IOSR Journal of Engineering
Apr. 2012, Vol. 2(4) pp: 640-643

Information Integration for Heterogeneous Data Sources

Er. Majid Zaman,
Scientist Directorate of Information Technology & Support Systems, University of Kashmir, Srinagar, J&K, India

Dr. S. M. K. Quadri
Head & Director, PG Department of Computer Science, University of Kashmir, Srinagar, Srinagar, J&K, India

Er. Muheet Ahmed Butt
Scientist Directorate of Information Technology & Support Systems, University of Kashmir, Srinagar, J&K, India

Abstract

Information Retrieval from heterogeneous information systems is required but challenging at the same as data is stored and represented in different data models in different information systems.Information integrated from heterogeneous data sources into single data source are faced upon by major challenge of information transformation- were in different formats and constraints in data transformation are used in data integration for the purpose of integrating information systems, at the same is not cost effective.

This paper introduces ideaof Information integration based on search criteria from heterogeneous data sources into single data source. Every element of information source such as entity, field, and relation is mapped to component of new single text source-created every time heterogeneous information systems are searched and result is saved into new text file.

This approach allows us to create new text file and delete existing file, modifiying wrapper, making modifications later and managing data retrieval in a simple unified style. This architecture is flexible enough to incorporate variety of data models and query capabilities by various protocols. It is possible to select logically tied information from all available legacy data sources.

Introduction

In a world of wide scale data sharing, coordination techniques are becoming more and more challenging. Information is expected to be found fragmented and distributed among multiple autonomous sources, making data retrieval a complicated procedure. The situation is further worsened if we take into account the significant heterogeneity, observed between these sources: Shared data is stored in different systems, described by various formats and entails different semantics. Data integration approaches are trying to solve these burdens, so that user queries will be able to retrieve the expected answers, combined correctly from multiple sources

Organizations, both governmental and business, have to manage large amount of information stored in some form of databases or files. One of the main problems to deal with information managing is the weak interoperability between various databases and information systems [17]. Especially this problem is serious when we want organize a collaboration between the information systems of various departments within the organization.

Data retrieval from different autonomous sources has become a hot topic during the last years. For instance, there are such data sources as employee data source, student data source, library data source etc within the same enterprise (talking of academic institution). When someone wants piece of information we need to execute n queries and possibly provide user with n such results, retrieved from n data sources.

Heterogeneous data sources are searched based on user criteria and result of n sources is integrated into single source, this data source is created every time heterogeneous information systems are to be searched & structure of this single data source is dynamic and not static as such structure of this source is variable and is defined a fresh every time

Distributed and Decentralized Data

Data storage is organized in a completely decentralized manner and information retrieval might involve querying multiple data sources. In large enterprises for example, where decisions are usually made based on data observations, each department might keep its own database system (HRM, Finance, Sales etc). Therefore accumulated information for the whole enterprise requires data combinations from various sources. The decentralization is further enhanced if we also take into account possible partners, vendors or competitors, whose data might be of company's interest. Another area, where this decentralization

worsens information retrieval is large scale scientific projects. Here not only the volume of data, but also the complexity has to be taken into account. Scientists nowadays, besides profound domain knowledge, require access to data and results provided by others. Therefore, querying individually different data sources leads to significant inefficiency in their work. Finally, for an effective search in Enterprise, user is required to look up for information in multiple data sources and collect the data individually.

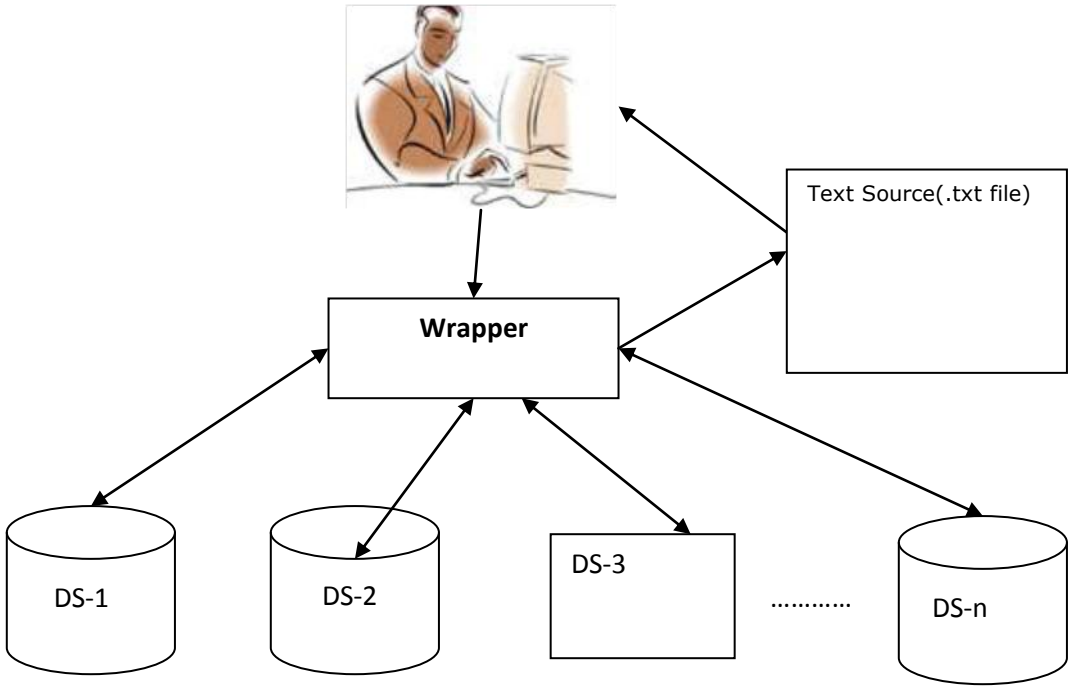
Heterogeneity of Data Sources

In addition to the decentralization, the effectiveness of information retrieval is further worsenedby the variety of heterogeneity present in the data sources. In each of these sources, data is organizedusing a different system (operating systems, SQL Vendor Implementations etc), based on different[17] conceptual models, and on different formats "system level heterogeneity" is considered to be nowadays much easier than before (e.g. viaODBC/JDBC connections on relational databases), much interest is laid on the so called "semanticheterogeneity", which appears every time there is a more than one way to structure a body of data.Semantic heterogeneity seems to be an unavoidable burden in data sharing and manipulation, sincepeople tend to model their data according to their own understanding of the reality. This of course isfundamentally different for each individual. In that sense, heterogeneity is to be found in data models,conceptual schemas, and of course the mind of the users.

Solution

The basicprinciple of data integration is to combine(integrate) selected information sources from aspecific domain, in a way that a whole newdata Source is generated. The enduser, when querying for data, has the illusion ofinteracting with one single system, which presents him a unified logical view of the data available. The first attempts to address information integration issues in enterprises where based primarily on data warehousing techniques, however our proposed architecture, is described graphically below.

Traditional solutions prescribe creation of new data source on Information integration from heterogeneous data sources, which is not cost effective.



The schema provides the user with a unified view of all data, on which queries can be posed. This particular schema is not designed to store any data; it is purely a logical schema. A user query (*Q*) is reformulated over the source schemas automatically based on the set of rules *M* (Source descriptions) and searched locally on the autonomous text sources. Hence, the reformulation of Q results in a set of source-specific queries *Qi*, whose combination will yield the answer to the initial query *Q*.

Wrapper

Performs following tasks

1. Query Reformulation

There are two main contexts in which the problem of answering queries using views has been considered. In the first context, where the goal is query optimization or maintenance of physical data independence, we search for an expression that uses the views and is *equivalent* to the original query. Here it is usually assumed that the number of views is on the same order as the size of the schema. The second context is that of data integration, where views describe a set of autonomous heterogeneous data sources. A user poses a query in terms of a mediated schema, and the data integration system needs to reformulate the query to refer to the data sources. In a subsequent phase, the queries over the sources are optimized and executed. The reformulation problem can be solved by algorithms for answering queries using views, though in this context, we usually cannot find a rewriting that is equivalent to the user query because of the data sources limited coverage.

In some data integration applications, the number of data sources may be quite large – for example, data sources may be a set of web sites, a large set of suppliers and consumers in an electronic marketplace, or a set of peers containing fragments of a larger data set in a peer-to-peer environment. Hence, the challenge in this context is to develop an solution that scales up in the number of views.

As such, user query posed in terms of a mediated schema, and the data integration system needs to reformulate the query to refer to the data sources. Since there are n heterogeneous data source, but user desired result may be present in m views where n>m, as such it is the responsibility of Wrapper to identify m sources and prepare resultant m queries.In a subsequent phase, the queries over the sources are optimized and executed.

To minimize data retrieval, Wrapper generates very selective SQL, returning only the data that is needed. To avoid retrieving rows that are not needed, the conditions in Where clauses and predicates are converted to Where clauses in the generated SQL. To avoid retrieving columns that are not needed, the generated SQL specifies the columns actually needed by the user.

***2.* Data Transformation**

Wrapper- sends queries to a data source, receives answers back, possiblyapplies basic transformations and creates new text source, this newly created text source is defined in accordance to the result generated as a result of executing query on heterogeneous data sources, and transformed data is stored in this text source, finally user is provided result from this text source-source text file is tab separated, as such user can be provided result in desired format e.g .pdf,.doc,.xls etc.

It is common that applications need to deal with data which is not available in a single format; and that's the context where dealing with a single query language, data model and interface which covers heterogeneous data sources becomes fundamental. Think about a scenario, for example, where a list of auctioned *ITEM*s is available in an XML document, as but details about the person who's offering the *ITEM* are available in a *USERS* table hosted on a relational database, including information about the user id, name, address and email. Now think about the need of creating an application that given a user's email address retrieves all the items that are being auctioned by that user.

The wrapper consuming the result is aware of the physical origin of the data returned as a result of execution of queries even if the result mixes information stored in a relational database and in an XML document. Since data received by the wrapper are in different formats is transformed into generic format, extracted data is transformed and saved into text source, before saving in text format-text source is created in accordance with data retrieved as a result of execution of n queries on n heterogeneous data sources, definition includes column definition.

Extracted, refined, cleaned, transformed, saved data in temp text source is passed onto user.

***3.* Query Processing**

Another issue that had to be redefined in Data Integration scenarios is query processing. In a traditional DBMS, query processing is comprised distinctively from a query optimization and a query execution phase. Query is optimized at compile time, generating a query execution plan at run time, which follows strictly the instructions of the optimization. However in Data Integration Applications, an optimized query execution plan cannot be constructed during compilation, because properties of the data sources are usually unknown beforehand (cardinalities, ordering information, histograms and other selectivity estimation aids, dependencies and uniqueness constraints). In addition the operating environment of each data source is also unknown (CPU speed, disk access time etc)[17].

Several works discussed extensions to query optimizers that try to make use of materialized views in query processing. In some cases, they modified the System-R style join enumeration component, and in others they incorporated view rewritings into the rewrite phase of the optimizer. These works showed that considering the presence of materialized views did not negatively impact the performance of the optimizer. However, in these works the number of views tended to be relatively small.

We consider the problem of finding the most efficient rewriting of the query using a set of views, in the context of query optimization, where query execution plan is being modified at run time.

Scheduling based Methods that preserve the logical structure of the query plan, but re-schedule the order in which operations are processed by the CPU. Redundant Computation Methods that use several query plans to process the same data. The most efficient plan is finally executed and the rest are abandoned.

Conclusion

In this paper we have discussed how Single Wrapper can be useful in providing data services which accomplish data integration tasks across heterogeneous data sources. In order to succeed in that task, Wrapper implementation must be optimized to deal with the peculiarities of the various supported data sources. Wrapper implements a variety of techniques when dealing with relational databases and XML documents; those include the ability to push SQL to the relational engine, to minimize the amount of data retrieved from the database, transform, refine, clean,& save data into text source finally is passed onto user.

Although Data Integration was considered to be "an area of intellectual curiosity"[1] at its early years, the advent of information sharing nowadays is calling for effective integration approaches realized in practice. Users are not compromising with low standards of information accuracy and are willing to find the right information at the right time. The research community, thus far, has shown excellent progress in dealing with the most crucial problems presented on the way of integrating data, however, further challenges arise constantly: The expansion of (semi -) & unstructured data (XML) for example implies that data sources are even more complex and difficult to handle. Coping with semantic heterogeneity in such scenarios seems almost impossible. However, research is getting even more intense and promising ideas are expected to develop.

References

1. S. Bergamaschi, S. Castano and M. Vincini, "Semantic integration of semi-structured and structured data sources," SIGMOD Rec., vol. 28, 1999, pp. 54-59.
2. M. Berry and G. Linoff, Data Mining Techniques For Marketing, Sales, and Customer Support. John Wiley & Sons, 1997.
3. A.W. Brown and G. Booch, "Reusing Open-Source Software and Practices: The Impact of Open-Source on Commercial Vendors," Proc. 7th Intl Conf on Software Reuse: Methods, Techniques, and Tools, Springer, 2002.
4. I.R. Cruz, X. Huiyong and H. Feihong, "An ontology-based framework for XML semantic integration," Proc. Intl. Database Engineering and Applications Symp., IEEE, 2004, pp. 217-226.
5. Alon Halevy, Anand Rajaraman and Joann Ordille. "Data Integration: The Teenage Years", In VLDBConference, pages 9-16, 2006.
6. Maurizio Lenzerini. "Data Integration: A Theoretical Perspective", In Symposium of Principles of Database Systems, 2002.
7. Alon Halevy . "Information Integration", In Encyclopedia of Database Systems, 2009.
8. Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying
9. Heterogeneous Information Sources Using Source descriptions.
10. *Proceedings of the International Conference on Very Large Databases*
11. *(VLDB)*, 1996.
12. Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying Heterogeneous Information Sources Using Source descriptions. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 1996.
13. Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying Heterogeneous Information Sources Using Source descriptions. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 1996.
14. Jason Bloomberg and John Goodson. Best Practices for SOA: Building a Data Services Layer. *SOA World Magazine*, May, 2008.
15. DataDirect Technologies. DataDirect XQuery Web Service Framework. *http://www.xquery.com*
16. S. Vajjhala and J. Fialli. The Java architecture for XML binding (JAXB) 2.0.*http://jcp.org/en/jsr/detail?id=222*.
17. *Hot Topics in Data Management System. Data Integration Underlying problems and Research Approaches, ETH Group 2010*