Providing a way for qualified XML schema design in RDB to XML conversion

Myint Myint Lwin, Thi Thi Soe Nyunt, Yuzana
University of Computer Studies, Yangon, Myanmar

Abstract

Extensible Markup Language (XML) is the standard format for the data exchanging over the Web. It is also compatible for Web-based application and the volume of XML data is increasing day by day. On the other hand, most of the data are still storing in the traditional database such as relational database (RDB). As a result, most of relational data are required to convert into XML format to be applicable to Web Application. The efficient RDB to XML conversion methods are essential and XML schema codes need to be qualified in today world. This paper provides a way of qualified XML schema design in RDB to XML conversion method by grouping the generally common attributes in the RDB. The resulted XML schema provides more modularity, more understandability for users and reduces maintainability efforts.

Keywords: relational database, XML schema, string similarity.

1 Introduction

Nowadays, Web applications are popular and most of the business organizations try to involve in the Web. However, most of the data are stored in RDB which are required to convert into Extensible Markup Language (XML) format to achieve the data exchanging over the Web. Therefore, the converting relational data to XML format becomes popular and interested field in today research trends. Many RDB to XML conversion methods have been proposed by many researchers with their point of views such as structural or semantic. But the conversion process is nontrivial task and need to be tedious to obtain the qualified results. First, choosing the suitable schema language for both source and target format. In the XML world, many schema languages such as document type definition (DTD), W3C schema language, and Schematron are available to

WIT Transactions on Information and Communication Technologies, Vol. 58, © 2014 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/ICTE130521

express the schema document. Consequently, the right choice of schema language is also important factor in the conversion process to maintain the original constraints of relational database (RDB). Among the schema language, W3C schema language is most suitable language for the RDB because it can provide the domain constraints of RDB especially data type for user defined data type.

Another factor is the converted XML schema to be satisfied with the XML quality factors—maintainability, user understandability, simplicity, code modularity, and integrate ability. In the investigation experience, most of the conversion method lacks the schema code modularity which will reduce the maintainability effort and user understandability. Therefore, this paper provides a way for these features by grouping the common attributes in the RDB. To achieve this task, the string matching algorithms: common characters in the string, maximum consecutive at right and maximum consecutive at any position in strings are proposed and applied to collect the most similar attributes. Then, the detected common attributes are converted into XML global element groups. The tables and attributes are converted into the associated XML format and finally the data in RDB are built into the XML document by obeying the rules of generated XML schema document.

This paper presents about introduction in Section 1. Section 2 describes the related works of the proposed method. Section 3 presents the architecture of proposed system. The experimental results of the proposed system are presented in Section 4. Finally, conclusion is given in Section 5.

2 Related works

RDB to XML conversion is essential in research trends because the volume of information within the today world is staggering, but the limitations of existing technology can make it difficult to access [1]. Many RDB to XML conversion methods are described in Ref. [9, 10] with the structural and semantic points of views. The earliest conversion method is flat translation (FT) [3] and each relation is converted as element and each attribute of the relation is either converted as subelement or attribute. It is the simplest conversion method but it does not consider the nesting idea and also did not present the relationship between the relations. Therefore, it is not an efficient conversion method. The nesting based translation (NeT) [3] was proposed to overcome nesting problem. It applied the nested structured by nested operator such as "*" and "+". That method is better than FT and useful for decreasing data redundancy. However, it considers one table at a time and does not cover for the whole RDB. It also does not include the relationship of all tables. Both FT and NeT are structurally conversion method and they did not consider the semantic aspects. Constraints-based translation [4] method was developed to solve the problem which occurred in NeT. It applied the inclusion dependency of relational schema which is based on the foreign key constraints. It is mostly associated with the usage of subelement and IDREF attributes for translation purpose. Moreover, it considers not only the structural part such as tables and columns but also the semantic part such as the constraints and referential integrity (RI). But it can only provide the explicit RI. If the implic-

it RI exists, it cannot extract RI and cannot generate the exact XML document. The ConvRel algorithm [5] detects the relationship between tables and extracts the RI by applying the idea of parent–child relationship. It also provide the N:M relationships modeled as a combination between a nested structure and keyref. All of the above translation methods did not consider code modularity, maintainability effort, and user understandability. When the databases are larger, some of the relations may have the common attributes. It may introduce the complex for the developers and less modularity in schema code.

This paper presents new RDB to XML conversion method with grouping the common attributes in a database. To achieve this purpose, string matching algorithms are applied to detect the common attributes in the relations. Final output of the proposed system is XML document with schema file. The results can provide the important XML quality factors.

3 The proposed system

The overall architecture of the proposed system is shown in Figure 1. The RDB is taken as input to convert into XML format. First, the keys and constraints of the database are investigated to keep them in the resulted XML schema.

The domain constraints and RI constraints are essential for both RDB and XML. In the proposed system, foreign keys constraints are mainly used to provide the RI constraints in the generated XML schema. To achieve the highly nested structure, the idea of referenced relations and referencing relations are defined as a tuple t1 in relation R1 said to reference a tuple t2 in relation R2 if t1[FK]=t2[PK].Then it is applicable to define the child and parent relation in RDB and outer or inner element in the XML schema.

The next process is to detect the common attributes in the database which is mainly emphasized in this paper. According to our previous investigation, some of the database design has the common attribute which can be detected and

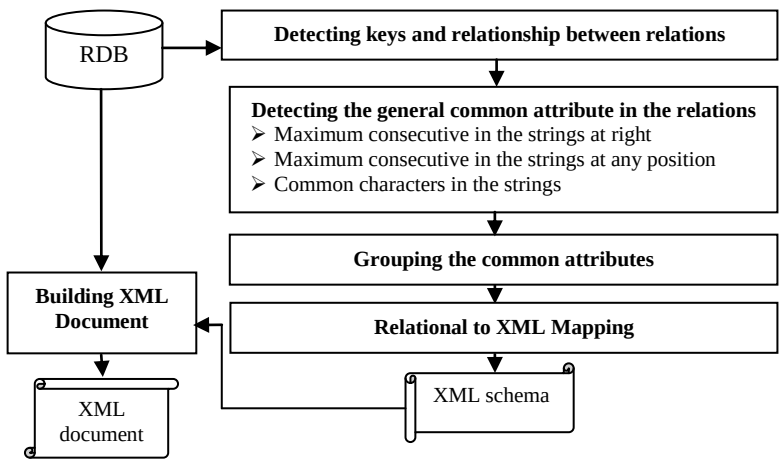


Figure 1: Architecture of the proposed system.

Algorithm 1: Maximum consecutive in the strings at right

```
Input  :S1, S2//two strings to compare
Output :SMCSn (ri) //maximum consecutive substring
Begin
    d ←| S1 |,  t ← | S2 |
    If  d > t then
        ri ← S2 , sj ←  S1
    Else
        ri ←S1 , sj  ← S2
    End if
    While  | ri | ≥ 1
        If  ri ∈ sj;  that is sj ∩ ri = ri
                Return ri
        Else
            ri ← ri \ r1  that is, remove
    the  left-most character from ri
        End if
    End while
End
```

(a)

Algorithm 2: Common characters in the strings

```
Input  : S1, S2 // two strings to compare
Output : Sc //common any character but not consecutive

Begin
    i ← 0
    While  (i < S1.length and S2.length>0)
        If  S1i ⊆ S2 then
            Sc ← Sc.concat(S1i)
            S2 ← S2\S1i
        End if
      i ← i+1
    End while
 Return Sc
End
```

(b)

Figure 2: (a) Maximum consecutive in the strings at right and (b) common characters in the strings.

grouped as the element group. In practical, attributes name of the relations is randomly defined by the database designers. As a result, these attributes may be same identically or partially. Therefore, the string matching algorithms (maximum consecutive in the strings at right, maximum consecutive at any position in the strings and common characters in the strings) are proposed in our previous work [2] to successfully detect these common attributes. The string matching algorithms are described as Figures (2) and (3). First, maximum consecutive in the strings from right is essential for detecting common attributes because most of the common characters are same at the right part of the attributes, for example, Sname, jname, and pname. All algorithms take two attributes from different relations as input and produce the associated outputs.

To get the total similarity values, the proposed system uses the normalized method which is proposed in Ref. [6] and the similar attributes are normalized using the following normalized equations to get the accurate similarity values. Equation (1) is applied for normalizing the similar attributes that are produced by algorithm 1 and NS is the normalized values, r_i and s_j are input attributes.

$$\mathbf{v_1} = \mathbf{NS_{MCSn}}(\mathbf{r_i}, \mathbf{s_j}) = \frac{\{\text{length}(S_{\text{MCS}n}(ri, sj))\}^2}{\text{length}(r_i) \times \text{length}(s_j)} \tag{1}$$

Equation (2) is used for normalizing the similar attributes which are produced by algorithm 2.

```
Algorithm 3: maximum consecutive in the strings at any portion
Input      : S1, S2 // two strings to compare
Output     : S_MCSany (Max)
 Begin
    Max ← ϕ, i ← 0
    If  S1.length < S2.length  then
          pattern ← S1
          target  ← S2
    Else
          pattern  ← S2
          target   ← S1
    End if
    While ( i < target. length)
          C ← pattern.Char(i)
          tempMax ← tempMax.concat (C)
      If tempMax ⊆ target  then
        If tempMax.length > Max.length  then
             Max ← tempMax
        End if
      Else
          tempMax  ← tempMax \ left-most-character
          i ← i+1
      End if
     End while
     Return Max
 End
```

Figure 3: Maximum consecutive in the strings at any position.

$$\mathbf{v_2} = \mathbf{NS_c}(\mathbf{r_i}, \mathbf{s_j}) = \frac{\{\text{length}(S_c(r_i, s_j))\}^2}{\text{length}(r_i) \times \text{length}(s_j)} \tag{2}$$

Equation (3) is used for normalizing the similar attributes that are produced by algorithm 3.

$$\mathbf{v_3} = \mathbf{NS_{MCSany}}(\mathbf{r_i}, \mathbf{s_j}) = \frac{\{\text{length}(S_{\text{MCSany}}(r_i, s_j))\}^2}{\text{length}(r_i) \times \text{length}(s_j)} \tag{3}$$

After normalization the strings, the normalized values are evaluated by using Eq. (4) to obtain the total similarity value.

$$\alpha = w_1 v_1 + w_2 v_2 + w_3 v_3 \tag{4}$$

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 RDB 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. They are grouped as the XML global element group to utilize in the generated XML schema.

The final process is mapping the relational to the XML format. In the proposed system, all tables are converted into the XML complex types. The attributes in the relation are converted into the subelement of associated complex type. The relationships of tables are created as nested structure according to the nested lists which are investigated by the referential key constraints. Some of the common attributes are converted as common substring structure. For example, jname is converted into name. Finally, the relational data are converted into the XML document according to the generated XML schema document.

4 Experimental result

In the experimental results, the proposed method is more suitable for the RDB with having common attributes. Therefore, if the more common attribute are detected, the proposed method can reduce the number of tags in the output schema document. There are 8 relational databases are used and experimental result of detecting the common attributes in the tested RDB is described in Figure 4.

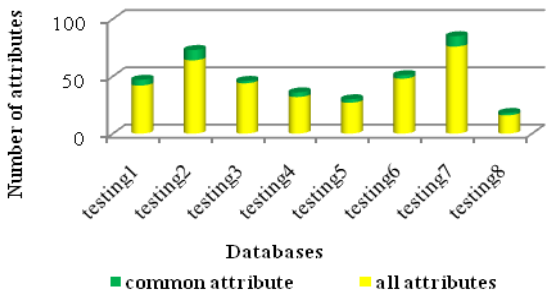


Figure 4: Results of detecting the common attributes in tested RDB.

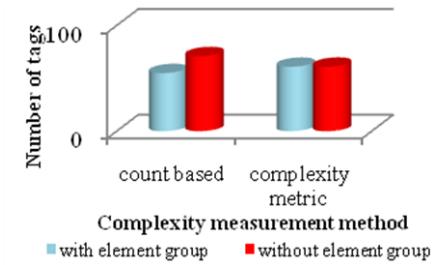
The next experimental result is the nested idea of RDB and is shown with sample database (SPJ). It is one of the tested RDBs in Figure 4. The schema file is taken as input and the outputs are shown in Table 1.

SUPPLIER (S#, SNAME, STATUS, CITY)
SPJ (S#, P#, J#, QTY)
PROJECT (J#, JNAME, CITY)
PART (P#, PNAME, COLOR, WEIGHT, CITY)

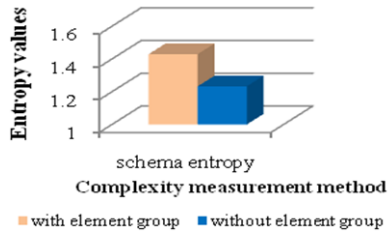
In Table 1, the table list is all tables in the example RDB. The nested list shows the outer: inner format. According to the nested list, 1:3 means part table is the outer element of spj table. Common attributes list describes the common attribute and shared table list. For example, name—1,2,4 means attribute "name" is common attribute and it is contained in part, project and supplier table.

Table 1: Output of Example SPJ RDB.

	Output
Table list	[1.part, 2. project, 3. spj, 4. supplier]
Nested list	[1:3, 2:3, 4:3]
Common attributes list (or) element group	name—1,2,4 city—1,2,4



(a)



(b)

Figure 5: (a) Complexity measurement of the resulted XML with count based and complexity metric (b): Complexity measurement of the resulted XML schema with schema entropy.

4.1 XML schema quality measurement methods

Many complexity measurement methods can be used to measure the quality of the XML schema documents (DTD or XSD). In this paper, the target XML schema is written with the XSD schema language and only XSD complexity measurement methods are used to show the complexity of the resulted XML schema. The count-based method is the earliest method which counts the number of element or attributes. The complexity metric [7] and schema entropy (SE) [8] are used to prove the quality of the resulted XML schema document.

4.2 Comparison of the complexity measurement of the output XML schema

According to the above experimental results, Figure 5(a) shows that the proposed system reduces number of element tags in the XML schema and (b) also shows that the schema entropy value of the proposed system is greater than other schema

without using element group. The SE values is increasing when the reusable component are increasing. As a result, the output XML schema is more modular and provides more reusability and understandability for the human readers (developers).

5 Conclusion

In this paper, the common attributes in the RDB are detected by applying the string matching algorithms and converted as the global element group in the XML schema for the RDB to XML conversion. The proposed method can provide the XML schema quality factors and more effective for some RDB having the common attributes. The generated XML schema design provides more understandability, more modularity, and reduces the maintainability efforts. The complexity of output XML schema document is also described with the complexity measurement methods. The correctness of the proposed method will be shown in the ongoing task.

References

[1] Erik T. Ray, *Learning XML*, First Ed., January 2001, ISBN: 0-59600-046-4, O'Reilly Media, 368 pages.

[2] Myint Myint Lwin, Thi Thi Soe Nyunt, Yuzana, *Generating the Good XML Schema from Relational Database by using String Matching Algorithms*. In: *10th International Conference on Computer Applications*, pp. 357–361, February 2012.

[3] Dongwon Lee, Murali Mani, Wesley W. Chu, Nesting-based relational-to XML schema translation. In: *Proceedings of International Workshop on the Web and Databases*, pp. 61–66, 2001.

[4] Dongwon Lee, Murali Mani, Wesley W. Chu, NeT & CoT: Translating relational schemas to XML schemas using semantic constraints. In: *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, pp. 282–291, 2002.

[5] Angela Cristina Duta, Ken barker, Reda Alhajj, ConvRel: relationship conversion to XML nested structure. In: *Proceedings of the 2004 ACM Symposium on Applied Computing*, ACM New York, NY, USA, pp. 698-702, 2004.

[6] Aminul Islam, Diana Inkpen, Iluju Kiringa, Applications of Corpus-based Semantic Similarity and Word Segmentation to Database Schema Matching, Springer-Verlag, New York, NJ, USA, **17(5)**, pp. 1293-1320, August 2008.

[7] Dilek Basci, Sanjay Misra, Measuring and evaluating a design complexity metric for XML schema documents. *Journal of Information Science and Engineering*, **25**, pp.1405–1425, 2009.

[8] Dilek Basci, Sanjay Misra, Entropy as a measure of quality of XML schema document. *The International Arab Journal of Information Technology*, **8(1)**, pp. 16-24, January 2011.

[9] Dogwon Lee, Murali Mani, Wesley W. Chu, Effective schema conversions between XML and relational models. In: *European Conf. on Artificial Intelligence (ECAI), Knowledge Transformation Workshop (ECAI-OT)*, Lyon, France, July 2002.

[10] Joseph Fong, Anthony Fong, H.K. Wong, et al., Translating relational schema with constraints into XML schema. *International Journal of Software Engineering and Knowledge Engineering IJSEKE*, **16(2)**, pp. 201–243, April 2006.