ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (124-128), Month: April - June 2015, Available at: www.researchpublish.com

Mapping XML Data to RDBMS

[1]Kushagra Sharma, [2]S Ganesh Kumar

[1,2] Computer Science Department of Engineering, SRM University, Chennai, India

***Abstract*: XML is commonly used for data exchange over internet so we need to store and query that data efficiently. Since large amount of data is stored in relational database system we need some mechanism for mapping xml data to RDBMS so that we can easily query that XML data.XML is supported by SQL database system. In this paper we propose an algorithm that maps XML data to relational schema by extracting information from XML DTD. Here we are using tree based approach in which nodes of tree will be mapped to a relational table. This enables reuse of technology and provide interoperability between and XML and SQL.**

Keywords: RDBMS; DTD; XML; SQL.

I. INTRODUCTION

sXML is emerging as standard to exchange data over internet so we need an efficient mechanism to store and query data. Since it is easy to store and query data by using RDBMS so we need some mechanism to convert XML data to SQL tables but both XML and relational data are not compatible to each other so it is not straightforward task to map XML data to RDBMS.

The diffusion of XML in most applicative fields sets a pressing need for providing the capability to query XML data to a wide spectrum of users, including those with minimal or no computer programming skills at all. There are many mapping algorithms are available for this purpose all have their advantage and disadvantage. Here we are proposing an algorithm which is DOM based algorithm and named as XMLalgo. In this algorithm we are giving DOMTree and DTDgraph as input and by using desirable relational schema we transform XML data into SQL tables.

Data model used in this algorithm Document object model of W3C.Here we consider XML values as data fields of XML nodes.

To store XML data into RDBMS following problem we have to resolve.

Schema mapping: In schema mapping we generate corresponding relational schema of input XML DTD. Various XML elements combine into one table.

Data Mapping: In data mapping XML data is inserted into relational tuples into the database.XML data is mapped to tuples and inserted into database according to schema generated in schema mapping. In this we transform XML INSERT to SQL INSERT query. Query Mapping: XML query is transformed to sequence of SQL queries.

The following mapping functions are used.

- α (e) maps an element type to corresponding relational table.
- β (a) maps an XML attribute to a relational attribute.
- μ (e) maps a leaf element to a relational attribute.

Even when using an RDBMS, there are many ways to store XML data. One approach is to ask user to define relational schema to store data. Another approach is to infer from DTDs how to map XML element to relational table.

The rest of the paper is organized as follows .Section 2 represents an overview of related work. Section 3 describe algorithm which is going to use in this project. Section 4 represents experimental result of this project. Finally Section 5 concludes the paper and shows features of our work.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (124-128), Month: April - June 2015, Available at: www.researchpublish.com

II. RELATED WORK

Different techniques have been proposed for storing and querying XML data. One approach is to design native XML databases that support XML data model and query languages [1]. Some examples are AG's Tamino XML Server and Sonic Software's extensible Information Server. Benifit of this approach is that XML data can be stored and retrieved in their original formats and no additional mappings are required. Most native XML database have the ability to full-text searches including word subbing and proximity searches [3]. The disadvantage is that, due to the document-centric nature of these database complex searches might be not efficient and hard to do.

The second approach is use the XML enabled database system [2]. Currently, major database such as SQL Server, Oracle and DB2 provide mechanisms to store and query XML data. A set of methods is also associated with XML data type to process, manipulate and query stored XML data.

The third approach is to use existing technology such as relational database management system or object oriented DBMS to store and query XML data[3]. The main issue with this approach is to resolve the conflict between XML data model and target data model, so main issue is to develop an efficient algorithm to perform these mapping. Different approaches have their pros and cons and selection has to be made based on need of application.

III. DATA MAPPING

One approach for mapping XML data to relational schema is mapping each node of XML document to a table. Although it is an easy process but it results in many relational table. This requires many joins among different tables which causes query processing to be inefficient.

The algorithm proposed in this paper is based on W3C's Document object model (DOM) .In our XML element DOM tree, XML values considers as data fields of XML element. Data mapping algorithm proposed in this algorithm is based on schema mapping. In our algorithm we suggest combining every child attribute to its parent node. Hence, we reduce the

number of tables and consequently the average number of

joins for queries. Our algorithm takes XML document as input and produce relational schema as output. In addition it produces mapping function between XML element and attributes in the input XML document.

Definition 3.1 (DOM Tree) We are designing an XML element document D as an element DOM tree T in which nodes represent XML element and edges represent parent child relationship between XML elements. For each XML element node *e* in *T*, we use the following notations:

- E.name name of the XML element.
- E. parents the parent node of e.
- e.children children of node e.
- e.attributes attribute of node e.
- e.value value of node e.

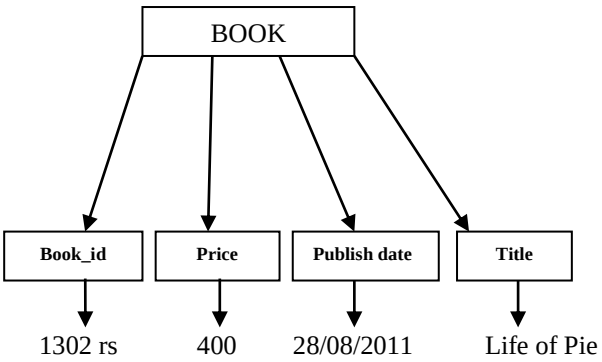

Figure: 1 Dom tree of sample xml doc

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (124-128), Month: April - June 2015, Available at: www.researchpublish.com

```
<catalog>
  <PART id="bk1061">
     <ITEM>Motherboard</ITEM>
    <MANUFACTURER>ASUS
    </MANUFACTURER>
      <MODEL>P3B-F</MODEL>
      <COST> 123.00</COST>
         </PART>
  <PART id="1">
      <ITEM>Video Card</ITEM>
      <MANUFACTURER>ATI</MANUFACTURER>
      <MODEL>All-in-Wonder Pro</MODEL>
       <COST> 160.00</COST>
  </PART>
        <PART id="bk1021">
            <ITEM>Sound Card</ITEM>
            <MANUFACTURER>Creative Labs
            </MANUFACTURER>
            <MODEL>Sound Blaster Live</MODEL>
       <COST> 80.00</COST>
  </PART >
  <PART id="bk1011">
      <ITEM>inch Monitor</ITEM>
      <MANUFACTURER>LG Electronics
      </MANUFACTURER>
      <MODEL> 995E</MODEL>
      <COST> 290.00</COST>
 </PART>
</catalog>
```

Figure 2 sample XML document book.xml

An XML element DOM tree of sample XML file is shown in figure 1.Each node e is labeled by e.name. e.value is value of node e.value is omitted when node is non-leaf node where e.value=NULL.

Our data mapping algorithm XMLalgo is based on the concept of inlinable elements introduced in the schema mapping phase. However we cannot find out whether a XML element instance is inlinable from a DOMTree itself.

Figure 3 describes our algorithm which inserts XML documents into relational database whose schema is derived from DTD previously. Given a DTD [7]graph G and nodes n1 and n2 in G1. Given an element instance *e*, *type*(*e*) denotes the corresponding element node in *G1*. We define a field EID which is associated with each element instance *e* in the algorithm. We introduce *parenEID* and *parentNodeType* fields in the algorithm to keep the parent-child relationship between the elements.

XMLalgo algorithm is composed of two nested While loops. The outer while loop maintains a Queue, *q*, to store the non-inlinable XML elements [4]. It obtains the typical information of the tuple, *tp*, corresponding to a noninlinable element, *e*, such as ID, nodeType, XML attribute values and the content [5]. Finally, it inserts the tuple *tp* into the table α(*e*) store the parent-child relationship.

If *e* is not a leaf element then the inner While loop is used to search for inlinable descendants of *e*. It maintains a Queue, *p*, to process the descendants of *e*. Firstly; it determines the inlinable descendants of *e* and fetch their data to complete the context information for the tuple *tp*. Secondly, it keeps the parent-child information.

IV. EXPERIMENTAL RESULT

We applied the data mapping algorithm introduced in this paper to XML schema mapping schemes. The XML schema mapping scheme that we use in our experiment is DomAlgo algorithm. These schemes map the XML elements to relational tables based on the operation of inlining child nodes into parent nodes.

We used a core i3 computer with 2.4 GHz processor and 2GB main memory. We ran our implementation using Microsoft DotNet framework using C#. We maintained DOM element tree using W3C's DOM specification and processed it using an on-shelf DOM API.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (124-128), Month: April - June 2015, Available at: www.researchpublish.com

Table 1 Result

Results | Messages

	ITEM	MANUFACTURER	MODEL	COST
1	Motherboard	ASUS	P3B-F	123.00
2	Video Card	ATI	All-in-Wonder Pro	160.00
3	Sound Card	Creative Labs	Sound Blaster Live	80.00
4	inch Monitor	LG Electronics	995E	290.00

Relational schema of above XML file

In the above table mapping of sample XML file is shown in which all XML attributes are mapped to column of relational table and values of nodes becomes tuples of table. We minimized the usage of system resources during the experiments to get more realistic spent time values. Loading the data into database take more time than execution of algorithm.

```
1 Algorithm XMLalgo
2 Input: DOMTree T, DTDGraph G1, Schema mappings α,β, μ
3 Output: elements in T are inserted into the relational database
4 Start
5 Queue q := EmptyQueue(), T.root.EID.value := genID(), q.enqueue(T.root)
6 While q.isNotEmpty() do
7 e := q.dequeue()
8 Table tb:= α (e)
9 create a tuple tp of table tb with all attributes initialized to NULL
10 tp.ID = e.EID.value
11 If nodeType ∈ tb.AttributeSet then tp.nodetype = e.name End If
12 For each XML attribute e.ai of e do tp. β(e.ai) := e.ai.value End For
13 If e is a leaf then
14 tp.μ(e) := e.value
15 Else /* e is not a leaf */
16 Queue r := EmptyQueue()
17 For each child e.ci of e do r.enqueue(e.ci) End For
18 While r.isNotEmpty() do
19 f := r.dequeue()
20 If f is not inlinable to e then
21 f.EID.value := genID()
22 f.parentEID.value := e.EID.value
23 f.parentNodeType.value := e.name
24 If l(type(f.parent), type(f), G) <> '*' then
25 tp.β(f.EID) := f.EID.value
26 End If
27 q.enqueue(f)
28 Else /* f inlinable to e */
29 for each XML attribute f.ai of f do tp.α(f.ai) := f.ai.value End For
30 If f is a leaf then
31 tp.μ(f) := f.value
32 Else /* f is not a leaf */
33 For each child f.ci of f do r.enqueue(f.ci) End For
34 End If
35 End If
36 End While
37 End If
38 Insert tuple tp into table tb
39 End If
40 End While
41 End Algorithm
```

Figure 3 the algorithm for mapping XML data to relational data

V. CONCLUSION

Several algorithms have been proposed for schema mapping, but the problem of data mapping has not been discussed thoroughly. In our study, we have addressed the problem of data mapping and defined an efficient and linear algorithm for mapping XML data to relational data. Our algorithm populates a relational database with the input XML documents, according to the relational schema which is generated by the schema mapping algorithm DomAlgo. based on inlining technique. We have not dealt with the referential and integrity constraints in our data mapping algorithm. These issues need to be checked as a potential future work.

REFERENCES

[1] Dongton Lee,Wasley W. Chu "CPI:Constraints-Preserving Inlining algorithm for mapping XML DTD to relational schema " 0169-023X/01/$ 2001 published by Elsveir science B.V.

[2] A. A. Abdel-aziz,H.Oakasha" Mapping XML DTDs to Relational Schemas" 0-7803-873S-X/OS/$20.00©200S IEEE.

[3] Rebecca J. Cathey, Steven M. Beitzel, Eric C. Jensen, David Grossman, Ophir Frieder "Relationally Mapping XML Queries For Scalable XML Search" 1-4244-1330-3/07/$25.OO 02007 IEEE.

[4] Zin Mar Kyu, Thi Thi Soe Nyunt "Storing DTD-Independent XML Data in Relational Database" 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6, 2009, Kuala Lumpur, Malaysia

[5] Liang Jeff Chen, Philip A. Bernstein, Peter Carlin, Dimitrije Filipovic, Michael Rys, Nikita Shamgunov "Mapping XML to a Wide Sparse Table" 1041-4347 c_ 2012 IEEE

[6] Daniela Florescu, Donald Kossmann "Storing and Querying XML Data using an RDMBS" Copyright 1999 IEEE.