INSTANCES NAVIGATION FOR QUERYING INTEGRATED DATA FROM WEB-SITES

Domenico Beneventano[1], Sonia Bergamaschi[1], Stefania Bruschi[1],
Francesco Guerra[2], Mirko Orsini[1], Maurizio Vincini[1]
[1] *DII - Università di Modena e Reggio Emilia*
via Vignolese 905, 41100 Modena
[2] *DEA - Università di Modena e Reggio Emilia*
v.le Berengario 51, 41100 Modena

Keywords: Semantic integration, wrapper HTML, query manager.

Abstract: Research on data integration has provided a set of rich and well understood schema mediation languages and systems that provide a meta-data representation of the modeled real world, while, in general, they do not deal with data instances.
Such meta-data are necessary for querying classes result of an integration process: the end user typically does not know the contents of such classes, he simply defines his queries on the basis of the names of classes and attributes.
In this paper we introduce an approach enriching the description of selected attributes specifying as meta-data a list of the "relevant values" for such attributes. Furthermore relevant values may be hierarchically collected in a taxonomy. In this way, the user may exploit new meta-data in the interactive process of creating/refining a query. The same meta-data are also exploited by the system in the query rewriting/unfolding process in order to filter the results showed to the user.
We conducted an evaluation of the strategy in an e-business context within the EU-IST SEWASIE project. The evaluation proved the practicability of the approach for large value instances.

1 INTRODUCTION

Integration of data from multiple sources is one of the main problems facing the database research community. One of the most common approach for integrating information sources is to build a mediated schema as synthesis of them. By holding all the data collected in a common way, such mediated schema allows the user to pose a query following a global perception. The system answers translating the query into a set of sub-queries for the involved sources by means of automatic unfolding-rewriting operations taking into account the mediated and the sources schemas. Results from sub-queries are then unified by exploiting data reconciliation techniques.

Research on data integration has provided a set of rich and well understood schema mediation languages and systems, which may be classified as the global-as-view (where the mediated schema is defined as a set of views over the data sources) and the local-as-view (where the contents of data sources are described as view over the mediated schema) formalisms (Halevy, 2003).

Following the GAV approach, we developed MOMIS (Mediator envirOnment for Multiple Information Sources) (Bergamaschi et al., 2001; Beneventano et al., 2003), a framework to perform information extraction and integration from both structured and semi-structured data sources, plus a query management environment to take incoming queries and process them through the exploitation of the mediated schema.

Two kinds of users interact with MOMIS: the integration engineer and the end user (or client/web service applications). The integration engineer is responsible for the integration process (the operation is performed by means of the MOMIS - Ontology Builder) giving rise to a Global Virtual View (GVV) of selected information sources; the end user queries the GVV classes (created by the integration engineer) aiming at obtaining a unified answer from the involved sources.

Several approaches emerged about the user supporting in querying. Such approaches may be summarized in three different schools (Broder et al., 2005): (a) the search-centric schools that guided navigation is superfluous since users can satisfy all their needs via simple queries; (b) the taxonomy navigation

Beneventano D., Bergamaschi S., Bruschi S., Guerra F., Orsini M. and Vincini M. (2006).
INSTANCES NAVIGATION FOR QUERYING INTEGRATED DATA FROM WEB-SITES.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 46-53
DOI: 10.5220/0001247900460053

school claims that users have difficulties expressing informational needs; (c) the meta-data centric school advocates the use of meta-data for large sets of results.

In this paper, we describe a method for supporting users in querying by providing meta-data about attributes of integrated GVV classes. Our approach aims at showing to the user semantic, synthesized and meaningful information emerged directly from the data. We claim such meta-data are necessary for querying classes result of an integration process: the end user typically does not know the contents of the GVV classes, he simply defines his queries on the basis of the names of classes and attributes. Such labels may be generic: the synthesis operation narrows in few classes data "semantically similar" coming from different sources. Consequently the name/description for a global class is often unspecific, especially for web sources where the user is highly involved in choosing the label for the elements descriptions. For example the integration of two local classes "`T-Shirt`" and "`Trouser`" could be a unique Global Class called "`Dress`". Such name does not allow a user to know which specific kinds of dresses are stored.

We proposed a partial solution to these issues in (Beneventano et al., 2003) where a semantic annotation of all the Global Classes with respect to the WordNet lexical database[1] provides each term with a well- understood meaning.

Our goal is now enriching the description of selected attributes specifying as meta-data a list of the "relevant values" for such attributes. Furthermore relevant values may be hierarchically collected in a taxonomy. In this way, the user may exploit new meta-data in the interactive process of creating/refining a query. The same meta-data are also exploited by the system in the query rewriting process in order to filter the results showed to the user.

Exploiting such new kind of meta-data is an interesting challenge: the literature about integration systems mainly focuses on creating/representing structures for heterogeneous data sources (Buneman et al., 1997; Nestorov et al., 1997; Halevy, 2004). Only recently, some techniques for combining data structure and data management were developed (Chaudhuri et al., 2005). The work closest to our is the "Malleable Schema" (Dong and Halevy, 2005), where a middle point between a collection of schemas/DTDs in a domain and a single strict schema for that domain is offered. In contrast with malleable schemas, our approach models a domain with a fixed semistructured model (ODM_{I^3}) where meta-data derived from extensional analysis are added.

Next section describes the MOMIS approach to data integration, section 3 defines our technique to calculate relevant values for selected attributes, section 3.2 shows the impact of relevant values in the querying process and section 4 gives an example of relevant values calculated for a real domain. Finally section 5 sketches out some conclusions.

[1] http://wordnet.princeton.edu/

2 THE MOMIS APPROACH

The framework consists of a language and several semi-automatic tools:

- The ODL_{I^3} language is an object-oriented language, with an underlying Description Logic; it is derived from the standard ODMG. ODL_{I^3} extends ODL with the following relationships expressing intra- and inter-schema knowledge for the source schemas: SYN (synonym of), BT (broader terms), NT (narrower terms) and RT (related terms). By means of ODL_{I^3} only one language is exploited to describe both the sources (the input of the synthesis process) and the GVV (the result of the process). The translation of ODL_{I^3} descriptions into one of the Semantic Web standards such as RDF, DAML+OIL, OWL is a straightforward process. In fact, from a general perspective an ODL_{I^3} concept corresponds to a Class of the Semantic Web standard, and ODL_{I^3} relationships are translated into properties.
- Information integration is performed in a semi-automatic way, by exploiting the knowledge in a Common Thesaurus (semi-automatically defined from the structural and lexical analysis of the information sources) and ODL_{I^3} descriptions of source schemas with a combination of clustering techniques and Description Logics. This integration process (performed by means of the MOMIS - Ontology Builder) gives rise to a GVV of the underlying sources. The GVV consists of a set of Global Classes, each of them made up of Global Attributes. Mapping rules connect the GVV with the original information sources and integrity constraints are specified to handle heterogeneity.
- The MOMIS Query Manager is the coordinated set of functions which take an incoming query, decompose the query according to the mapping of the GVV onto the local data sources relevant for the query, send the sub-queries to these data sources, collect their answers, perform any residual filtering as necessary, and finally deliver the answer to the requesting user. The unfolding and rewriting process is based on the full disjunction operation (Galindo-Legaria, 1994) and it is described with details in (Beneventano and Lenzerini, 2005).

2.1 The MOMIS Ontology Builder

The MOMIS integration process, shown in Figure 1, has five phases:

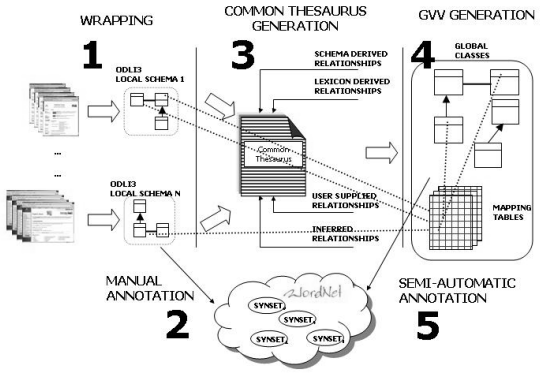

Figure 1: Functional representantion of the MOMIS Ontology builder.

1. **Local source schemata extraction**. Wrappers analyze sources in order to extract (or generate if the source is not structured) schemas. Such schemas are then translated into the common language ODL$_{I^3}$.
2. **Local source annotation with WordNet**. The integration designer defines a meaning for each element of a local source schema, according to the WordNet lexical ontology. A tool supports the integration designer: some WordNet synsets are suggested for each source element.
3. **Common thesaurus generation**. Starting from the annotated local schema, MOMIS constructs a set of relationships describing inter- and intraschema knowledge about classes and attributes of the source schemata.
4. **GVV generation**. The MOMIS methodology, applied to the common thesaurus and the local schemata descriptions, generates a global schema and sets of mappings with local schemata. The Global Schema is made up of a set of global classes. Several global attributed belong to a global class.
5. **GVV annotation**. Exploiting the annotated local schemata and the mappings between local and global schemata, the MOMIS system semiautomatically assigns name and meaning to each element of the global schema.

The Ontology Builder Tool supports the integration designer in all the GVV generation process phases.

2.2 Local Source Schemata Extraction

To enable MOMIS to manage web pages and data sources, we need specialized software (wrappers) for the construction of a semantically rich representations of the information sources by means of a common data model. A wrapper logically converts the source data structure into an ODL$_{I^3}$ schema. The wrapper architecture and interfaces are crucial, because wrappers are the focal point for managing the diversity of data sources. For conventional structured information sources (e.g. relational databases), schema description is always available and can be directly translated. For web content, this is mainly available in the form of HTML documents: such documents do not separate data from presentation and are ill-suited for being the target of database queries and most other forms of automatic processing. This problem has been addressed by much work on so-called Web wrappers, programs that extract the relevant information from HTML documents and translate it into a more machine-friendly format such as XML. The wrapping problem has been addressed by a substantial amount of work: in our approach we used Lixto (Gottlob et al., 2004) to translate the website data in XML format. The main feature of Lixto is its graphics intuitive interface that interactively guides the wrapper designer's intervention. The lixto wrapper is coupled with an XML wrapper, we developed, generating for each XML representation of a web-site the related XML Schema (an XSD file) and loads the XML data into a relational database. In this way, we are able to structure and query web-site sources. Our XML wrapper is based on MS .net framework and automatically updates the data extracted from the web by means of script daemons into the relational database.

3 PROVIDING INFORMATION ABOUT RELEVANT VALUES OF ATTRIBUTES

A global class contains data collected from different local sources by means of an integration process, and consequently its name (and the associated annotations) may not perfectly fit in its contents. Thus, a query written only on the basis of this information name may be misleading. Moreover, ignoring the values assumed by a global attribute may generate mistaken queries: a user that does not know the granularity of an attribute may write an exceedingly selective query, or a selection clause that does not really produce any result because semantically improper for that context. On the other hand, to know all the data

collected from a global class is not possible for a user: databases contain large amount of data which a user can not deal with.

For these reasons, we present a technique to provide the end user with the knowledge of the "relevant values" for global attributes selected by the integration designer in the GVV building phase.

Given an attribute At of a global class C, a *relevant value* RV for At is a pair $RV = \langle RVN, RVI \rangle$ where RVN is the *name* of the relevant value and RVI is a set of values assumed by At, i.e., $RVI \subseteq \prod_{A_t}(C)$.

A *set of relevant values* SRV is a set $SRV = \{RV_1, RV_2, ..., RV_v\}, n > 0$, such that $\bigcup_{RVi \in SRV} RV_i = \prod_{At} C$

The relevant values may be classified in a taxonomy by means of `ISA` relationships: RV_1 `ISA` RV_2; a taxonomy need to be *consistent*: if RV_1 `ISA` RV_2 then $RVI_1 \subseteq RVI_2$.

A relevant value $RV = \langle RVN, RVI \rangle$ is *representative* of all the associated values RVI: at the level of query management a condition At=RVN will be transformed in the equivalent condition At `IN` RVI.

Such relevant values are calculated by means of a semi-automatic process composed of the following steps:

1. **identification of the relevant values**: the set of relevant values SRV is calculated by applying clustering techniques to $\prod_{At} C$.

   There are different algorithms in literature to be applied for clustering values. For example in (Gibson et al., 2000) a proposal for clustering values that cannot be naturally ordered by a metric (i.e. categorical data) is described. Nevertheless, we claim that one single algorithm may not work in any domain. Therefore our goal is to develop and propose to the user a pool of techniques for selecting the most suitable method (or the combination of different methods) for the specific domain. In MOMIS, we developed a syntactic algorithm particularly customized for dealing with classifications of services and goods. It is a typical topic of the e-commerce, where enterprises propose their products by means of web-sites. In such sites, products are often grouped by means of categories called in different sites in different ways. Moreover categories are typically collected in taxonomies on the basis of specific criteria.We observed that the names of these categories are often qualified with multiple attributes in order to describe specific products. The proposed method exploits such features by means of the "Contains" function that shows if a single value for the selected attribute is contained in another one. We choose this function because typically implemented in RDBMS and anyhow easily developed. The list of relevant values is obtained by a stemming process on the $\prod_{At} C$ elements and applying the "Contains" function to the attribute values repeatedly until the achievement of a fixed point. In section 4 we show an example of relevant values set obtained by applying the algorithm on four web-sites.

2. **identification of the name of a relevant value**: the name associated to the relevant value is typically the most general value among the collected values, i.e. given $RV = \langle RVN, RVI \rangle$, RVN is the most general value of RVI. The name choice may be result of the clustering algorithm applied to identify the relevant values. Otherwise, the system proposes a name that has to be confirmed by the user. The method proposed in MOMIS, based on the "Contains" function, allows defining for special string domains a set of relevant values $SRV = \{RV_1, RV_2, ..., RV_v\}$, where in each RVI_i there is a "most general value" which can be used as name of the relevant value RV_i (see section 4).

3. **hierarchy definition**: Relevant attributes may be exploited for summarizing categories and classifications. In this context, data sources (e.g.: information systems, web-sites) typically provide partial/total hierarchies for supporting users in the querying phase. Our goal is exploiting such original hierarchies by applying to the SRV the hierarchical relationships being among the values RVI_i belonging to the RVI (see section 4 for an example). In addition, the MOMIS system provides a graphical interface helping users in the manually execution of this process.

3.1 Relevant Values Representation

The set of relevant values is represented according the proposal of the Ontology Engineering and Patterns Task Force in the Semantic Web Best Practices and Deployment Working Group (N. Noy, 2005), where five different approaches are suggested to represent OWL classes as property values on the Semantic Web. In particular, with reference to the first representation , OWL classes are directly used to describe the different relevant values belonging to the selected global attribute A_t. This assumption allows modeling an hierarchy of relevant values by means of the `rdfs:subClassOf` property. According to this approach, we represent each RVN as an OWL Class and the set of values $v_j, j : 1, ..., k$ of RVI as instances of RVN; finally each RVN (i.e. an OWL class) is then generalized by a root OWL Class (called as the A_t name) that becomes the property value of the A_t attribute.

For example, referring to the example domain described in Section 4 (Table 1), the Moulding relevant value is modeled as follows:

```
<owl:Class rdf:ID="CategoryName">
 <rdfs:subClassOf
   rdf:resource=
   "http://www.w3.org/2002/07/owl#Class"/>
</owl:Class>

<owl:Class rdf:ID="Moulding">
 <rdfs:subClassOf
   rdf:resource="#CategoryName"/>
</owl:Class>

<Moulding rdf:ID="Moulding" />
<Moulding rdf:ID=
   "Compression injection moulding" />
<Moulding rdf:ID="Insert moulding" />
<Moulding rdf:ID="Normal moulding" />
<Moulding rdf:ID="Intrusion moulding" />
<Moulding rdf:ID="Deep moulding" />
```

3.2 Querying Relevant Values

The MOMIS system provides the end user with a graphical interface where the global classes, the global attributes, the primary and foreign keys are shown (see figure 2). This interface enables the end user to write a query. The interface shows on the left the complete GVV with an E/R like formalism and a tree representation. By selecting a class, its WordNet annotations (i.e. the synsets associated to the class) are visualized on the bottom panel. Right on the top a box allows inserting the query.

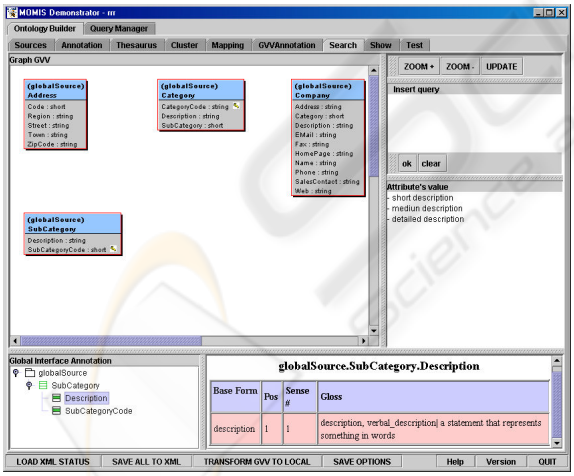

Figure 2: Screenshot of the MOMIS Query Manager.

On the basis of the knowledge about relevant values, the queries may exploit selection clauses fitting in the data:

1. **the user is interested to the instances assuming a specific known value**: relevant values do not improve the querying process, but, if an attribute value is called in different sources in different ways, the results may not include interesting instances.
2. **the user queries an attribute with relevant values**: the user expresses a query, by using the graphical interface, containing a relevant value, i.e. a condition of the form `At = RVN`. As stated before, this condition is equivalent to the condition `At IN RVI`. The discussion about how this query is executed is out of the scope of this paper; intuitively:
   - In a naive approach the condition `At = RVN` is rewritten into a local source `L` (for simplicity, we suppose that the global attribute `At` corresponds to the same local attribute `At` in the local sources `L`) as $\bigvee_{value \in RVI} At = value$.
   - On the other hand, if the function `Contains` is *executable/supported* by the local source `L` (this is a frequent case if the local source is a database), the condition `At = RVN` may be rewritten into `L` as `Contains(At,RVN)` $= true$.

4 EVALUATION ON A REAL DOMAIN

Within the EU SEWASIE project (IST-2001-34825), we collected information about enterprises working on the mechanical sector. Our goal was to integrate information coming from specialized web sites.

4.1 Building the GVV and a Relevant Attribute Set

Four portals containing data about italian companies were analyzed:

- `www.subforn.net`: provides access to a database containing more than 6.000 subcontractors of eight italian regions. Companies are classified on the basis of their production. Mechanichal and mould sectors are divided into 53 different categories. For each category, several specific kinds of production (almost 1000) are defined.
- `www.plasticaitalia.net`: the plasticaitalia database contains more than 12.000 italian companies. For each company, several kinds of production are indicated: this classification is based on a three levels hierarchy specializing each kind of production in more than 300 cases.
- `www.tuttostampi.com`: contains 4000 italian companies categorized in 58 different kinds of services.
- `www.deformazione.it`: more than 2000 companies are catalogued on the basis of 39 different sectors.

By means of the MOMIS system, a GVV representative of the four web sites was built. The simple structure of the original information sources generates a generic GVV composed of two main global classes: one storing data about companies, the second one containing all the production categories (see figure 3 where a third table allows mapping companies into categories).

```
Company(id, name, address, email, fax,
        telephone_number, country,
        foundation_year, ...  )
Category(id, name)
List_categories(company_id, category_id)
```

Figure 3: Some of the Global Classes for the mechanical GVV.

According with this representation, it is very difficult for a user querying for companies producing on a specific sector: the user does not have any idea about the more than 1000 possible categories (the union of all the different categories used in the four web-sites) and indicating a specific category may be misleading: similar categories may be called in different sources in different ways (and then the user will have no result or a partial result from his query) or the user may be interested to a result with a larger granularity. For example, if the user searches for companies producing "`mould`", the result will not include companies classified as producing "`Plastic castings`", that could be interesting for the user. For these reasons, we apply the "Contains" technique to the global attribute `name` of the global class `category`. The result was a set of 355 relevant values. Figure 4 shows the dimension of the obtained relevant values: 34% of the relevant values (120) represents 80% (845) of the instances of the selected attributes, while 235 relevant values contain only one instance. Table 1 shows some significant relevant values and the instances belonging to them.

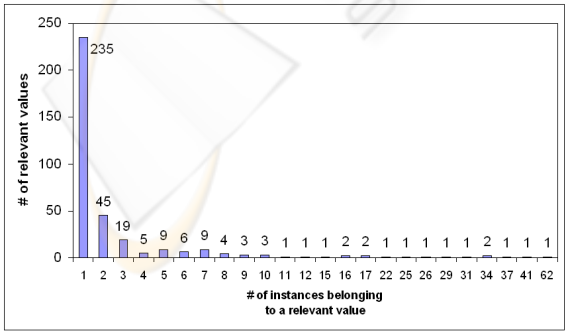

Figure 4: Attributes distribution in the relevant values set.

Table 1: An example of some relevant values.

Castings	Castings, Casting zinc and its alloys, Casting with pouring under pressure, Cast iron casting, Steel casting, Casting titanium and its alloys, Aluminum and magnesium casting, Casting copper and its alloys, Casting other metals, ...
Moulding	Moulding, Compression injection moulding, Insert moulding, Normal moulding, Intrusion moulding, Dip moulding, ...
Windings	Windings,Filament winding reinforced plastic, Transformer windings, Motor windings, Coil windings

The high number of relevant attributes made up of only one instance suggested to us to improve our technique by considering some semantic knowledge extracted from the original web-sites. E-commerce web-sites classify their products on the basis of hierarchical classifications. By automatically exploiting these taxonomies with customized wrappers, it is possible to build a relevant values set taking into account the semantic grouping of services and goods made in the original sources. In SEWASIE, the result was the identification of 135 relevant values. By exploiting again the original taxonomies, such relevant values were then classified in a three levels hierarchy. Figure 5 shows parts of the hierarchy: the first level divides the categories in two parts: the mechanical sector and the plastic and rubber sector; then, each sector is specialized in specific topics. For example, the "`mould_making`" relevant value is contained in the class "`plastic_and_rubber_processes`" that is part of the "`plastic_and_rubber`" sector. Moreover, the "`mould_making`" relevant value stands for 54 values in the local sources, for example large size moulds, castings, mould manufacturing, ... Figure 6 shows the dimension and the number of obtained relevant values.

4.2 Querying the GVV by Means of the Relevant Attribute Set

We suppose that an end user is searching for enterprises by using the MOMIS system applied to this domain. His query may assume different forms:

- The user is looking for enterprises without specifying their kinds of production. The result is a set of almost 30,000 enterprises and the related productions (several enterprises belong to more than one category).
- The user is looking for enterprises belonging to a specific category (e.g.:`mould`). The corresponding query is:

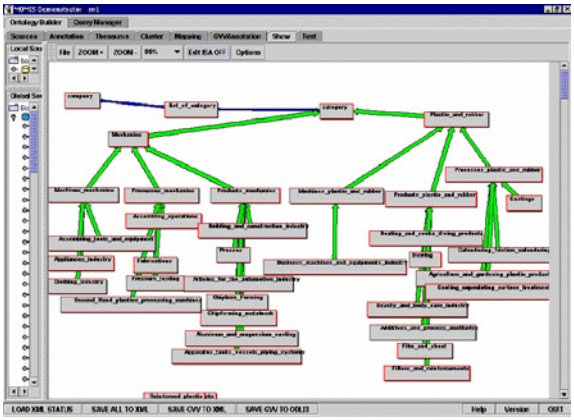

Figure 5: Part of the hierarchy of the relevant values for the category name attribute.

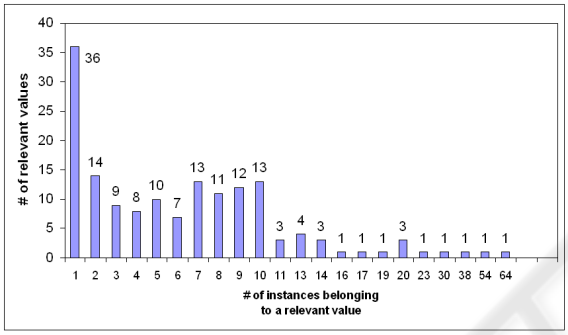

Figure 6: Attributes distribution in the relevant values set for the SEWASIE project.

```
select *
from Company C, List_categories L,
     Category Cat
where C.id = L.company_id
and L.category_id = Cat.id
and Cat.name like 'mould'
```

The result is a set of 89 companies.

- The user is interested to enterprises working on a category `mould manufacturing` by exploiting the relevant values set (e.g.: the `moulding` relevant value). In this case he has to simply indicate the selected relevant value and the selection clause of the query automatically changes including as predicate all the instances belonging to the relevant value. The choice of the `moulding` relevant attribute generates a selection clause "`... Cat.name in ('Moulding', 'Compression injection moulding', 'Insert moulding', ...)`". The result of this query is 943 companies if the relevant values set is calculated by the algorithm described in section 3, 1459 companies if relevant values set is calculated by means of the "semantic" version (i.e., by using the `mould making` value).

5 CONCLUSIONS AND FUTURE WORK

In this paper we proposed a method to identify the relevant values of selected global attributes. These values may be provided to the end user (classified in a hierarchy when it possible) in order to ease him having the knowledge about a Global Class and writing a query. There are few critical issues to be pointed out:

- the method is based on a data analysis. Working with instances is a limit of such technique: if data change, relevant values and the consequent hierarchy has to be updated. In specific contexts (databases not frequently updated, e.g. databases for e-commerce storing the products catalog for a company), data are almost static and then the calculus has to be only occasionally re-done. Nevertheless, the MOMIS wrappers were modified in order to periodically check the sources for verifying the relevant values consistency.
- the relevant values identification is a critical aspect: the integration engineer has to define a set of relevant values covering all the possible kinds of values, with a limited number of different values to be easily visualized and known by the end user. Otherwise, the end user does not know the Global Class contents and does not find the required information to write a query.
- the method allows a user to write specific queries having an organized knowledge of the sources contents.

The future work will be addressed on three directions:

1. to improve the relevant values selection by proposing to the user a multi-strategic approach for obtaining the most suitable relevant values set: for this goal, we think of it is possible to use industrial standards for classification of services and goods (e.g. UNSPSC, ecl@ss, NAICS, ...) inside specific domains both for creating the most suitable set of relevant values and for creating an automatic hierarchy of these attributes;
2. to calculate relevant values of multiple global attributes;
3. to improve the Query Manager graphical interface allowing users to query the GVV without writing any query.

ACKNOWLEDGMENTS

This work started in the EU SEWASIE project (IST-2001-34825). Now the research activity

continues within the Italian MIUR PRIN WISDOM project (2004-2006). Further information at http://www.dbgroup.unimo.it/wisdom.

REFERENCES

Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2003). Synthesizing an integrated ontology. *IEEE Internet Computing Magazine*, pages 42–51.

Beneventano, D. and Lenzerini, M. (2005). Final release of the system prototype for query management. *Sewasie, Deliverable D.3.5, Final Version, available at http://www.dbgroup.unimo.it/pubs.html*.

Bergamaschi, S., Castano, S., Beneventano, D., and Vincini, M. (2001). Semantic integration of heterogeneous information sources. *Data & Knowledge Engineering, Special Issue on Intelligent Information Integration*, 36(1):215–249.

Broder, A. Z., Maarek, Y. S., Bharat, K., Dumais, S. T., Papa, S., Pedersen, J., and Raghavan, P. (2005). Current trends in the integration of searching and browsing. In *WWW (Special interest tracks and posters)*, page 793.

Buneman, P., Davidson, S., Fernandez, M., and Suciu, D. (1997). Adding structure to unstructured data. In *Proc. of ICDT 1997*, pages 336–350, Delphi, Greece.

Chaudhuri, S., Ramakrishnan, R., and Weikum, G. (2005). Integrating db and ir technologies: What is the sound of one hand clapping? In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA*, pages 1–12.

Dong, X. and Halevy, A. Y. (2005). Malleable schemas: A preliminary report. In *Proceedings of he Eight International Workshop on the Web & Databases (WebDB 2005), Baltimore, Maryland, USA*, pages 139–144.

Galindo-Legaria, C. A. (1994). Outerjoins as disjunctions. In Snodgrass, R. T. and Winslett, M., editors, *SIGMOD Conference*, pages 348–358. ACM Press.

Gibson, D., Kleinberg, J., and Raghavan, P. (2000). Clustering categorical data: an approach based on dynamical systems. *VLDB Journal*, 8(3-4):222–236.

Gottlob, G., Koch, C., Baumgartner, R., Herzog, M., and Flesca, S. (2004). The lixto data extraction project - back and forth between theory and practice. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 1–12, Paris, France.

Halevy, A. (2003). Data integration: a status report. In *Proceedings of the German Database Conference, BTW-03*, Leipzig.

Halevy, A. Y. (2004). Structures, semantics and statistics. In *Proceedings of the 30th International Conference on VLDB, Toronto, Canada*, pages 4–6.

N. Noy, M. Uschold, C. W. (2005). Representing classes as property values on the semantic web. *Semantic Web Best Practices and Deployment Working Group, part of the W3C Semantic Web Activity. (http://www.w3.org/TR/swbp-classes-as-values)*.

Nestorov, S., Abiteboul, S., and Motwani, R. (1997). Inferring structure in semistructured data. *SIGMOD Record*, 26(4):39–43.