XML based Mediated Query Re-writing Framework

Jahangir khan, Muhammad Ahmed, Muhammad Khalid khan

Graduate Institute of Science &
Engineering, PAF-KIET,
PAF Base Korangi Creek
Karachi75190, Pakistan

ABSTRACT

To integrate the information from heterogeneous data sources and give it a unified representation to the users is known as Information Integration. There are many application architectures that are designed for Enterprise Information Integration for solving the problems of semantic heterogeneity (the modeling problem) and query optimization (the querying problem) in Integration Architecture. Architectures such as Mediator-based in which information is coming from disseminate sources, Agent-based Architectures that have various software agents specialized in specific tasks work together to provide various integration services. Federated Architectures in which data is integrated through message-oriented middle wares. Enterprise Information Integration depends on sophisticated technologies and complex architectures. However, Query Optimization/Management is the major area of research in XML based integration systems. Since XQuery precludes the features of traditional SQL or OQL as it deals with the structured and semi-structured data sources. Focus is to present a solution to the problem of query optimization in XML-based data integration in hybrid peer to peer data management environment. The contributions to this paper are: providing a conceptual frame work for Information Integration System based on XML query language, formulation of rewriting algorithm for XML query and implementation of the proposed algorithm.

General Terms

Data Integration Services, distributed database management system, enterprise and Information Services.

Keywords

Data Integration, Query Optimization, XML based Mediated Query Re-writing (XMQR).

1. INTRODUCTION

For the last ten to fifteen years the information systems have became increasingly decentralized and distributed. The challenges of information integration have increased as the volume of the data that most organizations are managing, the variety of data formats and users' requirements for accessing complex information have all increased. Likewise, information integration technology has advanced to meet the more challenging data consistency management requirements created by these demands for accessing the information. Keeping in view of past solutions and to overcome the deficiency of latency in data for real time solutions the newer Enterprise Information Integration (EII) come along and solved this problem by synchronizing changes across systems in real time. Enterprise Information Integration comes under the domain of Data Engineering. Enterprise Information Integration (EII) provides access to information regardless of information sources and storage format and presents the information as it is coming from single source. Data is what you run your business on and it is the ability to capture, dispense and utilize it at the right time and in the right ways that will help business to excel. As the organizations are more likely to have decentralized architectures due to rapid changes in e-commerce environment and organizations are more often require sharing information at inter-organizational level to business-to-business data interchange or to form virtual data repository. Information Integration can be used for the following kinds of applications: Creating a single view of business entities, Data integration and management at enterprise level, Real time reporting and analysis, Updating common information across information sources, Integrating unstructured data including documents, audio, video and other electronic media into applications, Providing an infrastructure for enterprise information management, Updating a data warehouse and Creating a virtual data warehouse.

There are two general approaches to this problem, materialized integration and virtual integration. *Materialized integration* is strongly related to materialize views in databases, by storing all data from the participant local sources and then querying them. Data warehousing is a well known example of materialized integration. It is suited for situations when data changes infrequently and a fast evaluation of complex queries are required. However it is not always possible or convenient to replicate and update all the data from a set of sources. There are situations when the size and volatility of data or the limitations imposed by the sources query interfaces makes materialization impossible. This is the reason why virtual integration has become of increasing interest in recent years as it has matured. *Virtual integration* aims to offer the same results without the constraint of having to store and update all data from all the sources. In pure virtual integration the global/ mediated schema is strictly a logical entity. Queries issued over it are dynamically re-written at runtime and re-directed to the underlying data sources. Resulting data is fetched from the sources through wrapper/middleware and merged to give a unified representation.

There are various classes of Data Integration System, the most common and widely used class of data integration is Mediator based Data Integration System. The general components are Mediated schema which may be mapped with the schemas of the local data sources. This schema correspondence is based on the Local-as-view (LAV) or Global-as-view (GAV) approaches. Both approaches have different benefits over each other such as GAV doesn't support integrity constraints and has exact views where as LAV keeps integrity constraints and mapping of relations among local schemas and global schema. Other components include *Query Processing* or *Query management* and *Result Integrator*. Components may be added or modified for specialized tasks such as keeping the knowledge base, virtual views, materialized views, access control mechanism etc. Other classes are Agent-based Data Integration System, Ontology-driven Data Integration System, Peer-to-Peer Data management and XML-based Data Integration Systems. The components of all classes of data integration systems are somewhat similar to each other. There are several problems related to data integration, but the main one are; the ability to present an global/ mediated schema for the user to query, or *the modeling problem,* ability to reformulate the query to combine information from the different sources according to their relationships with the global/ mediated schema, or *the querying problem*, and the ability of efficiently execute the query over the various local and remote data sources.

Xml based information integration does not support features like order insensitivity and fixed schema that results in formulation of inefficient global query plan in hybrid peer to peer data management environment, so Proposed 'XML based Mediated Query Re-writing (XMQR)' is mechanism, which formulates efficient global query plan for the hybrid peer to peer data management environment. The paper proceeds as, section 1 describes the overview, section 2 describe issues, section 3 shows optimization includes algorithm and implementation and the last one concludes the paper.

2. OVERVIEW

Today most of the organizations are having decentralized business systems and having considerable autonomy within their structures due to their increasing volume of data and dynamic information systems' architectures. Therefore they are facing many business and technology challenges in order to share their resources whether within the same organization or inter-organizational level. The need of sharing information may be due to the mergers of organizations of same business to form an enterprise, business to business (B2B) data interchange for achieving competitive advantage, or dynamism of e-business environment. Due to this decentralized nature of the organizational resources, there is need of not only more flexible and adaptable but also cohesive and value creating role of information systems infrastructures and their management as designed in figure 1.

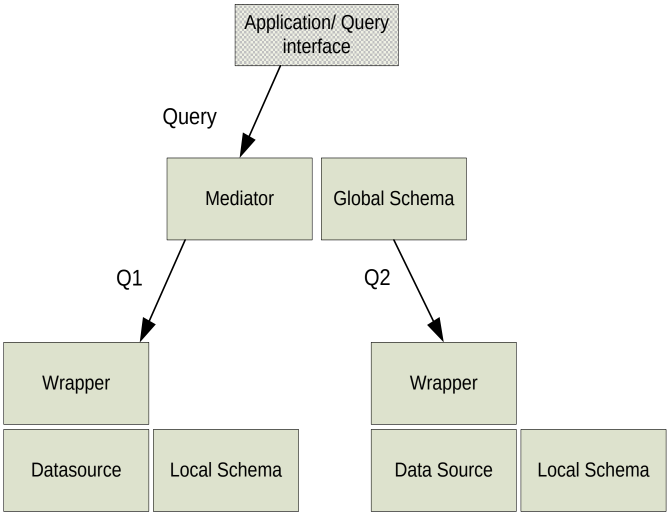


Fig1: Data Integration Architecture

Enterprise Information Integration gives benefits like transformation of data into useful knowledge, data analysis capabilities, and enterprise data management. There are many processes through which the data is processed to take the form of information such as aggregation, consolidation, cleansing, filtering, transformation, and validation. For successful implementation of information integration framework the organization must addresses following capabilities [24, 25].

Access: The frame work should have capability to provide access virtually to all kinds of data sources, including relational databases, flat file, mainframe legacy, XML web data and even packaged data from Enterprise Resource Planning (ERP).

Integrate: The framework should have the flexibility for deployment across all the architectures including client/server, Web/application server (XML, ADO/ASP, etc.) and distributed computing, supporting all major models of Microsoft COM, EJB, and CORBA. To integrate disparate data into information, the framework should have key data integration features including global business metadata catalog, ontology, or global schema.

Manage: Framework should be predictable for maintenance purpose. It should include for centralized management technologies including named services through LDAP or Microsoft Active Directory Services.

Secure: The framework must address security issues that an enterprise faces today or can expect to face in the future. The framework can enable centralized control of data access resource utilization and security. It should support multilevel authentication including database, application, and host and system level.

Scalability: The framework must support the performance and scalability for online systems, and should consider the new demands of e-business.

Now let us discuss some benefits of the information integration framework in terms of business and technology aspects.

The business benefits include:

- Ability to capitalize on new markets and revenue opportunities with flexible business architecture for rapid response to change markets.
- Accelerated transformation to e-business by integrating data, information and technology assets with the cost benefit of utilizing economical web infrastructure.
- Better customer service with better informed staff, partners and employees through improved efficiency of information retrieval and delivery and utilization.

And the technology benefits include:

- Flexible information integration architecture for today and future, spanning all major computing platforms, architectures and data stores.
- Interoperable through standards of past, present, and future including COM, CORBA, EJB as well as ODBC, JDBC, ADO/OLE DB and XML.
- Hide complexity of data and information for both end-use and development, while maintaining security and control.
- Alignment of Information Systems with the goals of the business.
- Support dynamic integration of new data sources.

3. ISSUES

One of the key issues faced in the data integration projects is locating and understanding the data, which is to be integrated. Often one would find that the data needed for a particular integration application is not even captured in any source in the enterprise. In other cases, significant effort is needed in order to understand the semantic relationships between sources and convey those to the system. Tools addressing these issues are relatively in their infancy. They require both a framework for storing the meta-data across an enterprise, and tools that make it easy to bridge the semantic heterogeneity between sources and maintain it over time. Efficiently integrating new information sources from outside the enterprise is often critical to success in a world of global competition, interdependency, and rapid market change. For this purpose the commercial applications are beginning to require rapid access to multiple data sources, and ability to rapidly fuse data from disparate formats. The next generation integration technology must handle this scalability issue and must have fast transformation engines. To connect with data source the application requires an adapter or connecting component that enables application to interact with the data source. When suitable adaptor for data source is not available then one must create, this activity is very costly and causes delays in providing information form specific data source. This stated problem could be solved by design such a mechanism that automatically locate the newly added and having universal adaptor that will connect the application with the data source and through this data is retrieved. There is another critical issue of efficient query processing and optimization [17].

The Query Reformulation and Management phase has several contemporary issues that impede the progress in this technology domain and are still in research phase. The query optimization issue is common in all classes of Data Integration Systems. Optimization is required for generating efficient query plans and fast execution of the sub-queries. Optimization issue addresses the problem of delays such as slow delivery, initial delays or bursty data. The solution of delays issue provides by different researchers as Query Scrambling which is the re-optimization or runtime optimization of the query plans. Query Scrambling has few limitations such as materialization, memory usage or resource overhead. Query plan may be defined as the sequence of data sources which are to be visited for retrieval of answers. The sub-issues in generating query plans is to find the cost effective query plan among the number of query plans generated during the phase of query process. There is another common issue for Query Execution Engine, which is to have knowledge of the data sources prior to execute query plans or sub-queries. This knowledge includes the description of source schema, access patterns to the data sources and relations among the local and global entities. The most intelligent data integration systems are based on agent and mobile agent based distributed computing. These integration systems possess strong features of intelligently searching the relevant information from the information sources and may perform transformation from information management to knowledge management. Agent based system contains agents that are specialized to perform different tasks such as negotiation process between ontology and local data sources, query execution or updating the knowledge base of the system. As it is relatively a new field so there is several issues in Mobile Query Environment such as integrity and confidentiality of the data which is being carried by the agents in the form of user queries or results, itinerary optimization of query mobile agents, Fault tolerance (system failures, agent crashes etc).

The primarily focus of this research is on the multiple XML queries optimization in XML-based data integration system based on hybrid peer-to-peer data management environment. Research on optimization of queries in XML Query language such as XQuery is in its infancy. This is due to the fact that XQuery precludes the features of traditional SQL/OQL. SQL is basically designed for highly tuned and well defined relational data and schema of these relational data is also fixed and well elaborated, whereas XQuery is designed for dealing with time varying schema and order sensitive data. XPath defines the path expressions for the nodes in XML data and XQuery is based on these path expressions. Therefore, generating the efficient query plans is a complex task in XQuery as we have to deal with order sensitivity and redefined the order semantics of path expressions. In addition to this, the nested structure of XQuery further aggravates the situation. In this dissertation the focus is on answering two questions 1) How an efficient processing of multiple queries in scalable integration environment is to be done; 2) What are the attributes for query management in the context of Information Integration based on hybrid Peer-to-Peer Data Model.

4. OPTIMIZATION

Primarily focus of this research is to present a solution to the problem of query optimization in XML-based data integration in hybrid peer to peer data management environment. The foundation lies on the Peer-to-Peer (P2P) data management model. The contributions to this research are: (i) providing a conceptual frame work for Information Integration System based on XML query language (ii) formulation of rewriting algorithm for XML query (iii) implementation of the proposed algorithm [17, 22, 23, 27, 28].

4.1 Query Management

The basic user query scenario we have in mind concerns a user at a workstation, looking for information on a topic within the context of broader-scoped task. The user may then issue a request expressed in some fixed language to the network. The user interface translates the user request into a query expressed in formal language (user query language), and sends a probe out for looking the answers.

4.1.1 An Overview of Peer-to-Peer Systems

A Peer-to-Peer (P2P) system primarily relies on the network bandwidth and computing power of individual computers that participate in the network. Each computer in network refers to as peer. Each peer can act as a client as well as server to provide and share information in scalable distributed environment. Contrary to this, client/server environment network computers rely on few numbers of servers. Early design of the internet is based on P2P systems, which is used for file sharing and data exchange among few corporate participants. As the number of users increases and some security issues greatly influence to develop new internet/network protocols and have completely replaced the P2P models. But today most of vendors specialized themselves for providing P2P environment in specialized business domain. Napster, that made the P2P system idea popular, avoids some this complexity by employing a centralized database with references to the information files on the peers. Gnutella, another well-known P2P system, it has no central database, and is based on a communication-intensive search mechanism. Gnutella draws on research in distributed and cooperative information systems to provide decentralized and scalable data access structures. P-Grid is a virtual binary tree that distributes replication over community of peers and supports efficient search. In particular, search time and number of generated messages grow as with number of data items in the network. Peers in P-grid perform construction and search/update operations without any central control or global knowledge. P-Grid's search structure exhibits the following properties: It is completely decentralized, all peers serve as entry points for search, Interactions are strictly local and it uses randomized algorithms for access and search.

At first glance, many of the challenges in designing P2P systems seem to fall clearly under banner of the distributed systems community. However, upon closer examination, the fundamental problem in most P2P systems is the placement and retrieval of data. Indeed, current P2P systems focus strictly on handling semantic-free, large-granularity requests for objects by identifier (typically a name), which both limits their utility and restricts the techniques that might be employed to distribute the data. These limitations arise because the P2P systems lack focus on the areas of semantics, data transformation, and data relationships. Yet, these are some of the core strengths of data management, where queries, views, and integrity constraints can be used to express relationships between existing objects [24, 25].

4.1.2 Mediated Schema

A mediated schema, when a data integration system employs a logical schema in order for several autonomous sources to interoperate. This kind of schema usually accompanied by the definition of semantic and structural mappings between the mediated schema(s) and the schemas of the underlying data sources. This correspondence will determine how the queries to the system are answered. There are three basic approaches for specifying the mappings in a data integration system [37], which are generally incorporated, namely local-as-view (LAV), global-as-view (GAV), and GLAV, brief descriptions of these approaches is given in subsequent sections.

Local-as-View (LAV)

In the LAV approach, the mapping associates to each element of the source schema 's' a query 'Q_g' over 'G'. In other word, an information source is described as view expression over the mediated schema. $s \rightarrow Q_g$

From the modeling point of view, the LAV approach is based on the idea that the content of each source 's' should be characterized in terms of view 'Q_g' over the mediated schema. LAV approach favors the extensibility of the system; adding a new source simply means enriching the mapping the new assertion, without other changes.

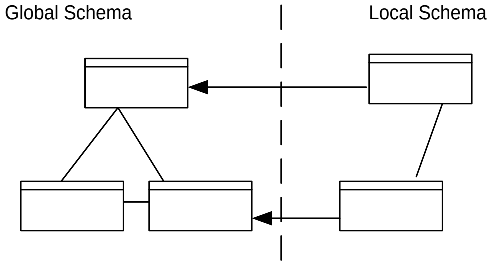


Fig 2: Local-as-view (LAV) approach

Global-as-View (GAV)

In the GAV approach, the mapping associates to each element in mediated schema a query 'Q_s' over 's'.

$G \rightarrow Q_s$

From the modeling point of view, the GAV approach is based on the idea that the content of each element of the mediated schema should be characterized in terms of view 'Q_s' over the sources. This approach is effective whenever the data integration system is based on a set of sources that is stable. The GAV approach favors the system in carrying out query processing, because it tells the system how to use the sources to retrieve data. However, extending the system with a new source may have an impact on the definition of various elements of the mediated schema, whose associated views need also to be redefined.

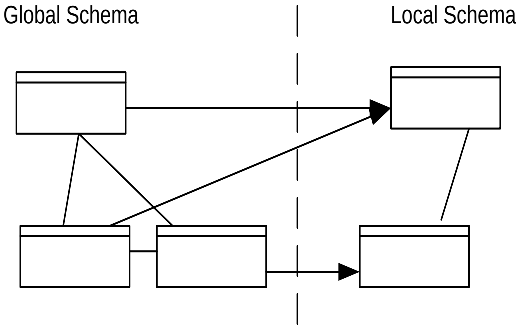


Fig3: Global-as-View (GAV) approach

GLAV (Combining LAV and GAV)

A third kind of mapping, combining the advantages of both LAV and GAV, is called GLAV, In GLAV the mapping between G and S is constituted by a set of assertion of the form.

$Q_s \rightarrow Q_g$

Where 'Q_s' and 'Q_g,' are the two queries, which have equal semantic meanings, respectively over the source schema 's', and over the mediated schema G. In the proposed integration framework, the characteristics of GLAV are explored for efficient query processing.

4.2 Integration Framework And Formulation of Mediated Schema(s)

In integration framework, there is central mediated XML schema(s). The sources are mapped transitively with a mediated schema(s). However, this framework is based on hybrid P2P data management model and thus has an attribute of scalability and dynamism. To deal with the scalability factor this data integration framework is also capable of performing path-to-path mappings. The formulation of mediated schema(s) and mapping rules established for this purpose are explained in subsequent sections [4, 18].

XML Sources

To illustrate the formal data integration framework, an example is considered representing the XML-based data sources. Formal definitions and details are deferred.

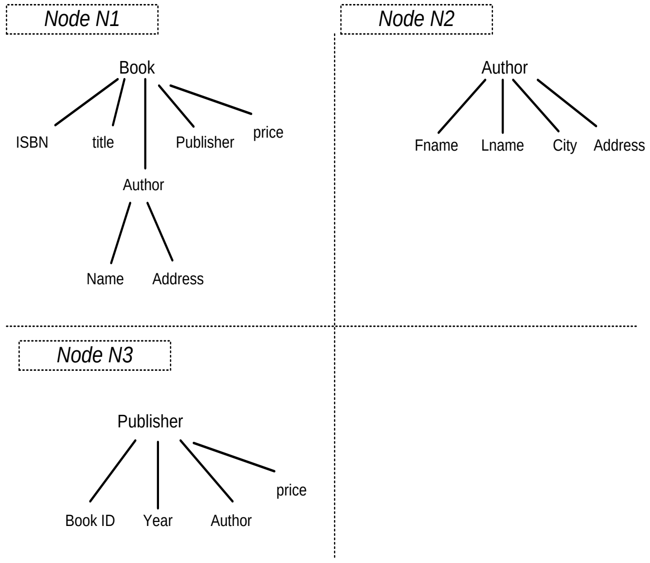


Fig4: Example XML Data Sources

It is common in data integration, a single source or peer may not be providing complete data on a subject. The sources given in figure 4. are different not only in terms of structure, but also in terms of contents they contain. As in source *N1* author is defined under a structure *Book*, whereas in source *N3* 'author' is defined under a structure *Publisher*. Similarly semantics differences can be depicted in *N1* and *N2*, in tree 'Author' contains name of the author as 'Fname' and 'Lname'. On contrary to this, in *N1*, author name is given by node 'Name'. Similar differences can also be seen in *N1* and *N3* as book is identified by ISBN and Book ID respectively. A XML schema representation of the sources defined in figure 5. can be depicted in as.

```
<databaseSchema dbname= "N1">
  <table name= "Book">
        <column name= "ISBN" />
        <column name= "title" />
        <column name= "publisher" />
        <column name= "price" />
        <primaryKey>
              <columnName>ISBN</columnName>
        </primaryKey>
  </table>
  <table name= "Author">
        <column name= "ISBN" />
        <column name= "name" />
        <column name= "address" />
        <column name= "price"
  </table>
</databaseSchema>

<databaseSchema dbname= "N2">
  <table name= "Author">
        <column name= "au_id" />
        <column name= "Fname" />
        <column name= "Lname" />
        <column name= "address" />
        <column name= "city" />
        <primaryKey>
              <columnName>au_id</columnName>
        </primaryKey>
 </table>
</databaseSchema>

<databaseSchema dbname= "N3">
  <table name= "Publisher">
        <column name= "pid" />
        <column name= "city" />
        <column name= "book_id" />
        <column name= "year" />
        <column name= "price" />
        <primaryKey>
              <columnName>pid</columnName>
        </primaryKey>
 </table>
</databaseSchema>
```

Fig 5: XML Schema for the sources

In proposed data integration framework, there is a unique interface for the users for querying the data and this interface is described as *Mediated Schema*. It is the unified view of the data, independent of the actual format and locations of the local data sources. In this framework, only XML based data sources are considered. To integrate a source in the system, only need to provide a set of mapping rules that describe the relationships between mediated schema(s) and local data sources' schemas. The specification of mappings is thus flexible and scalable. Two major issues are dealt for efficient query processing and data management: Structural heterogeneity and Semantic heterogeneity. *Structural heterogeneity* concerns the different representation of the data in XML document. Structural heterogeneities are addressed among XML data sources by associating paths in different schemas. Mappings are specified as path expressions that relate a specific element or attribute together with its path in the source schema to relate elements or attributes in the mediated schema(s). *Semantic heterogeneity* concerns the intended meaning of the described data. The mediated schema for the sources shown in figure 6. can be depicted as.

```
<databaseSchema dbname= “GS”  name=”Global Schema”>


<maps>
            <map dest= “GS:/GlobalSchema” />
            <map dest= “GS:/GlobalSchema/Books” />

            <map dest= “GS:/GlobalSchema/Books/@ISBN”/>
            <map dest= “GS:/GlobalSchema/Books/@ISBN”  source= “N1:/Book/ @ISBN” / >
            <map dest= “GS:/GlobalSchema/Books/@ISBN”  source= “N3:/Publisher/ book_id” / >

            <map dest= “GS:/GlobalSchema/Books/title”  source= “N1:/Book/title” / >
            <map dest= “GS:/GlobalSchema/Books/title”  source= “N3:/Publisher/bookname” / >

            <map dest= “GS:/GlobalSchema/Books/Authors/Person”  source= “N1:/Book/Author” / >
            <map dest= “GS:/GlobalSchema/Books/Authors/Person”
                        source=concatenate ( “N2:/Author/Fname, N2:/Author/Lname)” / >
</maps>


</databaseSchema >
```

Fig6: Mediated Schema

The source schemas show the structure of the local sources, where the actual data resides. Whereas, the mediated schema describes the unified and virtual representation of the underlying sources. This unified and virtual representation is formulated by asserting mapping rules on the elements and their relationships of local sources to the mediated or destination schema. When user poses a query, it is in fact, posed on the mediated schema. For processing of the query, it is decomposed into one or more queries on the bases of the mapping rules established to formulate the mediated schema. Since the system is based on Peer-to-Peer data model and as it is said earlier the formal P2P model is hybrid in nature, residing one super peer but not forgetting the fact that in P2P computing model, user can login from any node in overlay network. Therefore, it is imperative to say a query may be posed from any node in network over the mediated schema(s). So for this reason each peer has information of mediated schema(s). Considering the scalability and dynamism of P2P computing model, every element in schema S; where S={S1,S2,…,Si}, is associated by mapping rules M to Q (Query) over mediated schema.

4.3 Mapping Procedure between Mediated Schema and the Local Sources

In this integration framework, a source is integrated, by providing a set of mapping rules that describe the relationships between the source schemas and the mediated schema(s). Association of paths in the mediated schema(s) with paths in source schemas allows both to associate concepts with XML nodes in the data sources, and to associate relationships among concepts with XPath location paths in the XML sources. Paths in a source are described in terms of XPath location paths. An XPath location path is composed of sequence of location steps. Location steps have three parts: (i) an axis specify the relationship (child, descendant, ancestor, attribute etc.) between the nodes selected by the location step and the context node. (ii) a node test specifies a node’s XML type (element, attribute, so on) and possibly its name. (iii) Optional predicates which use XPath expressions to further refine the set of selected nodes. It is assumed that the sources are heterogeneous and autonomous; they do not provide the persistent object identifiers that are valid for all sources. The ID/IDREF XML attribute mechanisms are used for internal references, but cannot serve for a key mechanism to perform joins between objects that originate from different sources, i.e. sources might specify meaningful keys in terms of XML elements/ attributes, but it cannot be expected that different autonomous sources always use the same keys. For example a Book might be identified by its title in one source, and by its title and ISBN in another source. The mapping rules for the example mediated schema correspond to local sources spread over the different peers are illustrated hereunder:

1. N1/Book/ISBN → N3/Publisher/book_id, P4/Store/Product_id
2. N1/ Book/Author → N2/Author/Fname, N2/Author/Lname
3. N1/Book/title → N3/Publisher/bookname, P4/Store/Productname
4. N1/Book/Author/Address → N2/Author/Address, N2/Author/City

4.4 Query Processing

The resolution of query in a data integration system can be divided in two stages; *query reformulation* and *query processing*. Query reformulation corresponds to answer the queries using virtual views, and focus on the selection of the sources that can provide the best valid response to a given query. However, merely knowing which sources to query is not enough. The obtained re-written query of the first stage is declarative query which refers to the sources modeled as views. In a local system such a high level query would be translated to a syntactic tree and then optimized for execution. On contrary to this, in a data integration system some of the algebraic operators can be performed locally at the sources, while others must be performed in the mediator. The query processing stage aims to generate the best execution plan for a given query and executing that plan with the help of the mediator and the wrappers/middleware of the sources. As the target systems are distributed, autonomous and heterogeneous (hybrid peer-to-peer data model), achieving a good performance can be a difficult task [8-16].

4.4.1 Query Optimization Framework

The query processing approach uses the correlated schema, which is formulated by asserting the mapping rules. Sources are connected to the mediated schema(s) by semantic mappings. The query poses over the mediated schema(s), will be decompose into one or more sub-queries and these decomposed queries are the union of the query posed over the mediated schema(s). The decomposed queries fetch complete or partial results, and then these results are integrated to form a unified representation and presented to the user. The model of the query optimization/execution can be depicted in figure 7. User poses a single query over a mediated schema, the virtual representation of the schemas of local data sources. This query is processed by Query Execution module. At first, it decomposes query into sub-queries by semantically relating the elements/ path expressions of the query with those found in local schemas by implying semantic mapping rules. After reformulation of the query into sub-queries, these queries are physically executed in Query Execution Engine. Each sub-query executed over the source, which is semantically related to the path expression in a given query. After execution the sub-queries, the partial results of the sub-queries are retrieved from their respective data sources.

Redundancies in partial results are removed and then these partials results are integrated and presented to the user. The technique for reformulation of the query into sub-queries is presented in the subsequent section.

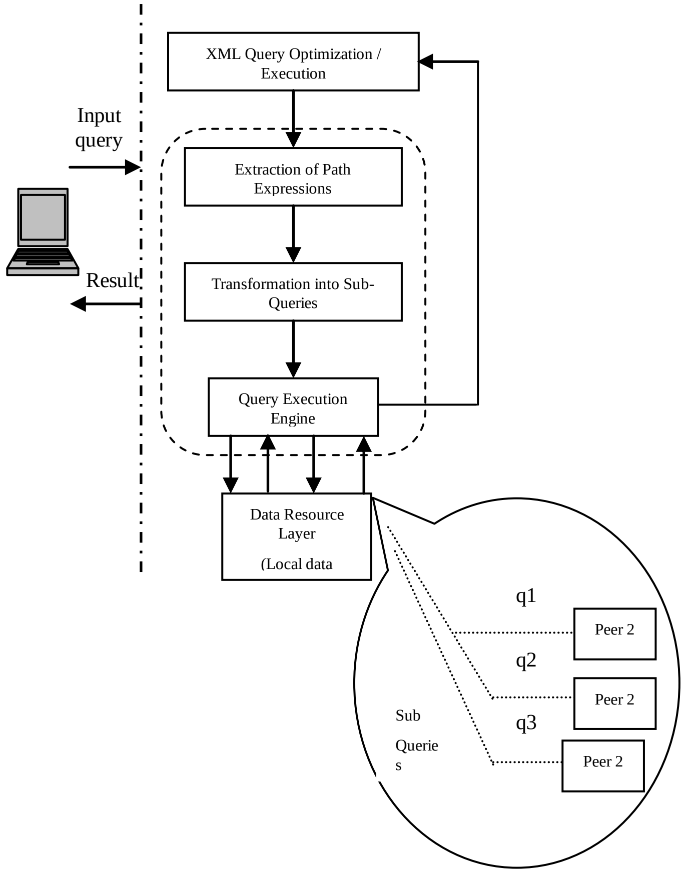


Fig7: Query Processing Framework

4.4.2 XMQR Algorithm

Steps of the algorithm for query re-formulation are given in this section. The assumptions which are used in the algorithm; an XML query '*Q*' as input over mediated schema(s) '*S*' with a set of mapping rules 'M_R'. Upon satisfying the mapping rules, one or more than one sub-queries 'Q_{Ri}' are formulated. The steps are given hereunder:

STEP 1: Determining the Path Expressions:

There may be one or more predicates in a query *Q*, which may lead to different paths or branching points in a tree pattern representing the XML source '*S*'. The paths identified through predicates in *Q* are inserted in to set 'P'.

STEP 2: Nomination and Pruning of 'Schemas':

a) By using mapping information provided with mediated schema(s) of '*S*', determine the path expressions corresponding to every P_i in P. These path expressions are semantically connected to the source schemas S_i. Identified paths P_i in Q, which connects expressions to the S_i of S, are termed as 'Candidate Paths' and schemas to which they are connected are termed as 'Candidate Schemas'.

b) Algorithm checks whether the candidate schema has atleast one candidate path for each path present in Q, further to this, it also ensures that each candidate path is used no more than once to reformulate the query Q in order to ovoid redundant paths. The source schemas that meet these conditions are the only ones that will be considered to obtain reformulated queries.

STEP 3: Composing Reformulated Queries:

In this step, one or more XPath queries over each candidate schema are initiated; following conditions must be conformed;

The number of reformulated queries depends on the possible path combinations. If each path P_i has a single correspondent path over the schema S_j, thus the output of the reformulation will be single query expressed over the candidate schema S_j. On the contrary, if there will be one P_i in P has more than one destination path over schema S_j, there will be more than one reformulated query over the schema S_j. The number of possible path combinations for S_j, is equal to the product of cardinality of each path $P_{i,j}$.

Case 1: If P_i belongs to 'P' provided $P_{i,j} > 1$ over S_j *then*

Return $Q_{R\,i,j} > 1$

Case 2: If P_i belongs to 'P' provided $P_{i,j} = 1$ over S_j *then*

Return $Q_{R\,i,j} = 1$

After determining the cardinality of the mapping and before initiating the sub-queries, the join conditions between the paths are validated. Number of reformulated queries depends on satisfied join conditions among the paths of each combination.

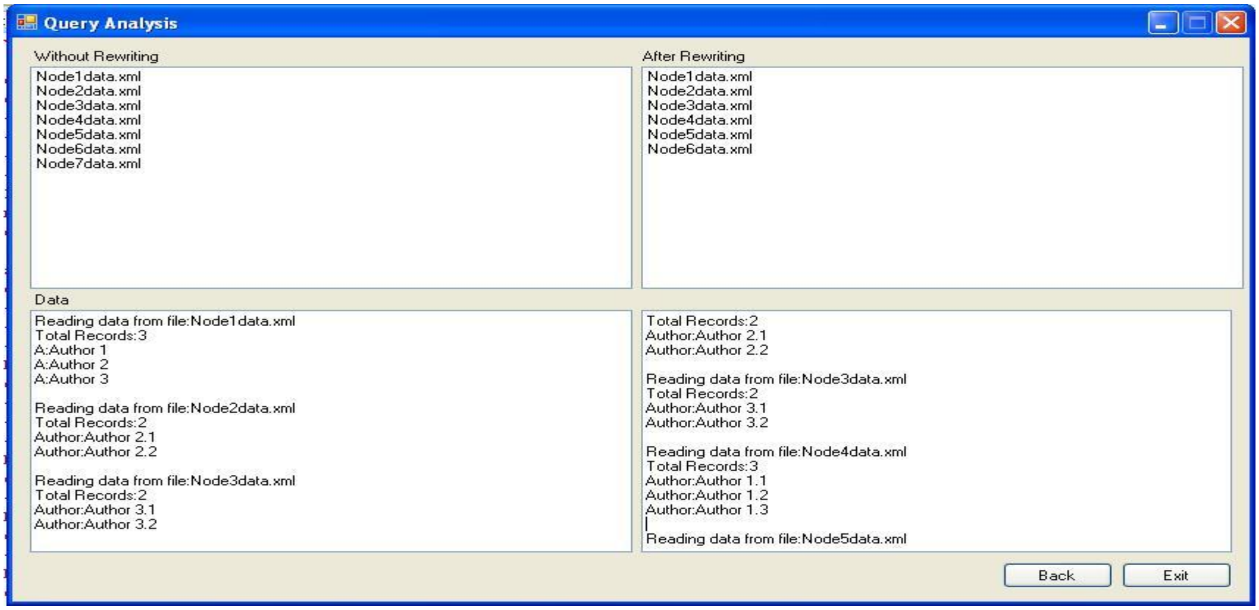
After verifying the join conditions between the destination paths, the actual generation of one or more sub-queries is initiated. These queries are the product of the query '*Q*' over the destination schema 'S_j'.

4.5 Implementation

Query re-writing algorithm is implemented in limited scope to prove the concept, and at this stage only XPath expressions are evaluated for rewriting. Prototype is only capable of rewriting the simpler XPath expressions, and it proves the efficiency of the rewriting algorithm

Four nodes are considered, which are mapped into the mediated schema. a screen is shown, user can select the location of the mediated schema(s) by browsing, and then Adding to the application. As it can be seen that query has same result in both Without Rewriting and After Rewriting List box, this is due to the fact that the element Author is found in the schemas of all four nodes that are selected. Therefore query is r-written according to the path expressions mapped in all four nodes. Whereas in figure 8. the answer of the query Book/Title is shown, in After Rewriting list box only those nodes are selected for the query execution where the element Title is found. Algorithm only reformulates the sub-queries over the local schemas, where the path expression mapped correspond to the element Title.

Similarly as depicted in figure 9. And figure 10. answer of the query Book/Publisher is shown, in After Rewriting list box only those nodes are selected for the query execution where the element Publisher is found.



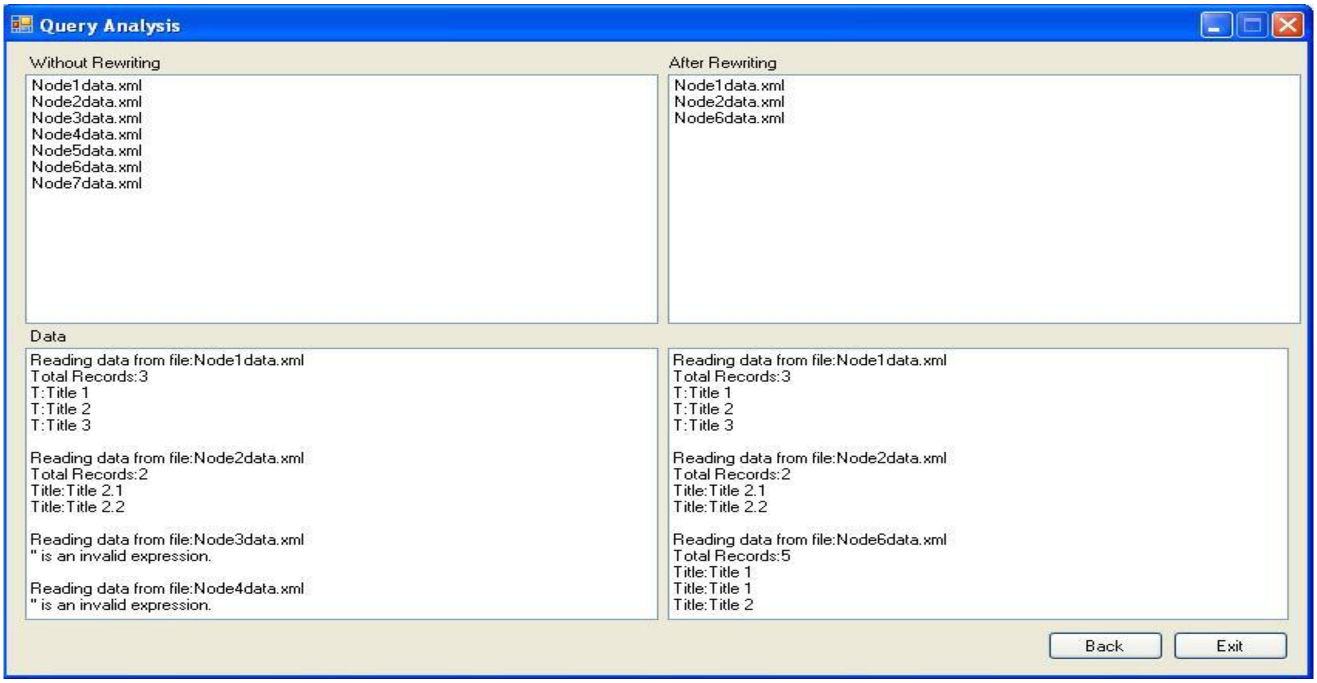
```
Node1data.xml
Node2data.xml
Node3data.xml
Node4data.xml
Node5data.xml
Node6data.xml
Node7data.xml
```

```
Node1data.xml
Node2data.xml
Node3data.xml
Node4data.xml
Node5data.xml
Node6data.xml
```

```
Reading data from file:Node1data.xml
Total Records:3
A:Author 1
A:Author 2
A:Author 3

Reading data from file:Node2data.xml
Total Records:2
Author:Author 2.1
Author:Author 2.2

Reading data from file:Node3data.xml
Total Records:2
Author:Author 3.1
Author:Author 3.2
```

```
Total Records:2
Author:Author 2.1
Author:Author 2.2

Reading data from file:Node3data.xml
Total Records:2
Author:Author 3.1
Author:Author 3.2

Reading data from file:Node4data.xml
Total Records:3
Author:Author 1.1
Author:Author 1.2
Author:Author 1.3

Reading data from file:Node5data.xml
```

Fig8: Result of first query



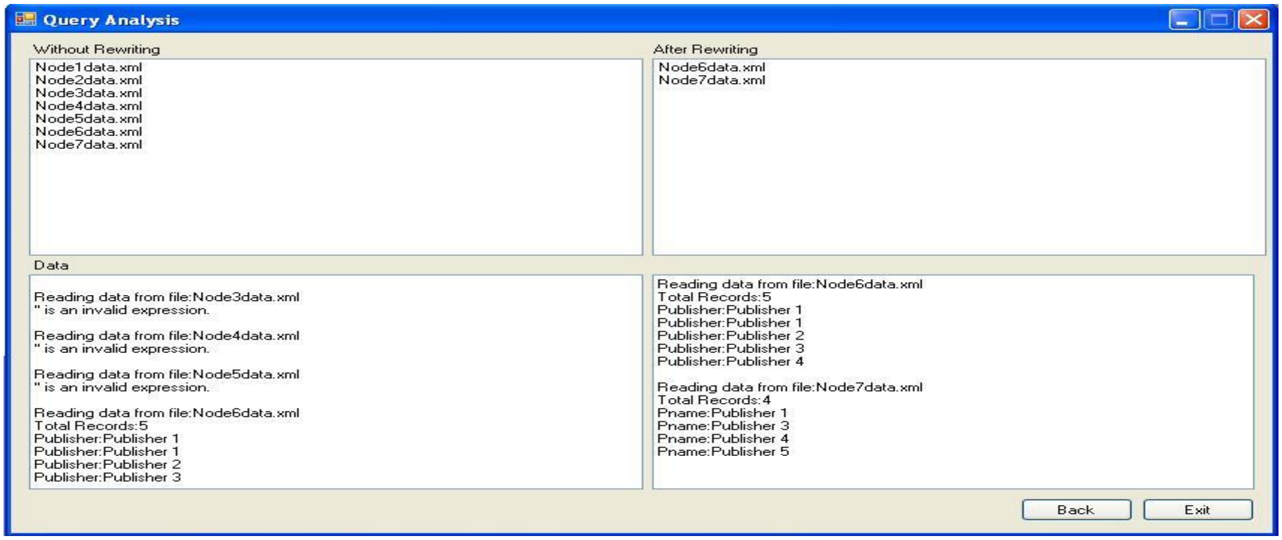
```
Node1data.xml
Node2data.xml
Node3data.xml
Node4data.xml
Node5data.xml
Node6data.xml
Node7data.xml
```

```
Node1data.xml
Node2data.xml
Node6data.xml
```

```
Reading data from file:Node1data.xml
Total Records:3
T:Title 1
T:Title 2
T:Title 3

Reading data from file:Node2data.xml
Total Records:2
Title:Title 2.1
Title:Title 2.2

Reading data from file:Node3data.xml
" is an invalid expression.

Reading data from file:Node4data.xml
" is an invalid expression.
```

```
Reading data from file:Node1data.xml
Total Records:3
T:Title 1
T:Title 2
T:Title 3

Reading data from file:Node2data.xml
Total Records:2
Title:Title 2.1
Title:Title 2.2

Reading data from file:Node6data.xml
Total Records:5
Title:Title 1
Title:Title 1
Title:Title 2
```

Fig9: Result of second query



```
Node1data.xml
Node2data.xml
Node3data.xml
Node4data.xml
Node5data.xml
Node6data.xml
Node7data.xml
```

```
Node6data.xml
Node7data.xml
```

```
Reading data from file:Node3data.xml
" is an invalid expression.

Reading data from file:Node4data.xml
" is an invalid expression.

Reading data from file:Node5data.xml
" is an invalid expression.

Reading data from file:Node6data.xml
Total Records:5
Publisher:Publisher 1
Publisher:Publisher 1
Publisher:Publisher 2
Publisher:Publisher 3
```

```
Reading data from file:Node6data.xml
Total Records:5
Publisher:Publisher 1
Publisher:Publisher 1
Publisher:Publisher 2
Publisher:Publisher 3
Publisher:Publisher 4

Reading data from file:Node7data.xml
Total Records:4
Pname:Publisher 1
Pname:Publisher 3
Pname:Publisher 4
Pname:Publisher 5
```

Fig10: Result of third query

5. CONCLUSION

The massive amount of data that today is available from distributed and heterogeneous data sources is making data integration as important as data mining and knowledge discovery for exploiting the value of such large and distributed data repositories. Integration and correlation of large data sets demand significant advances in query management. The XMQR algorithm presented in this paper is evaluates only those schemas which are selected as Candidate schemas. In case of large data sets the efficiency of the algorithm greatly improves as instead of flooding the queries over the complete datasets for evaluating initial query plan cost it directly execute the sub-queries after rewriting, only over those schemas that contain the candidate path expressions. Therefore, XMQR greatly reduces the requirement of network bandwidth and delays for answering the multiple queries in hybrid peer-to-peer environment. And also has the query execution engine processes the queries on the basis of transitive and path-to-path mappings, it would not produce redundancies in query plans and results are efficient, Scalable hybrid peer to peer models and architecture for distributed data integration. Schemas of new data sources can easily be incorporated into the mediated schema(s), by only providing a set of mapping rules and XMQR algorithm does not require query plans cost evaluation and sub-queries formulated can be directly executed on the local sources.

ACKNOWLEDGMENTS

Our thanks to the IJCA who have given the opportunity for exploring our ideas and research in the researcher platform.

REFERENCES

[1] Edited by Liam Quin, "The XML Data Model", www.w3.org/XML

[2] Munindar P. Singh, Krithi Ramamritham, "XML Processing and Data integration with XQuery", IEEE Internet Computing published by IEEE, 2007.

[3] Mary Fenandez, Ashok Malhotra, Jonathan Marsh, Marton Nagy, Norman Walsh, "XQuery 1.0 and XPath 2.0 Data Model (XDM)", W3C Recommendation 23rd January 2007, www.w3.org/TR/xpath-datamodel/

[4] Kangchan Lee, Jaehong Min, Kishik Park and Kyuchi Lee, "A Design and Implementation of XML-Based Mediation Framework (XMF) for Integration of Internet Information Resources", Proceeding of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35'02), 2002 IEEE.

[5] Allen Moulton, Staurt E. Madnick and Michel D. Siegel, "Semantic Interoperability in the Fixed Income Securities Industry: A Knowledge Representation Architecture for Dynamic Integration of Web-based Information", Proceeding of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2002 IEEE.

[6] Hasan Davulcu, Zoe Lacroix, Kaushal Parekh, I.V. Ramakarishnan and Nikeeta Julasana, "Exploiting Agent and Database Technologies for Biological Data Collection", Proceeding of the 15th International Workshop on Database and Expert Systems application (DEXA'04), 2004 IEEE.

[7] Isabel F.Cruz, Afsheen Rajendran and Nancy Wiegand, "Handling Semantic Heterogeneities Using Declarative Agreements", Proceeding of Conference on GIS, 2004 ACM.

[8] Caragea, D.; Jie Bao; Pathak, J.; Silvescu, A.; Andorf, C.; Dobbs, D.; Honavar, V., "Information Integration from Semantically Heterogeneous Biological Data Sources", Proceedings of Sixteenth International Workshop on Database and Expert Systems Applications, 22-26 Aug. 2005

[9] Ning Zhang; Hong Chen; Yu Wang; Shi-Jun Cheng; Ming-Feng Xiong, "Odaies: Ontology-driven Adaptive Web Information Extraction System", IEEE/WIC International Conference on Intelligent Agent Technology,3-16 Oct. 2003.

[10] Song Jun-feng; Zhang Wei-ming; Xiao Wei-dong; Li Guo-hui; Xu Zhen-ning, "Ontology-based Information Retrieval Model for the Semantic Web", Proceedings the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, 29 March-1 April 2005.

[11] Kuziemsky, C.E.; Lau, F.; Bilykh, I.; Jahnke, J.H.; McCallum, G.; Obry, C.; Onabajo, A.; Downing, G.M., "Ontology-based Information Integration in Health Care: A Focus on Palliative

Care", Eleventh Annual International Workshop on Software Technology and Engineering Practice, 19-21 Sept. 2003.

[12] Malucelli, A., Palzer, D., Oliveira, E., "Combining ontologies and agents to help in solving the heterogeneity problem in e-commerce negotiations", Proceedings of International Workshop on Data Engineering Issues in E-Commerce,9 April 2005.

[13] Rahimi, S.; Bjursell, J.; Ali, D.; cobb, M.; Paprzycki, M., "Preliminary Performance Evaluation of an Agent-based Geospatial Data Conflation System", IEEE/WIC International Conference on Intelligent Agent Technology, 13-16 Oct. 2003

[14] Hongwei Zhu; Madnick, S.E.; Siegel, M.D., "Effective Data Integration in the Presence of Temporal Semantic Conflicts", Proceedings of 11th International Symposium on Temporal Representation and Reasoning, 1-3 July 2004.

[15] Nyunt, P.P.; Ni Lar Thein, "Software Agent Oriented Information Integration System in Semantic Web", Proceedings 6th Asia-Pacific Symposium on Information and Telecommunication Technologies, 09-10 Nov. 2005.

[16] Castano, S.; De Antonellis, V., "Global Viewing of Heterogeneous Data Sources", IEEE Transactions on Knowledge and Data Engineering, Volume 13, Issue 2, March-April 2001.

[17] AnHai Doan, Alon Halevy, "Efficiently Ordering Query Plans for Data Integration", Found in 18th International Conference on Data Engineering (ICDE'02), February 2002.

[18] Jiuyang Tang, Weiming Zhang, Junfeng Song and Weidong Xiao, "Capabilities-Based Query Planning in Mediator Systems", Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS'04), 2004 IEEE.

[19] Bouganim, L.; Fabret, F.; Mohan, C.; Valduriez, P., "Dynamic Query Scheduling in Data Integration Systems", Proceedings 16th International Conference on Data Engineering, 29 Feb.-3 March 2000.

[20] Majkic, Z., "Querying with Negation in Data Integration Systems", Proceeding of 9th International Database Engineering and Application Symposium (IDEAS'05), July 2005.

[21] Chen Li and Edward Chang, "On Answering Queries in the Presence of Limited Access Patterns", IEEE.

[22] Epaminondas Kapetanios, David Baer, Glaus Glaus, Paul Groenewoud, "MDDQL-Stat: Data Querying and Analysis through Integration of Intentional and Extensional Semantics", Found in 16th International Conference on Scientific and Statistical Database Management (SSDBM'04), June 2004.

[23] Naphtali R., Alexander Vaschillo and D. Vasilevsky, " The Architecture for Semantic Data Access to Heterogeneous Information Sources", IEEE

[24] Majkic, Z.; "Massive Parallelism for Query Answering in Weakly Integrated P2P Systems", Proceedings 15th International Workshop on Database and Expert Systems Applications, 30 Aug.-3 Sept. 2004.

[25] Thanda Win and Khin Mar Lar Tun, "Mobile Agent Cooperation Methods in Hybrid Query Optimization", Proceedings of 6th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT'05), November, 2005 IEEE.

[26] Abdelkader Hameurlain, Franck Morvan, Philipp Tomsich, Robert Bruckner, Harald Kosch and Peter Brezany, "Mobile Query Optimization Based on Agent-technology for Distributed Data Warehouse and OLAP applications", Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02), 2002 IEEE

[27] LiangHuai, Shiwei Tang, Dongqing Yang and Lijun Chen, "Efficient XML Query Processing in Mediators", Proceedings of 12th International Workshop on Database and Expert Systems Applications, Sept. 2001 IEEE.

[28] Song Wang; Rundensteiner, E.A.; Mani, M., "Optimization of Nested XQuery Expressions with Orderby Clauses", 21st International Conference on Data Engineering Workshops, 05-08 April 2005 IEEE.

[29] Cheng Luo, Zhewei Jiang, Wen-Chi Hou, Feng Yan and Chih-Fang Wang, "Estimating XML Structural Join Size Quickly and Economically", Proceedings of the 22nd International Conference on Data Engineering (ICDE '06), 03-07 April 2006 IEEE.

[30] Tengjiao Wang, Dongqing Yang, Shiwei Tang and Yunfeng Liu, "Discovering and generating materialized XML views in data integration system", Proceedings of International Database Engineering and Applications Symposium (IDEAS '04), 7-9 July 2004 IEEE.

[31] Gilles Nachouki and Mohamed Quafafou, "MDSManager: A System Based on Multidatasource Appraoch for Data Integration", Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05).

[32] Shahram Rahimi, Norman F. Carver, "A Multi-Agent Architecture for Distributed Domain-Specific Information Integration", Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), January 2005.

[33] Joan Lu, Umair Rahman, Shaowen Yao, "An Agent Related Implementation on the Web Information Retrieval System", Found in: Third International Conference on Information Technology and Applications (ICITA'05), July 2005.

[34] Aijuan Dong; Honglin Li, "Ontology-based Information Integration in Virtual Learning Environment", Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, 19-22 Sept. 2005.

[35] Hui Yang; Minjie Zhang; "Ontology-based Resource Descriptions for Distributed Information Sources", Third International Conference on Information Technology and Applications, Volume 1, 4-7 July 2005.

[36] Ian Gorton2, Justin Almquist, Kevin Dorow1, Peng Gong3, Dave Thurman, "An Architecture for Dynamic Data Source Integration", Proceedings of the 38th Hawaii International Conference on System Sciences – 2005 IEEE

[37] Alon Halevy, Anand Rajaraman and Joan Ordille, "Data Integration: The Teenage Years", International Conference on Very Large Database (VLDB'06), September 2006.