Report on WEBVIEW: An SQL Extension for Joining Corporate Data to Data Derived from the Web

Yasmine Elglaly

Motivation

The motivation of this article is to eliminate the disadvantages and the difficulties of the existing data mining and knowledge management tools. One of the dimensions of this problem is how to collect up-to-date data from the web. The authors proposed how to retrieve data from the web sites and store them in a typical database using an effective and simple tool.

Problem statement

The problem that this article addressed can be stated as follows:

Given web sites and relational database, it is required to join these two forms of data into one database under the following conditions:

1. The data queried from the web pages must be at real time, and not previously collected and stored.
2. The developed tool should be easy to use, robust and affordable.

Objectives:

The authors defined their objectives of their research in three main points:

1. Design an SQL extension language that can manipulate data from web sites and join them to relational database.
2. Test their new tool, named WEB VIEW and show its robustness.
3. Implement the proposed tool and use it in practice.

Solution Approach

- The authors begin their proposed solution by defining five requirements that should be fulfilled by the SQL extension, which are:
  1. Does not depend on a specific web language.
  2. The ability to reconstruct the web data.
  3. Retrieve any kind of web data.
  4. Consistency with existing SQL databases.
  5. Efficiency.
- The extension they made to the SQL is programming a new command called "WEB VIEW" that has its own clauses.
- The function of the command "WEB VIEW" is to query data from the different web pages on the fly.
The clauses are used to determine the "WEB VIEW" attributes; such as specifying the queried web page using the clause "USING" and specifying URLs contained in one page using the clause "LINK".

The authors test their SQL extension using four different cases to prove its robustness and effectiveness.

The first case: they test the new SQL command on ODBC database engine (Oracle, Access, Sybase, etc.). They reported that using the command "WEB VIEW" with other regular SQL statements was performed successfully.

The second case they tested is using "WEB VIEW" to retrieve data from different web pages. In their test, they succeeded to retrieve data from "froogle.google.com" after defining a key word to search with. Using the clause "Link", "WEB VIEW" succeeded to retrieve data from 88 web pages within froogle web site.

The third case was to test the use of "WEB VIEW" with hierarchical data sets. In this test case, it was shown the powerful of the clause "NESTED" in manipulating XML data and storing the retrieved data in a relational format.

The forth case and the last one; it was shown how the "WEB VIEW" can join different views of relational database tables and other webviews. This task is performed using the "USING" clause.

**Strengths**

The main advantages of WEB VIEW are:

1. It is simple tool, unlike other complicated software programs that are used to retrieve data from the web.
2. The data retrieved from the web using WEB VIEW are not static.
3. It can retrieve all web data types such as HTML and XML.
4. It is an extension to SQL; so it inherited all the SQL advantages.
5. It can work on any SQL-based database, which is highly used by systems and are very familiar to users.
6. Cost is low.

**Note:**

- This tool can be considered as a first step in data mining or knowledge management system. It provides these systems with a simple tool to collect and corporate data which are ready for being analyzed.
- In order to use WEBVIEW command, the user must know prior information about the web site he wants to retrieve data from it. The user must specify the tables and the attributes that he will store the XML data into them.
Summary of "Storing DTD-Independent XML Data in Relational Database"

Motivation
Store XML data into relational database without being dependent on the DTD information of the XML document. This is because DTD-dependent approaches can lack accuracy in case of missing data.

Problem Statement
Given only XML document (no DTD information is provided); it is required to translate it into a relational schema. Then store the data extracted from the XML document into a relational database.

Objectives: Develop a Schema Mapping Algorithm and a Data Mapping Algorithm.

Solution Approach

- The basic idea of this paper's approach is that the XML document can be viewed as an ordered tree. The relational schema is defined by determining the tree structure and the position of elements in this structure.
- The authors used this idea to build their Schema Mapping Algorithm, which consists of the following steps:
  - Use SAX parser to parse the input XML and extract the elementList and the attributeList.
  - Extract distinct element's parent from the elementList to get the table names tableNameList.
  - Get the tableElementList by eliminating elements with the same parent.
  - Get the tableAttributeList by extracting attribute name with different parent name from the attributeList.
  - Create the database schema based on the information gathered from the previous steps.
- Then, the authors described their data Mapping Algorithm which can be simply summarized as follows:
  - Insert every element in the elementList in its suitable place.
  - The element could be a new table, new tuple in a table or a new value.

Strengths

1. Does not rely on DTD data.
2. Tables are created dynamically based on the input XML.
3. Can query XML data that are stored in relational database using SQL.
4. Original XML document can be re-constructed from the mapped data that are stored in the relational database.

Notes: The creation of the schema focused on the creation of the tables only. Further processing should be done to set the relations between the tables. As the authors choose to map the XML document to Relational database in particular, they should describe how the relations would be determined by their algorithm; or should they be set manually. Also, the resulted tables need to be normalized.

**Motivation**

Find the best mapping strategy for XML data in order to be stored in object relational database. The searching strategy should be adaptive and matches the current application.

**Problem Statement**

The problem this paper tries to solve is to decompose the input XML schema S into set of relations R={r₁,r₂,...} using mapping strategy f.

Given:

1. Set of XML schema transformations T
2. Set of sample data; XML documents and XML queries (To characterize the application)
3. Cost function f that evaluates the relational schema R for data sample D.

Goal: Find the optimal relational schema R, where its cost function is minimum

**Objectives**

1. Improve the process of searching the optimal mapping strategy.
2. Enhance the adaptation process using similarity of XML data.
3. Combine it with the idea of user-driven techniques.

**Solution Approach**

1. Use the Ant Colonies optimization technique to find the (near) optimal schema.
2. The transformations used are the inlining, outlining and unshredding.
3. The cost function is evaluated in terms of the number of joins needed to evaluate queries Q.
4. Borrow the idea of the system UserMap, and map similar fragments of the schema similarly.
5. Combine both partly cost-driven and partly user-driven strategies.
6. Represent the user-specified annotations by composite transformations to speed up the search.

**Strengths**

1. Applying the Ant Colony optimization technique highly improves the searching step results because it can avoid the local optima and reaches the global optima.
Comparison between the previous three papers

The common features between the three papers:

- They agreed that there is an increasing need to collect the tremendous amounts of data that are spread all over the Internet.
- Collecting these data should be dynamic.
- The best repository to store the collected data within; is the relational database.

The differences between the three papers:

- The first paper [1] is a pioneering research on mapping XML documents to relational database.
- Their product WEBVIEW is a great success and widely used.
- Its popularity is gained because of its simplicity. For database developers, they only have to learn one new command "WEBVIEW" and its clauses.

In the second paper [2] the authors proposed a new method to extract data from XML documents and store it in relational database. It is very different from paper [1].

In [2] the authors propose a new algorithm that can automatically create a relational schema based only on the XML document.

Moreover, they proposed an algorithm to load the data into the database.

- The third paper [3] is not actually a new tool for storing XML data in relational database. The authors' contribution was only in improving the search strategy of the mapping process.

- Although the Ant Colony (AC) is a famous heuristic in solving optimization problems, the authors did not mention any comparisons between the AC results and previously used strategies' results.

- It is well known that AC is more complex than greedy algorithms. It will be more convenient to show that the performance of the mapping process is highly enhanced so that it is acceptable to increase the system complexity in order to obtain improved results.

Table1. Summary for the comparison between the three papers

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Contribution</td>
<td>Join XML to relational database.</td>
<td>No need to DTD information.</td>
<td>Enhance mapping strategies.</td>
</tr>
<tr>
<td>Strongest Feature</td>
<td>SQL extension.</td>
<td>Automatic schema mapping.</td>
<td>Finds the global optimal mapping strategy.</td>
</tr>
<tr>
<td>Weakness Point</td>
<td>No automatic data-extraction.</td>
<td>Resulted tables need further processing and normalization.</td>
<td>Depend on user annotations</td>
</tr>
<tr>
<td></td>
<td>Prior information about the XML document is a must.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Experimental Results</td>
<td>Satisfactory</td>
<td>Satisfactory</td>
<td>Not Provided</td>
</tr>
<tr>
<td>Language used</td>
<td>SQL</td>
<td>Java</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>
References:

