Semantic Integration of Semistructured and Structured Data Sources

S. Bergamaschi\(^1,2\), S. Castano\(^3\) and M. Vincini\(^1\)

(1) University of Modena (2) CSITE-CNR Bologna (3) University of Milano

e-mail: sonia.vincini@dsi.unimo.it e-mail: castano@dsi.unimi.it

Providing an integrated access to multiple heterogeneous sources is a challenging issue in global information systems for cooperation and interoperability. In this context, two fundamental problems arise. First, how to determine if the sources contain semantically related information, that is, information related to the same or similar real-world concept(s). Second, how to handle semantic heterogeneity to support integration and uniform query interfaces. Complicating factors with respect to conventional view integration techniques are related to the fact that the sources to be integrated already exist and that semantic heterogeneity occurs on the large-scale, involving terminology, structure, and context of the involved sources, with respect to geographical, organizational, and functional aspects related to information use. Moreover, to meet the requirements of global, Internet-based information systems, it is important that tools developed for supporting these activities are semi-automatic and scalable as much as possible.

The goal of this paper is to describe the MOMIS \(4, 5\) (Mediator enVirOnment for Multiple Information Sources) approach to the integration and query of multiple, heterogeneous information sources, containing structured and semistructured data. MOMIS has been conceived as a joint collaboration between University of Milano and Modena in the framework of the INTERDATA national research project, aiming at providing methods and tools for data management in Internet-based information systems. Like other integration projects \(1, 10, 14\), MOMIS follows a “semantic approach” to information integration based on the conceptual schema, or metadata, of the information sources, and on the following architectural elements: i) a common object-oriented data model, defined according to the ODL\(_{P2}\) language, to describe source schemas for integration purposes. The data model and ODL\(_{P2}\) have been defined in MOMIS as subset of the ODMG-93 ones, following the proposal for a standard mediator language developed by the \(F^3\)/POB working group \(7\). In addition, ODL\(_{P2}\) introduces new constructors to support the semantic integration process \(4, 5\); ii) one or more wrappers, to translate schema descriptions into the common ODL\(_{P2}\) representation; iii) a mediator and a query-processing component, based on two pre-existing tools, namely ARTEMIS \(8\) and ODB-Tools \(3\) (available on Internet at http://sparc20.dsi.unimo.it/), to provide an \(F^3\) architecture for integration and query optimization. In this paper, we focus on capturing and reasoning about semantic aspects of schema descriptions of heterogeneous information sources for supporting integration and query optimization. Both semistructured and structured data sources are taken into account \(5\). A Common Thesaurus is constructed, which has the role of a shared ontology for the information sources. The Common Thesaurus is built by analyzing ODL\(_{P2}\) descriptions of the sources, by exploiting the Description Logics OLCD (Object Language with Complements allowing Descriptive cycles) \(2, 6\), derived from KL-ONE family \(17\). The knowledge in the Common Thesaurus is then exploited for the identification of semantically related information in ODL\(_{P2}\) descriptions of different sources and for their integration at the global level. Mapping rules and integrity constraints are defined at the global level to express the relationships holding between the integrated description and the sources descriptions. ODB-Tools, supporting OLCD and description logic inference techniques, allows the analysis of sources descriptions for generating a consistent Common Thesaurus and provides support for semantic optimization of queries at the global level, based on defined mapping rules and integrity constraints.

1 Providing a shared ontology

In order to illustrate the way our approach works, we will use the following example of integration in the Hospital domain. Consider the Cardiology and Intensive Care departments of a given hospital, needing to share information about their patients. The Cardiology department (CD) contains semistructured objects about patients with ischemic heart diseases, hypertension, and about physicians and nurses who have access in the line of duty to information concerning patient’s health. Fig. 1 il-
Figure 1: Cardiology department (CD) illustrates a portion of the data. There is one complex root object with four complex children objects, two patients, one physician, and one nurse. In the Intensive Care department (ID) there is a relational database containing information similar to information of the Cardiology department: it stores data both on patients with diagnoses of trauma, myocardial infarction and on medical staff. There are four relations: Patient, Doctor, Test, and Dis_Patient (see Fig. 2), where Dis_Patient instance is a subset of Patient instance and contains information about discharged patients. For integration and query, we consider schema descriptions of the sources. For structured data sources, the schema is already available and is used. To represent semistructured data at the intensional level, we associate an object pattern with each set of objects having the same label in the source graph [5]. Object patterns for all the objects in our semistructured source are shown in Fig. 3.

Schemata of structured data sources and object patterns are translated in the ODL Fr language. In our example, the object patterns defined for the CD source and the schema of the ID source are translated by defining a ODL Fr class for each object pattern and each relation, respectively. The main extensions introduced in ODL Fr are the union and optional constructors, to capture heterogeneities of semistructured data. In particular, the union constructor expresses alternative data structures in a class definition (this to capture, for instance, that the address can be defined as a string in one object of the source and as a non-atomic address object containing as its value three atomic objects (city, street, zipcode) in another object of the source). A detailed description of the union and its management in OLCD is given in [5]. The optional (*) constructor is introduced for expressing the fact that an attribute can be optional for some instances of the class.

As an example, the ODL Fr representation of the CD.Patients object pattern is as follows.

```java
interface Patient
{
    source semistructured

    Cardiology_Department

    { attribute string name;

```
ID.Dis_Patient and ID.Patient relations). In case of semi-structured sources, ODB-Tools extracts RT relationships, due to the nature of relationships defined in the semi-structured data model.

Another set of relationships can be automatically extracted exploiting the WordNet [12] lexical system. In this case, synonyms, hypernyms/hyponyms, and related terms can be automatically proposed to the designer, by selecting them according to relationships predefined in the lexical system.

**Example 1** Consider the CD and ID departments. The set of terminological relationships automatically extracted by ODB-Tools are the following:

- (ID.Patient RT ID.Doctor), (ID.Nurse RT ID.Patient), (ID.Patient RT CD.Physician), (ID.Patient RT CD.Exam), (ID.Patient RT ID.Dis_Patient), (ID.Patient RT ID.Test).
- The relationships derived from WordNet are the following: (ID.Doctor RT CD.Physician), (ID.Test SYNT CD.Exam), (ID.Patient-name RT ID.Patient_first_name), (ID.Patient-name RT ID.Patient_last_name).

**Integration/Revision of new relationships.**

In addition to the terminological relationships automatically extracted, other relationships can be supplied directly by the designer, interacting with the tool, to capture specific domain knowledge on the sources schemas (e.g., new synonyms).

**Example 2** In our Hospital domain, the designer supplies the following terminological relationships for classes and attributes:

- (ID.Doctor RT ID.Nurse), (ID.Patient SYNT ID.Patient), (ID.Patient.physician RT ID.Patient.doctor_id), (ID.Nurse.level SYNT ID.Doctor.position), (CD.Exam.outcome SYNT ID.Test.result).

Terminological relationships can, in general, correlate ODL/I3 classes whose types present structural conflicts with respect to the semantics of generalization and equivalence relationships. To promote terminological relationships to the rank of semantic relationships, that is, SYNT to equivalence, RT to generalization, and RT to aggregation, we need to solve structural conflicts producing a new “virtual schema” containing modified description of each local source. The virtual schema can then be used to enrich the Thesaurus with new relationships, by exploiting ODB-Tools inference techniques. To promote a SYNT relationship into a valid equivalence relationship it is necessary to “uniform” the types of both classes, that is, to give the same structure to both classes. The same problem arises for the RT relationship, whose transformation implies the addition of the attributes of the generalization class to the ones of the specialization class. Finally, when an RT relationship holds, a new aggregation attribute is defined between the two classes.

---

1 We use dot notation for specifying the source where a given term is used.
Inferred semantic relationships are represented as new terminological relationships enriching the Thesaurus. The result of the overall process is the so-called Common Thesaurus (see Fig. 4). A graphical representation of the Common Thesaurus for CD and ID departments is reported in Fig. 5, where solid lines represent explicit relationships (i.e., extracted/supplied), dashed lines represent inferred relationships, and superscripts indicate their kind.²

ODB-Tools performs validation and inference steps by exploiting subsumption (i.e., generalization) and equivalence computation. As we showed in [2, 6], the computation of subsumption and equivalence in OLCD is decidable. Furthermore, even if from a purely theoretical point of view this computation is PSPACE-hard (as proved in [6]), these problems can be efficiently solved by transforming a schema in a canonical form. These results imply that computing the canonical extension of a schema is difficult or that the canonical extension of a schema has a worst-case size that is exponential in the size of the original schema. However, the intractability previously mentioned rarely occurs in practice as a schema is generally formulated in such a way as to be “almost” canonical. Hence, we can conclude that transforming a schema to its canonical extension is feasible in polynomial time for most cases that appear in practice.

2 Building the mediator integrated view

In this section, we describe the process for the definition of the mediator global schema, that is the mediator integrated view of data stored in local sources. ODL₂ classes having a semantic relationship in different sources are identified. For this purpose, affinity coefficients (i.e., numerical values in the range [0,1]) are evaluated for all possible pairs of ODL₂ classes, based on the (valid) terminological relationships in the Common Thesaurus. Affinity coefficients determine the degree of semantic relationship of two classes based on their names (Name Affinity coefficient) and their attributes (Structural Affinity coefficient). A comprehensive value of affinity, called Global Affinity coefficient, is finally determined as the linear combination of the Name and Structural Affinity coefficients. Global affinity coefficients are used by a hierarchical clustering algorithm, to classify ODL₂ classes according to their degree of affinity. The output of the clustering procedure is an affinity tree, where ODL₂ classes are the leaves and intermediate nodes have an associated affinity value, holding for the classes in the corresponding cluster. The affinity-based evaluation and clustering procedures are performed with the help of the ARTEMIS tool environment (for a detailed see [4]). The affinity tree obtained for our example is shown in Fig. 6.

Clusters for integration are interactively selected from the affinity tree using a threshold based mechanism. For each selected cluster in the tree, a global class gcᵢ representative of the classes contained in the cluster (i.e., a class providing the unified view of all the classes of the cluster) is defined. The generation of gcᵢ is interactive with the designer. Let Clᵢ be a selected cluster in the affinity tree. First, the Global Schema Builder component of MOMIS associates to the gcᵢ a set of global attributes, corresponding to the union of the attributes of the classes belonging to Clᵢ, where the attributes with a valid terminological relationship are unified into a unique global attribute in gcᵢ. The attribute unification process is performed automatically for what concerns names according to the following rules:

- for attributes that have a SYN relationship, only one term is selected as the name for the corresponding global attribute in gcᵢ;
- for attributes that have a BT/NT relationship, a name which is a broader term for all of them is selected and assigned to the corresponding global attribute in gcᵢ.

For example, the attribute unification process for cluster Clᵢ of Fig. 6 produces the following set of global attributes:

- name, code, address, exam*, room, bed, therapy*, date, note, physician*

To complete global class definition, information on attribute mappings and default values is provided by the designer in the form of mapping rules. An example of ODL₂ specification for the global class

²For the sake of simplicity, only relationships between class names are reported.
Hospital_Patient (defined in correspondence of cluster C1) is shown in the following:

```java
interface Hospital_Patient {
  attribute name
  mapping_rule (ID.Patient.first_name and ID.Patient.last_name),
  CD.Patient.name;

  attribute physician
  mapping_rule CD.Patient.physician,
  ID.Patient.doctor_id;

  attribute dept
  mapping_rule CD.Patient = ‘Cardiology’,
  ID.Patient = ‘Intensive Care’,
  ID.Dis_Patient = ‘Intensive Care’;
}
```

A mapping rule is defined for each global attribute a and specifies: i) information on how to map a on the corresponding class attributes of the associated cluster and ii) default/null values defined for a based on values of attributes of cluster classes. For example, the mapping rule defined for the global attribute name in the global class Hospital_Patient above, specifies which attributes have to be considered in each class of the cluster C1; an and correspondence is defined for name, stating that the concatenation of the attributes first_name and last_name of ID.Patient have to be considered. The mapping rule defined for the global attribute dept specifies the value of this attribute for the instances of classes CD.Patient, ID.Patient and ID.Dis_Patient. The global schema of the mediator is composed of the global classes defined for all the clusters of the affinity tree.

Integrity constraint rules can also be specified for global classes of the mediator global schema, to express semantic relationships holding among the different sources. Let us suppose that in our Hospital domain a relationship exists between the result of the exam and the department of the patient. For example, the fact that all the patients with an exam result ‘Heart risk’ are ‘Cardiology’ patients can be expressed by the following integrity constraint rule in the global schema:

```
rule R1 for all X in Hospital_Patient:
  (X.exam.result=’Heart risk’) then X.dept=’Cardiology’;
```

### 3 Semantic optimization of global queries

The Query Manager module of MOMIS processes a global query Q by exploiting the semantic optimization techniques supported by ODB-Tools [3], in order to reduce the access plan cost of Q. Q is replaced by a new query, Q’, that incorporates any possible restriction which is not present in Q but is logically implied by Q on the global schema. The transformation is based on logical inferences from integrity constraints rules defined in the mediator global schema. Let us consider, as an example, query Q1: Retrieve the names of the patients with exam result ‘Heart risk’.

```
Q1: select name from Hospital_Patient
    where exam.result = 'Heart risk'
```

The Query Manager, using the query optimizer of ODB-Tools, executes the semantic expansion of Q1 by applying rule R1 and giving Q1:

```
Q1’: select name from Hospital_Patient
    where exam.result = 'Heart risk'
    and dept = ‘Cardiology’
```

Semantic expansion is performed in order to add boolean factors in the “where clause”: this process makes query plan formulation more expensive (because a heavier query has to be translated for each involved source) but single sources’ query processing overhead can be lighter in case secondary indexes on added predicates exist in the involved sources (i.e. dept in the example).

Furthermore, the introduction of a boolean factor can be useful for query plan formulation as it is in our ‘Heart risk’ example. Once the Query Manager has produced the optimized query, a set of subqueries for the local source wrappers is generated. For each source, the Query Manager expresses the subquery in terms of its local schema, by using mapping rules associated with the global class. In order to generate each local query, the Query Manager checks and translates every boolean factor in the where clause. In particular, a local query is generated only when all attributes of the where clause have a not-null correspondence in the local source. Referring to our example, the algorithm will exclude the ID.Dis_Patient class and the ID.Patient class, so that we derive only the following subquery for the ID wrapper:

```
select R.name from Patient R
where exists X in R.exam.X.outcome = 'Heart risk'
```

In such a way, an effective optimization is performed because only one local source is accessed.

### 4 Related work

MOMIS is in the line of the “virtual approach” and “read-only view” systems, that is, systems supporting read-only view of data that reside in multiple databases [9]. All of the virtual approaches are based on a model of query decomposition, sending subqueries to source databases, and merging the answers that come back. Projects close to MOMIS, based on description logics, are SIMS and Information Manifold. They are focused primarily on conjunctive queries (i.e., expressible using select, project and join), and have more the flavor of the Open World Assumption - the answer provided through an integrated view will hold a subset of the complete answer that is implied by the underlying databases. For the schema, a “top-down” approach is used: in essence a global schema encompassing all relevant information is created, and data held in the source databases is expressed as views over this global schema [16]. The SIMS project [1] proposes to create a global schema
definition using the LOOM Description Logics. Information Manifold [10] provides a source and query independent mediator. The GARLIC project [14] builds up on a complex wrapper architecture to describe the local sources with an OO language (GDL), and on the definition of Garlic Complex Objects to manually unify the local sources to define a global schema. The use of a global schema allows MOMIS and all the above systems to support every possible user queries on the schema instead of a predefined subset of them. In the OBSERVER system [11], metadata descriptions and ontologies for each different information source are considered, providing knowledge on the vocabulary used in the source. The focus of the system is on providing semantically rich queries on distributed information sources. Issues related to information integration are not taken into account. Rather, inter-ontology relationships have to be defined, under responsibility of the integration designer, to handle heterogeneity between different vocabularies for query processing. In [13], the SCOPE system is presented to perform semantic reconciliation of heterogeneous sources. Also in this system, thesauri and ontologies are used for identifying inter-schema semantic relationships, represented as assertions. Here the focus is on supporting dynamic and query-oriented integration, by constructing and refining contexts (i.e., sets of assertions) between the schema elements of the communicating systems, based on the knowledge acquired during the reconciliation process. In our project, we perform validation of the Common Thesaurus knowledge before starting the integration process, and we perform semantic integration of the sources based on selected affinity clusters, to generate the mediator integrated view of the sources. The idea of a validation and coordination mechanism as in SCOPE can be useful also in our approach, to manage the assimilation of new source schemas in the Common Thesaurus and in the mediator integrated view of the sources.

References