Journal of Computer Science

Original Research Paper

Comparative of Mediator Approach for Database Integration

[1]Mohd Kamir Yusof, [2]Md Yazid Mohd Saman and [1]Wan Nor Shuhadah Wan Nik

[1]*Universiti Sultan Zainal Abidin, Kampus Tembila, 22200 Besut, Terengganu, Malaysia*
[2]*Universiti Malaysia Terengganu, 21300 Kuala Terengganu, Terengganu, Malaysia*

Article history
Received: 27-07-2013
Revised: 29-03-2014
Accepted: 09-08-2014

Corresponding Author:
Mohd Kamir Yusof
Universiti Sultan Zainal Abidin,
Kampus Tembila, 22200 Besut,
Terengganu, Malaysia
Email: mohdkamir@unisza.edu.my

Abstract: Six applications which are based on mediator approach have been reviewed in this study. In practice, The mediator is used to integrate and access data from different data sources. The important characteristics for the implementation of mediator approach have been identified. This include types of data, file format and object data. These characteristics together with the advantages and the disadvantages of each implementation mediator approach described in section 4. This is important for other researchers to do the literature review. Indeed, this study highlights important issues that need to be addressed before future directions for the research in the area of database integration based on mediator is channelled.

Keywords: Database, Database Integration, Mediator

Introduction

A database can be defined as a collection of structured interrelated information units (Liao and Mcleod, 2001; Yusof and Safei, 2013). Database also can define as a collection of related data (Elmasri and Navathe, 2010). A package of information at various levels of granularity is represented as an information unit. An example of information units is a collection of data from experiments such as a string of letters. An increasing number of information units will affect the database size. In order to deal with unlimited and continuous growth of database size, a Database Management System (DBMS) is required. The DBMS is a set of programs that enables administrator to store, modify and extract the information from database (Geppert and Dittrich, 2002). DBMS also can defined as a collection of program enables users to create and maintain a database (Elmasri and Navathe, 2010). In addition, a DBMS also provides other functions such as to add, delete, access and analyze data in one location. Standard language used in DBMS is Structured Query Langague (SQL) (Benedikt and Senellart, 2011) for all operations such as insert, update and delete. Example of DBMS such as DB2 for IBM, Oracle, MySQL and Informix. A DBMS supports an abstract data model that consists of Data Definition Language (DDL) and Data Manipulation Language (DML) (Aberer, 2003). A DBMS coordinates the access by multiple users by supporting concepts of transactions (operation that moves the databse from one consistent state to another). Usually, a single application has a single database. Database is used to store information. However, a single application may has more than one database. In order to allow more than one database can be accessed, database integration is required. Therefore, database integration is defined as a combination of two or more databases from different sources. This integration provides a unified view of data from different sources (Robinson and Rahayu, 2004). Database integration is performed whenever two or more combined of databases either physically or virtually (Lim and Chiang, 2000). Physical database integration requires the original databases to be discarded after the integrated database has been constructed and all existing application software to be migrated to the database system operating the integrated database. Meanwhile, virtual database integration is deploys a multidatabase or data warehousing system to support queries on an integrated view constructed upon the original databases.

This integration allows users from different applications to make any data transaction such as search, view or send the data through a “suitable bridge”. A suitable bridge means an approach that can be used for database integration. One of the most popular approaches for database integration is the mediator. A mediator can be defined as a collection of functionality. Basic approach in mediator is translate query from the user, into query that is understood by the sources integrated into a system (Ishak and Salim, 2006). There are two functions; add and remove with minimal changes on each layers (Peng *et al*., 2004). One of the reasons why a mediator is popular in database integration is because of their functionality of high level of transparency to the user. A mediator accepts a query from the client/user, determine the sources needed to answer the query and decomposes the query into subqueries for each required sources (Bichutskiy, 2013). The purpose of mediator approach is to translate the global query into an executed query for each wrapper. This allows reconciliation between data from different databasewith different data source schemas before integrate them into a coherent global schema. Two important components in mediator are mediator and wrapper (Thiran, 2004). The main function of a mediator is to receive the formulated query from the global mediated schema. Then, the mediator will decompose the query into sub-queries that can map each data source's execution based on individual database. After that, the mediator will reformulate the queries into the respective execution plan based on the local models and schemas beforesend it to the wrapper. The wrapper is responsible to convert these queries into an expected execution format depending on the data source. The wrapper is also responsible to fetch data and respond from a specific data sources. In section 2, the example ofapplications that implement mediator approach has been described. This description summarizes the advantages, disadvantages, domain areas and any potential direction for future research works in the area of mediator approach.

Implementation of Mediator Approach

In this section, six different applications which implement mediator approach have been studied. The details are provided in view of different domains.

The Stanford-IBM Manager of Multiple Information Sources (TSIMMIS)

This project integrates heterogeneous information where structured and semi-structured data are involved (Shi, 2002). Figure 1 shows the architecture of TSIMMIS. In this Fig. 1, a wrapper acts as a translator where it converts the data object to a common information model. This wrapper also converts the queries over information in the common model into requests so that the data model can be executed. Then, the wrapper also converts the results returned by the data source into the common model.

In this project, the provided toolkit extracts the structured and semi-structured data sources. These data sources are required in modelling the Object Exchange Model (OEM). The string label represents the name of the object. The type of the object can be primitive such as integer, double or suite. where the value field may contains the real primitive value such as 234 or 'state'. However, if the type is 'set', the value field may contains a set of values which could be set or primitives.

This is an example of SQL language called OEM-QL:

SELECT Fetch-Expression FROM Object WHERE Condition. This statement results the following answer:<answer, set, {obj1, obj2 ...}>

GARLIC System

The GARLIC system act as a middleware system. The main function of this system is to provide an integrated view for heterogeneous legacy data sources. That is, any changes on how or where data is stored are transparent to users (Shi, 2002). Figure 2 shows the architecture of GRALIC.

Wrapper

The main function of a wrapper is to transform the underlying schema and data before the data sources are mapped to a Garlic Middleware model. The wrapper have a specialised facilities for query processing and translation of data from a particular class of external data sources (Risch *et al*., 2000). Then, the wrapper will model all collections of similar items in class interfaces from data sources and provides an object oriented view to the mediator. After that, each interface will be implemented. Each interface may have more than one implementation process if each class interface has some subset of that collection which is located in some data sources.

In this case, the wrapper is represented as an object. The wrapper will handle a query and return the result to Garlic middleware before send it to clients.

The Mediator

The mediator is represented as a middleware in Garlic. The heterogeneous data are stored in different sources and are assigned as objects (Eltabakh, 2012). These objects have a unified schema and common interfaces. One of the functions of Garlic middleware is to merge all schemas of individual data sources into a single, global schema. However, this process must go through a wrapper registration step. In the registration step, the wrapper produces their own data as "Garlic objects" and provides an "interface" definition which describes the behavior of these objects. The advantage of creating their own object is that, the wrapper can rename objects and attributes and also change their types and relationships.

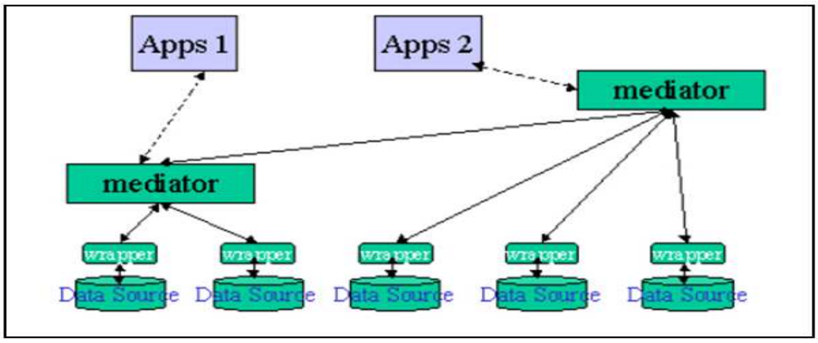

Fig. 1. TSMMIS architecture (Shi, 2002)

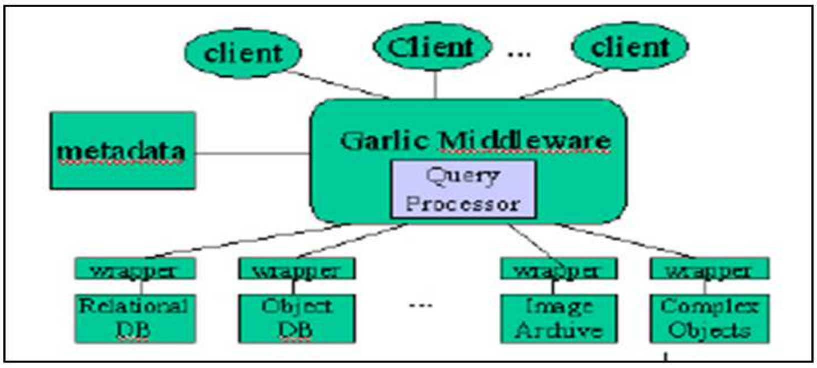

Fig. 2. GARLIC architecture (Shi, 2002)

Manual process in modelling is required for semi-structured and unstructured data sources. However, structured data sources can be modelled automatically. The formula above shows a common mapping relational model and the object oriented model (Garlic unified model):

Relational		Object oriented
Tuple	→	object
Column	→	attribute

Query

A query processor is provided in Garlic middleware. Sometimes, it is called as a query engine. The function of a query processor is to optimize and execute queries over different data sources posed in an object extended SQL. The query processor is required to communicate with “wrappers” for various data sources involved in the query.

Mediator Based Architecture Based for Integrated Access Biological Data

This research developed a tool to access multiple biological databases (Peng *et al*., 2004). The tool is used for data analysis. The data is analyzed based on different data sources. Then, the result are fed into various of application programs such as functional domain, protein structural prediction and motive identification. The results are important for certain biological research purposes.

A mediator-based architecture was proposed to solve the above problems. The main feature of this architecture is that data storage is not required to store data. Queries from users are executed remotely where searching and retrieving process are come from the original data sources. Then the results from both processes are sent to the users. Figure 3 shows the example of mediator-based architecture. There are three major components; GUI interface, Query transformation and Query execution. The GUI interface collects the information (declarative query) and formulate the query. The query transformation constructs a meaningful query (referred as a fully parameterized query that contain information on data source) based on the declarative query from the GUI interface. The execution query is responsible to facilitate the physical execution of queries against the data source, handle the communication with remote database, get the data and responds from different database sources.

This architecture is designed to access Swiss-Port (a protein sequence bank) and Gen Bank (a popular nucleotide sequence database) as a data sources. There are two types of schema; global and local schema. Each databank schemas are stored in a relational data model (MySQL).

The wrapper is responsible to receive query from data source relation. Then, the result is returned from the actual data source. Some experiments have been conducted to evaluate the performance of mediator-based architecture.

In Fig. 4, a response time for mediator approach is slower than GIM and SRS because of the delay of query translation, process of parsing and filtering the results in wrapper components, network propagation time for submitting the query and getting back the result. Meanwhile, Fig. 5 proved that a mediator approach is better as compared to SRS in term of response time. This is due to the less number of parameters involved. The another reason is that the mediator approach hasmore execution time as compared to a complicated query which has more number of parameters.

Integrated Heterogeneous Data Sources Using XML Mediator

The mediator act as a middleware between the collection of data sources and the user. There are ten main operations in the mediator (Rochlani, 2012):

- Query analysis
- Considered two types of analysis

  i. Syntactic analysis (accordance with grammar)
  ii. Semantic analysis (query schema)

- Query translation
- Translate the user query to XML
- Optimization of the query
- Divide users' query into several sub-queries
- Translation of the result to the user's format
- Reformulate the answer according users' format
- Mediator cache manager
- Manage the semantic cache of the mediator

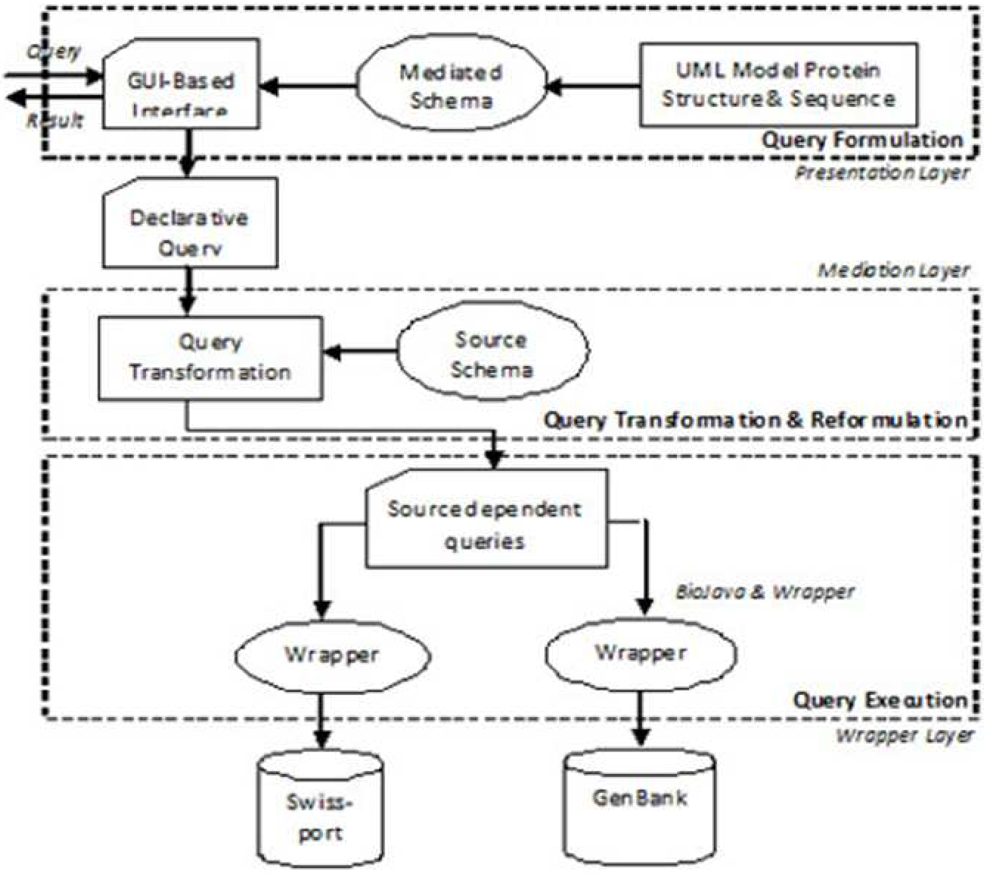

Fig. 3. Mediator-based architecture (Peng *et al*., 2004)

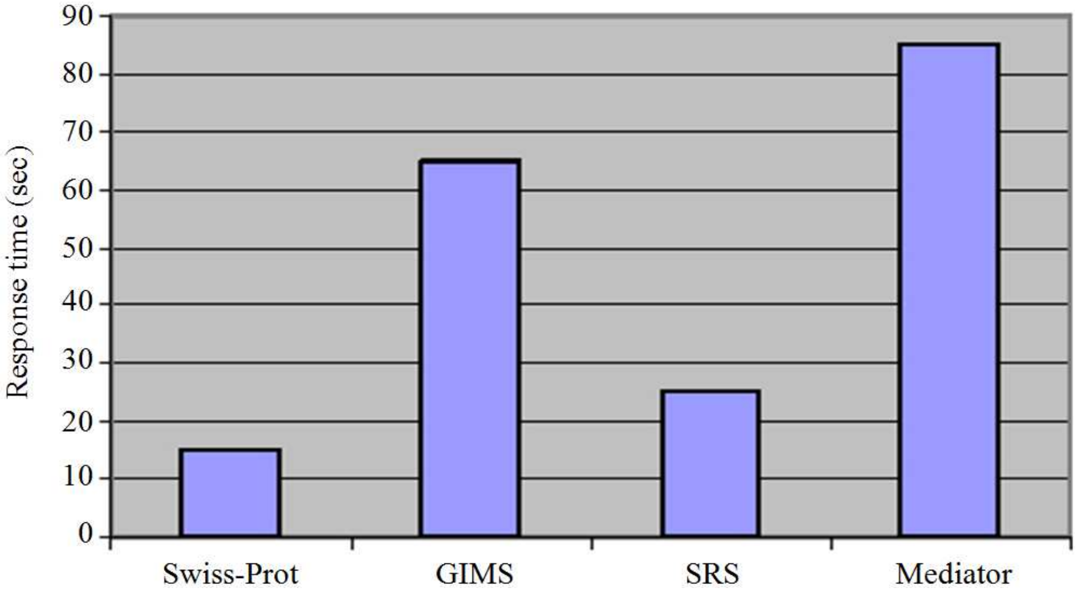

Fig. 4. Integration approach (Peng *et al*., 2004)

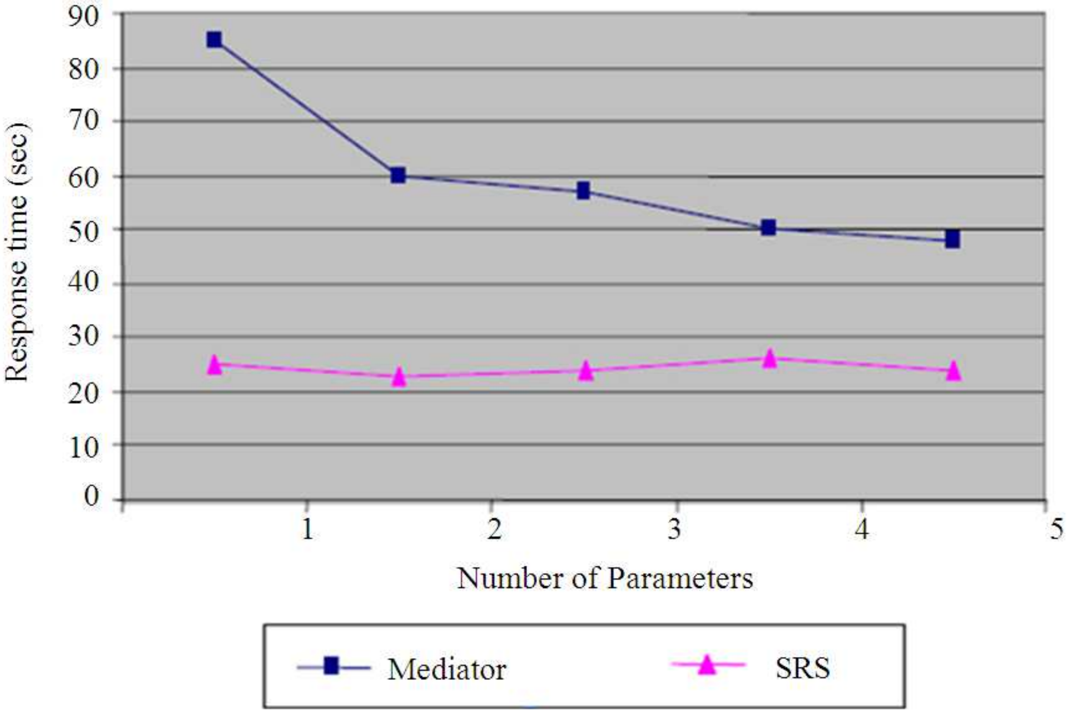

Fig. 5. Number of parameters (Peng *et al*., 2004)

Figure 6 shows the architecture of the mediator. In this architecture, each component has a role (please refer to Table 1).

The discussion on global and mapping schema are presented. In global schema, all domains are identified. These domains are modelled in hierarchical structure. Figure 7 shows the integration of domains in a tree structure. In Fig. 7, each node represents a group of sub domain. Meanwhile, global schema integration considers five criteria. There are a list of attributes and a description of the domain, a list of its attributes, a list of the integrated sources, a list of the integrated tools and list of sub domains generated by root domain.

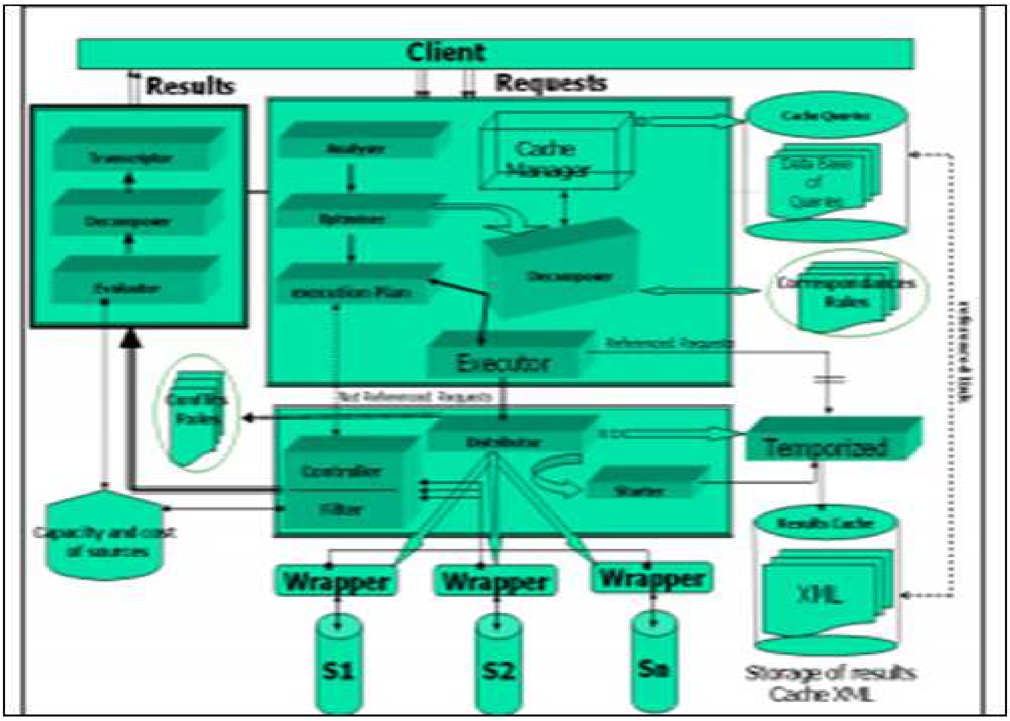

Fig. 6. Architecture of mediator (Rochlani, 2012)

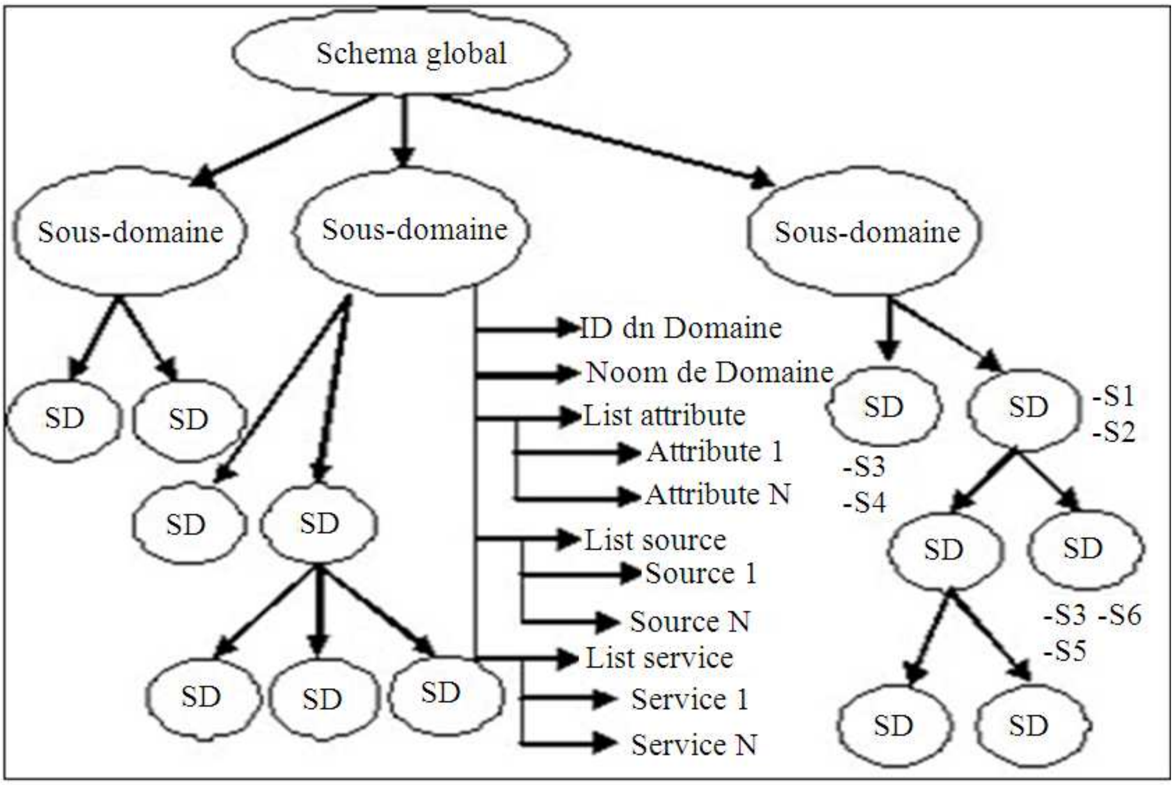

Fig. 7. Integration of tree structure (Rochlani, 2012)

Table 1. Components' role

Component	Role
Analyzer	Analyze user query based on syntactic and lexical checking
Optimizer	Optimizes the query
Decomposer	Carry out the operation of the query. Then, it generates a sub queries and send them to the specific Wrappers on local sources.
Execution plan generator	Set an execution order based on the sub query.
Queries executer	Transmission of sub queries are carried out by the wrappers and the manager of the semantic cache.
Temporization	Execute the sub queries on the local sources and the semantic cache query synchronously
Starter	Start the operations on the overlapped sub queries execution and data filtering
Controller/filter	Carry out the operations on the overlapped sub-queries execution and the data filtering
Evaluator	Control cost of various resources
Decomposer	Allow the combination of results from various queried local sources and those of the semantic cache
Transcriptor	Provide a final results to users
Cache results database	Cache queries the database. Keeps queries history submitted to the mediator. Keeps the users queries execution results
Correspondences rules	Bind the elements of the schema sources with those in Global Schema
Conflicts rules	Manage the mapping phase, solve the inter-Schema problem and establish the inter-Schema correspondences
Wrapper	Main task is to wrap a data source in sequence so that the data source can interact with the rest of the integration

The advantages of domains structuring are to facilitate and optimize, mediate a query by the users and generate the query execution plan. Another advantage is that users can easily explore the integrated global schema in order to determine the list of sources which can be queried by the mediator.

However, the main problem in data integration is the correspondence between a data source and global schema. The mediator cannot answer users query because of this problem. This problem occurs due to semantic conflict. It means that two databases containing semantically related information. As an example, consider a site X (Table 2) and site Y (Table 3) which contain tables named products and product list respectively. Some of examples on semantic discrepancies between these sites are below:

- Attribute conflicts
- Value to value conflicts
- Table to table conflicts

Table 2 and 3 are combined as a Product Data (pno, name, company, cost, dealer, location, manufactureyear) using XML approach. XSLT and template files are used to transform both of these table. Table 4 shows the product data table based on the combination of Table 2 and 3 using XML approach.

The XML approach is one of the solution for the semantic conflicts. As a result, the mediator is able to provide the following features:

- Provide different designs and architecture
- Integrate heterogeneous data sources

Mediator Approach for Grid Computing

GDMS technology was implemented in medical domain for treatment of traumatic brain injury at Viena. This project enable access to more than one data source over global schema with a subset of Structured Query Language (SQL) (Wöhrer *et al.*, 2005). Figure 8 shows the components involved in GDMS. In Fig. 8, GDMS provides a virtual data source to handle the heterogeneity of data sources. Three different data formats used are MySQL DBMS, Xindice XML DB and CSV file format. The GDMS provides services to three different applications such as eBusiness Mall System, Data Mining Software and other application.

This research focuses on integration between Open Grid Services Architecture Data Access and Integration (OGSA-DAI) and GDMS. One of the open issue in Grid is Virtualization which leads to the problem ofloss of data access performance. Gridprovide high performance for application in data access.

A mediator approach in Grid has been applied to achieve the following objectives; (1) to link data sources from different data structures, (2) to provide ability to use data in one resource based on matching criteria or conditions for retrieving data from another resource, (3) to enable the construction of distributed queries when the target data resources are located at different sites and support heterogeneous and federated queries when some data resources are accessed through different query languages.

Table 2. Product (Site X)

pNO	Name	Company	Cost	Dealer	Location	Components
1	Motherboard	Heiss	2000	IBM compnay	India	100
2	Micro controllers	Joe	2500	Microns company	USA	65

Table 3. Product (Site Y)

pID	Productname	Company	Cost	Dealer	Location	Manufactured year
1	Motherboard	Heiss	2000	IBM compnay	India	100
2	Micro Controllers	Joe	2500	Microns company	USA	65

Table 4. Product data

pID	Productname	Company	Cost	Dealer	Location	Manufactured year
1	Motherboard	Heiss	2000	IBM compnay	India	100
2	Micro controllers	Joe	2500	Microns company	USA	65

Figure 8 shows the whole components involved in Grid Data Mediation Service (GDMS) whichhandle and hide the heterogeneity of the involved data sources.

Three components involved in the mediation approachare query reformulation, query optimization and query execution. Figure 9 shows the architecture of the GDMS modules which has been integrated into OGSA-DAI.

Query reformulation

The query reformulation translate the mediated schema into an internal representation (a query graph). Then, partitions of the table must be selected to answer the query. The query reformulation will be considered if have any specified in the WHERE-clause of the query.

Query Optimizer

In this phase, the query will be optimized depending on the data required. The optimizer component must be able to choose another replica (if specified) when one replicated data resource is not responding. Assume that the required MySQL data source is not available but the data has been exported into a CSV file. The CSV is a replication of the MySQL database. In this case, the integration between MySQL database and CSV file is needed, but the leaf of the query tree has to be re-instantiated with other implementation for the selected operation.

Query Execution Engine

In this phase, operators acts as iterators. All iterators have the same interface. The advantages of iterators include it can be plugged together and support the pipelining of the results from one operator to another to achieve a good performance. XQuery engine, i.e.,SAXON is used in this operator for fast prototyping, ease of use and flexible adaptability with new data sources and functionalities.

Mediator Approach in Integrating Heterogeneous Multimedia Data Sources

Multimedia data management is known as Multimedia digital Library for On-line Search (MILOS). The feature of MILOS is to support the storage and content based retrieval for any multimedia documents. The metadata model for multimedia data is represented in XML format. XML supports XML query language standard such as XQuery and XPath. XML also offers advanced search and indexing of arbitrary documents. MILOS's XML data also provides other features such as full text search, automatic classification and feature similarity search (Beneventano *et al*., 2011).

A Framework of Multiple Information Sources (MIMOS) is designed for structured and semi structured information extraction and integration from local data sources. A semantic approach is used in the integration process. A Global Schema is developed to view all integrated data sources. In MIMOS, a query manager will process global queries from the Global Schema.

A methodology and system for building and querying a Data and Multimedia Global Schema (DMGS) resulting from the integration of traditional and multimedia data sources was proposed in this research. It is the extension from MOMIS and MILOS. The first step is to introduce the notion of a Data and Multimedia Source (DMS). Then, results of such queries are ranked based on issue with similarity predicates. Median Rank Algorithm (MEDRANK) method is used to rank the result. In MEDRANK, a similarity predicates can be expressed in a global query without requiring multimedia processing capabilities at the mediator level. Figure 10 shows the functional architecture of proposed methodology based on mediator wrapper architecture.

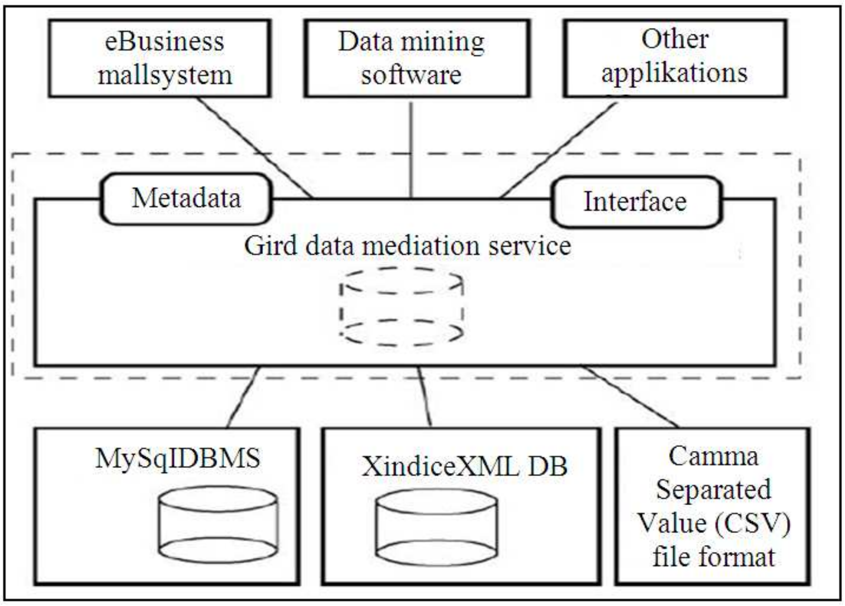

Fig. 8. Grid Data Mediation Service (GDMS) provides Virtual Data Source, hiding and handling the heterogeneity of the involved data sources

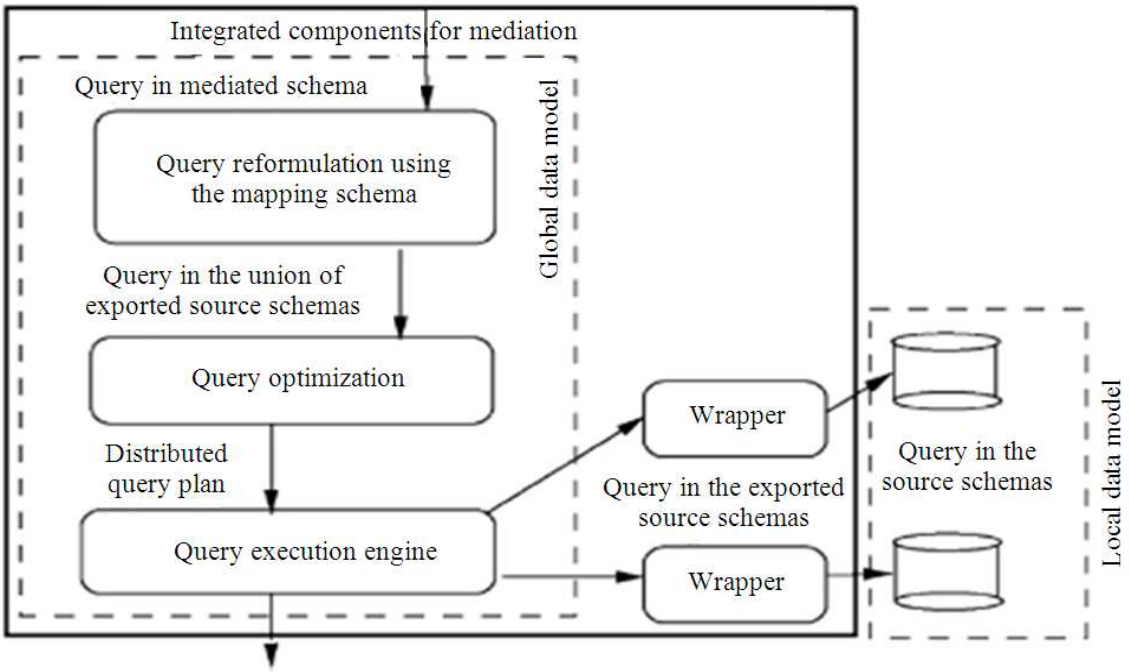
Integrated components for mediation
Query in mediated schema
Global data model
Query reformulation using the mapping schema
Query in the union of exported source schemas
Query optimization
Distributed query plan
Query execution engine
Wrapper
Query in the exported source schemas
Wrapper
Query in the source schemas
Local data model

Fig. 9. An Architecture of the GDMS modules integrated into OGSA-DAI

The wrapper and mediator are two important parts in Fig. 10. The wrappers are placed over each local source. The translation of metadata description on the local sources into a common language is translated by the wrapper. Then, the wrapper translates a global query into local queries for the local sources and export local query answers. Differ from MIMOS, the wrappers is available for commercial DBMSs, such as SQL SERVER and ORACLE. In this proposed methodology, the dedicated wrapper was developed to integrate multimedia local sources.

Meanwhile, the mediator performs two tasks. The first task is to create the Global Schema. The second is to execute a global query.

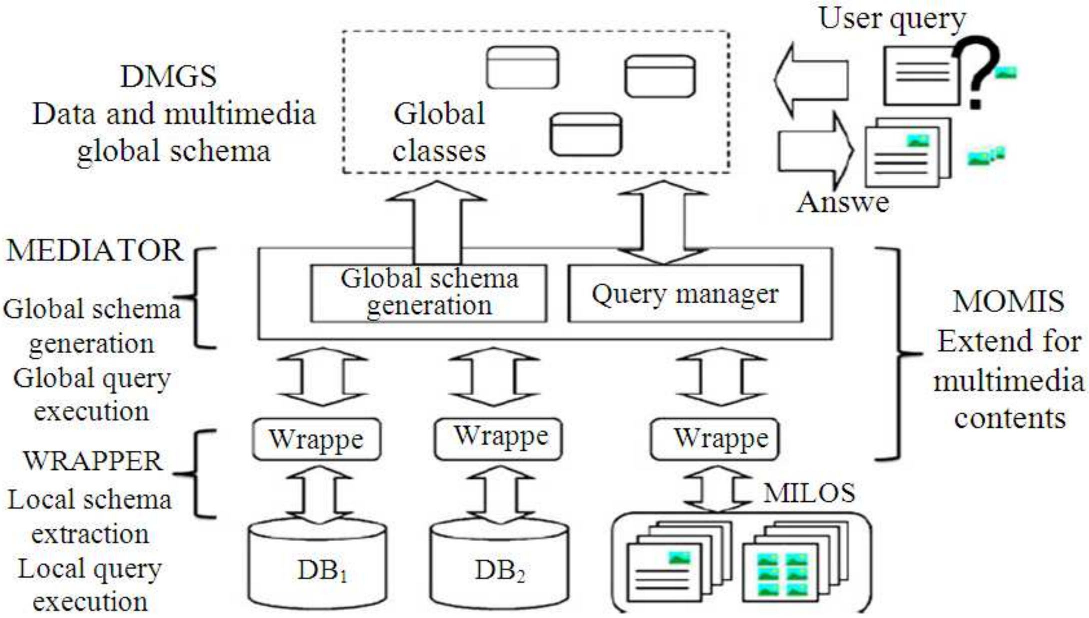

Fig. 10. The functional architecture (Beneventano *et al*., 2011)

Unified View of Data and Multimedia Source

Data and Multimedia Local Sources

The Data and Multimedia Source (DMS) is used to represent and query the data sources and multimedia sources. The DMS and MIMOS ODL language describes the heterogeneous schemas of structured and semi structured data sources. The function of ODL_I^3 language is to support multimedia document (e.g., Image) where it is represented in DMS object. There are two types of objects, i.e., simple textual attributes and complex structures. The attribute of simple text includes title, comment and date of creation. The standard of predefined ODL_I^3 is used for these attributes. ODL_I^3 supports different type of attributes such as string, double and integer. The ODL_I^3 language is also able to support selection predicates of structured and semi structured data such as = and<and>.

Meanwhile, multimedia standard MPEG-7 is used to support complex structures. The ODL_I^3 is also used to support multimedia attributes. The ODL_I^3 was chosen because this language is able to support similarity.

Implementation of Mediator Approach

SME trading for marble and granite was used as an example of simple scenario for this proposed methodology. In this scenario, two actors are involved, i.e., marble producers and customers. Usually, the customer finds out the type of marble from the internet or market. Then, the customer sends the requests (i.e., Queries) to marble producers through images of the desired materials and with short textual descriptions.

The marble producers manually analyze the material requested from the customer. In this case, the requests are typically sent to one producer at a time and each producer is needed to analyze the material requests manually.

The semantic multimedia system was designed to support the distributed resolution of such queries efficiently. This technique gives advantages to the financial department for analyzing the orders, find the best producer and time-and cost-consuming.

The marbles are stored in multimedia database in highquality images of every slab in their yard by producers. The category for each material types is represented in each slab.

In the transaction to find the marble based on customer needs, the customer need to submit a marble image to the producers. The submission must be through the form based on a mix of images and textual data. Then, the system will find all images from data sources which meet the customer request. Standard MPEG-7 features are used for image comparison. Finally, the results will be returned to the customer.

Query Data and Multimedia Local Sources

The ODL_I^3 describes two local classes i.e., marble and slab.

The mutltimedia attributes based on A local class M of a DMS is represented by S_1,..,S_m (e.g., phone and description in Marble class). Meanwhile, h represents a standard attributes by $A_1,...,A_h$ (e.g., width-cm and height-cm in the Marble class).

Example of SQL-like syntax as:
SELECT S_1,..A_1,... FROM M WHERE A_1 op_1 val_1 AND A_2 op_2 val_2 ORDER BY $S_1(Q_1)$, $S_2(Q_2)$,... LIMIT k

Where:

- WHERE refers to the conjunction of atomic predicates on standard attributes, for instance val_i constant and op_i is a relational operator (=, ≠, ≥, ≤, >, <)
- ORDER BY refers to a set of similarity predicates on multimedia attributes of the form $S_i(Q_i)$, where Q_i is a constant query object

This is an example of query on the Marble local class to find all objects on 3 cm of thickness:

```
SELECT      id, material, euro
FROM        Marble
WHERE       thickness = 3
ORDER BY    photo (“slab12345.jpg”)
LIMIT       100
```

Based on the above example, the clause ORDER BY photo (“slab12345.jpg”) and LIMIT 100 allows users to retrieve the most similar 100 Marble objects.

Querying a Global Class

A global class G based on a query on G (global query) is expressed as below:

```
SELECT      id, material, euro
FROM        Marble
WHERE       thickness = 3
ORDER BY    photo (“slab12345.jpg”)
LIMIT       100
```

Three processes are involved to process the above query. There are query unfolding, a fusion of local queries and application of resolution functions and residual predicates.

A. Query Unfolding

A global query on G to the equivalent set of queries (local quarries) can be expressed in the local classes belonging to G. The global class and its local classes is mapped.

Example

The predicate (Area = 2) is translated into (area_in_meter = 2) for the local class Slab.

The local queriesincludes the local join attribute with zero or more attributes have been generated as below:

```
LQ-Marble   =  SELECT id, material, euro, thickness
               FROM Marble
               ORDER BY photo (“slab12345.jpg”)
LQ-Slab     =  SELECT s-id, class, price
               FROM Slab
               ORDER BY img (“slab12345.jpg”)
LQ-MySlab   =  SELECT slb-id, thickness, cost
               FROM MySlab
               ORDER BY image (“slab12345.jpg”)
```

Fusion of Local Queries

It executes the local query on the related local source and fuse the local answer at global level. However, if no other similarity predicates in the global query, it uses a full outer join operation using the SQL engine in order to fuse the local answer.

Example

R_F (ID, LQ_Marble_material, LQ_Marble-euro, LQ_Marble_thickness, LQ_Slab-class, LQ_Slab_price, LQ_MySlab_thickness, LQ_MySlab_cost).

Application of Resolution Functions and Residual Predicates

All related resolution functions and residual predicates are computed on the attributes for each homogeneous attribute of the global query.

Evaluation of Mediator Approach for Integration of Heterogeneous Multimedia Sources

A tool was developed based on integration between MIMOS and MILOS. This tool provides a function to elaborate global queries with similarity predicates.

A web user interface was designed to test the methodology based MIMOS framework to integrate with a set of DMS. The user can perform an image similarity search beginning from one of the randomly selected image through this interface. Advanced search can be used for advanced searching, where an image is provided by the user (using similarity) or as a free text (full text search) or by stipulating restrictions on the basis of the metadata field of the slabs. For instance, the user can specify the class of material of the slab, the ID of the block and the thickness of the slab.

Figure 11 shows the interface where users can select an image by clicking the “Slogia” button. Also, user is able to use an advanced search by entering a full text search or value or field type or price. Then, the user need to click the search button. After that, this application will retrieve all images based on similar attributes. Figure 12 shows all images based on image query entered by user. Based on this results, 15 images were displayed where these images is quite similar to the query image (with specific thickness of 30 mm).

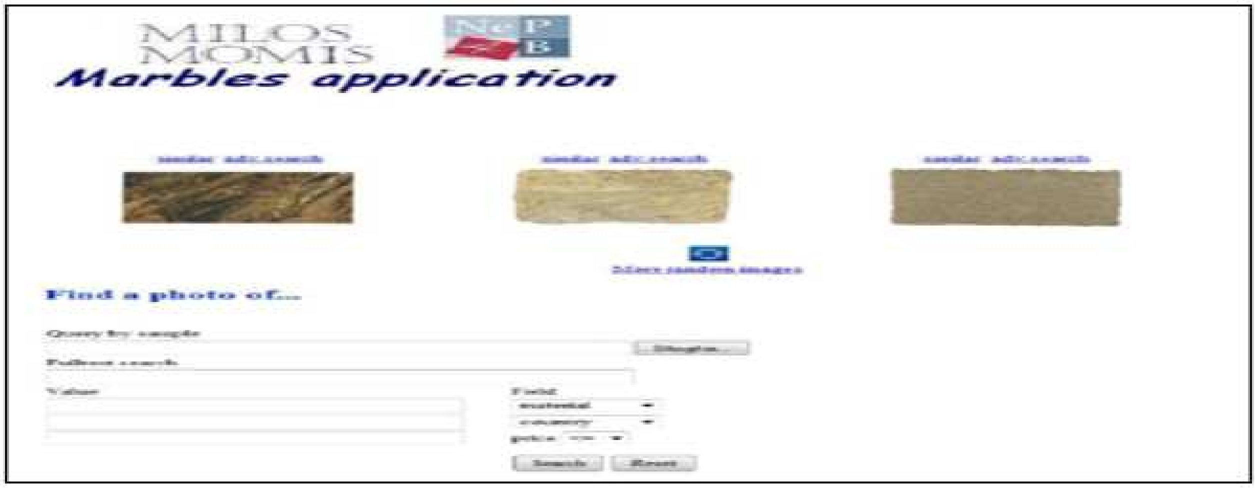

Fig. 11. Interface for query image (Beneventano *et al*., 2011)

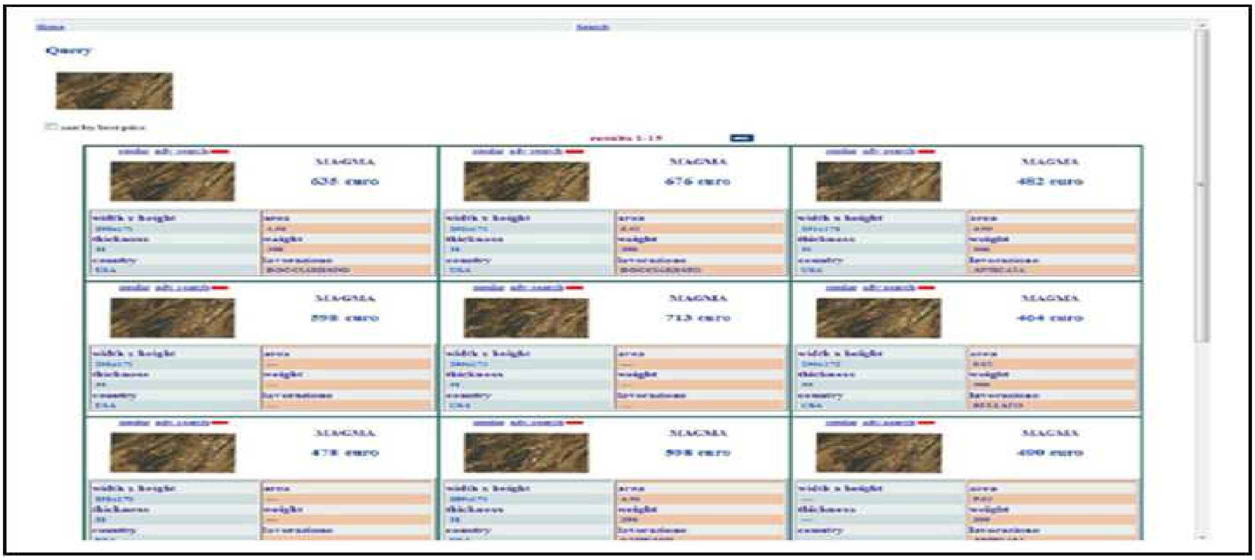

Fig. 12. Interface 2 (Beneventano *et al*., 2011)

Implementation of similarity predicates of this tool can make multimedia sources ranking process easier. The similarity predicates on multimedia attributes does not need to be executed at the mediator level but can be expressed in global query. This implementation is more efficient to produce a good result based on rank.

However, this technique doesnot support multi queries and similarity joints. Moreover, this technique can be improve by extending the query language to provide a capability to support complex queries.

In this research, the researchers focuses on mediation and Virtualization of heterogeneous data sources over Grid. XML was used to query transportation and result, describe database schema and data storing. GDMS acts as a platform and was integrated into OGSA-DAI. The result shows that query execution for retrieving data from data sources is more efficient as compared to GDMS.

Analysis of Current Implementation of Mediator Approach

This section summarizes the implementation of mediator approach for database integration in different application domain.

The characteristics of each application based on mediator approach are shown in Table 5. From this table, three applications based on mediator approach with more specified domain i.e., medical, biological and multimedia data., are presented. Other implementation of the mediator comes from more general domain. Two applications are developed to access data from heterogeneous database and other application area focuses to access homogeneous database.

Three types of data which are structured, semi structured and unstructured data have been considered in applications based on mediator. From Table 5, only TSIMMIS, GARLIC and Integration Heterogeneous Multimedia Sources are considered for semi structured data and other applications does not consider this type of data. Meanwhile, only GARLIC cater the unstructured data type but not for other applications.

All applications in Table 5 use XML for data exchange and wrapper as a mediator to receive query for searching and retrieving process and send a result to users.

Based on the review of characteristics, more opportunities for research especially in term of data types can be done for future work. However, the advantages and disadvantages of each mediator-based are also important. The advantages and disadvantages were described in this section.

Table 5. Characteristic of different application

Characteristics/application	2.1	2.2	2.3	2.4	2.5	2.6
Domain area	General	General	Biological data	General	Medical data	Multimedia data
Heterogeneous database	No	No	No	No	Yes	Yes
Structured data	Yes	Yes	Yes	Yes	Yes	Yes
Semi-Structured data	Yes	Yes	No	No	No	Yes
Unstructured data	No	Yes	No	No	No	No
Object Exchange Model (OEM)	Yes	No	No	No	No	No
XML	Yes	Yes	Yes	Yes	Yes	Yes
Wrapper	Yes	Yes	Yes	Yes	Yes	Yes

Table 6. Advantages and disadvantages

Application	Advantages	Disadvantages
TSIMMIS	Used OEM to model structured and semi structured data	Not support for unstructured data
GARLIC System	Provide a view of heterogeneous data sources without changing on how or where data is stored Provide collections of data in object oriented view Can model structured data automatically	Modelling process for semi-structured and unstructured should be done manually
Integrated access to biological database	Data storage is not required for storing the data Provide GUI interface	Take long time for query translation Delay in the process of parsing and filtering
Integrating heterogeneous data sources using XML mediator	Domains are modelled in hierarchical structure Allow to facilitate and optimize Mediate a user query and generate the query execution plan Easy to explore the integration global schema in order to determine a list of sources can queried by the mediator	Cannot answer the query from the user because of correspondence between a data source and global schema Semantic conflict (e.g., two databases containing semantically related information)
Grid information System	Access more than one data source over global Ability to link data source even it has different data structures Able to construct the distributed queries when the target data sources are located at different sites Support heterogeneous and able to federate queries when some data sources are accessed through different query language	Loss of data access performance schema with subset of Structured Query Language (SQL)
Integrating heterogeneous multimedia sources	Support the storage and content based retrieval for any multimedia documents Use XML for search and indexing multimedia data Used MEDRANK too easy to rank the result Can be expressed the similarity predicates in a global query without requiring multimedia processing capability at mediator level Support multimedia document (e.g., Image)	Not support multi queries Not support similarity joints

Conclusion

Based on the analysis of characteristics and advantages and disadvantages describd in Table 5 and 6, the following issues can be considered for future work:

- Design a mediator approach for unstructured data in database integration
- Develop an algorithm to support multi and similarity query in multimedia sources
- Design an algorithm for query translation and process of parsing and filtering the data from data sources

In conclusion, this study discussed current implementations of mediator approach in different domain. The characteristics for each implementation were identified and summarized. The advantages and disadvantages are also highlighted for potential improvement in future research especially in database integration.

Acknowledgment

The researchers would like to thank Universiti Sultan Zainal Abidin for providing facilities and services to do this research.

Author's Contributions

All authors equally contributed in this study.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Aberer, K., 2003. Conception of information systems part 2: Integration of heterogeneous databases. EPFL-SSC, Laboratore of Information System.

Benedikt, M. and P. Senellart, 2011. Databases. In: Computer Science: The Hardware, Software and Heart of It, Blum, E.K. and Aho, A.V. (Eds.)., Springer Science and Business Media, New York, ISBN-10: 1461411688, pp: 169-229.

Beneventano, D., C. Gennaro, S. Bergamaschi and F. Rabitti, 2011. A mediator-based approach for integrating heterogeneous multimedia sources. Multimedia Tools Applic., 62: 427-450. DOI: 10.1007/s11042-011-0904-0

Bichutskiy, V., 2013. Heterogeneous biomedical database integration using a hybrid strategy. UCI Undergraduate Res. J.

Elmasri, R. and S. Navathe, 2010. Fundamentals of Database Systems with Oracle 10g Programming: A Primer. 6th Edn., Prentice Hall, Addison-Wesley, ISBN-10: 0132165902, pp: 1172.

Eltabakh, M., 2012. Data integration. CS561-Spring 2012.

Geppert, A. and K.R. Dittrich, 2002. Component database system: Introduction, foundations and overview.

Ishak, I. and N. Salim, 2006. Database integration approaches for heterogeneous biological data sources: An overview. Proceedings of the Postgraduate Annual Research Seminar, May 24-25, UTM Skudai, pp: 202-206.

Liao, W.K. and D. Mcleod, 2001. Introduction to Databases. In: Computing the Brain: A Guide to Neuroinformatics2.

Lim, E.P. and R.H.L. Chiang, 2000. The integration of relationship instances from heterogeneous databases. Dec. Support Syst., 29: 153-167. DOI: 10.1016/S0167-9236(00)00070-1

Peng, T.C., W. Husain, R. Abdullah, R.A. Salam and N. Aini *et al.*, 2004. Mediator-based architecture for integrated access to biological databases. Proceeding of the 5th International Conference Parallel Dsitributed Computing Applications Technologies, Dec, 8-10 Singapore, pp: 13-16. DOI: 10.1007/978-3-540-30501-9_4

Risch, T., V. Josifovski and T. Katchaounov, 2000. Functional data integration in a distributed mediator system.

Robinson, A. and W. Rahayu, 2004. Genome database integration, Comput. Sci. Appli., 3045: 443-453. DOI: 10.1007/978-3-540-24767-8_46

Rochlani, Y.R., 2012. Integrating heterogeneous data sources using XML mediator. Int. J. Comput. Sci. Netw., 1: 1-9.

Shi, G., 2002. Data integration using agent based mediator-wrapper architecture. The University of Calgary.

Thiran, P., 2004. Wrapping and Iintegrating.

Wöhrer, A., Brezany, P. and A. Min Tjoa, 2005. Novel mediator architectures for Grid information systems. Future Generat. Comput. Syst., 21: 107-114. DOI: 10.1016/j.future.2004.09.018

Yusof, M.K. and S. Safei, 2013. Database integration via mediator approach for integrated applications with NFC technology. Proceedings of the 2nd International Conference on Advances in Computer and Information Technology, (CIT' 13), pp: 978-981. DOI: 10.3850/ 978-981-07-6261-2_1 2