SI-Designer: an Integration Framework for E-Commerce

I. Benetti[1], D. Beneventano[1,2], S. Bergamaschi[1,2], F. Guerra[1] and M. Vincini[1] *

(1) Università di Modena e Reggio Emilia, DSI - Via Campi 213/B, 41100 Modena

(2) CSITE-CNR Bologna V.le Risorgimento 2, 40136 Bologna

Abstract

Electronic commerce lets people purchase goods and exchange information on business transactions on-line. Therefore one of the main challenges for the designers of the e-commerce infrastructures is the information sharing, retrieving data located in different sources thus obtaining an integrated view to overcome any contradiction or redundancy. Virtual Catalogs synthesize this approach as they are conceived as instruments to dynamically retrieve information from multiple catalogs and present product data in a unified manner, without directly storing product data from catalogs.

In this paper we propose SI-Designer, a support tool for the integration of data from structured and semi-structured data sources, developed within the MOMIS (Mediator environment for Multiple Information Sources) project.

1 Introduction

The web explosion, both at internet and intranet level, has transformed the electronic information system from single isolated node to an entry points into a worldwide network of information exchange and business transactions. Business and commerce has taken the opportunity of the new technologies to define the e-commerce activity. An electronic marketplace represents a virtual place where buyers and sellers meet to exchange goods and services, by sharing information that is often obtained as hypertext catalogs from different companies. Companies have equipped themselves with data storing systems building up informative systems containing data which are related one another, but which are often redundant, heterogeneous and not always substantial. The problems that have to be faced in this field are mainly due to both structural and application heterogeneity, as well as to the lack of a common ontology, causing semantic differences between information sources. Moreover, these semantic differences can cause different kinds of conflicts, ranging from simple contradictions in names' use (when different names are used by different source to indicate the same concept), to structural conflicts (when different models/primitives are used to represent the same information).

Therefore one of the main challenges for the designers of the e-commerce infrastructures is the information sharing, retrieving data located in different sources thus obtaining an integrated view to overcome any contradiction or redundancy. Virtual Catalogs [Keller, 1995] synthesize this approach as they are conceived as instruments to dynamically retrieve information from multiple catalogs and present product data in a unified manner, without directly storing product data from catalogs. Customers, instead of having to interact with multiple heterogeneous catalogs, can interact in a uniform way with a virtual catalog.

In this paper we propose a designer support tool, called SI–Designer, for information integration developed within the MOMIS project. The MOMIS project (Mediator environment for Multiple Information Sources) [Bergamaschi *et al.*, 1999; Beneventano *et al.*, 2000; Bergamaschi *et al.*, 2001] aims to integrate data from structured and semistructured data sources. SI–Designer is a support tool for semiautomatic integration of heterogeneous sources schema (relational, object, XML and semi-structured sources); it carries out integration following a semantic approach which uses Description logics-based techniques, clustering techniques and an ODM-ODMG [Cattell, 2000] extended model to represent extracted and integrated information, ODM_{I^3}. Using the ODL_{I^3} language, referred to the ODM_{I^3} model, it is possible to describe the sources (local schema) and SI–Designer supports the designer in the generation of an integrated view of all the sources (Global Virtual View), which is expressed using XML standard. The use of XML in the definition of the Global Virtual View lets to use MOMIS infrastructure with other open integration information systems by the interchange of XML data files.

The outline of the paper is the following. Section 2 presents the MOMIS system architecture and ODL_{I^3} relationships together with a running example used in the remainder of the paper. Section 3 contains the MOMIS approach to data integration. In particular we will focus on the *Common Thesaurus* generation, analyzing the most significant relationships automatically extracted by MOMIS. In section 4 we present SI-Designer, a framework that represents a unified solution for the overall integration process and in section 5

*This research has been partially funded by the italian MURST ex-40% D2I project - Integrazione, Warehousing e Mining di sorgenti eterogenee.

we make comparison with related work and we give conclusion remarks.

2 System Architecture References

Like other integration projects [Arens *et al.*, 1993; Roth and Scharz, 1997], MOMIS follows a "semantic approach" to information integration based on the conceptual schema, or metadata, of the information sources, and on the the I^3 architecture [Hull and et al., 1995] (see figure 1). The system is composed by the following functional elements that communicates using the CORBA [OMG, 2000] standard:

1. a common data model, ODM_{I^3}, which is defined according to the ODL_{I^3} language, to describe source schemas for integration purposes. ODM_{I^3} and ODL_{I^3} have been defined in MOMIS as subset of the corresponding ones in ODMG, following the proposal for a standard mediator language developed by the I^3/POB working group [Buneman *et al.*, 1996]. In addition, ODL_{I^3} introduces new constructors to support the semantic integration process;
2. *Wrappers*, placed over each sources, translate metadata descriptions of the sources into the common ODL_{I^3} representation, translate (reformulate) a global query expressed in the OQL_{I^3} [1] query language into queries expressed in the sources languages and export query result data set;
3. a *Mediator*, which is composed of two modules: the *SI-Designer* and the *Query Manager* (QM). The SI-Designer module processes and integrates ODL_{I^3} descriptions received from wrappers to derive the integrated representation of the information sources. The QM module performs query processing and optimization. The QM generates OQL_{I^3} queries to be sent to wrappers starting from each query posed by the user on the Global Schema. QM automatically generates the translation of the query into a corresponding set of subqueries for the sources and synthesize a unified global answer for the user.

The original contribution of MOMIS is related to the availability of a set of techniques for the designer to face common problems that arise when integrating pre-existing information sources, containing both semistructured and structured data. MOMIS provides the capability of explicitly introducing many kinds of knowledge for integration, such as integrity constraints, intra- and inter-source intensional and extensional relationships, and designer supplied domain knowledge. A *Common Thesaurus*, which has the role of a shared ontology of the source is built in a semi-automatic way. The *Common Thesaurus* is a set of intra and inter-schema intensional and extensional relationships, describing inter-schema knowledge about classes and attributes of sources schemas; it provides the base reference for the identification of classes candidate to integration and subsequent derivation of their global representation.

[1] OQL_{I^3} is a subset of OQL-ODMG.

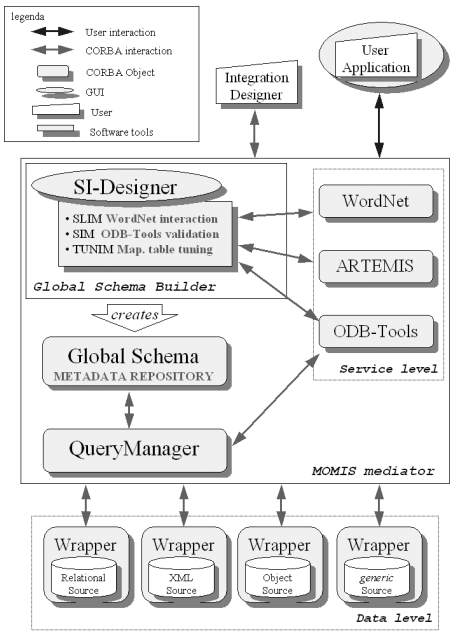

Figure 1: the MOMIS system architecture

MOMIS supports information integration in the creation of an integrated view of all sources (Global Virtual View) in a way automated as much as possible and performs revision and validation of the various kinds of knowledge used for the integration. To this end, MOMIS combines reasoning capabilities of Description Logics with affinity-based clustering techniques, by exploiting a common ontology for the sources constructed using lexical knowledge from WordNet and validated integration knowledge.

The Global Virtual View is expressed by using XML standard, to guarantee the interoperability with other open integration system prototype.

ODL_{I^3} relationships

For a semantically rich representation of source schemas inter relationships ODL_{I^3} introduces the following primitives:

Intensional relationships

They are *relationships* expressing intra and inter-schema knowledge for the source schemas. The following relationships can be specified in ODL_{I^3}:

- SYN (Synonym-of), defined between two terms t_i and t_j, with $t_i \neq t_j$, that are considered synonyms in every considered source (i.e., t_i and t_j can be indifferently used in every source to denote a certain concept).
- BT (Broader Terms), or hypernymy, defined between two terms t_i and t_j such as t_i has a broader, more general meaning than t_j. BT relationship is not symmetric. The opposite of BT is NT (Narrower Terms), or hyponymy.
- RT (Related Terms), or positive association, defined between two terms t_i and t_j that are generally used together in the same context in the considered sources.

An intensional relationships has no implications on the extension/compatibility of the structure (domain) of the two involved classes (attributes).

```
Vehicle(name, length, width, height)
Motor(cod_m, type, compression_ratio,
      KW, lubrification, emission)
Fuel_Consumption(name, cod_m, drive_trains,
      city_km_l, highway_km_l )
Model(name, cod_m, tires, steering, price)
```

Figure 2: Volkswagen database (VW)

Extensional relationships

Intensional relationships SYN, BT and NT between two classes C_1 and C_2 may be "strengthened" by establishing that they are also *extensional* relationships [Catarci and Lenzerini, 1993]. Consequently, the following extensional relationships can be defined in ODL_{I^3}:

- C_1 SYN_{ext} C_2: this means that the instances of C_1 are the same of C_2.
- C_1 BT_{ext} C_2: this means that the instances of C_1 are a superset of the instances of C_2.
- C_1 NT_{ext} C_2: this means that the instances of C_1 are a subset of the instances of C_2.
- C_1 DISJ_{ext} C_2: this means that the instances of C_1 are disjoint from the instances of C_2.

W.r.t. [Schmitt and Türker, 1998] we do not introduce an *overlap relationship* as we assume a default overlap relationships among two classes if no extensional relationship is specified. Moreover, extensional relationships "constrain" the structure of the two classes C_1 and C_2 , that is C_1 NT_{ext} C_2 is semantically equivalent to an "isa" relationship.

2.1 Running Example

In order to illustrate how the MOMIS approach works, we will use the following example of integration in the Car manufacturing catalogs, involving two different data-sources that collect information about vehicle. The first data-source is the `FIAT` catalog, containing semistructured XML informations about cars of the italian car factory.

The second data-source is the `Volkswagen` database (`VW`), a relational database containing information about this kind of car. Both database schemata are built by analyzing the web site of this factory.

3 Integration Process

The MOMIS approach to intelligent schema integration is articulated in the following phases:

1. *Generation of a Common Thesaurus.*
   The *Common Thesaurus* is a set of terminological intensional and extensional relationships, describing intra and inter-schema knowledge about classes and attributes of sources schemas. We express inter-schema knowledge in form of terminological and extensional relationships (em synonymy, *hypernymy* and *relationship*) between classes and/or attribute names. In this phase, to extract lexicon derived relationships the WordNet database is used [Gilarranz *et al.*, 1996; Miller, 1995].
2. *Affinity analysis of classes.*
   Relationships in the *Common Thesaurus* are used to evaluate the level of *affinity* between classes intra and inter sources. The concept of affinity is introduced to formalize the kind of relationships that can occur between classes from the integration point of view. The affinity of two classes is established by means of affinity coefficients based on class names, class structures and relationships in *Common Thesaurus*.
3. *Clustering classes* .
   Classes with affinity in different sources are grouped together in clusters using hierarchical clustering techniques. The goal is to identify the classes that have to be integrated since describing the same or semantically related information.
4. *Generation of the mediated schema*.
   Unification of affinity clusters leads to the construction of the predicted schema. A class is defined for each cluster, which is representative of all cluster's classes and is characterized by the union of their attributes. The global schema for the analyzed sources is composed of all classes derived from clusters, and is the basis for posing queries against the sources.

```
<!ELEMENT fiat(car*)>
<!ELEMENT car(name,engine,dimensions,tires,
      performance,price)>
<!ELEMENT engine(name,cylinders?,layout?,
      capacity_cc?,compression_ratio?,
      power_kw, fuel_system)>
<!ELEMENT dimensions(length,width,heigth,
      luggage_capacity)>
<!ELEMENT performance (urban_consumption,
      combined_consumption,speed)>
<!ELEMENT name (#pcdata)>
...
```

Figure 3: Fiat database (FIAT)

In the following we introduce the generation of the *Common Thesaurus* associated with the example domain, starting from the lexicon relationships definition by using Wordnet.

3.1 The WordNet database

WordNet is a lexical database which was developed by the Princeton University Cognitive science Laboratory. WordNet is inspired by current psycholinguistic human lexical memory connected theories and it is regarded as the most important researcher's available resource in the fields of computational linguistics, textual analysis and other related areas. The lexical Wordnet database, in the current 1.6 version has 64089 lemma which are organized in 99757 synonym sets (*synset*).

The starting point of lexical semantics derives from the observation of the existence of a conventional association between the words form (i.e. the way in which they are pronounced or written) and the concept/meaning they express; such association is of the many-to-many kind, giving rise to the following properties:

Synonymy property of a concept/meaning which can be expressed with two or more words. A synonyms group is named *synset*. Note that one and only *synset* exists for each concept/meaning. Later a *synset* will be indicated with s, while $\mathcal{S}$ will indicate the *synset* set.

Polysemy property of a single word having two or more meanings. The correspondence between the words form and

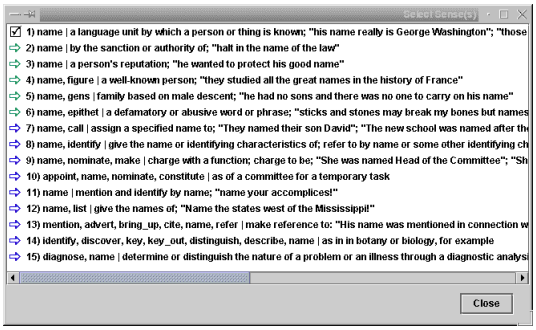

Figure 4: `name` Meanings

their meaning is synthesized in the so called *Lexical Matrix* $\mathcal{M}$, in which the words meaning are reported in rows (hence each row represents a *synset*) and columns represent the words form (form/base lemma).

Each matrix element is a(*entry*), $e = (f, m)$ definition, where f is the *base form* and m (*meaning*) is the meaning counter; for example (`address`, `2`) refers to the address where a person or an organization can be found; while (`address`, `1`) refers to a computer address in the informatics sphere. From here on the base form and the meaning of an element $e = (f, m)$ will be respectively indicated with $e.f$ and $e.m$. An element of the $\mathcal{M}$ matrix may be *null* or *indefinite*.

As only one $\mathcal{M}$ row is associated to a *synset*, from here on we will use $s \in \mathcal{S}$ as a $\mathcal{M}$ row indicator. In other words the non null elements of the $\mathcal{M}[s]$ row, represent each and every s element. In the same way, as only one $\mathcal{M}$ column is associated to a base form, from here on we will use the base forms as $\mathcal{M}$ columns index.

Semantic relationships between schema terms

With the concept of *term* we associate a definition to each class or attribute name. A *term* is formed by the $t = (n, e)$ couple, where n indicates a class or attribute name, and e indicates a definition. A class or attribute name n are qualified as follows a class name is qualified by the name of the source schema to whom the class belongs(`source_name.class_name`), an attribute name is moreover qualified with the name of the class to whom it belongs (`source_name.class_name.attribute_name`).
The classes and attributes names set is indicated by **N**; the set of words in **N** is indicated by $\mathbb{I}$. The relation between *synset* defined in Wordnet are the starting point to define semantic relations between words. Various relations are obtainable with the WordNet database; some of them are between single words others are between *synset*. In this context we will use the following relations between *synset*: **Synonymy**, **Hypernymy**, **Hyponymy**, **Olonymy**, **Meronymy**, **Correlation**.[2] As hyponymy and meronymy are inverse relations to hypernymy and olonymy, respectively, the set of relations between *synset* is the following: $\mathcal{W} = \{\mathbf{S}_{ynonymy}, \mathbf{H}_{ypernymy}, \mathbf{O}_{lonymy}, \mathbf{C}_{orrelation}\}$.Given the *synset* $\mathcal{S}$ set and the $\mathcal{W}$ relations set, The function $\phi : \mathcal{S} \times \mathcal{W} \longrightarrow 2^{\mathcal{S}}$ is inserted giving for each *synset* s the set of *synset* associated through the $r \in \mathcal{W}$ relation:

$$\phi(s, r) = \{s' \mid s' \in \mathcal{S}, r \in \mathcal{W}, \langle s' r s \rangle\}$$

Given a *synset* $\mathcal{S}$ set and a $\mathbb{I}$set of words, the function$\mathcal{H} : \mathcal{S} \longrightarrow 2^{\mathbb{I}}$ is defined associating, on the basis of the lexical matrix, a set of words to a given *synset* :

$$\mathcal{H}(s) = \{t = (n, e) | n \in \mathbf{N}, \mathcal{M}[s][t.e.f] = t.e\}$$

We can hence obtain the relations between the words using the relations existing between the *synset* that contain those words. Given a set of words $\mathbb{I}$, the set of relations between words $\mathcal{R}$, $\mathcal{R} \subseteq \mathbb{I} \times \mathcal{W} \times \mathbb{I}$, is defined as follows:

$$\mathcal{R} = \{\langle t_i r t_j \rangle | r \in \mathcal{W}, t_i, t_j \in \mathbb{I}, \exists s : t_i \in \mathcal{H}(s), t_j \in \phi(s, r), t_i \neq t_j\}$$

The relations deriving from are proposed as semantic relations to be inserted in the *Common Thesaurus* according to the following correspondence: Synonymy $\Rightarrow$ SYN, Hypernymy $\Rightarrow$ BT, Olonymy $\Rightarrow$ RT, Correlation $\Rightarrow$ RT. On the basis of these considerations, an algorithm has been developed which has as input the terms related to the schemata to be integrated, outputs the detected semantic relations:

$$\begin{aligned} \text{Input} \quad & \mathbb{I} = \{t_i | t_i.n \in \mathbf{N}\} \\ \text{Output} \quad & \mathcal{R} = \{\langle t_i\, r\, t_j \rangle, \quad r \in \{\text{SYN}, \text{BT}, \text{RT}\}\} \end{aligned}$$

Starting from the schema to be integrated, the designer must fix the $\mathbb{I}$ set. Given a name n the associated words must be chosen. This choice involves two steps:

1. **Base form choice.** The designer is supported in such a choice by the system which gives him the base form (word form) using the WordNet morphologic processor. For example, in Figure 4, by selecting the `car.name` attribute, we obtain the `name` base form from the morphologic processor. If a base form is not found, or there is an ambiguity[3], or it is not satisfactory, the designer can directly introduce it.
2. **Meaning choice.** The designer can relate a name to one, more than one, or no meaning. The choice of not relating a name to any meaning can be made for various reasons: (**a**) the concept is too complex and it can not be expressed with one word (e.g. `fuel_system`); (**b**) it belongs to the *tops*, i.e. to the generic concepts, therefore it would be related to the whole (e.g. `name`); (**c**) it is a substitute key, therefore it doesn't add any knowledge (e.g. `cod_m` of the table `vw.model`); (**d**) it is used as *foreign key*, therefore this relation has already been used during the extraction of relations from the schema structure (e.g. `cod_m` of the `model` table).

The designer selects one or more meanings from those found in WordNet starting from the base form chosen at step 1. Therefore, all the words that are related to the same name, share the same base form. For example, in Figure 4 all the 15 meanings that WordNet relates to the `name` base form are obtained. Selecting them all, i.e. considering 15 words for the `car.name` attribute, we could obtain "*wrong*" results, which are not suitable within the examined context. Some of them are shown in the following:

[2]Correlation is a relation which links 2 *synset* sharing the same hypernym, i.e. the same "father".

[3]For example 3 `axes` base forms are found: `ax` (1 sense), `axis` (5 senses), `axe` (2 senses).

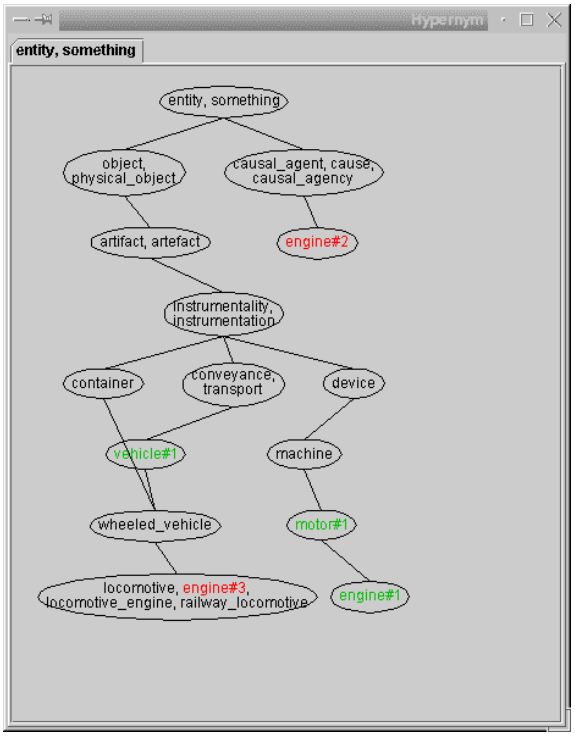

Figure 5: SLIM : hypernymy hierarchy of `engine`

⟨`fiat.car.name` SYN `vw.fuel_consumption.name`⟩
⟨`fiat.car.name` SYN `fiat.engine.name`⟩

Note that some of this relationships can look quite strange but they are true in some particular context. The problem, hence, is now resolving the meaning ambiguity so that a context-suitable couple (base form, meaning counter) can be supplied to WordNet for each concept of a source. To help the designer in the choice of the "right" meaning, for each couple (base form, meaning counter), a syntactic category (names - **N**, verbs - **V**, adjectives -**Aj**, Adverbs - **Av**) is indicated (see Figure 4).

This semi-automatic approach reduces the complexity of the designer task, in fact, a "*difficult*" problem (i.e. is finding the relations between all words), is divided in many "*easy*" ones, choosing each terms' meaning from a list. In practice this is an 80/20 problem, that is 80% of the words is worked out in the 20% of the time, just the time for reading the definitions, while the remaining 20% occupies the 80% of the time, because the choice is between very similar meanings. To speed up the 80% part a "cache" of the already selected couple (base form, meaning counter) is used (see Figure 4: the symbol ⇒ denotes the meaning already chosen by the designer for the `address` concept).

Furthermore, SI-Designer can show the generalization hierarchy of the meanings in order to help the designer in the most difficult choices. For example, (see Figure 5) in the case of `engine`: we see that "engine#2" inherits only from "causal_agent ...", "engine#3" from wheeled_vehicle and concern railway contest, whereas "engine#1" inherits also from "machine", "motor#1", ... thus we select "engine#1".

At the end of this phase, SI-Designer shows the relationships derived by using WordNet (see Figure 6). The designer

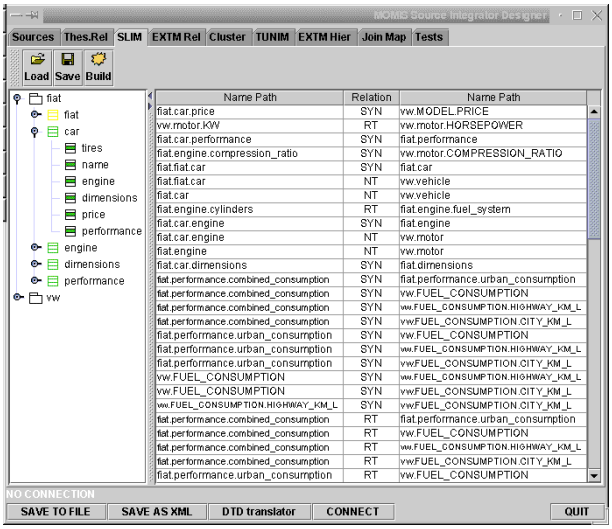

Figure 6: Inter-schema relationships extracted by SLIM

may delete any of the showed relationships and add new ones.

3.2 Generation of a *Common Thesaurus*

The *Common Thesaurus* is a set of terminological intensional and extensional relationships, describing inter-schema knowledge about classes and attributes of sources schemas; it provides a reference to define the identification of classes candidate to integration and subsequent derivation of their global representation. In the Common Thesaurus, we express inter-schema knowledge in form of terminological relationships (SYN, BT, NT, and RT) and extensional relationships (SYN_{ext}, BT_{ext}, and NT_{ext} between classes and/or attribute names.

The Common Thesaurus is constructed through an incremental process during which relationships are added in the following order:

1. *schema-derived relationships*: Intensional and extensional relationships holding at intra-schema level. These relationships are extracted by the SIM module by analyzing each ODL_{I^3} schema separately. In particular, intra-schema RT relationships are extracted from the specification of foreign keys in relational source schemas. When a foreign key is also a primary key both in the original and in the referenced relation, a BT/NT relationship is extracted We show the most significant intra-schema relationship automatically generated from MOMIS:
   ⟨`VW.Model` RT `VW.vehicle`⟩
   ⟨`VW.Model` RT `VW.motor`⟩
   ⟨`fiat.engine` RT `fiat.car`⟩

2. *lexical-derived relationships*: Intensional relationships holding at inter- schema level are extracted by the SLIM module by analyzing different sources ODL_{I^3} schemas together according to the Wordnet supplied ontology. Consider the `fiat` and the`VW` sources. The most significant lexical relationships derived using WordNet are the following:
   ⟨`fiat.car` SYN `VW.vehicle`⟩

⟨fiat.engine.compression_ratio SYN
VW.motor.compression_ratio⟩
⟨fiat.dimension BT VW.vehicle.width⟩

3. *designer-supplied relationships*: Intensional and extensional relationships supplied directly by the designer, to capture specific domain knowledge about the source schemata. Consider the VW source, in which the model entity can be considered as a specialization of the vehicle entity. This relationship can not be automatically extracted using both the lexical and the structural approaches, hence we supplied the following relationship: ⟨VW.Model NT fiat.car⟩ This is a crucial operation, because the new relationships are forced to belong to the Common Thesaurus and thus used to generate the global integrated schema. This means that, if a nonsense or wrong relationship is inserted, the subsequent integration process can produce a wrong global schema. Our system help the designer in detecting wrong relationships by performing a *Relationships validation* step with ODB-Tools. Validation is based on the compatibility of domains associated with attributes. This way, *valid* and *invalid* terminological relationships are distinguished. In particular, let $a_t = \langle n_t, d_t \rangle$ and $a_q = \langle n_q, d_q \rangle$ be two attributes, with a name and a domain, respectively. The following checks are executed on terminological relationships defined for attribute names in the Thesaurus:
   - $\langle n_t$ SYN $n_q \rangle$: the relationship is marked as valid if d_t and d_q are equivalent, or if one is a specialization of the other;
   - $\langle n_t$ BT $n_q \rangle$: the relationship is marked as valid if d_t contains or is equivalent to d_q;
   - $\langle n_t$ NT $n_q \rangle$: the relationship is marked as valid if d_t is contained in or is equivalent to d_q.

   When an attribute domain d_t (d_q) is defined using the union constructor, a *valid relationship* is recognized if at least one domain d_t (d_q) is compatible with d_q (d_t).

   Referring to the *Common Thesaurus* resulting from our example, we show some significant relationships (for each relationships, control flag[1] denotes a valid relationship, while [0] an invalid one):
   ⟨fiat.performance.combined_consumption RT
   vw.fuel_consumption.highway_km_l⟩ [0]
   ⟨fiat.dimensions BT vw.vehicle.height⟩ [1]
   ⟨VW.Model.name RT vw.vehicle.name⟩ [1]

4. *inferred relationships*: Intensional and extensional new relationships, holding at intra-schema level, inferred by exploiting inference capabilities of ODB-Tools. In the examined domain ODB-Tools infers the following relationships:
   ⟨VW.Model RT fiat.dimensions⟩
   ⟨VW.Model NT fiat.engine⟩
   ⟨VW.motor NT fiat.car⟩

All these relationships are added to the Common Thesaurus and thus considered in the subsequent phase of construction of Global Schema. For a more detailed description of the above described process see [Bergamaschi *et al.*, 2001].

Terminological relationships defined in each step hold at the intensional level by definition. Furthermore, in each of the above step the designer may "strengthen" a terminological relationships SYN, BT and NT between two classes C_1 and C_2 by establishing that they hold also at the extensional level, thus defining also an extensional relationship. The specification of an extensional relationship, on one hand, implies the insertion of a corresponding intensional relationship in the Common Thesaurus and, on the other hand, enable subsumption computation (i.e., inferred relationships) and consistency checking between two classes the C_1 and C_2.

Note that the inferred relationships show in this paper are simple and can be computed by a inheritance mechanism. On the other hand, description logic inferences are necessary to compute subsumption and to check consistency in more complex schemata and in the presence of extensional inter-schema relationships [Bergamaschi *et al.*, 2001].

Global Class and Mapping Tables

Starting from the output of the cluster generation, we define, for each cluster, a *Global Class* that represents the mediated view of all the classes of the cluster. For each global class a set of *global attributes* and, for each of them, the intensional mappings with the *local attributes* (i.e. the attributes of the local classes belonging to the cluster) are given [4].

Shortly, we can say that the global attributes are obtained in two steps: (1) Union of the attributes of all the classes belonging to the cluster; (2) Fusion of the "similar" attributes; in this step redundancies are eliminated in a semi–automatic way taking into account the relationships stored in the *Common Thesaurus*. For each global class a persistent *mapping-table* storing all the intensional mappings is generated; it is a table whose columns represent the set of the local classes which belong to the cluster and whose rows represent the global attributes. An element $MT[L][ag]$ represents how the global attribute ag is mapped into the local class L. Each element $MT[L][ag]$ of the table can assume one of the following values:

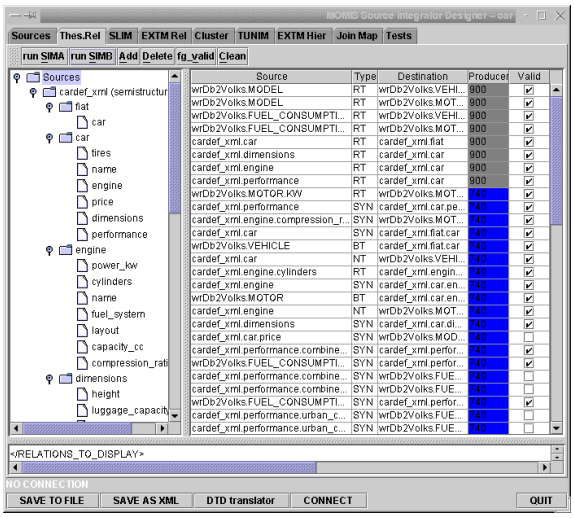

Figure 7: Example: the Common Thesaurus relationships

[4] For a detailed description of the mappings selection and of the tool SI-Designer which assist the designer in this integration phase see [Benetti *et al.*, 2001].

- $MT[L][ag] = al$: the global attribute ag maps into the al local attribute.
- $MT[L][ag] = al_1$ `and` al_2 `and...and` al_n: this is used when the value of the ag attribute is the concatenation of the values assumed by a set of attributes al_i belonging to the same local class L.
- $MT[L][ag]$ = `case of` al `const`$_1$: al_1,... `const`$_n$: al_n: this situation occurs when the ag global attribute can assume one value in a set of al_i belonging to the same local class L and the value choice depends on a third attribute, al,from the same class, which act as a selector.
- $MT[L][ag] =$ `const`: in this case a global attribute value does not refer to any local attribute and a constant value is set by the designer (see the Rank attribute).
- $MT[L][ag] =$ `null`: In this case no attribute of the class L corresponds to the global attribute ag.

The QM component will exploit each Mapping Table information to rewrite the user queries in a corresponding set of subqueries to be submitted to the local sources.

4 SI-Designer Framework

As described above, the integration process is composed by various steps actually implemented in separate module. SI-Designer is a framework that represents a unified solution for the overall integration process.

SI-Designer provides the designer with a graphical interface to reach the Global Virtual View, relating to each integration step a specific interaction with a software module. All the module involved are available as CORBA Object and interact using established `idl` interfaces.

In particular the SI-Designer performs this steps:

- **Source acquisition**: in this phase the user can select the sources to be integrated. A wrapper performs the translation from the source description model into ODL$_{I^3}$ description model. This step involves SAM module.
- **Intensional relationships definition**: in this phase, new relationships, *schema derived*, by interacting with SIM module and ODB-Tool system [Beneventano *et al.*, 1997], *lexicon derived*, by interacting with the WordNet [Miller, 1995] lexical database, and *designer supplied* are added to the Common Thesaurus (in Figure 7 the relationships involved by our example).
- **Extensional relationships definition**: Extensional relationships are defined by the interaction with the integration designer. This relationships are exploited to detect extensionally overlapping classes.
- **Clustering**: in this phase, based on the knowledge carried in the Common Thesaurus global classes are created. In our example (see Figure 8) we obtain mainly a cluster including car data contained in the sources, and a cluster for the motor and engine information.
- **Mapping table tuning**: for each global class generated in the previous phase, the user can modify the Global Virtual View proposed automatically from the system.

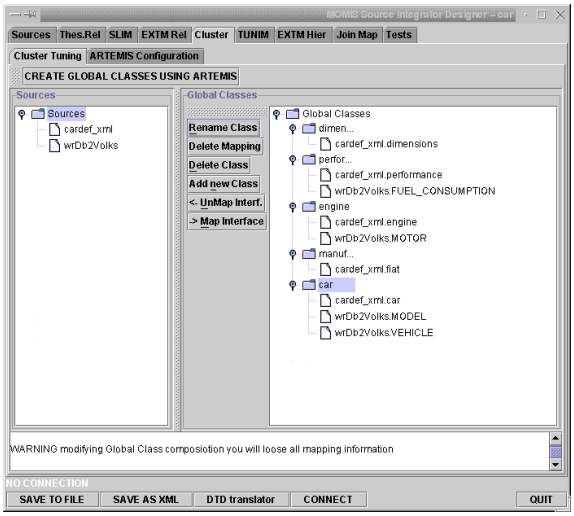

Figure 8: Example: the clustering definition

The final step of the integration process provides the export of the Global Virtual View into a XML DTD, by adding the appropriate XML TAGs to represent the mapping table relationships. The use of XML in the definition of the Global Virtual View lets to use MOMIS infrastructure with other open integration information system by the interchange of XML data files. In addition, the Common Thesaurus is translated into XML file, so that MOMIS may provides a shared ontology that can be used by different semantic ontology languages [Fensel *et al.*, 2000; Committee, 2000].

5 Related Work and Conclusion

In the area of heterogeneous information integration, many projects based on a mediator architecture have been developed. The most popular is TSIMMIS project [Chawathe *et al.*, 1994], which follows a 'structural' approach and uses a self-describing model (OEM) to represent heterogeneous data sources, the MSL (Mediator Specification Language)rule to enforce source integration and pattern matching techniques to perform a predefined set of queries based on a query template. Differently from MOMIS proposal, in TSIMMIS only the predefined queries may be executed and for each source modification a manually mediator rules rewriting must be performed.

The GARLIC project [Roth and Scharz, 1997] 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 SIMS project [Arens *et al.*, 1996] proposes to create a global schema definition by exploiting the use of Description Logics (i.e., the LOOM language) for describing information sources. The use of a global schema allows both GARLIC and SIMS projects to support every possible user queries on the schema instead of a predefined subset of them.

The Information Manifold system [Levy *et al.*, 1996] provides a source independent and query independent mediator. The input schema of Information Manifold is a set of descriptions of the sources. Given a query, the system will

create a plan for answering the query using the underlying source descriptions. Algorithms to decide the useful information sources and to generate the query plan have been implemented. The integrated schema is defined mainly manually by the designer, while in our approach it is tool-supported.

In the paper we have described a semi-automated approach to information extraction and integration, that has been implemented in the MOMIS system following a conventional wrapper/mediator architecture. The SI-Designer tool interfaces all the employed modules with the goal of allowing an interactive and customized use of MOMIS techniques by the designer, based on the specific requirements of a given integration process. The approach uses Description Logic module (ODB-Tools engine) to provide inference capabilities in the generation of a Common Thesaurus of inter-source terminological relationships. Future research will be devoted to the development of the Query Manager component of MOMIS with query optimization and "answer composition" functionalities, based on definition of extensional axioms and integrity constraints defined on global ODL_{I^3} classes. This problem is known in the literature as query rewriting and query answering using views, and has been studied very actively in the recent years. One of the original aspects of the Query Manager will consists in employing Description Logics based components (i.e., ODB-Tools engine) to perform semantic optimization steps on both on global and local queries, to minimize the number of accessed sources and the volume of data to be integrated as the result of sub-query execution.

References

[Arens *et al.*, 1993] Y. Arens, C.Y. Chee, C. Hsu, and C. A. Knoblock. Retrieving and integrating data from multiple information sources. *Int. Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.

[Arens *et al.*, 1996] Y. Arens, C. A. Knoblock, and C. Hsu. Query processing in the sims information mediator. *Advanced Planning Technology*, 1996.

[Benetti *et al.*, 2001] I. Benetti, D.Beneventano, S.Bergamaschi, A. Corni, F. Guerra, and G. Malvezzi. Si-designer: a tool for intelligent integration of information. *International Conference on System Sciences (HICSS2001)*, January 2001.

[Beneventano *et al.*, 1997] D. Beneventano, S. Bergamaschi, C. Sartori, and M. Vincini. ODB-QOPTIMIZER: A tool for semantic query optimization in oodb. In *Int. Conference on Data Engineering - ICDE97*, 1997.

[Beneventano *et al.*, 2000] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, and M. Vincini. Information integration: The momis project demonstration. In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 611–614. Morgan Kaufmann, 2000.

[Bergamaschi *et al.*, 1999] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Records*, 28(1), March 1999.

[Bergamaschi *et al.*, 2001] S. Bergamaschi, S. Castano, D. Beneventano, and M. Vincini. Semanitc integration of heterogenous information sources. *Journal of Data and Knowledge Engineering*, 36(3):215–249, 2001.

[Buneman *et al.*, 1996] P. Buneman, L. Raschid, and J. Ullman. Mediator languages - a proposal for a standard, April 1996.

[Catarci and Lenzerini, 1993] T. Catarci and M. Lenzerini. Representing and using interschema knowledge in cooperative information systems. *Journal of Intelligent and Cooperative Information Systems*, 2(4):375–398, 1993.

[Cattell, 2000] R. G. G. Cattell, editor. *The Object Database Standard: ODMG 3.0*. Morgan Kaufmann Publishers, San Mateo, CA, 2000.

[Chawathe *et al.*, 1994] S. Chawathe, Garcia Molina, H., J. Hammer, K.Ireland, Y. Papakostantinou, J.Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *IPSJ Conference, Tokyo, Japan*, 1994.

[Committee, 2000] DAML Joint Committee. Daml Project, 2000. Available at http://www.daml.org.

[Fensel *et al.*, 2000] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein. OIL in a nutshell. In *Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*. Springer, 2000. To appear.

[Gilarranz *et al.*, 1996] J. Gilarranz, J. Gonzalo, and F. Verdejo. Using the eurowordnet multilingual semantic database. In *Proc. of AAAI-96 Spring Symposium Cross-Language Text and Speech Retrieval*, 1996.

[Hull and et al., 1995] R. Hull and R. King et al. Arpa i^3 reference architecture, 1995. Available at http://www.isse.gmu.edu/I3_Arch/index.html.

[Keller, 1995] Arthur M. Keller. Smart catalogs and virtual catalogs. In *International Conference on Frontiers of Electronic Commerce*, Oct 1995.

[Levy *et al.*, 1996] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proc. of VLDB 1996*, pages 251–262, 1996.

[Miller, 1995] A.G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[OMG, 2000] OMG. Object management group. http://www.omg.org/, 2000.

[Roth and Scharz, 1997] M.T. Roth and P. Scharz. Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proc. of the 23rd Int. Conf. on Very Large Databases*, Athens, Greece, 1997.

[Schmitt and Türker, 1998] I. Schmitt and C. Türker. An Incremental Approach to Schema Integration by Refining Extensional Relationships. In *Int. Conf. on Information and Knowledge Management - ACM CIKM*, pages 322–330, New York, 1998. ACM Press.