Converting Relational Database into XML Schema and Document

Myint Myint Lwin, Thi Thi Soe Nyunt, Yuzana
University of Computer Studies, Yangon
lwin.myintmyint@gmail.com, thithisn@gmail.com, yuzana.yzn@gmail.com

Abstract

Today's world is mainly based on Internet and Web is the key medium of data exchange. Web services based on Extensible Markup Language (XML) technology. XML has emerged as the leading medium for data transfer over the World Wide Web due to its flexible nature and ease of implementation. Relational Database drives most business of any size today. A majority of data is still stored and maintained in relational database. Nevertheless, relational database cannot meet all the demands of electronic business because it processes data independently of the context. To overcome the problem of data exchanging between different platforms, converting from relational database (RDB) to XML becomes a popular field in research trends. The proposed system converts the relational database to XML by grouping the common attributes in the relational database to get the qualified XML schema. The string matching algorithms are proposed and applied to group the common attributes in RDB. The proposed system also solves the multi-parent relationship problem to avoid the data redundancy. The generated XML schema is more modular; more understand for the human readers and reduces the maintainability efforts.

1. Introduction

Since Extensible Markup Language (XML) was introduced in the late 90s, it has unleashed a torrent of new acronyms, standards, and rules that have left some in the Internet community wondering whether it is all really necessary. The standard process will figure prominently in the growth of this information revolution. XML itself is an attempt to rein in the uncontrolled development of competing technologies and proprietary languages that threaten to support the Web. XML creates a playground where structured information can play nicely with applications, maximizing accessibility without sacrificing richness of expression.

The use of XML is the common format for representing, exchanging, storing, and accessing data to database systems. Since the majority of everyday data is still stored and maintained in relational database systems, an increase in demand for tools to convert from relational databases to XML.

Consequently, if XML is to fulfill its potential, some mechanism is needed to convert relational data in the form of XML documents [9]. Given a language specification for converting relational tables to XML documents, an implementation to carry out the conversion raises many challenges. Relational tables are flat, while XML documents are tagged, hierarchical and graph-structured. On the other hand, the effective and proper implementation of XML in diverse domains requires well-designed XML schemas [2]. The design of XML schema plays an important role in the software development process and needs to be quantified for the ease of maintainability because XML schemas have been used in diverse fields of software industry and have been playing an important role in many such projects.

As a result, the important things and quality factors of XML schema are considered carefully in the RDB to XML conversion process. There are many things to fulfill in the generated XML schema. They are highly nested structure, schema code modularity, user understandability and maintainability effort.

However, many existing RDB to XML conversion methods do not fulfill the XML quality factors. Therefore, a new relational database to XML conversion method is proposed which provides the schema code modularity and the highly nested structure. Therefore, the three string matching algorithms: Non-consecutive string matching algorithm, Maximum consecutive string matching at right algorithm and Maximum consecutive string matching at any algorithm [11] are proposed to provide the schema code modularity. The consequence of the schema code modularity provides more understandability the developers and users. Moreover, it reduces the maintainability efforts. The proposed conversion method also considers the multiple relationships case in the relational database when converts the relational database into the XML by determining the maximum referencing times to the child relation. The quality of the generated XML schema is measured by the XML schema quality measurements.

2. Motivation

Nowadays, Web applications are interested in most business and organizations to exchange data. The XML is an essential role in Web based system. The quality measurements of the software development are popular to obtain the excellent software result. XML quality measurement metrics have also been developed to measure the XML schema quality. Therefore, the XML schema quality factors such as code modularity and user understandability should be emphasized when converts the relational database into XML.

The first important factor of generated XML schema in RDB to XML conversion is the highly nested structured. All of the RDB to XML conversion methods tried to obtain this factor with different techniques. However, some existing RDB to XML conversion methods have weaknesses in nested structure. The proposed conversion method considers the relation with the multiple parent relationship to nest in the XML schema. The previous conversion methods did not consider relation with the multi-parent relationship. When the multiple parent relationship occurs, the best parent is required to choose for the child XML element. To solve the problem, the proposed method applies the idea of maximum referencing times to choose the best parent of the child relation.

The next factor is the XML schema code modularity. Most of the relational databases collect the related relations which have the common behaviors and common features. For example, the university database includes student table, teacher table and staff table etc because they have same attributes such as name, phone, address, etc. However, their common attribute names have some variations such as SNAME, SADDRESS, SPHONE, TNAME, TADDRESS due to the database designers. These attribute names are variation from the NAME, PHONE and ADDRESS attributes. The proposed conversion method wants to group the common attributes and detect them by using string similarity algorithm to obtain the schema code modularity.

The previous RDB to Conversion methods generate various outputs based on their framework. Some methods produce only XML schema, only XML document using the user SQL query or XQuery, XML document for one table, and XML document for multiple tables with flat translation structure. Some methods are expert dependent, semi-automatic and fully automatic conversion. However, no RDB to XML conversion for the whole database automatically.

Therefore, the automatic relational database to XML conversion method is required to provide the technology changes and user demands.

3. Related Works

Database schema conversion is the process of converting between different models. The goal of schema conversion is to find the most common things between source and destination and convert the source format to target format.

As the volume of the XML data increasing, the conversions between XML and different sources are becoming popular tasks in research fields. There are two conversion approaches to XML: non-relational model to XML conversion and relational model to XML conversion. The first non-relational model to XML is UML to XML conversion. Dominguez et.al [6] presented the feature-based approaches and element-based approaches which widely emphasize on class, attribute, association, generation and built-in data type. Most of the feature-based approaches used meta model approach such as UML transformation. Krumbein and Kudrass [9] proposed an algorithm which implements set of rules using XSLT stylesheets that transform the UML class diagram into an adequate XML Schema definition language (XSD). The next one is Object-Oriented Database to XML and a novel method for OODB to XML conversion [5] is proposed using the object graph which includes inheritance as the intermediate process.

DB2's XML Extender [7] provides the ability to store and access XML documents, to generate XML documents from existing relational data, and to insert rows into relational tables from XML documents. XML Extender provides new data types, functions, and stored procedures to manage XML data in DB2. Tzvetkov et.al [18] introduced a software system, called DBXML that provides a transformation utility to connect XML with relational databases. The transformation utility can convert data both ways from XML-Schema to relational database schema and from relational database schema to XML-Schema. DB2XML uses an algorithm similar to Flat Translation. However, it allows user to choose table or tables to generate XML format and lacks the nested form. Andez et.al proposed SilkRoute [1]. In SilkRout, relational data is published in XML in three steps. First, the relational tables are presented to the database administrator in a canonical XML view.

XPERANTO [4] is introduced by Carey et.al, the success of the conversion is closely related with the quality of the target XML schema onto which a given input relational is mapped.

Lo, Alhajj and Barker [11] developed a user-friendly transformation tool called VIREX (Visual Relational to XML) which facilitates converting a selected portion of a given underlying relational database into XML. It extracts the required catalogue information by analyzing the underlying the database content. From the catalog information, whether available or extracted, VIREX derives and displays on the screen a graph similar to the entity-relationship diagram.

However, the mapping from the relational schema to the XML schema is specified by human experts. Therefore, when large amount of relational schemas and data need to be translated into XML documents, a significant investment of human expert effort is required to initially design target schema. Moreover, all of the above tools require the user to specify the mapping from the given relational schema to XML schema.

4. The Proposed System Architecture

The proposed system intends to get the highly nested structure, to avoid the data redundancy, to provide the schema modularity and user understandability. The architecture of proposed system is shown in Figure 1.

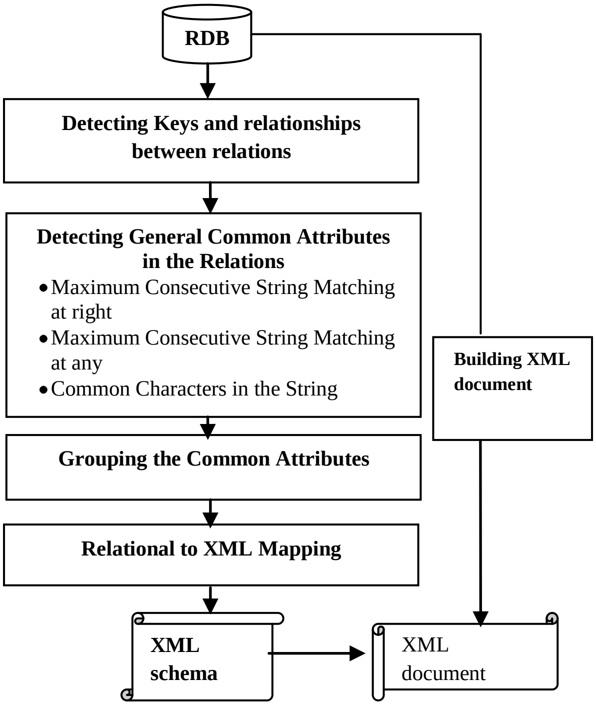
RDB
Detecting Keys and relationships between relations
Detecting General Common Attributes in the Relations
• Maximum Consecutive String Matching at right
• Maximum Consecutive String Matching at any
• Common Characters in the String
Grouping the Common Attributes
Relational to XML Mapping
Building XML document
XML schema
XML document

Figure 1: Architecture of the Proposed System

The proposed system includes the five main processes to convert from the relational database into XML. The relational database is taken as input to convert the relational database into XML. The proposed system produces both XML schema and data document.

Firstly, keys of the relations are detected to find the nested structure. The nested structure is built as the referential integrity constraints. In this step, the proposed system extracts the relationships between relationships which are based on the keys. The proposed system intends to avoid the data redundancy in the XML data document. The data redundancy is introduced by the multiple relationships to the single child table.

Therefore, the proposed system uses the algorithms to find the multi-parents relationships and solve the multi-parents relationship which are already introduced in my previous research work [15].The most suitable for the child relation is chosen by the maximum referencing times to the child relation.

The next process is the detecting the common attributes in the relations. In the relational database, many common attributes are included in some relations because they are collection of the related relations. However, these attributes varied due to the developers assumptions. Example, SNAME and SUPNAME are the NAME attribute. They are generally same, but they are different in relations. To detect these common attributes, the proposed system uses three string matching algorithms [12] to extract the common attributes in all relations. They are Maximum Consecutive String Matching at Right, Maximum consecutive String Matching at any and Non-Consecutive Common Character of two input strings.

Maximum Consecutive String Matching at Right is an algorithm which takes two relational attributes and finds the consecutive sub string in the two input strings. It is started at the right side of the input strings. Maximum Consecutive String Matching at any algorithm extracts the common substring from the two input strings. However, it finds the maximum common sub string which can be found in any position of input strings. The final string matching algorithm is the Non-consecutive Common String and it detects the common characters in the strings but they are not consecutive. The detected common sub strings are required to normalize and to calculate the total similarity values. The equation 1 is proposed by [8] and it is used to calculate the total similarity values of the attributes.

$$\alpha = w_1v_1 + w_2v_2 + w_3v_3 \tag{1}$$

where α is the similarity value of two strings. Then, w_1, w_2, w_3 are weights of each normalized value and $w_1+w_2+w_3=1$. The similarities of all of the attributes in the whole relational database are calculated by applying the string similarity and normalized equations. Then the common attributes are collected which are satisfied with the threshold value 0.5. The proposed system applies threshold value 0.5 by analyzing the eight relational databases. The analysis result is shown in Table 1.

Table 1. Analysis of the Threshold Values

Threshold		0.4		0.5		0.6	
RDB	Total common attribute	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
Testing 1	6	6	1	6	0	6	0
Testing 2	13	13	0	13	0	13	0
Testing 3	2	2	1	2	0	2	0
Testing 4	4	4	1	4	0	4	0
Testing 5	3	3	1	3	0	3	0
Testing 6	3	3	1	3	0	3	0
Testing 7	10	10	4	10	0	10	0
Testing 8	2	2	0	2	0	2	0

The next step is the grouping the detected common attributes. In this step, the common attributes are grouped as the XML global element groups. The usage of global element in the XML schema provides the schema code modularity.

The next process is the mapping the relational to XML. In this process, the relational tables are converted as the Complex type and attributes in the tables are created as the sub elements. Then, the proposed system generates the XML schema which has the code modularity, user understandability and reduces the maintainability efforts. Finally, the proposed system converts all relational data into the XML data by generating the XML data document. It is built according to rules of generated XML schema document. The XML document avoids the data redundancy.

5. Experimental Results

The output XML schema of the proposed system is measured by XML quality measurement metrics. The first method is the count-based metric and which is introduced by A. McDowell and C. Schmidt [16]. It is the common approach to measure the XML schema quality and counts the number of the schema components. To show the experimental result, the Northwind relational database is used which is the standard database of Microsoft access database. It includes 8 tables and 78 attributes. The comparison of complexity measurement is shown in Figure 2.

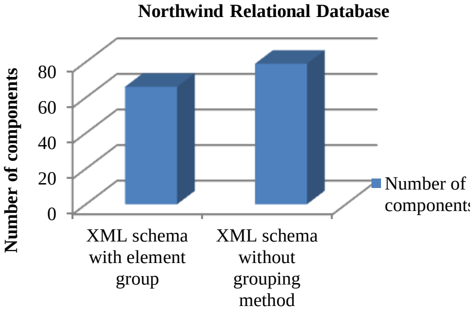

Figure 2.Complexity Measurement of Count based Method

The complexity of the proposed method is smaller than the complexity of XML schema without using common attributes.

The second measurement method is complexity measurement metric and which is proposed by [3]. This metric calculates the total of global element and attributes for referencing and no referencing, local elements and attributes for referencing and no referencing.

The comparison of the complexity of generated XML schema from the proposed system and XML schema without using common attributes is shown in Figure 3.

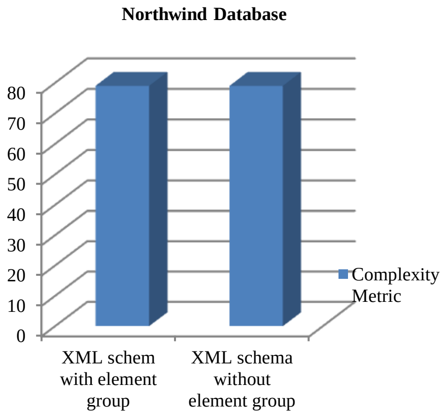

Figure 3. Comparison between the complexity of the generated XML schema and XML schema without element group

According to the Figure 3, the complexity values of both XML schemas are the same because the complexity measurement metric calculates the local element counts and global or referencing element counts. Nevertheless, the complexity of generated XML schema is not greater than the other XML schema.

The final XML quality measurement is Schema Entropy (SE) metric [2]. It is proposed by D. Basci and S. Misra. It is based on the entropy concept to measure the XML schema quality.

It said that the side effect of increasing number of reuse of same components is that increasing number of affected components that use the same reusable component. It also suggests that increasing reusability in XSD components may result in increasing complexity due to increasing number of affected components. The comparison of complexity for the generated XML schema and other method without group elements is described in Figure 4.

The quality of the generated XML schema is more than the XML schema without grouping common attributes. The increasing the SE value is directly proportional to the XML quality.

The SE metric provides more information about the understandability of the Schema documents. It is obvious that less understandable Schemas require more maintenance efforts. Therefore, the SE provides valuable information about maintainability. The SE metric may also be used to reflect the reusability of Schema components.

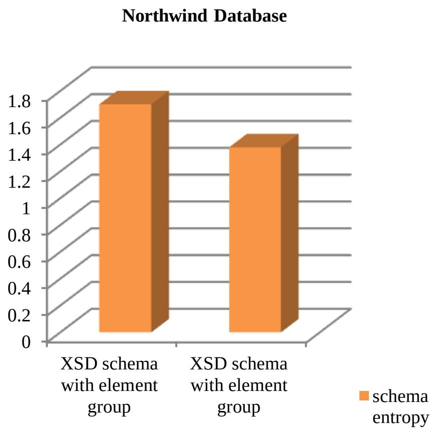

Figure 4. Comparison between the Schema Entropy of generated XML schema and XML Schema without grouping common attributes

The processing time of the proposed method is compared to XML schema which does not use XML global element groups. The processing time comparison is shown in Figure 5.

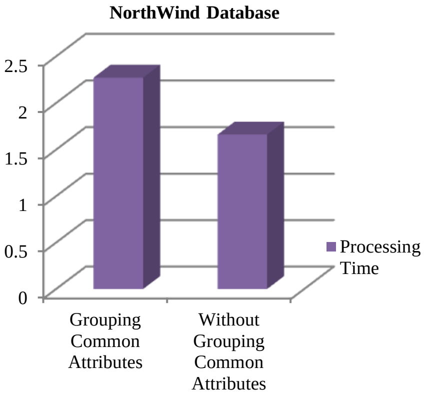

Figure 5. Comparison between processing time of generated XML schema and XML schema without grouping common attributes

The processing time of the proposed conversion method is greater than XML schema which does not use XML global element group. But the time of grouping the common attributes is very little for the processing time because it is about 1.1 second for the 78 attributes. The maintainability effort is very important for the software development and the grouping time can be ignored for the future software development.

6. Conclusion

This paper proposes a method for RDB to XML. It generates the both XML schema document with global common attributes and XML data document. It converts the relational database into XML automatically. The quality of the XML schema document is measured by three XML quality measurements. The generated XML schema provides code modularity, user understandability and reduces the maintainability efforts.

References

[1] M.F. Andez, Y. Kadiyska and D. Suciu, A. Morishima, W.C. TAN, “SilkRoute : A Framework for Publishing Relational Data in XML”, ACM Transactions on Database

Systems, Vol. 27, No. 4, pp. 1–55, December 2002.

[2] D. Basci and S. Misra, Entropy as a Measure of Quality of XML Schema Document, The International Arab Journal of Information Technology, Vol. 8, No. 1, pp. 16-24, January 2011.

[3] D.Basci and S.Misra, “Measuring and Evaluating a Design Complexity Metric”, Journal of Information Science and Engineering, vol. 25, No 5, pp 1405-1425, 2009.

[4] M.Carey, J. Kiernan, J. Shanmugasundaram, E. Shekita, and S. Subramanian, XPERANTO: Middleware for Publishing Object-Relational Data as XML Documents, Proceedings of the 26th International Conference on Very Large Data Bases, Pages 646-648, 2000.

[5] C.Desai , M.Patil and S. Shinde, “Representing object oriented database using XML-DTD”, Advances in Informatin Mining, Vol 1, Issue 2, pp 11-13, 2009.

[6] E.Dominguez, J.Lloret, B.Perez, A.Rodrguez, A..L. Rubio, and M.A. Zapata, “A Survey of UML Models to XML Schemas Transformations”, Web Information Systems Engineering (WISE) ,Lecture Notes in Computer Science Volume 4831, 2007, pp 184-195.

[7] Introduction to XML Extender, http://www.ibm.com/support/docview.wss?rs=71&uid=swg21370360

[8] A. Islam,D.Inkpen, I. Kiringa, Applications of corpus-based semantic similarity and word, The International Journal on Very Large Data Bases, volume 17, No. 5, pp 1293-1320, Aug 2008.

[9] T.Krumbein and T. Kudrass, “Rule-Based Generation of DTDs from UML Class Diagrams”, Advances in Database and Information Systems, LNCS Volume 2798, pp 339-354, 2003.

[10] D. Lee, M. Mani and W. W. Chu, “Effective Schema Conversions between XML and Relational Models”, In Proc. European Conf. on Artificial Intelligence (ECAI), Knowledge Transformation Workshop, July 2002.

[11] A. Lo, R. Alhajjand K. Barker, “VIREX: Visual Relational to XML Conversion Tool”, Visual Languages and Computing, Vol. 17, No. 1, pp. 25-45, 2006.

[12] M.M. Lwin, T. T. Soe Nyunt, Yuzana, “Generating the Good XML Schema from Relational Database by using String Matching Algorithms”, 10th International Conference on Computer Applications (ICCA), 28th-29th February, 2012, pp. 357-361.

[13] M.M. Lwin, T. T. Soe Nyunt, Yuzana, “Converting Relational to Qualified XML Schema with Referential Integrity Constraint”, 11th International Conference on Applications (ICCA), 26th-27th February, 2013, pp 114-118.

[14] M.M. Lwin, T. T. Soe Nyunt, Yuzana, “Providing a Way for Good XML Schema Design in RDB to XML Conversion”, International Conference on Information and Communication Technology for Education (ICTE 2013), 1st -2nd December, 2013.

[15] M.M. Lwin, T. T. Soe Nyunt, Yuzana, Nesting Idea for Relation with Multi-Parent in RDB to XML Conversion”, 1st International Conference on Energy, Environment and Human Engineering, 21st-23rd December, 2013.

[16] A. McDowell, C.Schmidt, K.B. Yue, “Analysis and Metrics of XML Schema”, In the Proceedings of the International Conference on Software Engineering Research and Practice, pp. 538-544, 2004.

[17] J. Shanmugasundaram, E. Shekita, R. Barr, M. Carey, B. Lindsay, H. Pira-hesh, and B. Reinwald, “Efficiently Publishing Relational Data as XML Documents”, In the proceedings of the 26th International Conference on very large data Bases, 2000, pp. 65–76.

[18] V. Tzvetkov and X. Wang, “DBXML-Connecting XML with Relational Databases”, in Proceedings of the 5th International Conference on Computer and Information Technology , Washington, DC, USA, 2005, pp. 130-135.