Nesting Idea for Relation with Multi-Parent in RDB to XML Conversion

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

In today world, information exchanging occupies as the essential role in Web application over the Internet. Some business and organization try to participate in the Web based applications. Traditionally, most of their useful data are stored in the relational database (RDB) and which can be a barrier for data exchanging on the Web. To overcome this problem, RDB to XML conversion becomes a critical field in research trends. In this paper, efficient nesting idea is presented for XML element with multi-parent element to provide the data accessing easily and accurately. To solve the multi-parent problem, the child element chooses the most suitable parent element by defining the referencing times. Most accessing data is directly associated with the referencing counts. As a reult, the child element chooses its main parent and locates under its parent by detecting the maximu referencing data. The other parents access their child element by referencing method. In this paper, nested idea for element with multiple parent elements is presented with Algorithms. Therefore, the design of converted XML schema is good for effecient data accessing.

1. Introduction

There are many database models exist in terms of the different data model and different vendor. They include relational database (RDB) which exports table type, XML database which represents XML tags and object oriented databases (OODB) which express data as object. Each data model has different data type representation such as MySQL, PostgreSQL and SQL Server belonging to the different vendor. These differences cause the problem of spending more time and labor during the development of the application.

Nowadays, Web technology becomes a popular field in information technology. Extensible Markup Language (XML) is tightly coupled with the Web and accepted as the common standard for data exchanging over the Web. However, most of the data are stored in traditional database model as relational database. These relational data are required to export into the Web application for data exchanging between heterogeneous platforms. It introduces a problem because relational data is platform dependent. To overcome this problem, RDB to XML conversion becomes an interested field in the research areas. Therefore, many researchers have been interested on the RDB to XML conversion to provide the heterogeneity problem in data exchanging XML can solve this problem and act as the most suitable format for representing the relational data. But the RDB to XML conversion process is non-trivial task due to their different nature. Many RDB to XML mapping and conversion methods have been proposed by many researchers with both structural and semantic point of views. The new conversion methods are continuously proposed to fulfill the weakness of the previous conversion methods. In RDB to XML conversion, many aspects are required to consider for getting the target format successfully. The target format should be not only applicable in its platform but also kept the important features of the original format such as integrity constraints, cardinality and relationships. Among them, one of the important things in RDB to XML conversion is XML schema design because it is now widely used to exchange information between different platforms or organizations. However, their representations are quietly different. Relations or tables are connected via keys in RDB and XML elements are referenced among them via the nested structure. In the

relational world, integrity constraints are the most important factor for accessing data efficiently whereas nesting idea is also an essential factor in XML world. Therefore the conversion method must provide data accessing effectively. The integrity constraints are defined on the primary key and foreign key relationship. Some relational table has composite key which are combined the foreign keys (FK) of other relations to access desired data. In this case, these FKs are referenced from other tables as a child with multiple parents. For the XML platform, these referential integrity constraints must be kept in the converted XML schema.

This paper proposed the RDB to XML conversion method for relation with the multi-parents relations which are frequently occurred in RDB. In this paper, three algorithms are proposed and applied in the proposed method. The proposed method chooses the main parent from multiple parents by determining the cardinality of each relation and chooses its parent with maximum cardinality to it. It can keep original integrity constraints and provide accessing data accurately.

This paper is organized as follows: The introduction of the proposed method is presented in Section 1. Related works and experiments are described in Section 2 and 3. The results and discussion are shown in Section 4. Finally, conclusion is provided in Section 5.

2. Related Works

Relational database (RDB) to XML conversion is an essential role in research trends because the volume of information within the today world is staggering, but the limitations of existing technology can make it as a barrier in information exchanging and accessing them [9]. Many RDB to XML conversion methods have proposed in the structural and semantic point of views. The earliest and simplest method is Flat Translation (FT). It lacks the nested structure and each relation is converted as element and each attribute in the relations is converted as sub element or attribute. To overcome the problem of FT, Nesting based Translation (NeT) [5] was proposed. NeT provides the nested structure by applying the nested operator such as "*" and "+". However, it can convert one table at a time and not cover for the whole relational database. Both FT and NeT are structural conversion method and lacked the semantic aspects. Constraints-based Translation (CoT) [6] is one of the RDB to XML conversion method which applies the Inclusion Dependency (INDs) of relational schema to solve the semantic problem of NeT. But it can only provide the explicit referential Integrity and cannot provide implicit referential integrity. In [1], the approach extends the conversion methods of NeT and CoT by analyzing the each relationship to find the suitable nested structure to represent it. The paper [3] proposed a method for conversion from relational conceptual model of entity relationship (EER) model to XML schema. It used DTD schema language. The method proposed by [4] acts a middleware that performs the mediation management between the source and target XML representations. It produces neatly nested structure without any redundancy.

All of the previous method did not describe the aspect of element with multi-parents. In the relational world, relation can exist with multiple relationships and can be referenced from multiple relations. It can be a problem in expression of XML format and accessing them. In this case, the derived element or child element needs to choose the best parent for it. In the proposed method, the child element chooses its main parent from multiple parents by analyzing the referencing times on it because it is directly proportional to the accessing times. When the child element chose one parent with maximum referencing times, the remaining parents are chosen as parents by using the referencing method. The proposed method can reduce the length of the XML data file and provide easily accessible data.

3. Experiment

The nested idea for relation with multi-parent in RDB to XML conversion is presented in this paper. The proposed method is the sub portion of the architecture of the whole system which is already proposed in [7, 8]. To generate the nested format, the proposed system performs five tasks. Firstly, the proposed system takes a relational database as input and extracts the integrity constraints such as primary key, foreign keys and domain constraints of relational model from metadata. Secondly, the relationships are defined by applying the referential integrity constraint to get the

nested format. The Algorithm 1 is applied to find the nested structure of all relations by checking the integrity constraints. The output of the Algorithm 1 is general nested structure of each relation.

Algorithm 1: **Nested Relations List**

Input: R, Relational database metadata
Output: Nested List

Begin

```
R={r1,r2,……,rn}      //all relations in the relational database
i ← 1                 //initialize the index for each relation
Nested_List ← ø       //initialize the empty set into nested list
//finding the child list of each relation
While (i ≤ n) do
    Key ← PK(ri)      //setting the primary key of each relation into key
    j ← 0
// checking the referential integrity constraint for all relations in R except ri
    While (j < (R\ri)) do
//checking the referencing key except the primary key of each relation
        If Key ϵ (rj\PK(rj)) then
        Parent (rj ) ← ri
        End If
    j ← j +1
    End While
// adding the nested list into nested of each relation
    Nested_List ← Nested_List U Child (ri)
i ← i+1
    End While
Return Nested_List
```

End

The output of Algorithm 1 is used to find the multi-parents list because the nested list is general and it is not ready for multi-parents element. Therefore, Algorithm 2 is used to find the relations with multiple parents and Algorithm 2 is shown in below.

Algorithm 2: Parents group of each relation

Input : Nested List
Output: Multi-parent_List

Begin

```
//collecting the parents each element
n ← size(Nested List)
    p ← 1                 //initialize the index of each relation
Multi-parent_List ← ø     //initialize the multi-parent list
    While ( p ≤ n) do
  Current_relation ← rp   //initialize each relation to group its parents
 //initialized the parent of each relation
      Parent_Set(rp) ← ø
          q ← 1
  While( q ≤ n) do
      If (rp found its new parent) then
       //adding new parent in its parent list
        Parent_Set(rp) ← Parent_Set(rp) U its new parent
      End
```

```
        q ← q+1
      End While
    // appending an element with multi parent into the set of multi-parent list
        Multi-parent_List ← Multi-parent_List U Parent_Set(r_p)
    p ← p+1
      End While
  Return Mulit-parent_List
End
```

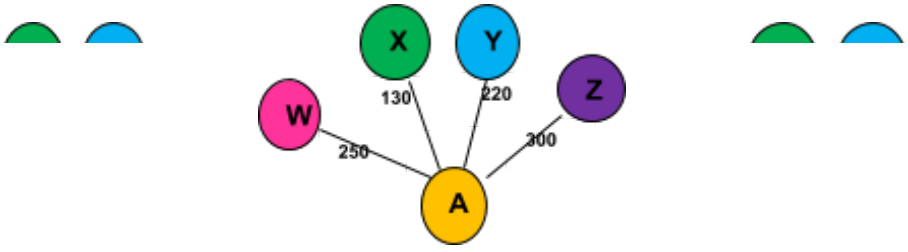
Finally, the proposed system detects an element with multi parents and chooses the most appropriate parent by determining the referencing times. In this task, Algorithm 3 is shown in below and used to find the best parent in the output of Algorithm 2.

Algorithm 3: Parent Choice

Input : R_{mp}, all relations with multi parents
Output: Final_Parent_List

```
Begin
  y ← 1, n ← size(R_mp)
  While (y ≤ n) do
    // list of referencing times of each relation from its parents
    Cardinality_List ← Parent_Set(r_y)
    // choose a parent with maximum referencing times
    Final_parent(r_y) ← Max (Cardinality_List)
    //adding the final parent of each relation into final parent list
    Final_Parent_List ← Final_Parent_List U Final_parent(r_y)
    //removing final parent from list of referenced parent List to get the referenced parent list
    Ref_Parent_List (r_y) ← Cardinality_List \ Final_parent(r_y)
  y ← y+1
  End While
   Return Final_Parent Lists     // returning the list of final parents for whole relational database
End
```

The processing steps of the Algorithm 3 are shown with following illustrations. In this illustration, there are four relations: A, W, X, Y and Z. It is shown in Figure 4(a). The parent-child relationships are described as the dotted lines. In this figure, the parents of relation A are W, X, Y and Z and they formed as multi-parents. Then, the proposed system finds the number of referencing times of each relation according to the cardinality relationship. The result of this step is shown in Figure 4(b). Figure 4(c) is the final result of choosing the parent among the multiple parents. Relation A chooses Relation Z as its parent because Z has the maximum referencing time. The relationship of the child and parent is described as straight bold line. Relation W, X and y are defined as the referenced parent. Therefore, Z is the main parent of A. In the converted XML document A is the sub element of Z. Relation W, X and Y reference the relation A by referencing method.



(a) (b) (c)

Fig 4 (a): Relation A with multi-parents (b) Parent Relations with referencing times (c) Parent-Child relationship

Then, the proposed system takes this output format as input and produces the XML nested structure which obeys the rules of input format.

4. Results and Discussion

The example of generated XML schema is shown in Figure 5. It is build according to the format or rules of the nested structure. In this schema document, element Z is the outer element of element A. Element W, X and Y are also the parents of A and they reference A by keyword "keyref".

```
<? Xml version="1.0" encoding ="UTF-8">
<xsd: schema xmlns: xsd="http://www.w3.org/2001/XMLSchema">
<xsd: element name="Z">
  <xsd: complexType>
  <xsd:sequence>
    <xsd:element name="A">
      <xsd: complexType>
      <xsd:sequence>
         <xsd:element name="column1" type="xsd:String/>
      <xsd:sequence>
      </xsd: complexType>
      <xsd:key name="A_ID">
      <xsd:selector xpath=".//A"/>
      <xsd:field xpath="A_ID"/>
      </xsd:key>
      </xsd:element>
      <xsd:element name="column1" type="xsd:Integer/>
      </xsd:sequence>
     </xsd:complexType>
     <xsd:key name="ID">
     <xsd:selector xpath=".//Z"/>
     <xsd:field xpath="ID"/>
     </xsd:key>
  </xsd:element>
  <xsd: element name="W">
     <xsd: complexType>
         <xsd:sequence>
                 <xsd:element name="column1" type="xsd:Integer/>
         </xsd:sequence>
    </xsd:complexType>
    <xsd:key name="W_ID">
          <xsd:selector xpath=".//W"/>
          <xsd:field xpath="W_ID"/>
    </xsd:key>
    <xsd:keyref name="W_ref_key" refer="A_ID">
          <xsd:selector xpath=".//A"/>
          <xsd:field xpath="W_ref_key"/>
```

```
    </xsd:keyref>
  </xsd:element>
  <xsd: element name="X">
  <xsd: complexType>
        <xsd:sequence>
              <xsd:element name="column1" type="xsd:Integer/>
        </xsd:sequence>
   </xsd:complexType>
   <xsd:key name="W_ID">
        <xsd:selector xpath=".//W"/>
        <xsd:field xpath="W_ID"/>
   </xsd:key>
   <xsd:keyref name="X_ref_key" refer="A_ID">
        <xsd:selector xpath=".//A"/>
        <xsd:field xpath="X_ref_key"/>
   </xsd:keyref>
  </xsd:element>
  <xsd: element name="Y">
  <xsd: complexType>
        <xsd:sequence>
              <xsd:element name="column1" type="xsd:Integer/>
        </xsd:sequence>
  </xsd:complexType>
  <xsd:key name="Y_ID">
        <xsd:selector xpath=".//Y"/>
        <xsd:field xpath="Y_ID"/>
  </xsd:key>
  <xsd:keyref name="Y_ref_key" refer="A_ID">
        <xsd:selector xpath=".//A"/>
        <xsd:field xpath="Y_ref_key"/>
    </xsd:keyref>
</xsd:element>
</xsd:schema>
```

Fig 5: Example of Generated XML Schema for the illustration

5. Conclusion

This paper proposes the idea for a relation with multi-parent to convert into the XML format. The child XML element chooses its main parent by determining the referencing times because it is directly proportional to the accessing data. The nested idea is presented by an algorithm and illustrated with example figures. This idea solves the problem of multi-parent relations in the RDB to XML conversion which are always occurred in the RDB world. The converted XML schema design is good for accessing XML element with efficiently and accurately.

References

[1] C. Duta Angela, B. Ken and A. Reda, *"ConvRel: relationship conversion to XML nested structure"*, Proceedings of the 2004 ACM symposium on Applied computing, ACM New York, NY, USA, 2004.

[2] C. Yang, J.Sun, "The Exchange from Relational Schemas to XML Schemas Based on Semantic Constraints", Vol.13. No.4, 2008, pp 485-489.

[3] F. Joseph, F. Anthony, W. HK and Y.Philip, "*Translating relational schema with constraints into XML schema"*, International Journal of Software Engineering and Knowledge Engineering IJSEKE, Volume 16, Issue 2, 2006, pp 201-243.

[4] J. Hossam, F. jocelyne and R. Paul, "An Automatic Approach to Generate XML schemas from Relational Models", 12th International Conference on Computer Modeling and Simulation, IEEE, 2010.

[5] L. Dongwon, M.Murali, C.Frank and W. Chu Wesley, "*Nesting-based relational-to XML schema translation",* In Proceedings of International Workshop on the Web and Databases, 2001, pp. 61-66.

[6] L. Dongwon, M.Murali, C.Frank and W. Chu Wesley, "*NeT & CoT: Translating relational schemas to XML schemas using semantic constraints",* In Proceedings of the 11th ACM International Conference on Information and Knowledge Management, 2002, pp. 282-291.

[7] L. Myint Myint, S.Nyunt Thi Thi and Yuzana, "*Generating the Good XML Schema from Relational Database by using String Matching Algorithms*", 10th International Conference on Computer Applications, February, 2012, pp 357-361.

[8] L. Myint Myint, S.Nyunt Thi Thi and Yuzana, "Providing a Way for Qualified XML Schema Design in RDB to XML Conversion", International Conference on Information and Communication for Education, 2013.

[9] T. Ray Erik, "*Learning XML*", First Edition, January 2001, ISBN: 0-59600-046-4.