Experimenting Ontology-based Data Access with MASTRO (Extended Abstract)

Domenico Fabio Savo[1], Domenico Lembo[1], Maurizio Lenzerini[1], Antonella Poggi[1], Mariano Rodriguez-Muro[2], Vittorio Romagnoli[3], Marco Ruzzi[1], Gabriele Stella[3]

[1] SAPIENZA Università di Roma
lastname@dis.uniroma1.it

[2] Free University of Bozen-Bolzano
rodriguez@inf.unibz.it

[3] Banca Monte dei Paschi di Siena
firstname.lastname@banca.mps.it

1 Introduction

While the amount of data stored in current information systems continuously grows, turning these data into information is still one of the most challenging tasks for IT. Specifically, the information systems of medium and large organizations are typically constituted by several, independent, and distributed data sources, and this poses great difficulties with respect to the goal of accessing data in a unified and coherent way. Such a unified access is crucial for getting useful information out of the system. This explains why organizations spend a great deal of time and money for the understanding, the governance, and the integration of data stored in different sources [5].

Below we report some reasons that make unified data access problematic: (i) despite the fact that the initial design of a collection of data sources (e.g., a database) is adequate, corrective maintenance actions tend to re-shape the data sources into a form that often diverges from the original conceptual structure; (ii) it is common practice to change a data source so as to adapt it both to specific application-dependent needs, and to new requirements, thus obtaining data sources strictly coupled to a specific class of applications; (iii) the data stored in different sources tend to be redundant, and mutually inconsistent, mainly because of the lack of central, coherent and unified data management tasks.

In principle, there are two alternative solutions to the above problems. One solution is the re-engineering of the whole information system, an approach clearly unfeasible in many situations, due to cost and organization problems. The other solution is to create a new stratum of the information system, posed over the data sources. Such new stratum is constituted by (i) a global (also called "mediated") schema, representing the unified structure presented to the clients, and (ii) the mapping relating the source data with the elements in global schema. One of the methods to realize such stratum resorts to a virtual approach in which data are not moved, and queries posed to the global schema are answered by suitably accessing the sources [6]. Such an approach, which is the one referred to in this work, is preferable in a dynamic scenario to the approach in which data are materialized in the global schema (a.k.a., datawarehousing), since sources may be updated frequently, and clients want to use up-to-date information.

To achieve this goal we recently proposed the notion of *ontology-based data integration*, also called *ontology-based data access* (OBDA) when data are managed by the same kind of data management systems, and are not distributed [1,8]. The basic idea

of OBDA is to express the global schema as an ontology, that is a semantically rich description of the relevant concepts and relationships in the domain of interest, with the mapping acting as the reconciling mechanism between the conceptual level and the data sources. Moreover, OBDA also exploits reasoning on the ontology in computing the answers to queries.

We point out that a distinguishing feature of OBDA is the possibility of connecting external and autonomous relational databases to an independently developed ontology, which makes OBDA different from other works that use a relational database to store the extensional level of an ontology [7,4]. Also, in such works query answering needs a data pre-processing step, which is not in general possible in OBDA, where query answering is completely intensional.

In this paper we report on an experimentation of OBDA carried out in a joint project by Banca Monte dei Paschi di Siena (MPS)[1], Free University of Bozen-Bolzano, and SAPIENZA Università di Roma, where we used MASTRO [8] for accessing, by means of an ontology, a set of data sources from the actual MPS data repository. MASTRO is an OBDA system extending the QUONTO[2] reasoner, which is based on *DL-Lite*$_{A,Id}$ [1], one of the logics of the *DL-Lite* family [3]. The OBDA scenario refers to a set of 12 relational data sources, collectively containing about 15 million tuples. The ontology comprises 79 concepts and 33 roles, and is expressed in terms of approximately 600 *DL-Lite*$_{A,Id}$ axioms. The relationships between the ontology and the sources are expressed in terms of about 200 mapping assertions. The results of the experimentation can be summarized as follows.

- In the context of MPS scenario, our approach has addressed many of the conceptual modeling and data access issues mentioned before: by using *DL-Lite*$_{A,Id}$ we fulfilled some lacks of the standard Web Ontology Language OWL 2 by using identification constraints and epistemic queries.
- MASTRO has shown very good performance in all the reasoning tasks, including query answering, which is the most important service required in the application.
- The experience in this project has shown that OBDA can be used for checking the quality of data sources. There are basically two kinds of data quality problems that our system is able to detect, one related to unexpected incompletenesses in the data sources, and the other one related to inconsistencies present in the data.
- Our work has pointed out the importance of the ontology itself, as a precious documentation tool for the organization. Indeed, the ontology developed in our project is currently used in MPS as a common specification of the relevant concepts in the organization.

The paper is organized as follows. Section 2 presents a brief description of MASTRO. Sections 3 illustrates the scenario of our experimentation. Section 4 presents the ontology and the mappings designed within the project. Section 5 illustrates the use of MASTRO in the scenario. Section 6 concludes the paper. The present work is an extended abstract of [9].

[1] MPS is one of the main banks, and the head company of the third banking group in Italy (see `http://english.mps.it/`).

[2] `http://www.dis.uniroma1.it/quonto/`

2 The MASTRO system

MASTRO is an OBDA system jointly developed at the SAPIENZA University of Rome and Free University of Bozen-Bolzano. MASTRO allows for the definition of *DL-Lite*$_{A,Id}$ ontologies connected through semantic mappings to external independent relational databases storing data to be accessed. Thus, differently from the common tools for ontology definition and reasoning, the extensional level of the ontology, namely, the instances of concepts and roles, are not explicitly asserted, but are specified by mapping assertions describing how they can be retrieved from the data at the sources. In the following we briefly sketch the architecture of the system, distinguishing between "Ontology Definition Module", "Mapping Manager", "Data Source Manager", and "Reasoner".

The *Ontology Definition Module* provides mechanisms for the specification of the ontology as a *DL-Lite*$_{A,Id}$ terminology (i.e., a TBox [3]). *DL-Lite*$_{A,Id}$ is a Description Logic (DL) belonging to the *DL-Lite* family, which adopts the Unique Name Assumption, and provides all the constructs of OWL 2 QL[3], a tractable profile of OWL 2, plus functionality and identification assertions, with the limitation that these kind of assertions cannot involve sub-roles. These features, while enhancing the expressive power of the logics, do not endanger the efficiency of both intensional reasoning, and query answering. In other words, the computational complexity of these tasks is the same as in OWL 2 QL, namely PTIME with respect to the size of the TBox, and LOGSPACE in the size of the data at the sources.

The *Mapping Manager* supports the definition of mapping assertions relating the data at the sources to the concepts in the ontology. The mapping assertions supported by MASTRO are a particular form of sound GAV mappings [6]. More specifically, a mapping assertion is an expression of the form $\psi \rightsquigarrow \varphi$ where ψ is an arbitrary SQL query over the database, and φ is a *DL-Lite*$_{A,Id}$ conjunctive query without existential variables. As described in [1], data extracted by means of query ψ are used, together with suitable Skolem functions, to build the logic terms representing the object identifiers, thus solving the impedance mismatch problem between data at the sources and instances of the ontology (see Section 4 for examples). The Mapping Manager interacts with the *Data Source Manager*, which is in charge of the management of the underlying relational sources, providing transparent access to a wide range of both commercial and freeware relational DBMSs[4].

Finally, the *Reasoner* exploits both the TBox and the mapping assertions in order to (i) check the satisfiability of the whole OBDA system, and (ii) compute the answer to the queries posed by the users. Such module is based on QUONTO, a reasoner for the *DL-Lite* family that uses query rewriting as a main processing technique. The MASTRO process to answer conjunctive queries is inspired by the one implemented in the QUONTO system. First, the query posed by the user over the ontology is reformulated according to the inclusion assertions expressed among concepts and roles; second, such rewriting is *unfolded* according to the mapping assertions in order to generate an SQL query which can be directly issued over the relational data source. It can be shown

[3] Main constructs of OWL 2 QL are inclusions between concepts and properties, which allow for modeling ISA and disjointness between concepts and properties, role typings, mandatory participation constraints, etc. (cf. `http://www.w3.org/TR/owl2-profiles/`)

[4] Non-relational data sources can be accessed by means of suitable wrapping tools.

that the answers to such an SQL query are exactly the answers logically implied by the OBDA system [1]. As a further powerful feature, MASTRO is able to answer EQL (Epistemic Query Language) queries [2], i.e., first-order logic queries over the ontology interpreted under an epistemic semantics. Finally, MASTRO provides the consistency check capability. By virtue of the characteristics of *DL-Lite*$_{A,Id}$, MASTRO reduces consistency check of the whole OBDA system to verifying whether queries generated for disjointness and functionality assertions or identification (IDCs) and epistemic constraints (i.e., specified as boolean EQL queries) return an empty result. To this aim, a boolean query is automatically generated for every such construct and then rewritten, unfolded and evaluated over the database.

3 Case study: The domain of experimentation

The data of interest in our case study are those exploited by MPS personnel for risk estimation in the process of granting credit to bank customers. A customer may be a person, an ordinary company, or an holding company. Customers are ranked with respect to their credit worthiness, which is established considering various circumstances and credit/debit positions of customers. In addition to customer information, data of interest regard company groups to which customers belong, and business relations between bank customers (in particular, fifteen different kinds of such relations are relevant). Hereinafter, such groups of customers will be called *Clusters of Connected Customers (CCCs)*. A 15 million tuple database, stored in 12 relational tables managed by the IBM DB2 RDBMS, has been used as data source collection in the experimentation. Figure 1 shows a summary of the data sources. Such data sources are managed by a specific application which is also in charge of guaranteeing data integrity (in fact, the underlying database does not force constraints on data). Not only this application performs various updates, but an automatic procedure is executed on a daily basis to examine the data collected in the database so as to identify connections between customers which are relevant for the credit rating calculus. Based on these connections, customers are grouped together to form CCCs. For each cluster, several data are collected that characterize the kinds of connections holding among cluster members (i.e., specifying juridical, economic, or financial aspects of connections).

Source name	Source Decription	Source size
GZ0001	Data on customers	3.463.083
GZ0002	Data on juridical connections between customers	157.280
GZ0003	Data on guarantee connection between customers	1.270.333
GZ0004	Data on economical connections between customers	104.033
GZ0005	Data on corporation connections between customers	1.021.779
GZ0006	Data on patrimonial connections between customers	809.321
GZ0007	Data on company groups	55.362
GZ0012	Customers loan information	5.966.948
GZ0015	Data on monitoring and reporting procedures	1.243
GZ0101	Data on membership of customers into CCCs	2.225.466
GZ0102	Information on CCCs	663.656
GZ0104	Data on bank credit coordinators for juridical CCCs	38.457

Fig. 1. Data sources

Data source schemas have undergone many changes in the years, trying to adapt to the changes in the application. The result is a stratification of the data source which causes an extended use of control fields, validity flags, and no longer used attributes in the source schemas. Consequently, an increasing effort for the management of the data sources is required, which has to be completely entrusted to the management applications rather than the domain experts. The aim of the experimentation was to show the effectiveness of the OBDA approach in providing a means to get useful information from the data through a conceptual specification of the domain.

4 Case study: ontology, mappings, and methodology

The process that led us to realize the OBDA system for the MPS case study has been carried out essentially in two main phases: in the first one, we have developed the ontology, whereas in the second one we have specified the mapping between the ontology and the data sources.

To be as much independent as possible from the actual source database, in the first phase we carried out an in-depth analysis of the business domain following a top-down approach. Therefore, after identifying the central concepts and the main relations between them, we iteratively refined the ontology, being supported in each development cycle by the experts from MPS. The top-down approach turned out to be fundamental for the success of the entire project, since in this way we were able to avoid that the data model provided by the schema of the data sources could affect the definition of the ontology, thus achieving complete separation between the conceptual layer and the logical/physical layer of the system. In fact, further information on the model coming from the analysis of the sources has been exploited only towards the end of the design process, in order to refine the realized ontology.

The final ontology comprises 79 concepts, 33 roles, 37 concept attributes, and is expressed in terms of about 600 *DL-Lite*$_{A,Id}$ axioms, including 30 identification constraints (IDCs), plus 20 EQL constraints (EQLCs). In Figure 2, we provide an excerpt

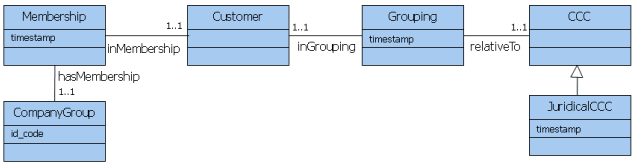

Fig. 2. Excerpt of the ontology (UML adaptation)

of the ontology, represented as UML class diagram. In this portion of the ontology, we model grouping of customers into CCCs (possibly Juridical CCCs), and their membership into company groups.

In the following, we report on a series of modeling issues we dealt with during the ontology definition phase. First, we observe that in the domain we have analyzed, several properties of individuals depend on time. It has been therefore necessary in the ontology to take trace of the changes of such properties, maintaining the information on the validity periods associated with each such change. Even though from a very abstract point of view, such properties might be considered roles or attributes, to properly model the temporal dimension, each such role or attribute needed to be in fact *reified* in the ontology. A timestamp attribute has been associated to each concept introduced by the reification process, together with a suitable identification constraint ensuring that no two instances of each such concept refer to the same period of time.

Example 1. The grouping of a customer into a cluster of connected customers is a time-dependent notion which is associated with a validity period (i.e., a timestamp). A crucial requirement is that a customer is not grouped into two clusters at the same time. To impose this, we specified over the ontology represented in Figure 2 the following identification constraint $(id\ \ Grouping\ \ inGrouping^{-}, timestamp)$ which

imposes that no two distinct instances of $Grouping$ exist that are connected to the same pair constituted by a value for the attribute $timestamp$ and an object filler for $inGrouping^-$[5]. The concept *Grouping* can bee seen as the reification of the notion of the grouping of a customer into a CCC. Without the reification and the use of an IDC, this constraint could have not been specified. Analogously, the IDC $(id\ Membership\ inMembership^-, timestamp)$ specifies that a customer is never member of two company groups at the same time.

Identification constraints turned out to be an essential modeling construct, not only for correctly modeling the temporal dimension through reification, but also for expressing important integrity constraints over the ontology that could not be captured otherwise.

Globally, we have specified more than 30 IDCs in the ontology. None of these presently correspond to integrity constraints at the data sources. This is because, as it is usual in practice, very few integrity constraints are explicitly asserted at the sources. Thus, our ontology plays an important role in representing business rules not explicitly reflected in the data repository of the organization.

EQLCs turned out to be another important means for correct domain modeling. Such constraints indeed permit to overcome some expressiveness limitations of *DL-Lite*$_{A,Id}$, without causing any computational blow up, since they are interpreted according to a suitable semantic approximation (cf. Section 2). In this experimentation we have heavily used EQLCs to express, e.g., completeness of hierarchies and other important business constraints, otherwise not expressible in our ontology. It is worth noticing that EQLCs, even though under the semantic approximation mentioned above, allow for imposing constraints that cannot be even specified in OWL 2, despite its expressiveness. Indeed, through EQLCs one can express constraints that involve general forms of joins over ontology predicates, which are typical of query languages but are not allowed in classical ontology languages (cf. [9]).

Let us now turn our attention to mapping specification. The mapping specification phase has required a complete understanding and an in-depth analysis of the data sources, that highlighted some modeling weaknesses present in the source database schema: various modifications stratified in the years over the original data schema have partially transformed the data sources, which now reveal some problems related to redundancy, inconsistency, and incompleteness in the data. Localizing the right data to be mapped to ontology constructs has thus required the definition of fairly complex mapping assertions, as shown in Example 2.

Example 2. Consider the following mapping assertion specifying how to construct instances of *JuridicalCCC* using data returned by an SQL query accessing both the table $GZ0102$, which contains information about CCCs, and the table $GZ0007$, which contains information about the company groups.

```
SELECT id_cluster, timestamp_val FROM GZ0102, GZ0007
WHERE GZ0102.validity_code = 'T' AND GZ0102.id_cluster <> 0
     AND GZ0007.validity_code = 'T' AND GZ0007.id_group <> 0
     AND GZ0102.id_cluster = GZ0007.id_group
```

$\rightsquigarrow\ JuridicalCCC(ccc(id\_cluster, timestamp\_val))$

[5] $inGrouping^-$ denotes the inverse of the role $inGrouping$, which is defined from the concept $Customer$ to the concept $Grouping$

From the data source analysis it turned out that each CCC that has an identifier ($GZ0102.id\_cluster$) coinciding with the identifier of a company group ($GZ0007.id\_group$) is a juridical CCC. Such a property is specified in the SQL query in the mapping through the join between $GZ0102$ and $GZ0007$ ($GZ0102.id\_cluster = GZ0007.id\_group$). Notice that invalid tuples (those with $validity\_code$ different from 'T') and meaningless tuples (those with $id\_cluster$ or id_group equal zero) are excluded from the selection. The query returns pairs of $id\_cluster$ and $timestamp\_val$, which are used as arguments of the function $ccc()$ to build logic terms representing objects that are instances of *JuridicalCCC*, according to the method described in [1].

The mapping specification phase has produced around 200 mapping assertions, many of which are quite involved. Their design has been possible by a deep understanding of the tables involved, their attributes, and the values they store. We initially tried to automate this process with the help of current tools for automatic mapping generation, but, due to the complexity of extracting the right semantics of the source tables, we failed. This is in line with our past experience on mapping design: the bulk of the work in mapping specification has to be essentially carried out manually.

5 The system at work

In this section we concentrate on two crucial aspects of our experience: the use we made of MASTRO to check the quality of the data sources, and the impact that certain characteristics of the MPS scenario have had on the evolution of the system in terms of its tuning and optimizations.

As mentioned in the introduction, we faced two main issues concerning the quality of the data sources, namely incompleteness and inconsistency. Detecting data incompleteness has been possible by exploiting the MASTRO query answering services, and more precisely, by inspecting the rewriting and the unfolding that MASTRO produces in the query answering process. Surprisingly, using the ontology to obtain all company codes (for example, by means of the query $q(Y) \leftarrow CompanyGroup(X), id\_code(X, Y)$), we actually obtain a larger answer set with respect to that obtained by directly querying the tables of the source database that were assumed to be complete with respect to such information. The reason for such a difference in the answers resides in the fact that the query that MASTRO asks to the source database and unfolding procedures of MASTRO, is much more complex than the query used by the MPS operators. By inspecting the unfolded query produced by MASTRO, it has been indeed possible to find out that some referential integrity constraints that would have made the MPS query complete were in fact missing in the source database.

In *DL-Lite*$_{A,Id}$, inconsistencies are caused by data that violate the assertions of the ontology. Also, causes of inconsistencies can be easily localized by retrieving the minimal set of data that produce each single violation. We actually modified the classical consistency check of MASTRO in order to identify the offending data, in particular exploiting EQL answering features (cf. Section 2) and the ability of EQL to express negation.

We finally notice that to solve some performance issues emerged during the MPS experimentation we revised some procedures exploited by MASTRO for query answering. In particular, we substantially modified the original unfolding technique of MASTRO in such a way that the final SQL query handed to the underlying DBMS assumes

a form that the DBMS query planner can manage in an efficient way. The gain we obtained with respect the previous technique has been enormous, and all the queries of interest for MPS tested during the experimentation have been executed in few seconds. More details on this aspects can be found in [9].

6 Conclusions

From the point of view of MPS, the project has provided very useful results in various areas of interest: (i) Data integration, providing the capability of accessing application data in a unified way, by means of queries written at a logical/conceptual level by end-users not necessarily acquainted with the characteristics of the application; (ii) Database quality improvement, providing tools for monitoring the actual quality of the database, both at intensional and extensional level, (iii)Knowledge sharing, providing, with the ontology-based representation of the domain, an efficient means for communicating and sharing knowledge and information throughout the company.

We are currently continuing our collaboration with MPS, and extending the work to other MPS applications, with the idea that the ontology-based approach could result in a basic step for the future IT architecture evolution, oriented towards Service-oriented architectures and Business Process Management.

References

1. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, and R. Rosati. Ontologies and databases: The *DL-Lite* approach. In S. Tessaris and E. Franconi, editors, *Reasoning Web Summer School 2009*, volume 5689 of *LNCS*, pages 255–356. Springer, 2009.
2. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. EQL-Lite: Effective first-order query processing in description logics. In *Proc. of IJCAI 2007*, pages 274–279, 2007.
3. 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.
4. J. Dolby, A. Fokoue, A. Kalyanpur, L. Ma, E. Schonberg, K. Srinivas, and X. Sun. Scalable grounded conjunctive query evaluation over large and expressive knowledge bases. In *Proc. of ISWC 2008*, volume 5318 of *LNCS*, pages 403–418. Springer, 2008.
5. L. M. Haas. Beauty and the beast: The theory and practice of information integration. In *Proc. of ICDT 2007*, volume 4353 of *LNCS*, pages 28–43. Springer, 2007.
6. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, pages 233–246, 2002.
7. C. Lutz, D. Toman, and F. Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. of IJCAI 2009*, pages 2070–2075, 2009.
8. A. Poggi, M. Rodriguez, and M. Ruzzi. Ontology-based database access with DIG-Mastro and the OBDA Plugin for Protégé. In K. Clark and P. F. Patel-Schneider, editors, *Proc. of OWLED 2008 DC*, 2008.
9. D. F. Savo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, M. Ruzzi, V. Romagnoli, and G. Stella. MASTRO at work: experience on ontology-based data access. In *Proc. of DL 2010*, 2010.