Query Processing in a Mediator System for Data and Multimedia *

Domenico Beneventano[1] and Claudio Gennaro[2] and Matteo Mordacchini[2] and R. Carlos Nana Mbinkeu[1]

[1] DII - Università di Modena e Reggio Emilia
Via Vignolese 905, 41100 Modena- Italy
firstname.lastname@unimore.it
[2] ISTI - CNR,
Via Moruzzi 1, Pisa - Italy
firstname.lastname@isti.cnr.it

Abstract. Managing data and multimedia sources with a unique tool is a challenging issue. In this paper, the capabilities of the MOMIS integration system and the MILOS multimedia content management system are coupled, thus providing a methodology and a tool for building and querying an integrated virtual view of data and multimedia sources.

1 Introduction

The research community has been developing techniques for integrating heterogeneous data sources for the last twenty years. One of the main motivations is that data often reside in different, heterogeneous and distributed data sources that users need to access in a single and unified way to gather a complete view of a specific domain. A common approach for integrating information sources is to build a mediated schema as a synthesis of them. By managing all the collected data in a common way, a global schema allows the user to pose a query according to a global perception of the handled information. A query over the mediated schema is translated 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 schemata. Results from sub-queries are finally unified by data reconciliation techniques.

Integration systems relying on mediator-based architectures have been developed, and most of them are able to integrate at the same time heterogeneous data sources, i.e. sources that represent their contents with relational, object-oriented and semi-structured (XML and its derived standards) models.

Similarity search for content-based retrieval (where content can be any combination of text, image, audio/video, etc.) has gained importance in recent years, also because of the advantage of ranking the retrieved results according to their proximity to a query. The systems usually exploit data types such as sets, strings, vectors, or complex structures that can be exemplified by XML documents. Intuitively, the problem is to find

* The work presented in this paper has been partially supported by the Italian FIRB project RBNE05XYPW NeP4B - Networked Peers for Business.

similar objects with respect to a query object according to a domain specific distance measure. However, the growing need to deal with large, possibly distributed, archives requires specialized indexing support to speedup retrieval. The common assumption is that the costs to build and to maintain an index structure are lower compared to the ones that are needed to execute a query.

In [5, 3] we proposed an approach that extends the action sphere of traditional mediator systems allowing them to manage "traditional" and "multimedia" data sources at the same time. The result is implemented in a tool for integrating traditional and multimedia data sources in a virtual global schema and transparently querying the global schema. We believe this is an interesting achievement for several reasons. Firstly, the application domain: there are several use cases where joining traditional and multimedia data is relevant (see Section 2 for a concrete scenario). Secondly, multimedia and traditional data sources are usually represented with different models. While there is a rich literature for transforming the differently modelled traditional data sources into a common model and it is possible to represent different multimedia sources with a uniform standard model such as MPEG-7, a standard for representing traditional and multimedia data does not exist. Finally, different languages and different interfaces for querying "traditional" and "multimedia" data sources have been developed. The former relies on expressive languages allowing expressing selection clauses, the latter typically implements similarity search techniques for retrieving multimedia documents similar to the one provided by the user.

In this paper we mainly focus on query processing in a such Mediator System for Data and Multimedia. First, we introduce the notion of DMS (*Data and Multimedia Source*) to represent and query data source and multimedia sources in a uniform way. Second, we discuss some cases where multimedia constraints can be expressed in a global query without requiring multimedia processing capabilities at the mediator level. This is an interesting result, since it upsets the usual paradigm on which mediator systems are based stating that the query processing power of a mediator is greater than the one of the integrated data sources [8]. Third, we face a new problem, that is how to optimize the mapping query associated to a global class to perform data fusion. In our framework, the mapping query is based on the full outer join operator which is considered a good candidate in todays integrating informa- tion systems, to perform data fusion [2, 11]. The problem is that full outer join queries are very expensive, especially in a distributed environment as the one of mediator/integration systems. Database optimizers take full advantage of associativity and commutativity properties of join to implement efficient and powerful optimizations on join queries; however, only limited optimization is performed on full outer join [11]. This paper reports the description of work-in-progress about query optimization techniques for full outer join queries in mediator/integration systems. Specially, we deal with only conjunctive queries that consist of conditions of terms connected with AND's.

This work is part of the NeP4B (Networked Peers for Business)[3] project, where we aim to contribute innovative ICTs solutions for SMEs, by developing an advanced technological infrastructure to enable companies of any nature, size and geographic location

[3] http://www.dbgroup.unimo.it/nep4b

to search for partners, exchange data, negotiate and collaborate without limitations and constraints. In the NeP4B project, we assume that data sources related to the same domain belong to the same semantic peer, i.e. a super peer exposing a semantic layer. Semantic peers are related by mappings, thus building a network.

In this paper we will focus on a single semantic peer and we will introduce a methodology and a system for building and querying a Semantic Peer Data Ontology (SPDO), i.e. a global schema including traditional and multimedia data sources related to a domain.

The paper is organized as follows: Section 2 describes an applicative scenario. Section 3 proposes an overview of the system; In Section 4 the notion of DMS (*Data and Multimedia Source*) is introduced to represent and query data source and multimedia sources in a uniform way; then the problem to query a DMS is discussed. Finally, the methodology to build the SPDO related to a set of DMSs is introduced. Section 5 introduces the problem of querying the SPDO and discusses how to optimize the mapping query associated to a global class. Finally, in Section 6, we sketch out some conclusion and future work.

2 An applicative scenario

Let us consider the tourist domain where we can find many portals and websites. The information about a tourist service, location or event is frequently widespread in different specific websites, due to the specialization of the information publisher. Thus, if a user wants to have a complete knowledge about a location, s/he has to navigate through several websites. This issue generates multiple queries with search engines, retrieving incomplete and overlapping information.

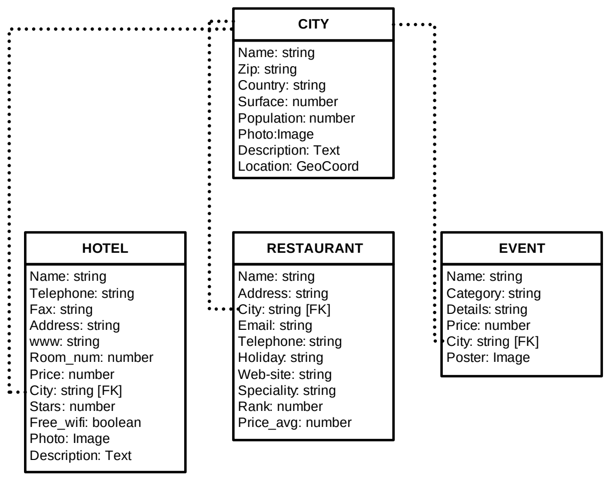

Fig. 1. The tourist integrated schema

An application, which provides a unified view of the data provided by different web sources, may provide the tourist promoters and travelers a more complete and easy to find information. Moreover, since our approach provides only a unified representation of data which still reside on specialized websites, the possibility of having out-of-date information is the same as from navigating the single specialized websites one at a time.

In Figure 1 we provide a graphical representation of the schema obtained from the integration of three data sources. It is composed of 4 classes: *hotel*, *restaurant*, *event* which are related to each other by means of the *city* class. Special attributes of the City, Hotel and Event classes are *Photo*, and *Poster* which are Multimedia Objects. These Multimedia Objects will be annotated with the MPEG-7 standard. Each data source contains traditional data types (e.g. city name in the Event class) together with multimedia data type (e.g. poster in the same Event class). Current search engines for multimedia content perform retrieval of similar objects using advanced similarity search techniques. The NeP4B project aims at combining the exact search on traditional data with a similarity search on multimedia data, exploiting the SPDO obtained by integrating the different data sources (traditional as well as multimedia). As an example, consider a user who wants to attend an event which involves fireworks. Using advanced search techniques over multimedia documents, in this case “poster”, it will be possible to search for posters that contain “festival” and related images of fireworks.

3 The system at a glance

The developed system is based on the MOMIS and the MILOS systems, where MOMIS is exploited for building and querying the SPDO, MILOS for managing the interaction with the multimedia sources (see Figure 2).

MOMIS (Mediator envirOnment for Multiple Information Sources) is a framework to perform integration of structured and semi-structured data sources, with a query management environment able to process incoming queries on the integrated schema [4]. The MOMIS integration process gives rise to a SPDO, in the form of global classes and global attributes, which may be considered as a domain ontology for the data sources.

MILOS [1] is a Multimedia Content Management System tailored to support design and effective implementation of digital library applications. MILOS supports the storage and content based retrieval of any multimedia documents whose descriptions are provided by using arbitrary metadata models represented in XML.

4 A unified view of data and multimedia sources

The notion of DMS (*Data and Multimedia Source*) is introduced to represent and query data source and multimedia sources in a uniform way. Then the problem to query a DMS is discussed.

4.1 Data and Multimedia Sources

A DMS is represented with a local schema defined in ODL$_{I^3}$ [6] and each class of a DMS schema, in general, includes a set of attributes declared using standard predefined ODL$_{I^3}$ types (such as string, double, integer, etc.). These attributes are referred to

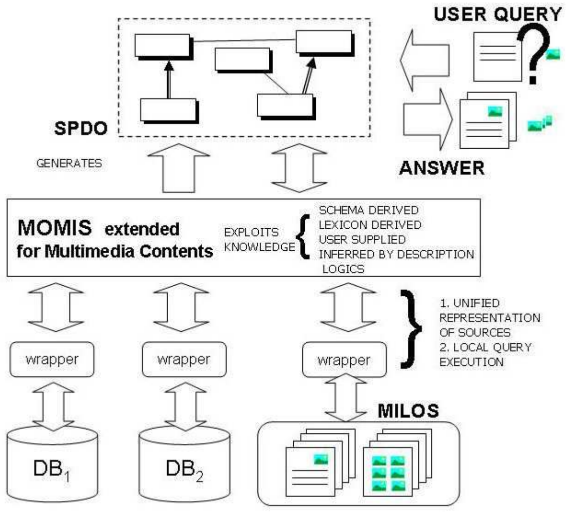

Fig. 2. The functional architecture

as *standard attributes* and support selection predicates typical of structured and semi-structured data, such as $=$, $<$, $>$, $\ldots$. Along with the standard attributes, DMS includes another set of special attributes, declared by means of special predefined classes in ODL$_{I^3}$ which support similarity based searches. We refer to these attributes as *multimedia attributes*.

Referring to our previous example of the tourist domain (see Figure 1), we can suppose two DMSs, DMS1 with a local class *resort* and DMS2 with a local class *hotel* with the following ODL$_{I^3}$ descriptions:

```
interface resort() {
   // standard attributes
   attribute   string      Name;
   attribute   string      Telephone;
   attribute   string      Fax;
   attribute   string      Web-site;
   attribute   integer     Room_num;
   attribute   integer     Price_avg;
   attribute   string      City;
   attribute   integer     Stars;
   // multimedia attributes
   attribute   Image       Photo;
   attribute   Text        Description;
}
```

```
interface hotel() {
   // standard attributes
   attribute   string      denomination;
   attribute   string      tel;
   attribute   string      fax;
   attribute   string      www;
   attribute   integer     rooms;
   attribute   integer     mean_price;
   attribute   string      location;
   attribute   boolean     free_wifi;
   attribute   integer     stars;
   // multimedia attributes
   attribute   Image       img;
   attribute   Text        commentary;
}
```

4.2 Querying Data and Multimedia Sources

A DMS M_i can be queried using an extension of standard SQL-like syntax SELECT clause. The WHERE clause consists of a conjunctive combination of predicates on the single standard attributes of M_i, as in the following:

```
SELECT $M_i.A_k, \dots M_i.S_l, \dots$
FROM $M_i$
    WHERE $M_i.A_x\ op_1\ val_1$
    AND $M_i.A_y\ op_2\ val_2$
    ...
    ORDER BY $M_i.S_w(Q_1), M_i.S_z(Q_2), \dots$
    LIMIT $K$
```

Where for the selection predicate in the form $M_i.A_f\ op_p\ val_p$, the val_p is a constant or a list of constants, and op_p is a scalar comparison operator $(=, \neq, >, \geq, <, \leq)$ or a set membership operator (IS IN, IS NOT IN). While $Q_1, Q_2, \dots$ are constants.

The query is logically processed as follows. The selection predicates (if any) applied to their corresponding attributes select as usually a subset of the objects of M_i. The ORDER BY clause sorts the entire result set coming from the selection predicates (if any) in order of decreasing similarity with respect to the given queries objects $Q_1, Q_2, \dots$. The LIMIT K clause specifies as K the maximum number of result objects desired. Note that, in contrast to what happen in traditional DBMS with standard attributes, the order in which the multimedia attributes appear in the LIMIT clause does not affect the order of the result set, as it is clarified later on.

We suppose that the following assumptions hold:

1. The way by which the returned objects are ordered is not known (black box);
2. The DMS does not return scores associated with the objects indicating the relevance of them with respect to the query;
3. If no ORDER BY clause is specified, DMS will return the records sorted in random order.

The rationale of the above assumptions is that our aim is to work in a general environment with heterogenous DMSs for which we do not have any knowledge of their scoring functions. The motivation is that the final scores themselves are often the result of the contributions of the scores of each attribute. A scoring function is therefore usually defined as an aggregation over partial heterogeneous scores (e.g., the relevance for text-based IR with keyword queries, or similarity degrees for color and texture of images in a multimedia database). Even in the simpler case of single multimedia attributes the knowledge of the scores become meaningless outside the context in which they are evaluated. As an example consider the $TF * IDF$ scoring function used by normal text search engines. The score of a document depends upon the collection statistics and search engines could use different scoring algorithms.

However, the above assumptions of considering a local DMS as a black box that does not return any score associated to result elements, do not presume that local DMSs do not use internally scoring functions for combing different multimedia attributes . Typically modern multimedia systems use fuzzy logic to aggregate scores of different multimedia attributes that are graded in the interval [0,1]. Classical examples of these functions are the min and mean functions.

To overcome this problem, as will we show in the next sections our mediator system exploits only the knowledge of the ranking ordering of the objects returned by the multimedia sources during the query in order to compute the final ranking of the merged objects.

4.3 Representing the SPDO

We build a conceptualization of a set of DMSs, composed of global classes and global attributes and mappings between the SPDO and the DMS schemata. We follow a *GAV* approach, thus this mapping is expressed by defining, for each global class G, a mapping query q_G over the local schemata. This mapping query is defined in a semi-automatic way (i.e. the designer is supported by the system to define the mapping query) as follows:

1. A *Mapping Table* (MT) is specified for each global class G, whose columns represent the n local classes $M_1, \dots, M_n$ belonging to G and whose rows represent the h global attributes of G. An element $MT(g, l)$ (with $1 \leq g \leq m$ and $1 \leq l \leq n$) represents the local attribute of M_l which is mapped onto a global attribute of G. The Mapping Table is automatically generated during the integration process, as described in [4].
2. Multimedia attributes can be mapped only onto Global multimedia attributes of the same type (for instance `Image` with `Image`, `Text` with `Text`, etc.).
3. *Data Conversion Functions* are manually specified and, for each not null element $MT(g, l)$, establish the operations that transform the values of the local attributes of M_l into the values of the corresponding the global attribute. Multimedia attributes do not need conversion functions.
4. *Join Conditions* are defined between pairs of local classes belonging to G and allow the system to identify instances of the same real-world object in different sources. Automatic object identification techniques (see for example [13]) or the designer knowledge may be exploited to define correct join conditions.
5. *Resolution Functions* [14, 7] are introduced for global attributes to solve data conflicts of local attribute values associated to the same real-world object. In [7] several resolution functions are described and classified; in our framework we consider and implement some of such resolution functions, in particular, the *PREFERRED* function, which takes the value of a preferred source and the *RANDOM* function, which takes a random value. If the designer knows that there are no data conflicts for a global attribute mapped onto more than one source (that is, the instances of the same real object in different local classes have the same value for this common attribute), s/he can define this attribute as an *Homogeneous Attribute*; of course, for homogeneous attributes resolution functions are not necessary. A global attribute mapped into only one local class is a particular case of an homogeneous attribute. For what concern the multimedia attributes, we introduce a new resolution function, called *MOST_SIMILAR*, which returns the multimedia objects most similar to the one expressed in the query (if any).

By using the introduced concepts, the mapping query is defined on the basis of the *full outerjoin-merge* operator introduced in [14]: for a global class G, the mapping query q_G is obtained by performing the *full outerjoin-merging* of the local classes $L(G)$ belonging to G.

Let us consider for example the ODL$_{I^3}$ description of the global class *Hotel* of Figure 1. This class is the result of the integration of the two local classes *resort* and *hotel*, which are defined above. The mapping table of such Global Classes takes into account the multimedia attributes `Commentary, Description` (*Text*) and `Photo, img` (*Image*) introduced by multimedia sources.

Hotel	resort	hotel
name (join)	Name	denomination
telephone	Telephone	tel
fax	Fax	fax
www	Web-site	www
room_num	Room_num	rooms
price (RF)	Price_avg	mean_price
city	City	location
stars	Stars	–
free_wifi	–	free_wifi
photo	Photo	img
description	Description	commentary

According to the mappings, we specify the *name* as join attribute intending that instances of the classes *resort* and *hotel* having the same *Name* (*denomination*) the same real object. Moreover, a resolution function may be defined for the global attribute price, i.e. the value of the global attribute price is the average of the values assumed by both the local sources.

5 Querying the SPDO

Given a global class G with m attributes of which k multimedia attributes, denoted by $G.S_1, \ldots, G.S_k$ (as `photo` and `description` in the class `Hotel`) and h standard attributes, denoted by $G.A_1, \ldots, G.A_h$ (as `name` and `city` in the class `Hotel`), a query on G (*global query*) is a conjunctive query, expressed in a simple abstract SQL-like syntax as:

SELECT $G.A_l, \ldots G.S_j, \ldots$
FROM G
  WHERE $G.A_x$ op_1 val_1
  AND $G.A_y$ op_2 val_2
  ...
  ORDER BY $G.S_w(Q_1), G.S_z(Q_2), \ldots$
  LIMIT K

To answer a global query on G, the query must be rewritten as an equivalent set of queries (*local queries*) expressed on the local classes $L(G)$ belonging to G. This query rewriting is performed by considering the mapping between the SPDO and the local schemata; in a GAV approach the query rewriting is performed by means of `query unfolding`, i.e., by expanding the global query on G according to the definition of its mapping query q_G.

As explained in the previous section, the ORDER BY clause is used to ideally reorder all the objects on the basis of their similarity expressed by the multimedia conditions. The final result will be a ranked list of the objects of each multimedia source. At a global level, the system needs to combine all this local results. We choose to fuse the results by using a Full Outerjoin-merge operation between the lists. Multimedia result lists are merged by means of both ranks and full join conditions in order to obtain a unique multimedia ranked list of results. The join conditions are defined by a set of join attributes, which are standard attributes of the local DMS.

The case of "standard" local data sources, as for instance a relational database, can be seen as a special case of multimedia source which has no multimedia attributes.

As an example consider the example of the query on the *city* class give in the previous section.

```
SELECT Name
FROM city
WHERE   Country = "Italy"
ORDER BY Photo("http://www.flickr.com/32e324e.jpg"),
         Location(41.89, 12.48)
LIMIT 100
```

Where '`http:www.flickr.com32e324e.jpg`' is the URL of the query image and (41.89, 12.48) are the coordinate of Rome.

5.1 Query unfolding

The process for executing the global query consists of the following steps:

1. **Computation of Local Query conditions:** Each atomic predicate P_i and similarity predicate in the global query are rewritten into corresponding constraints supported by the local classes. For example, the constraints `stars = 3` is translated into a constrain `Stars = 3` considering the local class `resort` and is not translated into any constraint considering the local class `hotel`.
   The ORDER BY clause for global multimedia attributes such the following
   ORDER BY $G.S_w(Q_1), G.S_z(Q_2), \ldots$
   is translated in a corresponding ORDER BY on the local class M_l restricting it only for its supported multimedia attributes (i.e., where $MT(w,l)$ is not null):
   ORDER BY $MT(w,l)(Q_1), MT(z,l)(Q_2), \ldots$
2. **Computation of Residual Conditions:** Conditions on not homogeneous standard attributes cannot be translated into local conditions: they are considered as *residual* and have to be solved at the global level. As an example, let us suppose that a numerical global attribute (as for instance `price` in our example) GA is mapped onto M_1 and M_2, and an AVG function is defined as resolution function, the constraint (GA = value) cannot be pushed at the local sources, since the AVG function has to be calculated at a global level and the constraint may be globally true but locally false. It is easy to verify that the same happens defining for `price` the *PREFERRED* or the *RANDOM* resolution function. As mentioned in previous section for multimedia attribute we use the *MOST_SIMILAR* resolution function, which

returns the multimedia objects most similar to the one expressed in the query (if any). For example, suppose we are searching for images similar to one specified in the query by means of 'ORDER BY' clause. If we retrieve two or more multimedia objects with one or more corresponding images, *MOST_SIMILAR* function will simply select the image that is more similar to the query image. There are a least two possible implementations of the *MOST_SIMILAR* resolution function:
  – **Mediator Bases Solution:** The mediator implements its similarity functions one for each multimedia attribute and hence it is able to decide which local multimedia attribute value is more similar to the query. This solution has the disadvantage that requires the evaluation of the similarity for each object returned by the sources, which can be, depending on the type of multimedia attribute, computational expensive. Moreover, the similarity implemented at level of Mediator can strongly differ from the ones implemented in the local sources.
  – **Rank Based Solution:** Another approach is to simply exploit the rank of the objects in the returned list as indicator of similarity between the attributes values belonging to the objects. This solution is better explained in the next section and it is adopted in our implementation.

3. **Fusion of local answers:** for each local source involved in the global query, a local query is generated and executed on the local sources. The local answers are fused into the global answer on the basis of the mapping query q_G defined for G, i.e. by using the Full Outerjoin-merge operation [14]. We assume that: (1) each M_i contains a key, (2) all the join conditions are on key attributes, and (3) all the join attributes are mapped into the same set of global attribute, say $G.A_k$. Objects with the same value of the key attribute represent the same real world objects.
   Intuitively, this operation is substantially performed in two steps:
   (a) Computation of the **full outer join** of local answers (`FOJ`). The result of this operation is ordered on the basis of the multimedia attributes specified in the query, this aspect is deeply examined in the next section.
   (b) Application of the **Resolution Functions :** for each attribute GA of the global query the related Resolution Function is applied to `FOJ` thus obtaining a relation R_FOJ ; in our example the result is the relation `R_FOJ(name, www, price)`. Notice that if the `www` attribute does not have any Resolution Function applied, i.e. we hold all the values, all the `www` values for a certain hotel X are retained: we will have only one record for X in `R_FOJ`, where the value of the `www` attribute is the concatenation of the `www` values (obviously adequately separated, e.g. by a comma).
4. **Application of the Residual Condition:** the result of the global query is obtained by applying the residual condition to the `R_FOJ`:

```
select name, www, price from R_FOJ
 where price < 100
```

5.2 Query Fusion: Ranking

As discussed in the previous Section, DMSs have the capability of ranking the result of a query on the basis of the similarity with respect one or more multimedia attributes.

We have pointed out that in general, we are interested in considering the general case in which we do not know anything about how the Multimedia Sources rank the result. This approach allow us to open our work towards a real scenario of integration of heterogenous sources.

In case we do not specify any multimedia attribute in the global query, local queries will not exploit the ranking and then the multimedia records coming from the sources will be fused using the FOJ operator and presented in unspecified order.

If at least one multimedia attribute is specified in the global query, then, depending on the Mapping Table, one or more local M_i will receive a multimedia query using the ORDER BY clause and hence will return ranked result lists. Let us neglect, for now, the LIMIT statement in the LIMIT clause (implicitly considering it as we had specified 'LIMIT *infinity*'). In this case, we will consider only the DMSs that receive a local query with multimedia attributes. Let $R_i = \langle r_{1,i}, \ldots, r_{n_i,i} \rangle$ be n_i be the set records satisfying the sub-query for the M_i source, sorted in decreasing order of relevance from 1 to n_i. The Unfolding process will produce a (fused) set of n_G global class records $R_G = \langle r_{1,G}, \ldots, r_{n_G,G} \rangle$ (global answer) sorted in decreasing order of relevance from 1 to n_G.

Several possible approach can be adopted to evaluate the order of the global answer, our solution is, at least in principle, to exploit an optimal rank aggregation method based on a distance measure to quantify the disagreements among different rankings. In this respect the overall ranking is the one that has minimum distance to the different rankings obtained from different sources. Several different distance measures are available in literature. However, as discussed in [12], the difficult of solving the problem of distance-based rank aggregation is related to the choice of the distance measure and its corresponding complexity that can be even NP-Hard in some cases. For instance, The rank aggregation obtained by optimizing the Kendall distance (called Kemeny optimal aggregation) for full liste is NP-hard even when the number of list four [9].

However, fortunately, our case falls into this category of the partial rank aggregation problems, in which we measures the distance between only the top-K lists rather than fully ranked lists. In fact: First, for definition of `FOJ`, a list returned by a source, even if complete, can contain records for which certain values of the key attribute $G.A_k$ are not found in the other list, i.e., the DMSs partially overlap. Second, more in general if the number of records of source is very large the use of the full lists becomes unpracticable. To work around this problem, as suggested in [10], a simple yet effective aggregation function for ordinal ranks is the *median* function. This function takes as the aggregated score of an object its median position in all the returned lists. Thus, given n multimedia sources that answer to a query by returning n different ranked lists $r_1, \ldots, r_n$, the aggregated score of an object o will be $median(r_1(o), \ldots, r_n(o))$. The aggregated list will reflect the scores computed using the $median$ function The $median$ function is demonstrated [10] to be near-optimal, even for top-k or partial lists. Then, it is possible to produce an aggregated list even if the different sources return only $top - k$ lists as their local answers to the query. For all these reasons, we take the median as the *MOST_SIMILAR* aggregation function for multimedia objects at the global level.

The objects coming from DMSs that do not receive a multimedia query or do not have multimedia attributes at all, will not influence the final rank of the global result list.

5.3 Full Outer Join Optimization Techniques

In this section we will show, by means of some examples, how to optimize the computation of the full outer join of local answers (step **3.a** of the query unfolding process).

To show how the optimization method works, for the sake of simplicity and without loss of generality, we restrict ourselves to a global query without multimedia global attributes involved, i.e. without the ORDER BY clause. Let us consider the following global query:

```
select Name, www, price from Hotel
where price < 100 and stars = 3 and free_wifi= TRUE
```

For this global query, the following local queries are generated:

```
LQ_resort = select Name, Price_avg, Web-site from resort
where Stars = 3

LQ_hotel = select denomination, mean_price, www from hotel
where free_wifi= TRUE
```

Then the Computation of the full join of local answers is:

```
FOJ = LQ_resort  join LQ_hotel on
 (LQ_resort.Name=LQ_hotel.Denomination)
```

In this example, we can optimize the FOJ expression with the following one by replacing the Full Outer Join with the join:

```
LQ_resort join LQ_hotel on
LQ_resort.Name=LQ_hotel.Denomination
```

Intuitively, since the predicate stars = 3 (free_wifi = TRUE) can be satisfied only in the class resort (hotel), an object of the result must be in LQ resort (LQ hotel) and then we can avoid the Right Outer Join on LQ resort (the Left Outer Join on LQ hotel).

As another example of simplification: If the predicate stars = 3 is not present in the initial global query, we can optimize the FOJ expression replacing the Full Outer Join with the Left Outer Join.

This simple example shows as constraints for global attributes mapped into only a local class (star and free wi in the example) can be pushed at the local level and allow the simplication of the FOJ expression. It can be demonstrated that this kind of optimization, i.e. the constraint pushing at local level and the simplification of the FOJ expression, is valid in the case of more than two local classes but only if we consider constraint on homogeneous attributes.

On the other hand, the general case is more than two local classes with not-homogeneous attributes. In this case, as we will show in the following, the simplification of the FOJ expression is still valid but constraint cannot be pushed at the local level.

Hotel	resort	hotel	myhotel
name(join)	Name	denomination	name
telephone	Telephone	tel	phone
fax	Fax	fax	fax
www	Web-site	www	link
room_num	Room_num	rooms	bedroom
price (RF)	Price_avg	mean_price	best_price
city	City	location	town
stars	Stars	-	stars
free_wifi	-	free_wifi	-
photo	Photo	img	image
description	Description	commentary	-

Fig. 3. Mapping Table of the global class Hotel

In order to consider an example of more than two local classes with not-homogeneous attributes, let us consider a new local class (called myhotel) added to the global class Hotel with the mapping table shown in table 3.

Let us consider the following global query:

```
select Name, www, price from Hotel
where stars = 3 and free_wifi= TRUE
```

for each local source involved in the global query, a local query is generated:

```
LQ_resort = select Name, Price_avg, Web-site from resort

LQ_hotel  = select denomination, mean_price, www from hotel
            where free_wifi= TRUE


 LQ_myhotel  = select name, best_price, link from myhotel
```

Computation of the **Full Join** of local answers (`FOJ`):

$FOJ_1 =$ `LQ_resort  FOJ LQ_hotel`
`on  LQ_resort.Name=LQ_hotel.Denomination`

$MFJ_1 = RF(FOJ_1)$

$FOJ_2 =$ `MFJ FOJ myhotel on MFJ.Name=LQ_myhotel.name`

$MFJ_2 = RF(FOJ_2)$

The final computation of full join for ours three classes is MFJ_2. In this case we suppose the attribute `stars` non homogeneous therefore it is not translated at the local level. Intuitively, since the predicate `stars = 3` can be satisfied only in the classes `resort` and `myhotel` the predicate `free_wifi = TRUE` can be satisfied only in the class `hotel`: an object of the result must be obtained combining tuples of `LQ_resort` and tuples of `LQ_hotel` or combinig tuples of `LQ_hotel` and tuples of `LQ_myhotel`. This means that we can optimize the `FOJ` expression for these three classes in the following way:

$FOJ_1^{'} =$ `LQ_resort left outer join LQ_hotel`
`on  LQ_resort.Name=LQ_hotel.Denomination`

$MFJ_1^{'} = RF(FJ_1)$
$FOJ'_2 = MFJ_1^{'}$ `right outer join myhotel`
`on MFJ.Name=LQ_myhotel.name`
$MFJ'_2 = RF(FJ_2)$

The final computation of full join for ours three classes is MFJ'_2 optimized because the partial result of $FJ_1^{'}$ is less than FJ_1 and $FJ_2^{'}$ is less than FJ_2. Finally the application of residual predicate on $MFJ_2^{'}$ will be fast than MFJ_2 because the number of tuple in $MFJ_2^{'}$ involved in the selection operation is less than a number of tuple in MFJ_2.

Even if the non-homogeneous attribute `stars`= 3+ has not been translated to the local level, we have seen how it is always possible to optimize the FOJ expression. So the fact that an attribute is not homogeneous does not affect the optimization of the FOJ expression. As in the first example, we can note that full outerjoin was replaced with a left(right) outerjoin or a join.

6 Conclusion and Future Work

We presented a methodology implemented in a tool that allows a user to create and query an integrated view of data and multimedia sources. The methodology joins and extends the capabilities of two previously developed tools: the MOMIS integration system and the MILOS multimedia content management system. In particular, concerning the query processing, we discussed some cases where multimedia constraints can be expressed in a global query without requiring multimedia processing capabilities at the global level. This is an interesting result, since it upsets the usual paradigm on which mediator systems are based, stating that the *query processing power* of a mediator is greater than the one of the integrated data sources [8]. We will evaluate the effect of this aspect in our future work.

Future work will also be devoted to experiment the tool in real scenarios. In particular, our tool will be exploited for integrating business catalogs related to the area of "tiles". We think that such data may provide useful test cases because of the need of connecting data about the features of the tiles with their images.

References

1. G. Amato, C. Gennaro, P. Savino, and F. Rabitti. Milos: a Multimedia Content Management System for Digital Library Applications. In *Proceedings of the 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2004)*, volume 3232 of *Lecture Notes in Computer Science*, pages 14–25. Springer, September 2004.
2. D. Beneventano and S. Bergamaschi. Semantic search engines based on data integration systems. In *Semantic Web Services: Theory, Tools and Applications.* Idea Group., 2006. To appear. Available at http://dbgroup.unimo.it/SSE/SSE.pdf.
3. D. Beneventano, S. Bergamaschi, C. Gennaro, F. Guerra, M. Mordacchini, and A. Sala. A mediator system for data and multimedia sources. In *International Workshop on Data Integration through Semantic Technology at the 3rd Asian Semantic Web Conference. 08-11 december 2008, Bangkok*, 2008.

4. D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Synthesizing an integrated ontology. *IEEE Internet Computing*, 7(5):42–51, 2003.
5. D. Beneventano, C. Gennaro, and F. Guerra. A methodology for building and querying an ontology representing data and multimedia sources. In *International VLDB Workshop on Ontology-based Techniques for DataBases in Information Systems and Knowledge Systems, ODBIS 2008, Auckland, New Zealand, August 23, 2008, Co-located with the 34th International Conference on Very Large Data Bases*, 2008.
6. S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration of heterogeneous information sources. *Data Knowl. Eng.*, 36(3):215–249, 2001.
7. J. Bleiholder and F. Naumann. Conflict handling strategies in an integrated information system. In *IIWEB '06: Proceedings of the WWW Workshop in Information Integration on the Web (IIWeb)*, 2006.
8. K. C.-C. Chang and H. Garcia-Molina. Mind your vocabulary: Query mapping across heterogeneous information sources. In *SIGMOD Conference*, pages 335–346, 1999.
9. C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM.
10. R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *PODS '04: Proceedings of the twenty-third symposium on Principles of database systems*, pages 47–58, New York, NY, USA, 2004. ACM.
11. C. A. Galindo-Legaria and A. Rosenthal. Outerjoin simplification and reordering for query optimization. *ACM Trans. Database Syst.*, 22(1):43–73, 1997.
12. I. F. Ilyas, G. Beskales, and M. A. Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Comput. Surv.*, 40(4):1–58, 2008.
13. F. Naumann, A. Bilke, J. Bleiholder, and M. Weis. Data fusion in three steps: Resolving schema, tuple, and value inconsistencies. *IEEE Data Eng. Bull.*, 29(2):21–31, 2006.
14. F. Naumann, J. C. Freytag, and U. Leser. Completeness of integrated information sources. *Inf. Syst.*, 29(7):583–615, 2004.