Implementing of XML and Intelligent Algorithm for Improving Web Query Processing in Heterogeneous Database Access

Mohd Kamir Yusof, Ahmad Faisal Amri Abidin, Sufian Mat Deris, Surayati Usop
Fakulti Informatik
Universiti Sultan Zainal Abidin
Terengganu, Malaysia
mohdkamir@unisza.edu.my

Abstract

Web query processing is hot issue in heterogeneous database access. Performance of web query processing is become slowly due to increasing number of data. A good methodology is needed to improve web query processing in heterogeneous database. In this research, intelligent algorithm and XML was created and implemented for accessing heterogeneous database. The heterogeneous database means data sources allocated at different places. Then, these data source will integrate in order to allow web users access all data. The main function of the intelligent algorithm is to search and retrieve only related data based on queries from web users. Meanwhile, the main function of XML is to map directly to data sources that contain related data. Prototype architecture based on the intelligent algorithm and XML was designed. This prototype architecture was carried out with one application. This application was tested for heterogeneous database access. The result indicates, implementation of intelligent algorithm and XML able to improve web query processing in heterogeneous database access.

Keywords: *Heterogeneous database, Intelligent algorithm, Web query processing, XML*

1. Introduction

Most of internet applications provide lot of useful information to users. Internet application is important tool for user to gain information such as education, entertainment, health, etc. The information on the World Wide Web has lead to the need processing intelligently to address more of the user's intended requirements than previously possible [1]. The main challenge of internet applications is to ensure web query be able to search information reflects the user's need information. The issue in internet applications is web query processing. This problem will due possibility of user to get relevant information is low [1]. Increasing number of data also effected to web query processing. This problem occurs caused by less intelligent platform or methodology in web query processing engine. In this research, two purposes have been identified and need to solve in order to improve web query processing. First is integrating heterogeneous database. Second, create intelligent algorithm and implement XML in web query processing for heterogeneous database access. This implementation is important to improve web query processing. When web query processing performance was increased, internet applications will able to display possible relevant information to web user. In this research also, prototype architecture will design and come out with internet application for experiment and testing purpose.

2. Related Works

In this section will describe about heterogeneous database and XML approach. The heterogeneous database concept will use in this research. Meanwhile, XML approach will implement in order to improve web query processing.

2.1. Heterogeneous Databases

Heterogeneous database is new challenge in database domain. Nowadays, most of research is ongoing efforts to develop more intelligent and robust methods for querying and integrating data from heterogeneous data sources. World Wide Web (WWW) has generated an urgent need for a new and robust methods that simplify the querying and integration data due to unprecedented increase in the availability of information [2]. Based on the past researches, researchers focus on developing methodologies or technique for databases integration. The purpose is to integrate data from existing databases in a distributed environment while minimizing the impact of operations on the databases [3]. One of the approaches is to use a unified global integration schema, such as the relational schema, to facilitate efficient global processing. This approach is efficient for web query and data integrate but their global schemas become hard to manage as the number and types of data sources increase [2]. Another approach for database integration is mediators and wrapper. This approach is remarkably scalable, and allows the integration of an increasing number of data sources. Mediation does not store any data on its own rather it provided a virtual view of the integrated sources [4]. In this research, heterogeneous database approach will use to handle and manage huge of data and to improve web query processing.

2.2. XML

Extensible Markup Language (XML) is accepted as a standard for internet data representation as a mark-up language and for data exchange over the internet [7][8]. Most of internet applications use XML approach for storing and exchange the data. XML is powerful technique where this technique allows users to ask very powerful queries on the web.

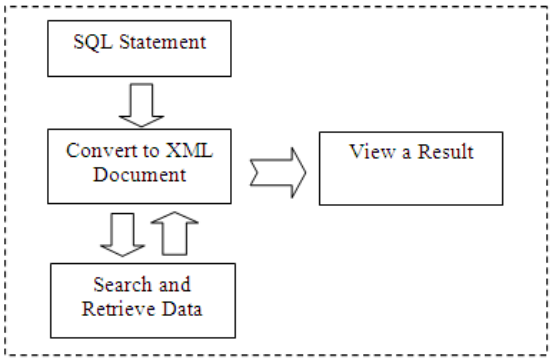

Figure 1: XML Works

In figure 1, application will receive query from web user and convert into SQL statement. For instance, query equal to “Information Retrieval”, and convert to in SQL statement as “SELECT * FROM tablename WHERE fieldname=Information Retrieval”. The SQL statement will convert again into XML document. In XML, document will

save as a xml format. For example this SQL statement will convert to search.xml (Fig. 2). This XML document will communicate with database server (that contains data sources) for searching and retrieving process.

```
<Query>
        <entry>
        <query>Information Retrieval</query>
        </entry>
</Query>
```

Figure 2: search.xml

After searching and retrieving process have been done, the result will convert to XML document before display to web user. The main advantage of XML is powerful technique for search and retrieves data from database. In this research, XML will use in order to improve web query processing for heterogeneous database access.

3. Implementation of Intelligent Algorithm and XML

In this section, details about implementation of intelligent algorithm and XML were described.

3.1. Structure of Intelligent Algorithm

In this section, structure for an intelligent algorithm was described. This structure was designed in order to integrate with system architecture before implementation phase. In figure 3, four core processes in intelligent algorithm are initial query, exploit a query, assign possible queries, and match a query.

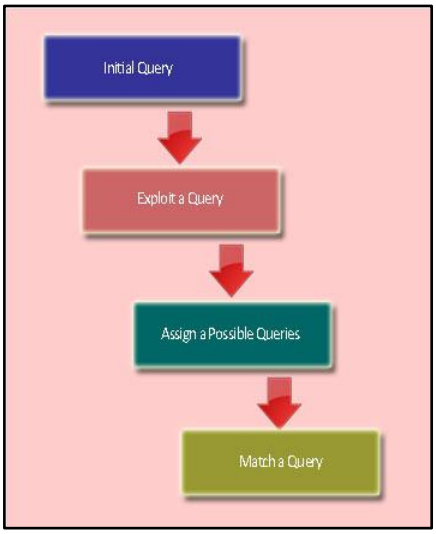

Figure 3: Structure of Intelligent Algorithm

a) Assign Initial query

Initial query is a keyword or information was received from web user. The web user is needed to enter a keyword or information through the user interface. Through this intelligent algorithm, assign initial query equal to X.

X → Initial Query, for example web user request about "Information Retrieval". If request equal to "Information Retrieval", assign X → Information Retrieval.

$Y \in \{X_i, X_{i+1}, X_{i+2} \ldots X_n\}$, where i = 1,2,3, n

Number of initial query (Y) is depending on number of query send by web users.

b) Exploit a query

An initial query will exploit one by one. The process of exploitation is based on algorithm below:

```
Start
assign x and i
x = "Information Retrieval"
i = exploit[x," "]
for (j=0; j<count(i); j++) // looping process until data equal to null
{
   m = i[j]; n = n.m  }
End
```

Figure 4: Exploit Initial Query Algorithm

Example

Based on algorithm above, consider initial query, X → "Information Retrieval". The results after executed this algorithm is i[0] → "Information" and i[1] → "Retrieval". However, this result can write as X ∈ {Data, Mining}.

c) Assign Possible Queries

After exploit process, these data will combine in order to produce any possible queries. The algorithm to produce any possible queries was created as below:-

```
Start
assign a,b
for (i=0; i<count(n); i++)
{
   a =a." ".a[i]
   for (j=0; j<count(TempFile); j++)
   {
     b = b →b[j]; a ≠ b;  i ≠ j
   } }
End
```

Figure 4: Assigning Any Possible Queries Algorithm

Example

Based on algorithm above consider an initial query, X→ {Information, Retrieval} have been exploited. The results produce two possible queries; "Information Retrieval" and "Retrieval Information".

m[0] → Information Retrieval
m[1] → Retrieval Information

d) Match a Query

This process is important to ensure only relevant data will select from different data sources. The most important in these processes are to match keywords in temporary file match among possible queries. In figure 6 shows the algorithm for match a query.

```
Start
assign a,b,c
y ∈ {x_i, x_i+1, x_i+2, x_i+n} // i equal to 0, 1, 2, ..., n
for (i=0; i<count(x),i++) { // x represent number of possible queries
  fileopen (tempFile) // temporary file contains list of keywords
  if (y[i]==tempFile[i])
    go to mapping data schema
  else if (y[i]!=tempFile)&&(y[i]==data[i])
    store new keywords and data schema // refer to new data sources
  else
    loop until y equal to null
}
End
```

Figure 6: Matching, Searching and Retrieving Algorithm

Example

Suppose we have 3 possible queries, $M \in \{x_1, x_2, x_3\}$. Firstly, find and match these queries in temporary file. Matching process will loop until number of possible queries equal to null, $M \neq$ and $M \geq 1$. If $M \leq 0$, hence number of possible queries is null. In this theorem, keyword (x_i) will store with data schema (y) and data source (destination, z). So, we can write $X_i \rightarrow Y \rightarrow Z$. In this case, if X_1 equal to Xi, hence go directly to specify data source. In second cases, if X2 equal to Xi, hence go directly to specify data source. Otherwise, find and match again until number of keyword (x_i) equal to null. If not found, we assign a new keyword, x_i and store into temporary file. Next process is to assign a data schema and data source.

3.2. XML Mapping

In this implementation, initial query from web user will convert to SQL statement. Then, this SQL statement will convert into XML document.

Example

Suppose web user request about "Information Retrieval". Two possible queries are "Information Retrieval" and "Retrieval Information". Here, two XML document will produce. First is search1.xml and second is search2.xml.

```
<Query><entry>
<query>Information Retrieval</query>
</entry></Query>
```

Figure 7: search1.xml

```
<Query>
<entry><query>Retrieval Information</query>
</entry>
</Query>
```

Figure 8: search2.xml

After that, xml document will search and find keywords in temporary file. The temporary file keep data sources destination. After, xml document will map directly to data sources (that contains information) for searching and retrieving process.

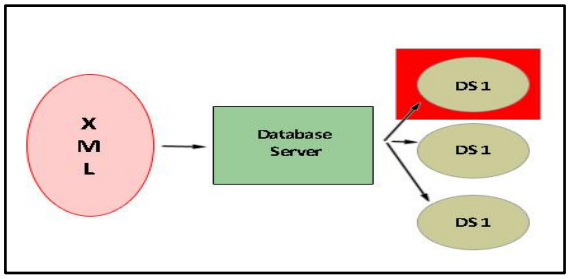

Figure 9: XML Mapping

In figure 9, XML will communicate with database server. Based on keywords has been found in temporary file, only data source 1 (DS 1) contains the data about "information retrieval" or "retrieval information". In this case, searching and retrieving process only occurs at data sources 1. After data is found, the result will convert into xml document. Finally the result will display to web user by loading xml document. Based on example below, implementation of XML is efficient and effective in web query processing for heterogeneous database access.

4. System Architecture and Implementation

J2EE technology was implemented in this prototype architecture in figure 10. Simple application was develop using JSP (Java Server Pages) based on architecture in figure 10.

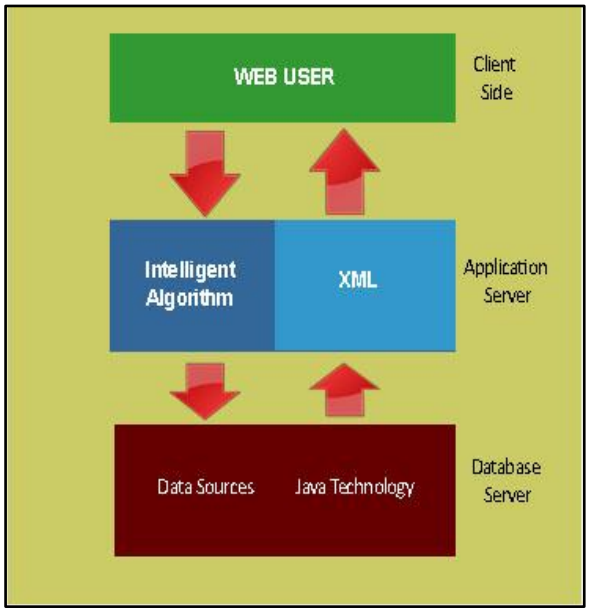

Figure 10: System Architecture

In figure 10, three sections are divided which is client side, application server and database server. In client side, web user will enter a keyword such as “Data Warehouse”, “University of UK”, etc. This keyword will pass to application server. In application server, two components are intelligent algorithm and XML. Four components core process in intelligent algorithm: (1) initial query, (2) exploit query, (3) assign possible query and (4) match query. The process for each component already explained in section 3. The main function of XML is to map directly data sources based on assigned identifier based on keyword. In database server, two components are data sources and java technology. Data sources contain data and allocate at different places.

4.1. Implementation

In this implementation, web application was develop using JSP (Java Server Pages). This environment was chosen because it would make the system portable and easily accessible through the World Wide Web (WWW). J2EE (Java 2 Platform Enterprise Edition) was chosen as a platform for server programming in Java programming language. The J2EE platform simplifies enterprise applications by basing them on standardized, modular components, by providing a complete set service to those components, and by handling many details of applications behavior automatically, without complex programming [5]. This platform also supports JavaBeans components, Java Servlets API, JavaServer Pages and XML technology. XML is standard for data exchange on the World Wide Web [2][6]. Figure 11 shows an enterprise application model involves with J2EE platform. In this model, 3 components are divided into client-side presentation, server-side business logic and enterprise information system. J2EE is the middle part between client side and enterprise information system. Four components involved are client-side presentation, server-side presentation, server-side business logic and enterprise information system. Interaction between all these components is needed to ensure all process can be executed efficiently.

4.2. Sample query

This section illustrates how our system works using a sample query. Suppose that a web user is looking to buy books. Assume that a web user is looking to buy “Data mining” book.

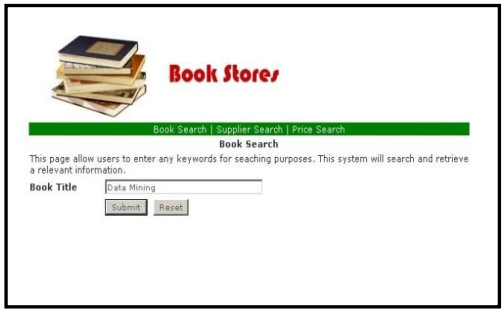

Figure 11: Search Form

The user interface is a Web site which is allows the web users to enter initiate keywords (Fig 11). The user requests information through queries submitted to the system via an HTML form. Once the query has been submitted, it is sent to the client side. Once this query has been submitted, this query will define as an *initial query*. After that, this initial query will exploit into two words; data and mining. Then, the next process is refinement possible query. This process will refine and display any possible query, for instance “data mining” or “mining

data". Last module in client side is matching process. This process will communicate to data warehouse in order to find and match a possible query with keywords in data warehouse. Meanwhile, on server side, keywords in data warehouse have been assigned to data schema. The code processes in client side are searching and retrieving. In this concept, data schema was stored in data warehouse, but physical data store in data sources. Once any changing occurs in data sources, automatically data schema in data warehouse also changed. Based on the data schema, system directly maps to certain data sources. System automatically retrieves any possible and relevant information.

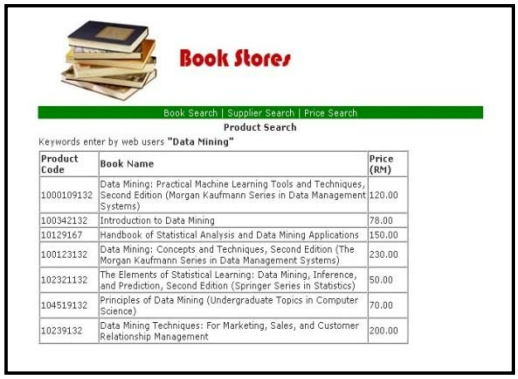

Figure 12: Result

In figure 12, all relevant information will display to web user. These all relevant information is based on keywords entered by web user and through processes or modules on client side and server side.

4.3. Analysis

In section 4.2, sample queries ("Data Mining") have been executed with the results shown in Figure 12. Based on figure 10, only relevant information will display to web users.

Table 1: Query Result

Sample Query	Number of Relevance Data	Assign to Possible Queries	Response Time
Data Mining	5	(Data Mining) or (Mining Data)	4 sec
Introduction to Data Mining	6	(Introduction to Data Mining) or (Mining)	3 sec
Information System	10	(Information System) or (System Information)	5 sec
Heterogeneous Database	11	(Heterogeneous Database) or (Database Heterogeneous)	5 sec
Biological Data	3	(Biological Data) or (Data Biological)	2 sec
Multimedia Data	10	(Multimedia Data) or (Data Multimedia)	4 sec
Science and Technology	4	(Science and Technology) or (Technology and Science)	2 sec

Table 2: Query Result

Sample Query	Number of Relevance Data	Query	Response Time
Data Mining	15	Data Mining	7 sec
Introduction to Data Mining	9	Introduction to Data Mining	5 sec
Information System	14	Information System	7 sec
Heterogeneous Database	16	Heterogeneous Database	8 sec
Biological Data	14	Biological Data	5 sec
Multimedia Data	17	Multimedia Data	7 sec
Science and Technology	10	Science and Technology	4 sec

Table 1 shows the number of queries have been executed and the number of relevance data was displayed to web users. Table 1 show the results based on implementation of intelligent algorithm and XML approach. The results indicate these approaches able to improve web query processing in term of response time compare to table 2. Based on the results in table 1, performance of response time for searching and retrieving process decrease about 10% to 20% compared to results in table 2. Meanwhile, the results number of relevance data in table 1 decrease about 70% to 30% compare to table 2. Based on this analysis, performance of web query processing in heterogeneous database access was improved by implementation of intelligent algorithm and XML.

5. Conclusions

In conclusion, implementation of intelligent algorithm and XML approach can be used for improving web query processing in heterogeneous database access. Methodology for improving web query processing in heterogeneous database access is important to ensure web user can access the important information from different data sources. Four major components in this intelligent algorithm are assigning initial query, exploit query, assign possible query, matching query and illustrated them with examples implementation. This methodology can implement in real situation and for future can implement in different domain such as biomedical, biotechnology, etc.

Acknowledgement

Special thanks to Universiti Sultan Zainal Abidin for Research Grant and Che Mat Ismail for advice and support.

References

[1] Jordi Conesa, Veda C. Storey, Vijayan Sugumaran. (2008). *Improving web-query processing through semantic knowledge. Data & knowledge engineering*, 66, 18-34.

[2] Samueal Robert Collins, Shamkant Navathe and Leo Mark (2002). *XML schema mapping for heterogeneous database access. Information and software technology*, 44, 251-257.

[3] Bright M, A. Hurson and S. Pakzad (1992). *A taxanomy and current issues in multidatabase systems. IEEE computer* 25(3), 50-60.

[4] Majid Kazemian, Behzad Moshiri, Hamid Nikbakt and Caro Lucas (2005). *Architecture for Biological Database Integration. AIML 05 Conference*, 19-21 Dec 2005, Cairo Egypt.

[5] Askar S. Boranbayev. *Defining methodologies for developing J2EE web-based information systems. Nonlinear analysis* 71(2009), e1633 – e1637.

[6] Java 2 Platform, Enterprise Edition (J2EE) Overview. http://java.sun.com/j2ee/overview.html

[7] Arzucan Ozgur, Taflan I. Gundem (2006). *Efficient Indexing Technique for XML-Based Electronic*

Product Catalogs. Electronic Commerce Research and Application 5 (2006), 66-77.
[8] Maged El-Sayed, Katica Dimitrova, Elke A. Rundensteiner (2005). *Efficiently Supporting Order in XML Query Processing*. Data and Knowledge Engineering 54 (2005) 355 – 390.
[9] Jie Cao, Wen Hou, Tinyou Cai (2008). *Research of Heterogeneous Database Integration System Based on E-Business*. 186-189.
[10] Nikos Bikakis, Nektarios Gioldasis, Chrisa Tsinaraki, and Stavros Christodoulakis (2009). *Querying XML Data with SPARQL*. DEXA 2009, LNCS 5690, pp. 372-381, 2009.
[11] Walter Sujansky (2002). *Heterogeneous Database Integration in Biomedicine*. Journal of Biomedical Informatics 34, 285-298 (2001).
[12] Carole Goble, Robert Stevens (2008). *State of the Nation in Data Integration for Bioinformatics*. Journal of Biomedical Informatics 41 (2008), 687-693.
[13] Iskandar Ishak, Naomie Salim (2006). *Data Integration Approaches for Heterogeneous Biological Data*. Proceedings of the Postgraduate Annual Research Seminar 2006. 202 – 206.

Authors

Mohd Kamir Yusof obtained her Master of Computer Science from Faculty of Computer Science and Information System, Universiti Teknologi Malaysia in 2008. Currently, he is a Lecturer at Department of Computer Science, Faculty of Infomatics, Universiti Sultan Zainal Abidin (UniSZA), Kuala Terengganu, Terengganu, Malaysia. His main research areas include information retrieval, database integration and web semantics.

Ahmad Faisal Amri Abidin obtained his Master of Computer Science from Faculty of Computer Science and Information Technology, Universiti Putra Malaysia in 2008. Currently, he is a Lecturer at Departmen t of Computer Sciences, Faculty of Informatics, Universiti Sultan Zainal Abidin. His main research areas include computer security, mobile computing and computer networks.

Nor Surayati Mohamad Usop obtained her Master of Science Computer from Faculty of Computer Science and Information Technology, Universiti Putra Malaysia in 2009. Currently, she is a Lecture r at Department of Information Technology, Faculty of Informatics, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.

Mohd Sufian Mat Deris obtained her Master of Education (educational technology) from Faculty of Education, Universiti Teknologi Malaysia in 2006. Currently, he is a Lecturer at Department of Multimedia, Faculty of Infomatics, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.