

RXML: PATH-BASED AND XML DOM APPROACHES FOR INTEGRATING BETWEEN RELATIONAL AND XML DATABASES

Husam Ahmed Al Hamad

Information Computer Systems Department,
Amman Arab University, Amman, Jordan

ABSTRACT

The Semi-structure Extensible Markup Language (XML) databases become more popular and more efficient to exchange data and information. Although traditional models such as relational database a long time ago still widely used in most of databases sectors, the XML is more flexible and easy for exchange and share data. Consequently, this paper proposes a middleware relational storage for converting between XML and relational databases, the technique uses path-based relational storage approach and XML Document Object (DOM) Model, a middleware relational storage is developed to store XML tree structures and relation tables after reengineering. Likewise, the technique applies 1-index method to reduce size of the stored data. Moreover, the paper conducts a series of experiments to evaluate the performance of the proposed technique.

KEYWORDS

XML, RDBMS, relational database, conversion database, middleware schema, data mapping.

1. INTRODUCTION

XML model becomes an important database structure for storing, presenting and exchanging the data and information through mobile applications and internet. XML schema easy to understand the meaning of the text contents, it includes many features such as querying, use small volume of text and simple structure. It is a strong markup language for modelling of data and information such as extensibility, heterogeneity and flexibility. Format of XMLs can easily add additional information using the elements, attributes or ignore any information or element. This means that XML document can interpreted in multiple ways, as well as filter, and restructure its contents in order to fit application needs.

Many researchers investigate and study transformation and conversation theories of XML schema and relational model. Many problems such as message losing, relationship misjudging and field attribute changing during conversion data can be solve [1]. Other technologies and programing languages use Document Type Definition (DTD) language to define structure of XML documents and their legal elements and attributes. DOM method is also applied for translating data into an XML structure tree, it considered a platform that allows programs to access and manipulate the elements, content, structure, and style of XML documents. DOM handles set of XML elements operations such as fetch, change, add, and delete [2, 3, 4].

2. RELATED WORKS

Many researches use DOM and DTD for transformation between relational database and XML documents. Wang *et al.* [5] used path-based XML documents and storing method using a fixed relationship model. Ye Feng *et al.* [6] developed a method for mapping XML and DTD into relational schema. Teng Lv *et al.* [7] transformed relational schema to XML and DTDs then preserve semantics of data by functional dependencies of relational schemas keys. Ye Feng [8] proposed a technique for creating and optimizing DTD in order to express semantic constraints, elements and attributes of XML-DTD for converting XML-DTD to Relational Schema..

Indexing technique is also used by other researches [9] where it increases the efficiency of the result and decreases time of the query. On the other hand, determining physical paths helps to clarification of the individual path expressions in the input XML query tree [10]. Generate XML data out of relational data sources using SQL and XML queries has evaluated by Böttcher *et al.* [11], their technique supports different output formats ranging from XML documents. Semantics based keyword search in XML and relational databases identified and proposed by Tok *et al.* [12], the technique constructs the object tree for XML and the object-relationship-mixed data graph.

A mediated schema for integrating between heterogeneous XML schemas investigated by many researches. For example, Hamad [13] extracted and matched subschemas of the XML sources to find candidate subschemas, the technique creates a mediate schema for unifying and storing heterogeneous databases. Machkour *et al.* [14] converted a XML schema into an object-relational schema using DTD in order to exchange data between heterogeneous databases.

Moreover, other researchers query the database catalog, re-engineer the database tables, and convert entity relationship into XML documents [15]. SQL query has used to extract relational database catalog, transform schema catalog into XML schema (XSD) and transform contents catalog into document tree [16]. Fong [17] investigated applying schema translation technique and transformed the Extended Entity Relationship (EER) into XML Schema (XSD). Yan-Feng Zhang *et al.* [18] proposed converting XML Schema into UML diagram by clustering, restructuring, and unification of relationships, for providing a unified model for users.

Tzvetkov *et al.* [19] connected of both XML and relational databases and converted the data between XML files and relational database tables. In addition, Hong *et al.* [20] used a virtual collaboration system for converting the data between XML and relational databases. Some other researchers such as Zhou [21] parsed XML file into XML DOM tree by extracting DTD of the XML file and mapped XML document tree into a relational database. In addition, Fong [3] reconstructed relational databases into XML document trees by evaluating structure of XML DOM and functional dependency of the database. In terms of business, Oracle developed a new database technology called XML DB, the database stores and manages both of relational databases and XML files in an interoperable way [22, 23, 24].

3. TECHNIQUE DETAILS

As known, any relational database includes interrelated tables and uses SQL to access and manipulate the data. SQL query language considered an effective and wide-reliable language in the relational database model, which uses many application supports security, integrity, and concurrently accessing the data. For these reasons, store contents of XML documents in the relational databases provides many advantages.

The paper developed a fixed middleware of relational databases storage called RMXL, this middleware stores the contents of relational databases and XML documents in one sources. The technique uses path-based XML storage approach and standard relational database storage for extracting schema and contents of XML document and relational database; the techniques implements 1-index method for normalizing the extracted data before loading object instance of XML DOM data. Figure 1 illustrates the structure of the fixed RMXL middleware relational storage.

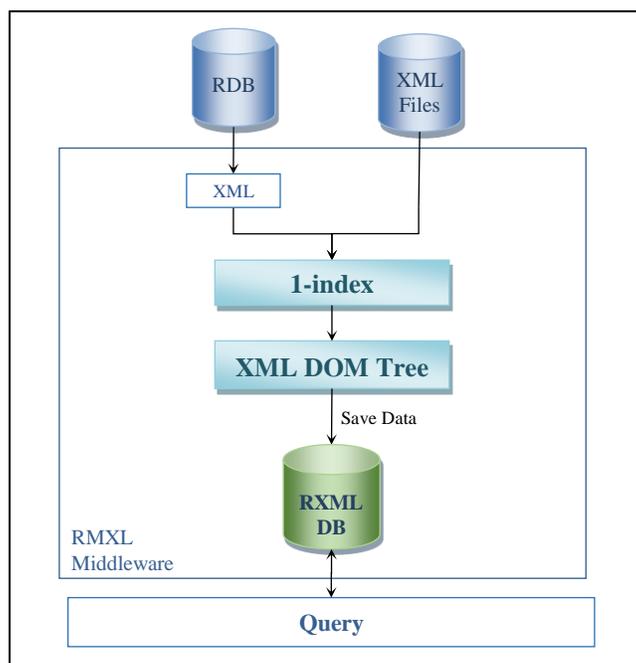


Figure 1. Structure of RMXL middleware relational storage.

Regarding XML files, the technique reads XML document tree and creates a document object first, then transform the elements and attributes values into the object instance in XML DOM, list 1 shows an example of XML document; Figure 2 illustrates an example of the DOM XML document. To make relational database compatible with XML DOM schema tree, the technique reconstructs relational tables into RXML database using SQL query, transforms all data as object instances of the XML DOM, and integrates XML DOMs into the middleware relational storage.

List 1: An example of XML Document `uni.xml`

```

<?xml version="1.0"?>
<University>
  <Student type="regular">
    <Name>Saeed</Name>
    <Id>17021</Id>
    <Age>20</Age>
  </Student>
  <Student type="transfer">
    <Name>Khalid</Name>
    <Id>17125</Id>
    <Age>19</Age>
  </Student>
</University>
  
```

```

<Student type="regular">
  <Name>Faisal</Name>
  <Id>17135</Id>
  <Age>22</Age>
</Student>
</University>

```

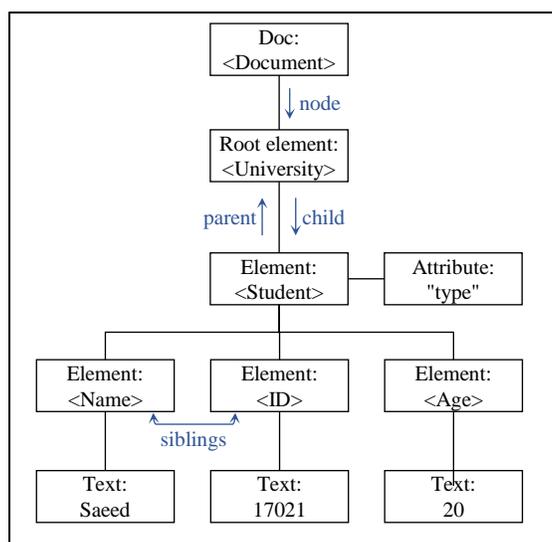


Figure 2. An example of XML DOM structure tree of XML document.

The technique uses path-based relational storage method; it uses SQL query to establish the database. The fixed middleware RXML database defines a table contains set of attributes, these attributes store contents of XML document and relational database. Before converting the data into XML DOM, the technique uses 1-index for reducing amount of stored data, it eliminates the duplicates and reduces number of nodes information; it merges the same nodes names and places at the same level based on equivalence relation (siblings). Figure 3 shows the idea behind 1-index [25], Figure 3a shows the original XML document tree and Figure 3b shows the XML document tree after applying 1-index method.

In practice of 1-index method, the technique scans names of all nodes that located at the same level, if two or more nodes have the same name the algorithm merges them using. Each combined group at the same level generates an indexID as a reference of all changes; indexID is a new identifier stored in the RXML middleware, it helps to reconstruct the original contents of XML documents. The name of indexID is the same node identifier name at the level, for example, as shown in Figure 2b, the letter "s" that considers node number, will become indexID name of the merged of the nodes "2, 3, 4, 5, and 6", and so on for the other.

The essential implementation steps of the proposed RXML middleware are proceeded through three steps, first, extracting XML schema and contents using DOM; Second, extracting relational schema and contents using DOM, and Finally, mapping the extracted information from first and second steps into the fixed middleware RXML relation.

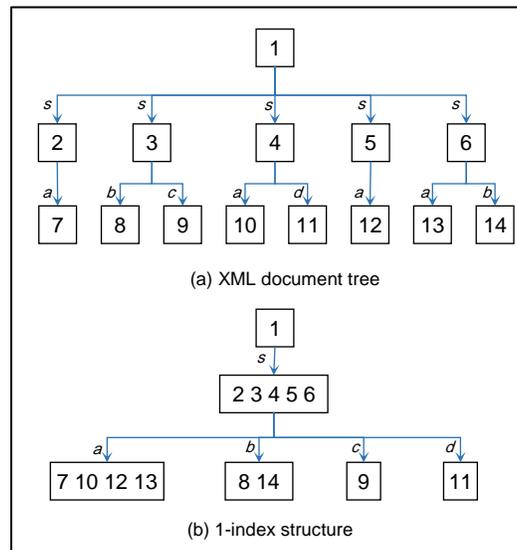


Figure 3. (a) XML document tree, (b) 1-index structure.

3.1. Extract XML schema and contents using XML DOM

This step uses path-based XML relational storage to extract and record information of each node in the XML tree document. Middleware relational storage uses 1-index method for normalizing the tree, as well as for reducing the amount of stored data. Thereafter, the technique classifies each attribute and reconstructs the extracted information after indexing into XML DOM structure tree, the goal is to access the information contents and represent the elements, attributes, and values of the document. Figure 2 and 3 clarify this step.

List 2 shows part of the code that reads "uni.xml" file that shown before; the code maps XML document "uni.xml" into XML DOM file "uniDom" and accesses all XML elements through the XML DOM.

List 2. Code of exacting XML content using DOM Tree

```
<script>
...
Var xdom = new XMLHttpRequest();
xdom.open("GET", " uni.xml", true);
xdom.send();
...
var uniDom = uni.responseXML;
var x = uniDom.getElementsByTagName("Name");

for (i = 0; i <x.length; i++) {
    var xdata = x[i].childNodes[0].nodeValue;
    ...
}
...
</script>
```

The code `getElementsByTagName ("Name")` returns all `<Name>` elements of the XML document, `length` property finds number of retrieved elements in order to loop through all the elements, and saves values of the extracted elements using `x[i]`. As for the code `childNodes[0].nodeValue` the `x[i]` returns the first `<Name>` element, `childNodes[0]` returns first child of the `<Name>` element, `nodeValue` returns the value of the node. The function `DOMParser` uses for retrieving the text string from the XML file and store it in the DOM tree as shown in the code below.

```
var parser = new DOMParser();
var uniDom = parser.parseFromString(text, "text/xml");
```

The algorithm uses some XML DOM attributes properties to navigate the content and schema of XML document, where `nodeName` attribute property extracts name of the element, `nodeValue` attribute property extracts value of the element, `parentNode` attribute property extracts parent element, and `childNodes` attribute property extracts child element.

3.2. Extract relational schema and contents using XML DOM

The technique extracts schema and instance of all tuples and transforms the table into object instances of XML documents using SQL query and database catalog, it reengineers schema and contents of the relational database. Al Hamad [16] proposed a conversion method from relational database into XML document using catalog-base. Fong [3] proposed a conversion technique from relational database into XML documents using XML DOM structure tree.

After transform the relational tables into XML document, the technique uses the same method that discussed in the point 3.1 above. In other words, the approach extracts schema and contents of the attributes from the tables and store them as XML documents then transforms them using XML DOM structure tree. The fixed middleware stores the tables using XML DOM into XML document, the technique initially extracts schema catalog and relation instance, then maps relation instance into XML document, thereafter transforms attributes and contents.

3.3. Transform XML DOM into the fixed middleware RXML Relation

As mentioned before, RXML middleware is a relational database stores XML information using XML DOM structure tree. Each tuple in the RXML relation represents node name, path, value, type, level and other additional information. All nodes will be stored in RXML relation at the same structure and contents. Each XML DOM document contains three types of the node: element, attribute, and values. RXML relation records all relevant information of each node and their sub-nodes. List 3 shows part of code that reads and navigates a XML DOM document "uniDom" and save the result in the RXML Middleware sequentially.

List 3. Code of navigating XML DOM data

```
<script>
...
var x = uniDom.documentElement.childNodes;
for (i = 0; i <x.length; i++) {
  xname = x[i].nodeName;
  xvalue = x[i].nodeValue;
}
...
</script>
```

To return name and value of parent node, we used `nodeName` and `nodeValue`. The algorithm uses length property code to loop through the child (element) of the root node, where `documentElement` property refers to root node. Some of XML DOM navigation properties that used by the algorithm to navigate parent, child and sibling nodes are `parentNode`, `childNodes`, `firstChild`, `lastChild`, `nextSibling`, and `previousSibling`.

In order to represent the relevant information of paths, elements, and attributes from all resources, the technique construct the RXML relation with eleven attributes fields (columns) to represent XML tuples. List 4 shows schema of RXML relation.

List 4: RXML Schema of XML document

```
RXML (DocID, DocName, DocPath, NodeId, Name, Path, Type, Value, ParentID, Level, IndexID)
```

The details of stored values of each RXML attribute are:

1. `DocID`: indicates to sequence number of XML document.
2. `DocName`: indicates to name of the XML document.
3. `DocPath`: indicates to path of the document starting root folder to the present folder.
4. `NodeId`: indicates to sequence number of each node that shows its order in document tree.
5. `Path`: indicates to path of node starting of root node to the current node.
6. `Name`: indicates to element or attribute name.
7. `Type`: indicates to type of node whether it is an element or an attribute.
8. `Value`: indicates to element or attribute value.
9. `ParentID`: indicates to the parent ID node of the element or attribute.
10. `Level`: indicates to the layer number of node location.
11. `IndexID`: indicates to the ID of combined nodes using the 1-index method.

In summary, the technique stores mapping structure and information of the XML DOM nodes. Primary key of the table is `DocID` because each document contains one XML DOM tree and RXML table represents many XML DOM documents, this key is not allowed to be empty. RXML includes all elements and attributes of the nodes, the relevant information of the elements and attributes are recorded into the table wither containing value or not. For example, root element and parent element (intermediate elements) have non-value and considered empty fields, as well as duplicate values in many attributes. This proposal would be more effective when normalize the RXML relation into four relations in order to eliminate the duplicate and empty fields as shown in list 5.

List 5: RXML relation after normalization

```
Doc (DocID, DocName, DocPath)
RXML (DocID, NodeId, PathID, Name, Type, ParentID, Level)
Value (DocID, NodeId, IndexID, Value)
Path (PathID, NodePath)
```

4. TECHNIQUE EVALUATION AND DISCUSSION

Many researches performed several experiments and proposed integration solutions between database models. Some techniques focused on querying XML files then storing the results as relational database, while others focused on converting and storing relational databases as XML files. This technique proposes a comprehensive solution by read XML documents and relational database, and reduce size of stored data, then store both of them in a middleware database.

Many experiments has performed for ten XML documents and ten relational tables, some of these experiments conducted using 1-index method; the algorithms has executed using a computer with 1.80GHz processor (Core i3-3217U) and 4GB memory operates by windows 10 operating system. The technique records the spent time for parsing and converting of relational database into XML documents until save the result in RXML middleware. Time of loading files and data is not considered. The technique repeated each experiment three time and calculated the mean value of time to obtain more precision results. XML DOM tree has constructed based on W3C recommendation. Figure 4 shows the average CPU time of the convention in millisecond.

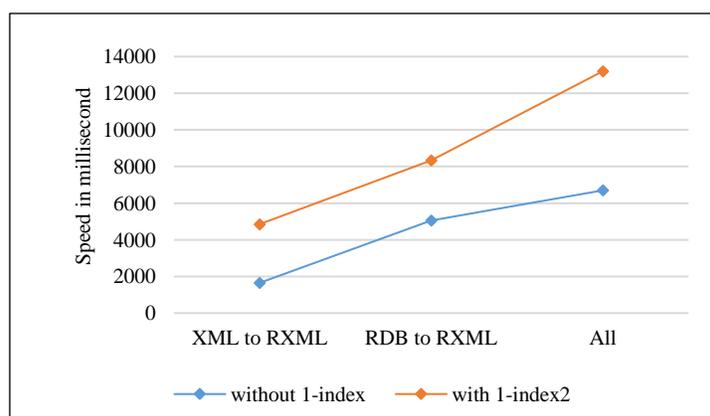


Figure 4. Performance of convention data into RXML.

5. CONCLUSIONS

We presented in this research a novel middleware data for integrating between relational tables and XML documents using two well-known XML parsers 1-index and XML DOM, the technique stores integrated data into a specific unified design of RXML relational database. Respectively, while 1-index normalizes the XML tree, XML DOM reconstructs data as object instances. The technique used path-based storage, catalog based technique, XML DOM, and 1-index method for converting the original relational databases and XML documents. Several experiments has compared and performed to measure and report precision and speed of the algorithm.

ACKNOWLEDGMENT

The author would like to acknowledge that the financial support provided by the Deanship of Scientific Research at Amman Arab University, under research project reference number (R.D./1/2/185), entitled, "Web-Based Model for Conversion between Relational Databases and XML Language".

REFERENCES

- [1] Qian L., (2010) "Research and implementation of XML-based data transformation for heterogeneous Database", Master's Thesis, Shenyang University of Technology.
- [2] Bin L., Xin Z., Zhongliang D., (2011) "Database Conversion Based on Relationship Schema Mapping", Internet Technology and Applications (iTAP), pp. 1-5.
- [3] Fong, J., Wonga, H.K., Chengb, Z., (2003) "Converting relational database into XML documents with DOM", Information and Software Technology Journal, Vol. 45, pp.335– 355.
- [4] Al Hamad H. A., (2017) "A literature Survey on Conversion between Relational and XML Models", 2nd International Computer Sciences and Informatics Conference (ICSIC 2017), Jordan, pp.217-222.
- [5] Wang Q., Ren Z., Dong L. and Sheng Z., (2012) "Path-based XML Relational Storage Approach", International Conference on Medical Physics and Biomedical Engineering.
- [6] Feng, Y. and Jingsheng, X., (2009) "Mapping XML DTD to Relational Schema", Database Technology and Applications, First International Workshop, pp.557-560.
- [7] Teng L., and Ping Y., (2007) "Schema Conversion from Relation to XML with Semantic Constraints", Fuzzy Systems and Knowledge Discovery, FSKD 2007. Fourth International Conference, pp. 619-623.
- [8] Feng, Y., (2009) "Converting XML DTD to Database", Intelligent Systems and Applications, ISA International Workshop, pp.1-4.
- [9] Karthigal D., Gunasekaran S., (2013) "Optimization of Query Processing in XML Document Using Association and Path Based Indexing", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, No. 2.
- [10] Chang Y., Iee C., and Iee W., (2007) "A Path-based Approach for Efficient Evaluation of Twig Queries over XML Data", Jordan of Information Science and Engineering, Vol. 23, 1523-1540.
- [11] Böttcher S., Hartel R., Wolters D., (2016) "S2CX: From relational data via SQL/XML to (Un-) Compressed XML", Information Systems Vol. 56, pp. 198–213.
- [12] Wang T., Ling, Zeng Z., Le T, and Lee M., (2016) "ORA-Semantics based Keyword Search in XML and Relational Databases", ICDE Workshops.
- [13] Al Hamad, H. A., (2015) "XML-Based Data Exchange in the Heterogeneous Databases (XDEHD)", International Journal of Web & Semantic Technology (IJWesT), Vol. 6, No. 3, pp.11-24.
- [14] Machkour M., Afdel K, Khamlichi Y., (2016) "A reversible conversion methodology: Between XML and object-relational models", 7th International Conference on Information and Communication Systems (ICICS).
- [15] Chunyan W., Lo A., Alhadj R., Barker K., (2005) "Novel Approach for Reengineering Relational Databases into XML", 21st International Conference of Data Engineering Workshops, pp. 1284.
- [16] Al Hamad, H.A., (2015) "Catalog-based Conversion from Relational Database into XML Schema (XSD)", International Journal of Date Engineering (IJDE), Vol. 6, No. 2, pp. 9-22.
- [17] Fong, J. and Cheung, S., (2001) "Translating relational schema into XML schema definition with data semantic preservation and XSD graph", Information and Software Technology, Vol. 47, No. 7, pp. 437-462.
- [18] Zhang, Y. and Liu, W., (2002) "Semantic integration of XML Schema", International Conference of Machine Learning and Cybernetics, pp. 1058-1061.
- [19] Tzvetkov, V. and Xiong W., (2005) "DBXML - Connecting XML with Relational Databases", The Fifth International Conference of Computer and Information Technology (CIT), pp. 130-135.

- [20] Hong, S. and Song, Y., (2007) "Efficient XML query using Relational Data Model", Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD, Eighth ACIS International Conference, pp. 1095-1100.
- [21] Zhou, A., Lu, H., Zheng, S., (2001) Liang, Y., Zhang, L., W. Ji, Tian, Z., "VXMLR: a visual XML-relational database system", Proceedings of the 27th International Conference on Very Large Data Bases, pp. 719-720.
- [22] Oracle, XML DB Developer's Guide, <https://docs.oracle.com/database/121/ADXDB/toc.htm>.
- [23] Oracle, (2013) "Oracle XML DB: Best Practices to Get Optimal Performance out of XML Queries", Oracle White Paper January, <http://www.oracle.com/technetwork/database-features/xmldb/xmlqueryoptimize11gr2-168036.pdf>
- [24] Oracle, (2017) "Oracle XML DB in Oracle Database 12c Release 2, Oracle White Paper January", <http://www.oracle.com/technetwork/database-features/xmldb/overview/xmldb-twp-12cr1-1964803.pdf>
- [25] Wong K., Yu J. & Tang N., (2006) "Answering XML Queries Using Path-Based Indexes: A Survey", World Wide Web Journal, Vol. 9, pp. 277-299.

AUTHORS

Dr. Husam Al Hamad, Assistant Professor of Computer Information Systems at Amman Arab University, in Amman, Jordan. He has got the BS, MS, and Ph.D. in the fields of Computer Science and Information Systems in 2001, 2003, and 2008 respectively. His research interests are data engineering, pattern recognition, image processing, artificial intelligence, and neural networks. hhamad@aau.edu.jo, hushamad@yahoo.com

