# Ontology-based Data Management for the Italian Public Debt

Natalia Antonioli [a] Francesco Castanò [b] Spartaco Coletta [a] Stefano Grossi [a]
Domenico Lembo [c] Maurizio Lenzerini [c] Antonella Poggi [c] Emanuela Virardi [b] and
Patrizia Castracane [a]

[a] *SOGEI*
[b] *Dip. del Tesoro – Ministero dell'Economia e delle Finanze*
[c] *DIAG – Sapienza Università di Roma*

**Abstract.** In this paper we present an ontology-based data management (OBDM) project concerning the Italian public debt domain, carried out within a joint collaboration between Sapienza University of Rome and the Department of Treasury of the Italian Ministry of Economy and Finance. We discuss the motivations at the basis of this project and present the main characteristics of the ontology we have built. We also describe the mechanisms we used to link the ontology to the actual data and the tools we have adopted for supporting ontology development and maintenance, as well as exploiting OBDM services. Finally, we provide a thorough evaluation of the ontology we produced and discuss in detail the role that it plays within the whole information system of the ministry department responsible for managing Italian public debt data.

## 1. Introduction

*Ontology-based data management* [16] (OBDM) is a new paradigm for accessing, integrating and managing 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, specified in terms of formal descriptions of concepts, binary relations between concepts, and attributes. 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.

The notion of OBDM derives from principles and methodologies developed in the last decades in several different disciplines, including Formal Ontology [19,12,11], Conceptual Modeling [2,13], Data integration [10,15], and Description Logics [1]. However, in the context of the above mentioned disciplines, the main role of the ontology (or, conceptual model) is to provide a formal representation of the domain of interest, used essentially as a design-time artifact with documentation or knowledge-sharing purposes. In these contexts, once the design phase is over, such conceptual model is often compiled into databases and/or software and communication modules, and keeping the model synchronized with the computational resources remains merely a desire. OBDM faces this problem in a fundamental way: the ontology is a run-time artifact that is not compiled, but used and interpreted directly during the operation of the information system. In other words, the ontology becomes the heart of the whole system, which is therefore committed to function coherently with the domain model. This is made possible by virtue of techniques that, given on-the-fly queries or processes formulated over the ontology, use the axioms of the ontology and the mapping to translate such queries/processes into appropriate, concrete operations over the data sources.

In this paper we present what we believe is the first experience of a comprehensive OBDM 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 SOGEI S.pA., an in-house IT company owned by MEF. When we started the project in 2011, the OBDM paradigm was in its very early days. At that time, the studies on OBDM were focused on scenarios in which data are not stored in independent databases but in a so-called *ABox*, a specialized structure representing a set of membership assertions on concepts and relations. Also, they concentrated on designing query answering algorithms, thus dealing with only one aspect of OBDM, called ontology-based data access (OBDA). The outcome of such early investigations on OBDA was twofold.

On the one side, it allowed to single out the ontology language expressivity boundaries for achieving query answering tractability [5,17,14]. In particular, initial studies on OBDA made it clear 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 [5], 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 reorganize the logical structure of the information system in order to make it compliant with the ontology, or to devise an *Extract-Transform-Load* process similar to the one used in Data Warehousing. Clearly, both these solutions would be very expensive in terms of initial investment and overall information system efficiency.

In [18] and [7], the authors point out the need of addressing the case of ontology-based access to pre-existing data sources, based on a set of mappings from the data sources to the ontology. These works follow the tradition of data integration and are at the origin of MASTRO [3], the OBDA system that we use in our experimentations.

As we said before, the aim of this paper is to report on a comprehensive OBDM project. In Section 2 we briefly describe the general scenario of our experimentation, by illustrating the main characteristics of the domain underlying the information system of the Department of Treasury, and the motivations leading to the decision of an OBDM project. In Section 3 we provide an overview of the ontology we have produced. We con-

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

sider a partition of the ontology into seven modules and describe their main characteristics. In Section 4 we presents a description of the mapping between the ontology and the physical data managed by the information system, whereas in Section 5 we describe the techniques and the tools we have devised in order to support the development and the deployment of the ontology in the organization. Then, in Section 6 we present an extensive evaluation of our ontology. We base our evaluation on both logically formalizable and informal criteria, referring, for the latter, to the Ontology Summit 2013 Communiqué[2]. Finally, in Section 7 we conclude the paper by highlighting future developments of the project.

## 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 Debt Central Administration, 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. Before this project, each office had to address its information management problems with specialized applications, none of which was general enough to represent the whole issue of the public debt.

Within the above scenario, various critical aspects of the information systems led to the decision of designing and developing an ontology for the Italian public debt. The following motivations were crucial for this decision.

- Although each sub-unit of the department had 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 was missing. The result was that different offices managed the same information according to different purposes, and this often caused misunderstandings about some, even crucial, business aspects, as well as about the nature and theoretical definition of peculiar information and data over which business management relies.
- From the information system point of view, there was a clear need to coordinate and integrate the data of the various sub-units. Data were indeed managed by different systems, and their structure had been heavily modified and updated during the years, often to serve specific application needs. Consequently, the original modeling of the data was hidden in the data structures used by applications and processes, and only few IT experts had the skill to access data according to a unified view. Domain users were instead forced to access the information system only by means of pre-defined queries. Hence, when a new information need arised, the managers of the Public Debt Directorate had to resort to complex processes, typically requiring several weeks and a considerable investment to be accomplished. The ontology was perceived as a solution

[2] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Communique

to this problems, being a conceptual layer acting as a mediator among various data sources, allowing the users to query integrated data in a flexible way, which is exactly the purpose of OBDA.

- Integrity constraints on the data were often not enforced in the running systems, mainly because of application performance reasons. Business rules were therefore hidden within software processes. The result was that the data quality was hampered, or difficult to assess. What was missing was a unified mechanism allowing the experts to effectively carry out data governance in general, and data quality tasks in particular. The OBDM paradigm was seen exactly as an answer to these problems.
- The euro area debt crisis that hit Italy in last years had 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. In this respect, the ontology was conceived as a powerful means to support the management of requirement changes and to govern the realization of new functionalities, once suitably deployed within the design and development processes.

## 3. The Italian Public Debt Ontology

The *Public Debt Ontology* we developed within this project formalizes the whole domain of the Italian public debt. In a nutshell, it describes both the public debt composition, namely the state liabilities and assets, and the financial instruments used by the Italian public administrations to manage the public debt. Importantly, it provides an historical view of the public debt, by focusing not only on the current state, but also on its evolution through past states. Such evolution is caused by several events, the most important of which are financial transactions.

The portion of the ontology that we are concerned with in this paper is expressed in OWL[3]. It is worth noting, however, that in order to completely and correctly capture the domain, we further resorted to non-OWL assertions, namely identification and denial assertions, as well as expressive integrity constraints (see [4] for details), which all turned to be very useful in practice. The OWL portion of the ontology is specified over a signature containing about 359 concept names, 157 binary relation names, 205 attribute names, and 2738 OWL assertions. The structure of the ontology reflects the partitioning of the domain into 7 related areas, singled out by the domain experts. To each of these areas, we associate a module of the ontology. Such modules are not intended to be specific to a given application. Rather, each of them aims at modeling one specific area of the domain. As a consequence, they can be potentially reused in every application tailored to the corresponding domain area.

In the following, we start by presenting the three modules of the ontology that constitute its core, namely the one for financial instruments, the one for liabilities and assets, and the one for financial transactions. Then we will briefly describe the remaining modules.

[3]In this section, when we mention the ontology elements, we use the Italian names actually occurring in the OWL file representing the official ontology.

*Financial instruments.* One of the main features of the Public Debt Ontology is the modeling of the fundamental distinction between financial instruments and liabilities and assets, which they generate. Initially, this distinction was not clear to the domain expert. And data about financial instruments was not separated from data about liabilities and assets within the information system. However, such a distinction clearly emerged from the interviews and from the deep analysis of the domain concepts, and became soon an overall guideline for the design of the core of the ontology.

In general, a financial instrument represents any contract that transfers money. Thus, the ontology concept `Strumento_finanziario` represents all financial instruments that are used by the Italian public administrations to raise finances. Clearly, several different types of financial instruments are used. Hence, `Strumento_finanziario` is specialized into several subconcepts, according to distinct criteria, among which, notable examples are: *(i)* the direction of the money transfer, e.g., whether it is an investment instrument, which generates an asset, or a raising instrument, which generates a liability; *(ii)* the legal type of the contract defining the instrument, e.g., loan, security, bilateral agreement; *(iii)* the individual debtor, e.g., central or local authority; *(iv)* the terms of interests payment, e.g., with or without coupons. Of course, depending on the specific configuration of criteria by means of which an instrument can be classified, it has a different impact on the public debt. In particular, it can be possibly affected by different types of financial transactions.

Another crucial aspect of financial instruments is that some of their properties, e.g., the expiry date, may change over time, and the ontology has to suitably represent such evolution. To this aim, we resorted to a modeling pattern which relies on the notion of *state*, where a state, intuitively, represents a "snapshot" of a set of evolving features, valid within a given period of time. More precisely, `Strumento_finanziario` is connected to the concept `Stato_di_Strumento_finanziario` through the relation `ha_stato`, where `Stato_di_Strumento_finanziario` represents the set of *past* states of financial instruments. Hence, `Stato_di_Strumento_finanziario` has all the evolving properties of financial instruments, and is identified by the financial instrument to which it is connected, together with the values of the attributes `inizio` and `fine` that indicate, respectively, the start and the end date of the period of validity of the state. Note that properties of `Stato_di_Strumento_finanziario` are also properties of `Strumento_finanziario`. Indeed, the concept `Strumento_finanziario` represents simultaneously the set of financial instruments and their *current* state, i.e. the current snapshot of the set of their evolving properties. By virtue of this choice, a user that is interested only in this current snapshot can avoid to look at the portion of the ontology modeling past states (e.g., can ignore `Stato_di_strumento_finanziario` ), thus simplifying the inspection of the ontology. In other terms, our modeling choice facilitates the construction of a view of the ontology referring only to its current state.

Note that, within the Public Debt Ontology, we used the modeling pattern based on the notion of state for all the elements whose evolution is relevant for the domain. This is the case, for example, of liabilities and assets, which we discuss below.

*Liabilities and assets.* The Italian public debt at a certain point in time is a stock quantity defined as the sum of the amount of all state liabilities at that time, net of the sum of the amount of all assets at the same time. Hence, liabilities, assets, and their evolution are crucial aspects of the public debt domain. Liabilities are represented by the concept

Passivita and are generated by raising financial instruments. Assets are represented by the concept Attivita and are generated by investment financial instruments.

Most of the properties of liabilities and assets evolve. Notable exceptions are *(i)* the financial instrument that generates the liability (respectively, asset), which is represented by (an instance of) the relation genera_passivita (respectively, genera_attivita), which connects Strumento_finanziario to Passivita (respectively, Attivita); and *(ii)* the date of generation of the liability (respectively, asset). Among the properties that evolve, the most relevant is the amount of the liability (or asset). As already mentioned, in order to keep track of the way in which evolving properties change, we used the notion of state, and hence introduced, respectively, the concepts Stato_di_Passivita and Stato_di_Attivita, related respectively to Passivita and Attivita by the relation ha_stato.

Observe that, so far, we described the modeling pattern we used to keep track of financial instruments, liabilities, and assets evolution. However, we have not mentioned yet anything about the events that trigger such evolution. This is precisely what financial transactions, presented next, are about.

*Financial transactions.* The Public Debt management is a complex issue, which is tackled by accomplishing a series of financial transactions, aiming at increasing or decreasing the debt, e.g., by issuing new securities or extinguishing current loans. Within the Public Debt Ontology, we are interested in modeling financial transactions that have an impact on financial instruments, liabilities, and/or assets. Specifically, financial transactions are represented by the concept Operazione, which is specialized into several subconcepts, according to the type of the financial transaction, e.g., security selling, mortgage opening or closing, etc.. Financial transactions are also classified according to the impact they have on the state of liabilities, assets, or financial instruments. Hence, in the ontology, we introduced concepts for modeling transactions that generate, update or extinguish liabilities or assets, as well as concepts for modeling transactions that create, update or close financial instruments. Finally, we introduced a further classification criterion for financial transactions, distinguishing between transactions that have occurred, those that are scheduled, and those that are forecast. Note, in particular, that from the interviews with the domain experts it emerged that financial transactions that are forecast look very similar, in terms of their properties, to transactions that have occurred, by virtue of the fact that a forecast transaction simulates the execution of a transaction.

*Remaining modules.* Besides the main modules described above, the Pubic Debt ontology aims at accurately capturing aspects concerning: *(i)* the actors, e.g., banks or public administrations, represented by the concept Soggetto_nel_debito_pubblico; *(ii)* financial flows and corresponding accounting entries, respectively represented by the concepts Flusso_finanziario and Movimento_contabile; *(iii)* auctions within the primary and the secondary markets, represented by Asta; and, finally, *(iv)* forecastings, represented by Previsione.

## 4. Mapping the Ontology to the Data Sources

According to the OBDM paradigm, the ontology we realized has been suitably linked to selected (source) databases that are part of the information systems currently in use at the Department of Treasury for the management of the Italian public debt. In our framework, such linkage has been specified in terms of a set of mapping assertions, each one associating a conjunctive query over the ontology with an SQL query over the underlying databases (cf. [18]). In the data integration terminology [15,10], this actually corresponds to a form of GAV mappings, which are the most popular kinds of mappings in practical applications of data integration.

The possibility of exploiting the full power of SQL in mapping assertions turned out to be crucial in our project. Indeed, the structure of the source data 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, but that this effort is in general well rewarded. Notably, besides the OBDM services it enables, at the Department of Treasury the mapping now constitues also a precious documentation of the current information system for public debt management.

Within this project, we mapped a large 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.

To define mappings, we started with an accurate analysis of 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 documentation on many of the sources we considered. On the other hand, such an analysis has to be carried out only once during an OBDM project, and faced again only in case new sources need to be added to the system.

Our analysis made it clear that data about distinct ontology concepts were often mixed together within the sources. This concerns, 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 an example, consider the mapping assertion below[4].

```
SELECT L.ID, L.RS,100 AS P
FROM D_LOAN L                                Loan(ln(ID)),
WHERE NOT EXISTS (SELECT *            ⇝     Entitlement(e(ID,RS)),
          FROM D_SHARED S                    percentage(e(ID,RS),P),
          WHERE S.LOAN_ID=L.ID)
```

In the left-hand side of the assertion, we select identifiers (L.ID) of unshared loans, i.e., loans with only one borrower, together with the loan beneficiary who is responsable for it (L.RS), which in this case coincides with the only loan borrower (whereas for shared loans is one among various borrowers). Since the table D_LOAN contains both shared and unshared loans, whereas the table D_SHARED contains all and only shared ones, we obtain data on unshared loans computing the difference between all loans in D_LOAN and shared ones in D_SHARED. In the SELECT clause of the SQL query we additionally specify the constant 100, and refer to it with the alias P. Then, we map tuples returned

[4] In the mapping, original Italian names have been translated in English for presentation purposes.

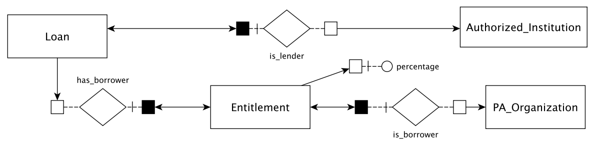


**Figure 1.** Graphical representation of an excerpt of the ontology

by the SQL query to Loan, Entitlement, and percentage, which respectively denote loans and mortgages, the entitlement relative to each loan borrower, and the percentage of such entitlement (cf. the description of the fragment of the ontology containing such concepts given in Section 5 and depicted in Figure 1). Intuitively, there is one instance of Entitlement per each loan-borrower pair. Since the mapping refers to loans with only one borrower, in this case we construct only one such pair, and assign the value `100` to the percentage. In the right-hand side of the mapping, **ln** is a function symbol used to construct objects denoting shared loans starting from IDs, whereas **e** constructs instances of Entitlement from each pair `ID`-`RS` (cf. [18]).

## 5. Supporting Development and Deployment of the Ontology and the Mapping

In dealing with a complex real world scenario, like that of the Italian public debt, we needed to face several critical issues, which are in fact typical of any project having similar objectives and dimensions. Some of them 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 typically 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 by both parts. A useful solution to this problem was the adoption, within this project, of a *graphical language* for ontology representation. According to this new formalism, called *Graphol*, the graphical representation of the ontology has a graph-like structure, similar to that of an Entity-Relationship diagram, but at the same time able to capture the main modeling features of OWL[5]. As an example, in Figure 1 we provide an excerpt of the ontology, translated in English for presentation purposes[6]. In such portion, we specify that each loan has one lender, which has to be an authorized institution, and has one or more borrowers, each one holding a particular percentage of the loan. Borrowers are public administration organizations, whereas the percentage is specified as an attribute of the concept Entitlement, which represents the benefit that a borrower has for a certain loan (Entitlement can be seen as the reification of a relation between Loan and PA_Organization).

[5]For further details about the graphical language, we refer the reader to the Graphol web site `http://www.dis.uniroma1.it/graphol/`.

[6]In the original ontology, Loan is denoted as `Mutuo`, is_lender is `est_mutuante`, Authorized_Institution is `Soggetto_autorizzato_ad_erogare_mutui`, has_borrower is `ha_beneficiario_mutuo`, is_borrower is `est_beneficiario_mutuo`, Entitlement is `Beneficio`, PA_Organization is `Soggetto_della_PA`, and percentage is `percentuale`.

We point out that the adoption of the graphical language has been 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 it and for identifying mistakes (cf. also Section 6). To this aim, we resorted to off-the-shelf systems for reasoning over ontologies.

As for the ontology documentation issue, we adopted a structured wiki-like documentation, 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. For each element of the ontology, the documentation includes a wiki page containing a hyper-text description of the element, and various structural properties derived from the ontology (for example, all subconcepts of a concept). Such documentation is accessible at `www.dis.uniroma1.it/~ontodeb`[7].

Actually, this is only one of the features of the system we adopted to support the development and the deployment of the ontology and the mappings in our project. The system, called MASTRO STUDIO, provides a comprehensive software environment where users can take advantage of the wiki-like documentation of the ontology, access both its graphical representation and its OWL specification, use intensional reasoning services for advanced ontology analysis, inspect the mapping towards the source databases, and exploit various types of OBDM services.

MASTRO STUDIO is a web-application based on Drupal[8], 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 ontology specification, as well as the invocation of intensional reasoning services over it and the analysis of their results.

Besides the above features, MASTRO STUDIO offers several utilities, including a tool for the translation of the graphical representation of the ontology, which is originally encoded into a standard XML format for graphs, into the OWL functional-syntax representation required by the components of the reasoning layer. Also, it provides a tool for the automatic generation and update of the documentation, starting from the ontology specification.

As for OBDM functionalities, MASTRO STUDIO relies over the reasoner MASTRO [3], through a web-service interface. We notice that MASTRO is an OBDA reasoner for *DL-Lite* ontologies equipped with mappings of the form described in Section 4. These choices are motivated by computational aspects related to query answering and reasoning in general: the expressiveness for ontologies and mappings adopted in MASTRO is essentially the maximal possible to have first-order rewritable answering of conjunctive queries, i.e., reducible to evaluation of an SQL query over the source databases [18,6], which has an evident practical impact.

In order to invoke MASTRO services, we thus needed to produce a *DL-Lite* version of the ontology starting from the OWL one. To this aim, MASTRO STUDIO provides a dedicated component, which implements the semantic approximation algorithm described in [8].

[7] To get credentials for login, please contact the authors.
[8] `http://drupal.org`

## 6. Ontology evaluation

The aim of this section is to assess the quality of the Public Debt Ontology. In particular, we consider both logically formalizable and informal criteria, referring, for the latter, to the Ontology Summit 2013 Communiqué[9], which proposes five high-level dimensions for ontology evaluation: *intelligibility*, *fidelity*, *craftsmanship*, *fitness*, and *deployability*.

In the following, we separately comment on each such informal criteria, as well as provide some final discussions on logical formalizable criteria for the Public Debt Ontology.

**Intelligibility.** The possibility of representing the ontology through the graphical language used within the MASTRO STUDIO system (cf. Section 5) allowed us to easily communicate with all the intended users of the ontology, even business stakeholders that are not knowledgable about ontology languages. The use of the graphical representation turned out to be crucial also in the training phases: ontologically naive speakers have become quite easily familiar with both the (graphical) syntax and the semantics of the ontology language, so that they have been soon able to understand the intended models of the ontology. Comprehension of the ontology by this kind of users has been fostered also by the wiki-like documentation associated to the ontology. Indeed, every documentation page of an ontology element (an atomic concept, a relation, or an attribute) provides both a description of the element and some additional structured information obtained automatically from the OWL ontology specification. For example, in the page associated to a concept, the user is provided with the list of asserted subconcepts and superconcepts, and the list of relations and attributes to which the concept can participate (with the specification if such participation is mandatory or not). Besides such documentation, users that are able to understand OWL can easily access OWL axioms (in functional syntax) through the MASTRO STUDIO web interface. Furthermore, OWL axioms that are logically inferred by the OWL specification are listed in a dedicated section of the system, which is in general used by more expert users for sophisticated analysis. Both asserted and inferred OWL axioms are visualized in pages associated to ontology elements (that is, the page documenting an element $E$ lists only axioms that mention $E$). Such a direct and focused access to the OWL code turned out to be very effective to test ontology intelligibility.

**Fidelity.** The wiki-like component of the MASTRO STUDIO system strongly facilitated the review of the ontology content by domain experts. Furthermore, it provided an effective collaborative environment where the ontology could be inspected, documented, and commented both by the ontologists and the domain experts. The collaborative process thus fostered the creation of a shared documentation (e.g., definitions of business elements and descriptions of related ontology entities), which brought together important competencies available at the Department of Treasury.

In the development of the public debt ontology, the collaborative process has been initiated by ontologists, which provided draft descriptions of the ontology elements singled out by a first analysis. Domain experts have then refined such descriptions and added additional comments, which often led to a restructuring of the ontology by the ontologists. To manage changes in the ontology and at the same time to preserve documenta-

[9] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Communique

tion already developed, we exploited the MASTRO STUDIO features for documentation versioning: when a new version of the ontology is loaded in the system, the documentation is updated accordingly, preserving the parts that did not change, and highlighting the portions that need manual intervention after the changes on the ontology automatically performed by the system.

As a further aspect regarding fidelity, we point out that in our design methodology we did not start from the analysis of the information systems currently in use at the Department of Treasury. This has been a deliberate choice to avoid the risk of shaping the ontology based on the data represented in the current system, rather than on the semantics of the domain. Nonetheless, some documentation on the conceptual schema of the data sources have been proved useful, especially to check whether the ontology completely covered the information content stored in the data sources. Actually, these checks allowed us to add various attributes (data-properties in OWL), which were neglected in the first versions of the ontology. This process guaranteed a complete coverage of the existing databases constituting the current information system.

**Craftmanship.** The public debt ontology is the product of a rigorous engineering process. This is witnessed by a set of formal properties it satisfies, such as syntactic correctness and logical consistency. The former has been verified through a MASTRO STUDIO software component that is in charge of parsing the graphical ontology specification provided as a graphml file[10]. The same component also translates such specification into an OWL functional syntax encoding, which has been then used to verify logical consistency through state-of-the-art OWL reasoners (for this task, we used both Pellet[11], and Fact++[12]).

From the point of view of formal ontological analysis, our ontology is a *descriptive ontology*: its goal is to make an already formed conceptualization by the domain experts, explicit. Also, our ontology has been designed according to the decision of restricting the attention to *particulars*. As usual, universals do appear in the ontology, at the levels of concepts, relations, and attributes, but the universe of discourse is the set of individuals in the domain. It is interesting to observe that the possibility of explicitly modeling universals as instances of other universals could in principle help in the design of the ontology. However, we avoided treating universals as individual objects in the ontology because of the lack of appropriate modeling primitives in the ontology language officially used in the project, namely OWL.

Finally, due to the nature of the domain of interest, we have both endurants and perdurants in our ontology. Indeed, as we said in Section 3, we modeled several concepts whose instances evolve in time. In order to faithfully represent such evolution, we resorted to various concepts modeled as *perdurants*, and in particular of type "time-snapshot".

**Fitness.** In the design and development of the Public Debt Ontology, a significative portion of the requirements was constituted by the main reports regularly produced by the Department of Treasury for the analysis of the public debt. In this respect, it is worth

[10]Graphml is a file format widely used for encoding graphs (`http://graphml.graphdrawing.org/`).
[11]`http://clarkparsia.com/pellet/`
[12]`http://owl.man.ac.uk/factplusplus/`

noticing that one of the most important goals of the entire project has been to realize a system that allowed users to obtain their reports by means of queries posed over an ontology representing the public debt domain. Such a system is indeed able to shift query specification at the conceptual level, and provides a more declarative mechanism for report production, in principle tailored to domain experts, rather than database administrators. Thus, information needs at the basis of the reports acted as specific *competency questions*, i.e., questions that the ontology, once developed, must be able to answer. From the very beginning of the project, we were provided with several sample reports, and in the ontology development we took into account competency questions formulated on the basis of such reports. A significant (and successful) fitness test has been then formulating competency questions in terms of queries over the ontology and verifying that such specification covered all information needs expressed by such questions.

We then point out that at the end of the project, the Department of Treasury decided to use the ontology in the call for tenders for the realization of a new information system for the management of data and processes relative to the public debt domain. In the call, the ontology as been referred to as a formal specification of the data requirements, and precise commitments have been requested to applicants for the final system to be compliant with such requirements. We believe that this is the best possible acknowledgment of the fitness quality of the ontology we realized, as well as an unquestionable element of innovation in tenders for the realization of an information system. Below, when discussing about ontology deployability, we further comment on the role of the ontology in the mentioned call for tender of the Department of Treasury .

**Deployability.** The operational engine of MASTRO STUDIO is the reasoner MASTRO for OBDA. Thus, in our context, deployability is the ability to deploy the ontology in the MASTRO reasoner, to use it at run-time for specific services for the users. We recall that MASTRO is a reasoner for *DL-Lite* ontologies equipped with mappings towards relational databases (cf. Section 4). Thus, in our project we had to produce a *DL-Lite* version of the Public Debt Ontology, starting from its OWL specification. As said, this has been done through the semantic approximation algorithm described in [8]. Therefore, we distinguished between a *reference* OWL ontology, that provides a more faithful representation of the domain, and a *DL-Lite operational* ontology, which allows us to exploit the efficient query answering services offered by MASTRO. The operational ontology has been then deployed in the MASTRO reasoner (and suitably equipped with mappings towards some selected databases managing Italian public debt data). Queries corresponding to competency questions, suitably encoded in the SPARQL standard query language[13], have been processed by MASTRO and correctness of the returned results has been verified through a comparison with official data published by the ministry (see [9] for examples of such queries, and information about query execution times).

On the basis of the success of this deployment, in the above mentioned call for tenders, the Department of Treasury decided to explicitly ask for the specification of a mapping from the Public Debt Ontology to the databases realized for the new public debt information system. On the one hand, this mapping will enrich the project documentation, thanks to its ability to specify the relationship between the future database and the ontology used as requirement specification. On the other hand, it will enable OBDM services on top of the new system, thus providing an important tool for verifying its quality.

[13] http://www.w3.org/TR/rdf-sparql-query/

Summarizing, the Public Debt Ontology is currently used at the Department of Treasury for two main aims. On the one hand, the ontology is used as a means for querying the data through the concepts and relations of the ontology. Domain experts rely on their expertise on the ontology and pose queries over the ontology signature. The OBDA service directly computes the results by relying on the query rewriting facilities of MASTRO STUDIO, which automatically translates the query over the data sources. This is a breakthrough with respect to the previous situation, where queries over the data sources were derived by IT people on the basis of requirements expressed by the domain experts, often by means of a long and costly process. On the other hand, as we said before, the ontology is currently used as a formal specification for the design of the databases which will constitute the basic building blocks of the new information system.

**Logically formalizable criteria.** As already said, logically formalizable criteria have been verified through the use of state-of-the-art Description Logic reasoners (as mentioned, for the reference ontology, we adopted the reasoners Pellet and Fact). More specifically, we used such tools to verify satisfiability of the overall Public Debt Ontology, i.e., the existence of interpretations satisfying it, and in case of unsatisfiability we exploited the justifications for the inconsistency provided by such tools, i.e., set of axioms that cause unsatisfiability. Such justifications have been then used to identify errors in the ontology and to correct them. Justifications have been also useful to find out inconsistent entities of the ontology, i.e., atomic concepts, relations, or attributes that have always an empty interpretation in every ontology model, and to correct the ontology to avoid this kind of situations. The use of reasoning tools has been also helpful to avoid the presence in the ontology of equivalent concepts, relations, or attributes, which, even though do not represent an anomaly from the logical point of view, can compromise the clarity and accuracy of the representation provided by the ontology. Furthermore, the intensional reasoning services, such as subsumption checking, provided by MASTRO STUDIO helped us in verifying the correctness of ontology. Finally, we point out that the deployment of the operational ontology in the MASTRO reasoner provided us with further mechanisms to verify whether the ontology satisfies its intended consequences, by means of the query answering tests we described above. Finally, the mapping we specified allowed us to verify the adequacy of the ontology to be instantiated with actual data (suitably transformed) stored in the databases currently in use at the Department of Treasury.

## 7. Conclusion

The collaboration between Sapienza and the Department of Treasury of the Italian Ministry of Economy and Finance is going on in several directions. In particular, we are currently working together on extending the ontology in order to cover other areas of interest for the Department. We are also collaborating in the definition of a formal process for the maintenance of the ontology and for the management of its evolution, being it a lively artifact that has to be continuously aligned to new possible requirements. Finally, we are investigating the issue of modeling the processes of the organization formally. This issue led us to the research problem of defining a formalism that allow specifying the dynamic aspects of the processes in a way to be fully coherent with the modeling structures used in the definition of the ontology. We believe that this issue represents an important direction for the research on ontology-based information systems.

## References


[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, 2007.

[2] A. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. K. Yu, editors. *Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos*, volume 5600 of *Lecture Notes in Computer Science*. Springer, 2009.

[3] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, and D. F. Savo. The Mastro system for ontology-based data access. *Semantic Web J.*, 2(1):43–53, 2011.

[4] 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 the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 274–279, 2007.

[5] 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.

[6] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. *Artificial Intelligence*, 195:335–360, 2013.

[7] 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.

[8] M. Console, V. Santarelli, and D. F. Savo. From OWL to DL-Lite through efficient ontology approximation. In *Proc. of the 7th Int. Conf. on Web Reasoning and Rule Systems (RR 2013)*, volume 7994 of *Lecture Notes in Computer Science*, pages 229–234. Springer, 2013.

[9] 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 the 16th Int. Conf. on Extending Database Technology (EDBT 2013)*, pages 561–572, 2013.

[10] A. Doan, A. Y. Halevy, and Z. G. Ives. *Principles of Data Integration*. Morgan Kaufmann, 2012.

[11] N. Guarino. Formal ontology, conceptual analysis and knowledge representation. *Int. J. of Human-Computer Studies*, 43(5–6):625–640, 1995.

[12] N. Guarino and C. A. Welty. An overview of OntoClean. In S. Staab and R. Studer, editors, *Handbook on Ontologies*. Springer, 2004.

[13] G. Guizzardi. Ontological foundations for conceptual modeling with applications. In *Proc. of the 24th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2012)*, volume 7328 of *Lecture Notes in Computer Science*, pages 695–696. Springer, 2012.

[14] R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to query answering in *DL-Lite*. In *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 247–257, 2010.

[15] M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2002)*, pages 233–246, 2002.

[16] M. Lenzerini. Ontology-based data management. In *Proc. of the 20th Int. Conf. on Information and Knowledge Management (CIKM 2011)*, pages 5–6, 2011.

[17] H. Pérez-Urbina, I. Horrocks, and B. Motik. Efficient query answering for OWL 2. In *Proc. of the 8th Int. Semantic Web Conf. (ISWC 2009)*, volume 5823 of *Lecture Notes in Computer Science*, pages 489–504. Springer, 2009.

[18] 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.

[19] B. Smith. Ontology. In *The Blackwell Guide to the Philosophy of Computing and Information*, pages 153–166. Blackwell Publishing Ltd, 2003.