Ontology-based Data Integration with MASTRO-I for Configuration and Data Management at SELEX Sistemi Integrati

Alfonso Amoroso[1], Gennaro Esposito[1], Domenico Lembo[2], Paolo Urbano[2], Raffaele Vertucci[1]

[1] SELEX Sistemi Integrati, via Circum. esterna di Napoli, 80014 Giugliano (NA)
aamoroso@sesm.it, rvertucci@selex-si.com

[2] Dip. di Informatica e Sistemistica, SAPIENZA Univ. di Roma, via Ariosto 25, 00185 Roma
lembo@dis.uniroma1.it, paolo.urbano@gmail.com

Abstract. In this paper, we report on a case study conducted with the MASTRO-I system at SELEX Sistemi Integrati, a world's leading company in the provision of integrated defence and air traffic systems. MASTRO-I is a tool for ontology-based data integration that combines advanced services for reasoning over ontologies with the capabilities provided by relational data federation tools, used for wrapping physical data sources. In our case study, we focused on the Configuration and Data Management (C&DM) domain, where data concerning the design and the production of components of complex systems are combined with data pertaining customer support and obsolescence monitoring services. At the current stage, C&DM data in SELEX-SI are fragmented over various and autonomous sources, managed by different systems under heterogeneous data models, and data integration is conducted manually by domain experts. To simplify and automatize this process, we developed an ontology for the C&DM domain and customized the MASTRO-I system to integrate C&DM data through such ontology. The effectiveness of the ontology representation, combined with the efficiency of the integration features of MASTRO-I, compared with the current manual integration process, evidenced the benefits achieved by SELEX-SI from this solution.

1 Introduction

Data integration provides organizations with a coordinated and centralized access to their distributed and heterogeneous data, through the use of a global, virtual domain view, called global schema, which is connected to data sources via suitable mappings [7]. An ontology is a representation scheme that describes a formal conceptualization of a domain of interest, and is nowadays advocated as the best means to provide a unified conceptual model of an organization. Ontologies are therefore best candidates to act as global schemas of integration applications at the enterprise level.

MASTRO-I is a tool developed at SAPIENZA Università di Roma for ontology-based data integration [3]. In MASTRO-I, the global schema is expressed in terms of a TBox, i.e., a set of assertions representing intensional knowledge, of the tractable Description Logics *DL-Lite*$_{\mathcal{A}}$ [8], which is particularly suited for conceptual modeling of organization domains and for dealing with large amounts of data, as typically required in data integration. The sources are represented through a relational schema, and the

mapping language allows for expressing Global-As-View mappings, which associate each element of the global schema with view over the sources. By virtue of the above mentioned design choice, in MASTRO-I, answering unions of conjunctive queries posed over the global schema can be done through a very efficient technique which reduces this task to standard SQL query evaluation. Thanks to this property, MASTRO-I can make use of commercial relational data federation tools for wrapping physical data sources, which can be indeed heterogeneous and non-relational, but that by means of data federation, MASTRO-I sees as a unique relational database schema.

In this paper, we report on a case study conducted with MASTRO-I at SELEX Sistemi Integrati (SELEX-SI), a Finmeccanica Company that is world leader in the provision of integrated defence, air traffic and paramilitary mission critical systems. SELEX-SI has customer base in over 150 countries, a fifty-year track record as radars and systems supplier, and more than 3000 employees in Italy, USA, Germany and UK. In our case study, we considered Configuration and Data Management (C&DM) in SELEX-SI, and focused on a significative portion of the data manipulated in this ambit.

C&DM is a technical management model that is central to all SELEX-SI activities since it governs the entire products' life cycle, by maintaining functional and physical attributes of the product consistent with its requirements, design, and operational information. C&DM data we focused on mainly regard design and production of components that are used to realize complex systems, physical deployment of such components, and analysis of their obsolescence. Currently, such data are stored in various, partially overlapping sources, and managed by different systems under diverse data models. Nonetheless, an integrated access to these data is often required, especially for those activities that regard on-site customer support and in-service management.

At the present stage, data integration is performed manually by experts of the C&DM domain, with great efforts in terms of time and resources, and without guarantees that the retrieved integrated data are sufficiently reliable and effective for the business aims. This motivated the use of MASTRO-I for ontology-based data integration for C&DM at SELEX-SI. Our experimentation proceeded as follows:

- We first carried out an ontological analysis of the domain of interest, operating both top-down, focusing on the needs and the requirements expressed by domain experts and bottom-up, analyzing single data sources and defining preliminary source ontologies, iteratively refined and fused in the final domain ontology. The output of this phase has been the definition of a *DL-Lite*$_{\mathcal{A}}$ TBox for C&DM at SELEX-SI.
- We then constructed a federated schema acting as source schema of our data integration system, and specified mappings relating the global and the source schema. To construct the source schema, we used a commercial data federation tool, whereas mappings have been specified through MASTRO-I facilities.
- We finally tested a set of queries considered significative by SELEX-SI experts for C&DM. In this testing we could experiment the effectiveness of ontology-based integration through MASTRO-I, in terms of ease of query specification, which does not require a domain expert, performance gain, especially if compared to manual data integration currently carried out, and quality of the answers, produced by a sound and complete query answering algorithm that properly reasons over the knowledge of the domain formalized through the *DL-Lite*$_{\mathcal{A}}$ TBox.

The rest of the paper is organized as follows. In Section 2 we describe C&DM at SELEX-SI. In Section 3 we recall the main characteristics of MASTRO-I. In Section 4 we discuss the use of MASTRO-I at SELEX-SI, and in Section 5 we conclude the paper.

2 Configuration and Data management in SELEX-SI

Configuration management (CM) is a technical management model that focuses on establishing and maintaining consistency of a product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life [9]. For information assurance, CM can be defined as the management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the life cycle of an information system. These concepts have been widely adopted by numerous technical management models, including systems engineering, integrated logistics support, Capability Maturity Model Integration (CMMI), ISO 9000, Information Technology Infrastructure Library (ITIL), product life cycle management, and application life cycle management. Traditionally CM has four elements:

- Configuration identification, that is the process of identifying the attributes that define every aspect of a configuration item. A configuration item is a product (hardware and/or software) that has an end-user purpose. These attributes are recorded in configuration documentation and base-lined.
- Configuration change control/management, that is a set of processes and approval stages required to change configuration item's attributes and to re-baseline them.
- Configuration status accounting/editing, that is the ability to record and report on the configuration baselines associated with each configuration item at any moment.

CM is widely used by many military e civilian organizations to manage the technical aspects of complex systems. Strictly connected to it, there is Data Management (DM), which comprises all the disciplines related to managing data as a valuable resource, i.e., development of architectures, policies, practices and procedures that properly manage the full data life cycle needs of an organization.

In SELEX-SI, the C&DM process is "The hub of the wheel". Indeed, C&DM is central to all company activities, mainly because the systems delivered by SELEX-SI have a very long life cycle (more than 20 years), and under this temporal perspective it is fundamental to have always a correct configuration management after delivery. The main activities in SELEX-SI based on the C&DM process are: Manufacturing, quality assurance, program management, logistic support, product development, technical publication, and contract management.

Basically there are three main configuration management processes that are related to different phases of product life cycle management:

- Project & Product Configuration Management, i.e., management of knowledge and requirements in the product design phase;
- Manufacturing Configuration Management, i.e., management of identification, manufacturing, and updates of data, to administrate product building chain
- In-Service Configuration Management, i.e., on-site installation checking, failures reporting, items changing or removing, and maintenance.

In the phases above, configuration management activities are carried out using different data management systems that are weakly integrated one another, and that manage partially overlapping data. Main systems adopted are:

- UGS Teamcenter, RDBMS-based tool for Project & Product Management;
- SAP R3, RDBMS-based tool for Enterprise Resource Planning (ERP) and Manufacturing Configuration Management;
- SAP Customer Support (CS), RDBMS-based tool for In-Service Configuration Management;
- eDEA xSCC Module, XML-based tool for In-Service Configuration Management;
- Odb, proprietary tool relying over Microsoft SQLServer DBMS, developed for Obsolescence Management.

Especially during In-Service Configuration Management, which is an activity that begins from the conclusion of the warranty period through the entire life cycle of the product, there is the need to use all DM systems and merge information coming from different sources. This data fusion task is performed manually by experts of the C&DM domain with great efforts in terms of time and resources. At the same time, the way in which this task is performed does not always guarantee reliability and effectiveness of the retrieved information. Obviously, this is a big deal, in particular for customer support activities. As a further aspect, various organization groups involved in the C&DM process often adopt different conventions in denoting, encoding and representing items they manage. Also, items' representations and encodings have been changed (in some cases several times) during the years, in order to adapt the representation of the information to always new business requirements and needs, thus further complicating the data integration process. The above described scenario calls for new data integration solutions allowing SELEX-SI to access all its C&DM fragmented data, create an accurate and consistent view of its information assets, possibly based on a common conceptualization of the C&DM domain, and leverage those assets to drive C&DM decisions and operations.

3 The MASTRO-I System

MASTRO-I is a Java tool for Ontology-based data integration, developed at SAPIENZA Università di Roma. It allows for the specification and management of integration applications relying on a conceptual architecture [7], according to which, the main components of a data integration system are the global schema, the sources, and the mapping. Thus, a data integration system is seen as a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where:

- $\mathcal{G}$ is the *global schema*, providing both a conceptual representation of the application domain, and a reconciled, integrated, and virtual view of the underlying sources.
- $\mathcal{S}$ is the *source schema*, i.e., the schema representing the data sources.
- $\mathcal{M}$ is the *mapping* between $\mathcal{G}$ and $\mathcal{S}$, i.e., a set of assertions establishing the connection between the elements of the global schema and those of the source schema.

The theoretical foundations at the basis of MASTRO-I have been presented in [5, 4, 8], whereas a description of the tool is given in [3]. For the sake of completeness, however, we recall in the following the main characteristics and the main features of

the system. In particular, we show how the above conceptual architecture is instantiated in MASTRO-I, and discuss reasoning over data integration systems in MASTRO-I.

The global schema. In MASTRO-I, the global schema is specified in terms of an ontology expressed in *DL-Lite*$_\mathcal{A}$ [8], a Description Logic of the *DL-Lite* family [5, 4].

We recall that Description Logics (DLs) [2] are logics that represent the domain of interest in terms of *concepts* (sets of objects) and *roles* (binary relations between concepts), and that allow for the definition of ontologies composed by a terminological component (*TBox*), specifying the intensional knowledge, and an assertional component (*ABox*), specifying the extensional knowledge. DLs have been widely used in the last years as formal language for specifying ontologies, for their ability of combining modelling power and decidability of reasoning [6].

As the other DLs of the *DL-Lite* family, *DL-Lite*$_\mathcal{A}$ has the expressive power of basic ontology languages (e.g., it captures the basics of UML class diagrams and Entity-Relationship diagrams), and at the same time guarantees tractable reasoning. Furthermore, it distinguishes concepts from *value-domains*, which denote sets of (data) values, and roles form *attributes*, which denote binary relations between objects and values.

A global schema in MASTRO-I is a *DL-Lite*$_\mathcal{A}$ TBox, whose syntax is given below[1].

$$B ::= A \mid \exists Q \mid \delta(U) \quad C ::= B \mid \neg B \quad Q ::= P \mid P^- \quad F ::= T_1 \mid \cdots \mid T_n$$

$$\text{TBox Assertions:} \quad B \sqsubseteq C \quad \rho(U) \sqsubseteq F \quad (\mathsf{funct}\ Q) \quad (\mathsf{funct}\ U)$$

According to the above syntax, *DL-Lite*$_\mathcal{A}$ TBoxes are given in terms of inclusion and functionality assertions. A *concept inclusion*, $B \sqsubseteq C$, expresses that a *basic concept* B is subsumed by a *general concept* C. B can be an *atomic concept* A, the unqualified existential quantification of a *basic role* Q, that can be in turn either an *atomic role* P or the inverse of an atomic role P^-, or the *domain* $\delta(U)$ of an *attribute* U. In words, $\exists P$ (resp. $\exists P^-$) denotes the first (resp. the second) component of P, whereas $\delta(U)$ denotes the set of objects that U relates to values. C can be in turn a basic concept or its negation (i.e., we allow for negation in the right-hand side of concept inclusions, but not in the left-hand side). *Value-domain inclusions* $\rho(U) \sqsubseteq F$ have an analogous meaning as concept inclusions. In particular, $\rho(U)$ denotes the *range* of U, i.e., the set of values related to objects by U, whereas F denotes a *value-domain*, that can be one among $T_1, \ldots, T_n$, which denote n pairwise disjoint unbounded data types used in our logic. Finally, $(\mathsf{funct}\ Q)$ (resp. $(\mathsf{funct}\ U)$) denotes a functionality assertion on a basic role (resp. on an attribute), i.e., it specifies that Q (resp. U) is functional.

As shown in [8], reasoning over a *DL-Lite*$_\mathcal{A}$ ontologies is tractable, and in particular query answering can be reduced to first-order (FOL) query evaluation over the ontology ABox seen as a plain database. Since FOL queries can be expressed in SQL syntax, this property allows us to make use of well-established relational database technologies to manage data (ABoxes) and answer queries posed over *DL-Lite*$_\mathcal{A}$ ontologies. In the following, we will give some hints on how this can be done for data integration systems managed by MASTRO-I. For more details and formal proofs on this aspect we refer the reader to [8, 5].

The source schema. The source schema is assumed to be a flat relational database schema, representing the schemas of all the data sources. Such a schema can be seen as

[1] In fact, we present here only *DL-Lite*$_\mathcal{A}$ expressions that are mentioned in the present paper. We refer the reader to [8] for the complete *DL-Lite*$_\mathcal{A}$ syntax.

the result of the wrapping of a set of heterogeneous data sources, not necessarily specified in relational format. Wrapping is actually delegated to an external data federation tool, to which MASTRO-I is connected via JDBC API, and which is in charge of presenting to MASTRO-I all the data sources as if they were a single relational database. In the experimentation discussed in the next section, to construct the source schema we have adopted Websphere Federation Server[2], the IBM tool for data federation.

The mapping. The mapping in MASTRO-I is specified according to the Global-As-View (GAV) approach, in which each element of the global schema is associated with a view (query) over the sources. Even though other, even more expressive, forms of mappings have been proposed in the literature [7], GAV mappings turn out to be effective enough when combined with global schemas specified in terms of ontologies.

Notably, the mapping in MASTRO-I establishes how data stored at the sources are linked to instances of the concepts and the roles in the global schema. To this aim, the mapping specification takes suitably into account the impedance mismatch problem, i.e., the fact that data at the sources are actually tuples of values, whereas instances of the global schema are given in terms of objects (and values connected to objects through attributes). Then, mapping assertions keep data value constants separate from object identifiers, and construct identifiers as (logic) terms over data values. More precisely, object identifiers are *terms* of the form $f(d_1, \ldots, d_n)$, where f is a function symbol of arity $n > 0$, and $d_1, \ldots, d_n$ are data values stored at the sources. The mapping is then a set of assertions of the forms

$$\Phi(\boldsymbol{v}) \rightsquigarrow A(\boldsymbol{v}); \qquad \Phi(\boldsymbol{v}) \rightsquigarrow Q(f_1(\boldsymbol{v}'), f_2(\boldsymbol{v}'')); \qquad \Phi(\boldsymbol{v}) \rightsquigarrow U(f(\boldsymbol{v}'), v))$$

where $\Phi(\boldsymbol{v})$ is a FOL query over the source schema $\mathcal{S}$ with distinguished variables [1] $\boldsymbol{v}$ (we will write such query in SQL syntax); A is an atomic concept, Q is a basic role, and U is an attribute; $\boldsymbol{v}'$ and $\boldsymbol{v}''$ are sequences of variables occurring in $\boldsymbol{v}$, v is a variable occurring in $\boldsymbol{v}$, and f, f_1, f_2 are function symbols. In words, such mapping assertions are used to map source relations (and the tuples they store), to concepts, roles, and attributes of the ontology (and the objects and the values that constitute their instances), respectively[3]. Examples of mappings are given in Section 4.

Semantics. As usual in data integration [7], we define the semantics of a data integration system $\mathcal{J} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ with respect to a given database instance for the source schema, i.e., a source database D. Then, the *semantics of $\mathcal{J}$ w.r.t. D*, denoted $sem_D(\mathcal{J})$, is the set of first-order interpretations for $\mathcal{G}$ that both satisfy the TBox assertions of $\mathcal{G}$, and satisfy the mapping assertions in $\mathcal{M}$ with respect to D.

The notion of interpretation for $\mathcal{G}$ is the usual one in DL [2]. We consider interpretations that assign the standard formal meaning to all expressions and assertions of the logic *DL-Lite*$_\mathcal{A}$, by suitably distinguishing between objects and values, and interpreting each term of the form $f(d_1, \ldots, d_n)$ with a different object in the interpretation domain, as described in detail in [8][4]. As for the interpretation of mapping assertions in $\mathcal{M}$ w.r.t.

[2] http://www-306.ibm.com/software/data/integration/federation_server/.

[3] In mappings involving attributes, it is also necessary to specify the data type of the value v, but for ease of presentation we do not discuss this aspect here.

[4] Note that we adopt the unique name assumption on both constants denoting values and terms denoting objects.

D. we adopt the notion of sound mapping [7], as we treat each mapping assertion as an implication, from left to right. (See [8] for a formal definition).

Finally, we consider unions of conjunctive queries (UCQs) [1] posed over the global schema, and define the semantics of query answering. Given a data integration system $\mathcal{J} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, and a database D for $\mathcal{S}$, the set of *certain answers* to a (UCQ) query q over $\mathcal{G}$ w.r.t. D is the set of all tuples of constants $\boldsymbol{t}$ such that $\boldsymbol{t}^{\mathcal{I}} \in q^{\mathcal{I}}$ for every $\mathcal{I}$ in $sem_D(\mathcal{J})$, where $q^{\mathcal{I}}$ and $\boldsymbol{t}^{\mathcal{I}}$ respectively denote the evaluation of q and $\boldsymbol{t}$ in $\mathcal{I}$.

Reasoning. Among the various reasoning services provided by MASTRO-I, we concentrate on query answering: Queries (UCQs) are posed in terms of the global schema, and their certain answers are to be computed by suitably reasoning on the global schema, and exploiting the mappings to access data at the sources.

In a nutshell, for answering a query q posed over the global schema $\mathcal{G}$, we proceed according to three main steps: (a) *Rewriting step*, in which q is reformulated according to $\mathcal{G}$ into a new query q_r over $\mathcal{G}$, such that q_r compiles the knowledge of $\mathcal{G}$ needed to answer q; (b) *Unfolding step*, in which q_r is further reformulated according to the mapping specification into an SQL query q_s, specified over the source schema $\mathcal{S}$ (this step generalizes, by making use of logic programming techniques, the unfolding procedure usually adopted in GAV data integration systems [7]); (c) *Evaluation step*, which consists in simply delegating the evaluation of q_s to the underlying data federation tool, which therefore evaluates q_s over D.

4 Experimenting MASTRO-I for C&DM at SELEX-SI

In this section we describe the use case conducted with MASTRO-I for data integration in the C&DM domain at SELEX-SI. As already said in Section 2, C&DM is central to all SELEX-SI activities, and involves huge amounts of data, fragmented over various and autonomous data sources, under the control of data management systems that adopt different data models (relational, XML-based, ect.). In our experimentation, we focused on a significative portion of the C&DM data, currently managed by the five systems mentioned in Section 2, and we used MASTRO-I to integrate such data, in such a way that relevant queries connected to important C&DM informational needs could be automatically processed by MASTRO-I.

For security and operative reasons, we worked on data exports obtained from the systems in which data are stored. These exports actually result in five different data sources (called with the name of the system from which data are exported):

- UGS Teamcenter, providing data for Project & Product management, mainly concerning apparatus components and their configuration states, seen at the design level, exported in HTML format;
- SAP R3, providing data for Manufacturing Configuration Management, partially overlapping with those exported by USG Teamcenter, with additional information on the obsolescence of components and substitutions, exported in Excel format;
- SAP Customer Support (CS), providing data for In-Service Configuration Management, mainly concerning physical components realized from design items, exported in Excel format;

– eDEA xSCC, providing data for In-Service Configuration Management, partially overlapping with those exported by SAP CS, mainly concerning deployment of physical components. We were provided with a complete backup of these data, that are managed in XML format;
– Odb, providing data concerning obsolescence of apparatus components, possible substitutions, and requests of purchasing or producing new components. We were provided with a complete backup of these data, that are managed in relational format under Microsoft SQLServer RDBMS.

Starting from the analysis of the above sources, we first realized the source schema, i.e., the schema representing the data sources managed by the underlying data federation tool. Then, we concentrated on the design of the global schema, i.e., a *DL-Lite*$_{\mathcal{A}}$ ontology for C&DM in SELEX-SI. Our third step has been the definition of the mappings connecting the source and the global schema. Finally, we tested a set of significative queries for C&DMat SELEX-SI. We comment below on each of the above steps.

C&DM source schema. As already said, to construct the source schema we adopted Websphere Federation Server, the IBM tool for data federation. Websphere Federation Server provides mechanisms for wrapping different kinds of data sources. In particular, data represented in (structured) text, HTML, XML or Excel format, as well as relational data managed by possibly different RDBMSs, can be easily wrapped in semi-automatic way. Websphere, indeed allows to associate a virtual relational view, called *nickname*, to external resources of these kinds. For example, for Excel data sources, a nickname is associated to each Excel sheet, and columns selected as relevant in the Excel sheet become attributes of the nickname. For external relational sources, a nickname is associated to each relational table, possibly projecting out non-relevant attributes. XML data sources are instead wrapped by associating to each XML document a nickname representing data at the nodes of the document. Further nicknames are then used to re-construct the father-child relation between nodes.

Since our data sources are in the formats above, we were able to wrap them by configuring some Websphere wrapping parameters. We point out that for eDEA and Odb data sources, having a local export of their data has not actually simplified the wrapping, being the local export an exact backup copy of the data managed by those systems. In all the other cases, the only modification required to connect Websphere Federation Server directly to the C&DM systems, rather than to locally materialized exports, is to define an ad-hoc wrapper for each specific data source, exploiting Websphere facilities.

A the end of the federation process, we produced a relational schema with around 50 relational tables in fact "virtual" nicknames, with an average of 15 attributes each, managed by Websphere. Having such federated schema frees us from the problem of having to locate physical sources, interact with them according to the data format they adopt, transform retrieved data in relational format. This however is not sufficient to achieve "real" data integration. Indeed, the federated schema is simply the union of the (relational representation of the) schemas of the data source, and it does not provide any information on how data in different sources are related one another, or have to be aligned and merged together. In other words, even in the presence of a federated schema, integration of data has to be done, to a large extent, manually.

C&DM global schema. Specifying the C&DM global schema means designing the C&DM ontology in terms of a *DL-Lite*$_{\mathcal{A}}$ TBox. On the basis of our analysis, we pro-

duced an ontology that models concepts concerning the design and the production of components that are used to realize complex systems, as, e.g., air traffic systems like radars. Furthermore, the ontology models aspects concerning the physical deployment of such components, and the analysis of their obsolescence, considering also possible substitutions. In particular, the ontology has to distinguish between *(virtual) items*, that are components seen at the design level, and *physical parts*, that represent implementations of virtual items, i.e., physical components, possibly characterized by a serial number (if the corresponding virtual item is "serializable"). An item can be associated with one or more configuration states, which apply also to the corresponding physical part. When physical parts are deployed, they are called *physical items*. Actually, they become part of more complex components, which can be in turn seen as physical items. In this case we are interested in representing the position that a physical item holds within the physical item to which it is connected by a part-of relationship. Items can be *obsolete*, and for these items a substitution might be indicated. In this way, in case a physical item that is also obsolete has to be substituted, it is possible to easily have a compatible item that can be used for substitution.

In some cases, according to the requirements of the domain experts, it is necessary to model at the ontology level information on the provenance of the data. Then, for some relevant cases, we have defined specific ontology elements to represent the data source from which the information is retrieved. For example, we defined a concept that represents items that are declared obsolete in the SAP R3 system, and attributes representing the configuration state and the description associated to an item, as specified in the eDEA system, or in the SAP CS system, etc. By virtue of this choice, it is possible to query the ontology to compare data as they are stored in different systems.

The overall *DL-Lite*$_{\mathcal{A}}$ ontology for C&DM contains around 40 concepts, 30 roles, and 50 attributes. Due to space limits, we cannot comment here all the TBox assertions contained in the C&DM ontology, and therefore we give below only a small (but significative) portion of them[5].

$$\mathit{obsolete} \sqsubseteq \mathit{item} \quad (1)$$
$$\mathit{obs\_sap\_r3} \sqsubseteq \mathit{obsolete} \quad (2)$$
$$\mathit{item} \sqsubseteq \delta(\mathsf{part\_number}) \quad (3)$$
$$(\mathsf{funct}\ \mathsf{part\_number}) \quad (4)$$
$$\rho(\mathsf{part\_number}) \sqsubseteq \mathtt{String} \quad (5)$$
$$\delta(\mathsf{part\_number}) \sqsubseteq \mathit{item} \quad (6)$$
$$\mathit{physical\_item} \sqsubseteq \mathit{physical\_part} \quad (7)$$
$$\exists \mathit{SUBST} \sqsubseteq \mathit{obsolete} \quad (8)$$
$$\exists \mathit{SUBST}^{-} \sqsubseteq \mathit{item} \quad (9)$$
$$\exists \mathit{SUBST}^{-} \sqsubseteq \neg \mathit{obsolete} \quad (10)$$
$$(\mathsf{funct}\ \exists \mathit{SUBST}) \quad (11)$$
$$\exists \mathit{IMPL} \sqsubseteq \mathit{item} \quad (12)$$
$$\exists \mathit{IMPL}^{-} \sqsubseteq \mathit{physical\_part} \quad (13)$$
$$(\mathsf{funct}\ \exists \mathit{IMPL}^{-}) \quad (14)$$
$$\mathit{physical\_part} \sqsubseteq \exists \mathit{IMPL}^{-} \quad (15)$$

In the assertions above, *item*, *obsolete*, *obs_sap_r3*, *physical_part*, and *physical_item* are atomic concepts that respectively represent items, obsolete items, items considered obsolete in the SAP R3 data source, physical parts, and physical items. Furthermore, *SUBST* is an atomic role representing the relationship between obsolete items and their substitutions, whereas *IMPL* is an atomic role representing the relationship between

[5] The overall ontology can be found at `www.dis.uniroma1.it/~lembo/CDM-Ontology.pdf`.

an item and its physical implementations, i.e., physical parts. Finally, part_number is an attribute representing codes (called part numbers), used to identify items. In words, Assertion (1) says that obsolete items are also items; Assertion (2) says that items considered obsolete in the SAP R3 system are also obsolete in the general sense; Assertion (3) and (4) respectively say that each item has one part number and that such part number is unique (i.e., part_number is mandatory and functional); Assertion (5) says that each part number is a string, whereas Assertion (6) says that any object that has a part number is an item. Also, Assertion (7) says that physical items are particular physical parts. Assertions (8) and (9) are the typings of the role *SUBST*. They say that only obsolete items may occur as the first component of instances of *SUBST* and only items may occur as the second component. This actually represents the fact that everything that has a substitution is an obsolete item, and that such substitution is an item. Furthermore, Assertion (11) says that the role *SUBST* is functional, i.e., everything that has a substitution cannot have more than one substitution, and Assertion (10) says that obsolete items cannot occur as the second component of *SUBST*, i.e., they cannot be used as substitutions. Similarly, Assertions (12) and (13) are the typings of the role *IMPL*. Furthermore, Assertion (15) says that each physical part occurs as second component of the role *IMPL*, thus imposing that each physical part is the implementation of an item. Finally, Assertion (14) says that the inverse of the role *IMPL* is functional.

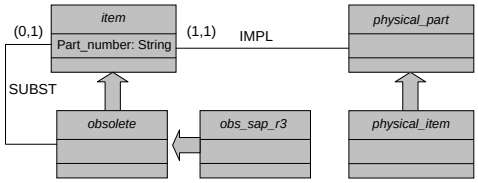

Fig. 1. Fragment of the C&DM ontology (UML approximation)

In Figure 1 we provide an UML class diagram that approximates the fragment presented above of the *DL-Lite*$_\mathcal{A}$ C&DM ontology. In particular, we notice that such diagram does not capture Assertion (10).

C&DM Mappings. We now show two mapping assertions specified to connect the C&DM ontology to the source schema. In the first mapping assertion

```
SELECT pn        ⇝ part_number(f(pn), pn),
FROM SAPR3_T1
```

we map the source table SAPR3_T1 to the ontology attribute part_number. More precisely, according to the mapping assertion above, instances of part_number are pairs of the form $(\mathbf{f}(\mathtt{pn}), \mathtt{pn})$, where pn denotes values extracted by the column pn of the table SAPR3_T1, and $\mathbf{f}(\mathtt{pn})$ is an object term constructed through the function symbol **f** and the constant pn. In a second mapping assertion,

```
SELECT P.pn, P.def                    ⇝ SUBST(f(P.pn), f(P.def)),
FROM SAPR3_T1 P
WHERE P.type='O' AND P.def IN
   (SELECT P2.pn FROM SAPR3_T1 P2)
```

we associate the role *SUBST* with an SQL query over the source schema. To understand this assertion we have to explain how information we are interested in is represented at the source. In the `SAPR3_T1` table, the attribute `def` may store strings with different meanings. In general, for each tuple it provides a description of the item which the tuple refers to. However, when the attribute `type` is valued by the character `O`, which means that the item at hand is obsolete, `def` *might* store the part number of a substitutive item. To check if this is the case, and then take such substitution and map it to a corresponding ontology object, we make use of the nested query that returns the set of items stored in `SAPR3_T1`. As for the first query, values extracted from the source are mapped to (pairs of) objects instances of the ontology through the use of a function symbol **f**.

Note that mappings are transparent to the user and that he can query the ontology ignoring completely how data are organized at the sources and are connected to ontology element. Of course mapping definition, as well as ontology definition, is a time-consuming process. However, mapping are defined only at design time, and they need to be modified only in case new data sources are added to the integration system, or some change in the domain requirements call for a modification of the ontology (which in general does not happen frequently).

Querying the C&DM data integration system. We finally provide an example (in Datalog syntax) of query posed over the global schema[6]:

$$q(it, sub) : -\mathsf{part\_number}(X, it), \mathit{SUBST}(X, Y), \mathsf{part\_number}(Y, sub), \\ \mathit{obsolete}(X), \mathit{IMPL}(X, Z), \mathit{physical\_item}(Z).$$

This query returns pairs of constants $\langle p, p' \rangle$, such that p is the part number of an obsolete item for which there exists an implementation that is a physical item, and p' is the part number of an item indicated as a substitution. Notice that this is an interesting query in the SELEX-SI C&DM domain, since it returns substitutions for physical components that result obsolete, but that are physically installed in some systems.

To get the same information without using MASTRO-I, a user should query separately each data source, and manually combine the single answers. Obviously, this can be done only by a domain expert, who knows how to localize data, retrieve it, and combine it by exploiting his/her domain knowledge. As already said, even in the presence of a federation of the data sources, actual integration should be performed manually, since the federated schema only helps the user in the source localization step. As a practical evidence of this aspect, we point out that, to access the same information returned by MASTRO-I for the query above, by directly querying the data federation tool, we have to specify a union of several non-trivial SELECT-PROJECT-JOIN queries.

Currently, at SELEX-SI queries as the one above require an unpredictable time to be answered, depending on the ability of the domain expert and the availability of the data. In MASTRO-I we can get those answers in few seconds, regardless of the expertise of the user, who has to simply know the ontology alphabet.

As a final remark, we highlight the importance of the reasoning in MASTRO-I. Reasoning for query answering actually means exploiting the knowledge provided by the ontology to obtain the complete set of certain answers to each query. To test this, we executed in MASTRO-I a set of queries over the C&DM ontology, both enabling and disabling reasoning features, which in the latter case means skipping the rewriting step

[6] Other queries can be found at the link: `www.dis.uniroma1.it/~lembo/CDM-Ontology.pdf`.

in the query answering algorithm (cf. Section 3). In almost all cases, the first strategy returned a greater number of tuples, and in some cases the gain in the answer has been enormous (e.g., from 577 tuples to 1562 tuples).

5 Conclusions

As already said, the use of MASTRO-I for C&DM at SELEX-SI can be considered successful from different points of views, including easy of access to distributed data, efficiency in query answers computation, quality of the answers returned by the system.

Starting form these basis, the case study can be extended in several directions. In particular, we foreseen to add other data sources to the current implementation, as well as to enrich the ontology to model also other aspects of C&DM that have not been taken into account in our first study, and consider also other application domains within SELEX-SI. Furthermore, we are currently working on the use of MASTRO-I also for tasks that go beyond query answering, as, for example ontology-based data update.

Acknowledgments. This research has been partially supported by the MIUR FIRB 2005 project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT), and by the FET project TONES (Thinking ONtologiES), funded by the EU under contract number FP6-7603.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995.
2. 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, 2003.
3. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. MASTRO-I: Efficient integration of relational data through DL ontologies. In *Proc. of DL 2007*, volume 250 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/Vol-250/`, pages 195–202, 2007.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*, pages 260–270, 2006.
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. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.
7. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, pages 233–246, 2002.
8. 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.
9. AA. VV.˙ Configuration management guidance. Technical Report MIL-HDBK-61A(SE), Department of Defense – USA, 2001.