in real-world use cases. The needs encountered in our experience pushed therefore the theoretical study of denial assertions in *DL-Lite* languages. In [8] we indeed show that it is possible to use them in conjunction with all other features of *DL-Lite* and maintaining query answering first-order rewritable. Consequently, support to the use of denial assertions has been incorporated in the MASTRO system.

4 Mapping the Ontology to the Data Sources

In MASTRO, the mapping is a set of assertions, each one associating an element of the ontology with an SQL query over the sources, which actually corresponds to GAV mapping in the data integration terminology [6]. This form of mappings, combined with an ontology specified in *DL-Lite*, even equipped with identification and denial assertions, guarantees that query answering is first-order rewritable, i.e., is reducible to evaluation of an SQL query over the sources. The possibility of exploiting the full

[2] The graphical language we adopt resembles ER diagrams, but it provides the right flexibility to represent *DL-Lite* ontologies. For further details, we refer to a technical report available at `http://www.dis.uniroma1.it/~savo/mef/GraphicalLanguage.pdf`.

power of SQL in mapping assertions turned out to be crucial in our experience at the Department of Treasury. Indeed, the structure of the data at the sources has been heavily modified during the years, often to serve specific application needs. The result is that the original modeling of the data is hidden in the form in which data are currently organized, and the "distance" between such form and the conceptual representation of the domain provided by the ontology is critically difficult to bridge. In this respect, our experience showed that mapping definition has to be essentially carried out manually.

Within our project, we mapped a portion of the ontology through the definition of 800 mapping assertions, which involve 80 relational tables stored in various separate databases, all managed by Microsoft SQL Server. These databases contain around 250 tables, storing approximately 5 million tuples, for an overall size of 2.7 gigabytes.

In order to define mappings, we started by deeply analyzing the structure of the data sources, to understand the meaning of relational tables they store and the dependencies among them. This has been a very time consuming phase of our project, due to the lack of accurate documentation, at least on some of the sources we considered. On the other hand, such an analysis has to be carried out only once during an OBDA project, and faced again only in case new sources need to be added to the OBDA system.

From our analysis of the sources clearly emerged that data about distinct ontology concepts were often mixed together within the sources. This regards, for instance, financial instruments, characterized by properties like the type, the maturity, or the interest rate they offer, and the amount of the actual debt they produce. As a specific example consider the mapping assertion below.

```
SELECT L.ID FROM D_LOAN L
WHERE NOT EXISTS (SELECT *
          FROM D_SHARED S                ⇝ Unshared_Loan(ln(ID))
          WHERE S.LOAN_ID=L.ID)
```

In the assertion, we select at the sources identifications of unshared loans, i.e., loans with only one borrower, and map them to the corresponding concept in the ontology, i.e., Unshared_Loan, which, together with Shared_Loan, partitions the concept Loan. Since the table D_LOAN contains both shared and unshared loans, we obtain IDs of unshared loans as the difference between all loans and shared ones, all occurring in the table D_SHARED. In the right-hand side of the mapping, **ln** is, intuitively, a function symbol used to construct objects denoting unshared loans starting from IDs (cf. [11]).

5 OBDA Services

The ultimate goal of our project is to provide the user with various services for accessing the data through the lens of the ontology. In this section, we briefly describe how MASTRO has been used and improved for providing an effective query answering service in our scenario. To this aim, note that query answering in MASTRO is divided into two steps: *(i) ontology rewriting*, where the original query is rewritten with respect to the ontology into a new query; *(ii) mapping rewriting*, where the query thus obtained is reformulated over the source database schema using the mapping assertions.

While in recent years we have seen many approaches to the ontology rewriting step and its optimizations, (see, e.g., [2, 12]) very little has been done towards the optimization of the mapping rewriting step. At the beginning of our experimentation, mapping rewriting was mainly realized through unfolding, i.e., by a method based on substituting every ontology element in the ontology query with the source query associated to it by the mapping. However, the very first experiments with this method showed that a pure unfolding method is too expensive. For instance, if the query to be rewritten has 10 atoms, and the predicate of each atom is mapped to 4 SQL source queries, then the unfolding produces a rewritten query which is the union of 4^{10} SQL queries, i.e., over 1 million SQL disjuncts. That is, even if the ontology query is not very large, the size of the rewriting might be too large to be handled by any SQL engine.

The first experiments of computing query answering on our scenario showed that the above mentioned combinatorial explosion could be often avoided. Indeed, we noticed that in many cases, several subqueries of the final union of SQL subqueries were actually redundant, i.e., contained into other subqueries. So, we reached the conclusion that in principle it is possible to optimize the mapping rewriting phase by limiting the combinatorial explosion, in particular if we could exploit the knowledge on containment between the SQL subqueries used in the mapping. Note that checking containment of arbitrary SQL queries is an undecidable problem. However, even sound and incomplete containment checking algorithms (i.e., algorithms that do not guarantee to discover all containments) make sense in this context. Indeed, discovering even a small number of containments between SQL subqueries can drastically lower the size of the final SQL query, thus making query answering over the OBDA system feasible.

Based on the above considerations, our experience in real world OBDA scenarios led us to realize the following optimizations:

1. We added *view inclusions* to the OBDA specification, i.e., inclusion assertions between the SQL queries used in mapping assertions, or, more precisely, between the corresponding view predicates. Based on such inclusions, we were able to eliminate subqueries contained into other subqueries of the rewritten query.

2. Given that the above optimization is applied only after generating the unoptimized mapping rewriting of the query, we have designed a further optimization of the whole query rewriting process. Intuitively, this has been realized through a kind of semantic caching approach. When the algorithm discovers that the query to be answered contains subqueries that have been already processed in previous executions, and for which the mapping rewriting has been stored, it directly uses such mapping rewritings to reformulate the query at hand, thus saving time.

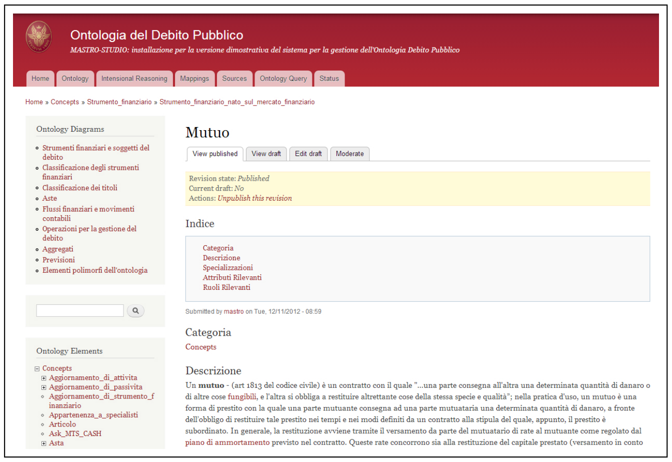
With the above optimizations, we have been able to provide the users with an effective query answering service,which allows the users to obtain various reports on the data stored in the sources by directly querying the ontology. For details see [5].

6 Supporting Ontology and Mapping development

Dealing with a complex real world scenario like the Italian Department of Treasury helped to identify several critical issues that are general enough to be taken into account in the OBDA approach regardless of the organization, and that can be summarized in: (i) communication with the experts of the domain of interest; (ii) development and refinement of the ontology; (iii) documentation of the ontology.

As for the first aspect, we notice that developing an ontology requires to exchange knowledge with people that are generally not expert of logical formalisms, but that have a deep understanding of the domain of interest. Exchanging this kind of knowledge requires a common tool that is understood on both parts. A useful solution to this problem was the adoption of a novel *graphical language* for ontology representation (cf. Fig. 1). According to this new formalism, the graphical representation of the ontology has a graph-like structure, similar to that of an ER diagram, that is both easy to understand and also easy to translate into other logical formalisms suitable for automatic reasoning, like OWL and *DL-Lite*.

This approach was an improvement also for ontology development and refinement, since it effectively supports the definition, the update, and the analysis of the ontology. In particular, the analysis of the ontology is crucial both for validating and for identifying mistakes. To this aim, alongside with the tool for the inspection of the graphical representation of the ontology, we developed a novel tool for intensional reasoning, to verify formal properties of the ontology, such as concept satisfiability.



Ontologia del Debito Pubblico

MASTRO-STUDIO: installazione per la versione dimostrativa del sistema per la gestione dell'Ontologia Debito Pubblico

Home » Concepts » Strumento_finanziario » Strumento_finanziario_nato_sul_mercato_finanziario

Ontology Diagrams

- Strumenti finanziari e soggetti del debito
- Classificazione degli strumenti finanziari
- Classificazione dei titoli
- Aste
- Flussi finanziari e movimenti contabili
- Operazioni per la gestione del debito
- Aggregati
- Previsioni
- Elementi polimorfi dell'ontologia

Ontology Elements

- Concepts
  - Aggiornamento_di_attivita
  - Aggiornamento_di_passivita
  - Aggiornamento_di_strumento_finanziario
  - Appartenenza_a_specialisti
  - Articolo
  - Ask_MTS_CASH
  - Asta

Mutuo

Revision state: *Published*
Current draft: *No*
Actions: *Unpublish this revision*

Indice

Categoria
Descrizione
Specializzazioni
Attributi Rilevanti
Ruoli Rilevanti

Submitted by mastro on Tue, 12/11/2012 - 08:59

Categoria

Concepts

Descrizione

Un **mutuo** - (art 1813 del codice civile) è un contratto con il quale "...una parte consegna all'altra una determinata quantità di danaro o di altre cose fungibili, e l'altra si obbliga a restituire altrettante cose della stessa specie e qualità"; nella pratica d'uso, un mutuo è una forma di prestito con la quale una parte mutuante consegna ad una parte mutuataria una determinata quantità di danaro, a fronte dell'obbligo di restituire tale prestito nei tempi e nei modi definiti da un contratto alla stipula del quale, appunto, il prestito è subordinato. In generale, la restituzione avviene tramite il versamento da parte del mutuatario di rate al mutuante come regolato dal piano di ammortamento previsto nel contratto. Queste rate concorrono sia alla restituzione del capitale prestato (versamento in conto

Fig. 2. MASTRO STUDIO GUI screenshot

As for the ontology documentation issue, we resorted to a structured wiki-like documentation (one wiki-page for each ontology concept, attribute and role), where various contributions are gathered together with the help of collaborative tools. This enables the cooperation of all the parts participating to the ontological analysis.

In order to tackle the above mentioned issues, during this project the collection of OBDA services offered by MASTRO evolved into a more complex suite of tools, called MASTRO STUDIO. MASTRO STUDIO is a web-application based on Drupal[3], an open source CMS (Content Management System), and thus comprises: *(i)* Drupal core modules; *(ii)* contributed Drupal modules, for the management and the moderation of collaborative editing of the ontology wiki-like documentation; and *(iii)* custom modules (i.e., extensions of the CMS) for the loading and the analysis of the OBDA specification, as well as the invocation of reasoning services over it, and the analysis of their results.

Beside the web-gui, MASTRO STUDIO offers several utilities, including a tool for the translation of graphical representations of ontologies into the OWL functional-syntax representation required by components of the reasoning layer, and a tool for the automatic generation and update of the wiki-like documentation, starting from the OBDA specification. An example of wiki-like documentation page (in italian) of the concept *Loan* is shown in Figure 2. Finally, MASTRO STUDIO provides a reasoning

[3] `http://drupal.org`

layer, which comprises modules that invoke and exploit MASTRO reasoning services, through a web-service interface.

7 Conclusions

We have illustrated how the application of the OBDA principles in a real-world scenario have led us to devise new techniques for making the whole approach effective. It is our opinion that this is one of the most important outcome of the project: on the one hand, the Department of Treasury benefits from the effective use of a new technology in various critical tasks, and on the other hand, the experience has guided the research activities towards the goal of solving real-world problems arising in practice.

Acknowledgments. The research on OBDA, and in particular on mapping management, at the basis of the experience described in this paper has been partially funded by the EU under FP7 project Optique – Scalable End-user Access to Big Data (grant n. FP7-318338).

References

1. A. Amoroso, G. Esposito, D. Lembo, P. Urbano, and R. Vertucci. Ontology-based data integration with MASTRO-I for configuration and data management at SELEX Sistemi Integrati. In *Proc. of SEBD 2008*, pages 81–92, 2008.
2. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
3. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Path-based identification constraints in description logics. In *Proc. of KR 2008*, pages 231–241, 2008.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, and M. Ruzzi. Using owl in data integration. In *Semantic Web Information Management - A Model-Based Perspective*, pages 397–424. Springer, 2009.
5. F. Di Pinto, D. Lembo, M. Lenzerini, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, and D. F. Savo. Optimizing query rewriting in ontology-based data access. In *Proc. of EDBT 2013*, 2013.
6. A. Doan, A. Y. Halevy, and Z. G. Ives. *Principles of Data Integration*. Morgan Kaufmann, 2012.
7. R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to query answering in *DL-Lite*. In *Proc. of KR 2010*, pages 247–257, 2010.
8. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant first-order rewritability of DL-Lite with identification and denial assertions. In *Proc. of DL 2012*, volume 846 of *CEUR,* `ceur-ws.org`, 2012.
9. M. Lenzerini. Ontology-based data management. In *Proc. of CIKM 2011*, pages 5–6, 2011.
10. H. Pérez-Urbina, I. Horrocks, and B. Motik. Efficient query answering for OWL 2. In *Proc. of ISWC 2009*, volume 5823 of *LNCS*, pages 489–504. Springer, 2009.
11. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
12. M. Rodriguez-Muro and D. Calvanese. High performance query answering over *DL-Lite* ontologies. In *Proc. of KR 2012*, pages 308–318, 2012.
13. D. F. Savo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, V. Romagnoli, M. Ruzzi, and G. Stella. MASTRO at work: Experiences on ontology-based data access. In *Proc. of DL 2010*, volume 573 of *CEUR,* `ceur-ws.org`, pages 20–31, 2010.