The Design and Development of a New Storage Mode Supported Dynamic XML Documents

Bai Tao, Chang Jian-xia and Wang Jing
Key Lab of Northwest Water Resources and Environment Ecology of MOE, Xi’an University of Technology Xi’an, China

Abstract: For the problem that XRel model can not effectively support the storage of dynamic XML document, the following improvements and extensions were made in this paper. First, a new encoding scheme called NewDietz, which supported the storage of dynamic XML document, were proposed based on the Dietz encoding. Second, the relationships logic model which can storage NewDietz coding node was designed, not only ensuring the effective storage of the new node, but also taking into account the problem that the relationship of nodes between ancestors and descendants must be judged firstly when the dynamic XML document restructured in this pattern. At last, a concrete example was given to verify the effect of the practical application of new model.

Key words: Dietz encoding, newdietz encoding scheme, relational storage of dynamic XML documents, vector method, XRel mode

INTRODUCTION

XML (extensible Markup Language), as a kind of semi-structured language, has a good information presentation function. This makes it have a wide application in the information exchange, data storage and heterogeneous data integration, and become the main standard in the current Internet information exchange.

XRel is a kind of storage and retrieval method of an XML document using by relational data base (Masatoshi et al., 2001). It adopted logic storage mode and index method, both of which are compatible with relational database. It storages the XML document information through by the SQL statements. XRel method can also well support XPath query based on node path relationship. But it does have the following faults:

- That it stores the structure of the nodes by former sequence - later sequence coding in the XRel mode led to low efficiency in the update process of document.
- As relations storage model is structured, such characteristic in storage process is easy to cause the loss of information for some XML document which did not provide the internal structural descriptions.

The fundamental reason is that former sequence-later sequence coding cannot be effectively judge the dynamic structure relation of XML document relations in relation mode.

Coding scheme supporting dynamic XML document storage-vdietz: In the field of new model coding design this paper improved Dietz coding scheme, and designed a NewDietz coding scheme which can support the dynamic XML document by introducing vector method. Dietz coding (Dietz, 1982) is an early kind of strategies which focus on former sequence-later sequence coding in the XML document tree node. In the dietz coding, each nodal points coding numerical value showed with a couple of integers: <Preorder, Postorder>. Preorder means the nodal points numerical value teaches by the XML document tree the preorder traversal. Postorder means the nodal points numerical value teaches by the XML document tree the postorder traversal. In the coordinate axis, each node coding can be expressed as the ancestors of the node coding subset. For any two nodes V1<Pre1, Post1> and V2<Pre2, Post2>, if V1 is longer than V2, which means that Pre1<Post2 and Pre2 < Post1, then V1 for V2 ancestors node.

Dietz coding has been widely used in the field of XML document retrieval and storage. Its advantage is that through simple traversal mechanism of coding value obtained coding value, obtained any of the two nodes ancestors-seed relations by obvious identifying method, and documents stored in the relational database can be easily realized easily. Because the continuity among XML document tree nodes, Dietz encoding's operational efficiency is very low for dynamic XML documents. When a piece of new data is inserted, the entire document trees need to encode again. This not only changed the
original node coding value, the more reduced document storage and retrieval efficiency. In dynamic XML document coding strategies aspects, Extended Dewey Coding scheme's deficiency in the aspect of XML data inserted dynamically, paper (Shi et al., 2007) proposes the IFED coding scheme. Based on Dewey codes, using its binary decimal characteristics, paper (Chen et al., 2008) puts forward BSC coding scheme. Paper (Jun-Ki et al., 2009) puts forward a kind of coding scheme EXEL which supports dynamic document updating. Considering the current native database system lack of the unified industry standard (Feng et al., 2006), relational database system has mature and wide applications (Tang et al., 2003) in the aspects of data storage and retrieval and query, and the actual development needs, we introduce vector method (Xuliang and Wang., 2007), and inherit coding schemes' advantages in relations storage (Olli, 2005). And then, we design a kind of coding scheme which could support dynamic XML documents stored in relation mode-New Dietz (New Dietz encoding), in order to solve the above coding schemes' insufficiencies in relationship models' application.

The support from NEWDIETZ coding for dynamic XML documents coding: When we encode the dynamic XML document to coding, we need to consider coding issues in the process of insert or delete nodes for the document. As space is limited, we mainly discussed the XML document when inserted encoding process in the new node New Dietz. According to the different position inserted, we can divide the node insertion process is into four kinds of cases (dashed line shown in Fig. 1):

- Insert new brother node between the two brothers nodes
- Insert the new left child node next under the father node. Insert the new right child node next under the father node
- Insert a new child node under a leaf nodes

In order to be able to judge the ancestors-descendents relationship of new inserted node seed, and ensure the node reasonable storage length in the relation mode, the following four definitions of coding come into being for the new inserted nodes: First of all, definition any of two nodes NewDietz coding: 

\[ V_1 = ((x_1, y_1), (x_2, y_2)) \] and 

\[ V_2 = ((x_3, y_3), (x_4, y_4)) \]  

- If the V1 and V2 are two brothers nodes for a XML document tree, then define the new insert node NewDietz coding is 

\[ V_3 = ((x_1+x_3, y_1+y_3), (x_2+x_4, y_2+y_4)) \]  

- If V1 is a node of the XML document tree, and V2 is the V1's far left child node before inserting new node, then NewDietz coding expressed as: 

\[ V_3 = ((x_1+x_3, y_1+y_3), (2x_1+x_3, 2y_1+y_3)) \] after inserting a new far left child node V3 under the V1  

- If the V1 is a node in the XML document tree, and V2 is the V1's far right child node before inserting new node, then NewDietz coding expressed as: 

\[ V_3 = ((x_2+x_4, y_2+y_4), (2x_2+x_4, 2y_2+y_4)) \] after inserting a new far right child node V3 under the V1  

- If the V1 is a leaf node, then NewDietz coding expressed as: 

\[ V_3 = ((x_1/2, y_1/2), (x_2/2, y_2/2)) \]  

From the new insert node's coding situation we can see, NewDietz coding scheme provides a good support to the XML document dates' update. It only needs to take NewDietz coding itself without all of the node re-encoding for each new inserted node. In addition, the definition of NewDietz coding for the new inserted node is in line with the coding ancestors-descendents judgment. And such judgment is also apparent in relation mode.

**Newdietz coding algorithm design**: For any one single XML document tree after Dietz coding, it would obtained the node's preorder traversal value: 1, 2... n, 1 and n will be given themselves starters (1, 0) and (0, 1) respectively. The method NewDietzProcess of NewDietz coding for the nodes is shown by the following:

Input: int [2][n]code //used for storage vector coding value of the node  

int start //the first node preorder traversal value, here is 1  

int end //the last node preorder traversal value, here is n

Output: int [2][n] code // Return to the matrix code which contains the node's preorder and postorder coding and subsequent vector codes

**Void NewDietzProcess (int [] [] code, int start, int end) {**

Code [0] [start] = 1;

Code [1] [start] = 0;

Code [0] [end] = 0;

Code [1] [end] = 1;
The relationship logic pattern which supports NEWDIETZ coding node storage: In the XML document elatedly research, the interaction with relational database is an important research direction. But there is existing irreconcilable conflict between strictly binary structure characteristics in elements-attributes of Relational data base and XML document half structural characteristics. So, when we establish storage mode, we need to provide the structure of an XML document description. And according to ancestors-descendants judgment algorithm and relationship model of XML documents node, it should have well complementary, and then realize document storage. At present the relationship stored pattern for the XML document mainly has two. One is based on the node, the other is based on document code. By establishing a DTD or Schema ideograph, the former establishes storage index algorithm of the XML document in order to further realize the relationship storage of the XML document by describing the node structure course in the ideograph; Based on the document code storage, According to documents-elements-attributes-text structure it split and memory the documents, then adopt the researchers pre-defined coding algorithm to judge arbitrary nodes between the relations of father-son and ancestors- descendants, and reconstruct the document. This is also the way XRel mode to use. This paper studied the 2nd kind of circumstance, namely, designed the storage mode of NewDietz coding in the relational database, and gave the method of inserting new node method in the new relationship model.

The mode construction: Firstly, according to documents -elements-attributes-text structure, it constructed logic pattern relationship based on NewDietz coding:

```java
int middle = (int) Math.Floor ((start + end)/2);
if ((middle < end) && (middle > start)) {
    Code[0][middle] = Code[0][start] + Code[0][end];
    Code[1][middle] = Code[1][start] + Code[1][end];
    NewDietzProcess (code, start, middle);
    NewDietzProcess (code, middle, end);
//Change the star value tand end value of the input parameters, recursion invoke the method of NewDietzProcess, up to all the elements are assigned value in matrix code
} else {
    return;
}
```

Relations logic pattern based NewDietz coding Is made up of four parts. Document list (Used to store the basic information of the XML documen). Element list (Storage XML document tree element node information, Including, node name, node's preorder and postorder vector coding and , father node preorder and postorder vector coding and etc, and take this list as the analytical basis of XML document structure). Attribute list(The information stored attribute node, such as the name of attribute node, the value of attribute node, and etc). Text list(information storage numerical node).

All of element list, Attribute list and Text list provide the foreign key identifier (documented) which used for document table connection. And Attribute list and Text list also provides the foreign key identifier (elementprevectorencode, elementpostvectorencode) used for element table connection. Through the foreign key, Relations logic model could keep the integrity of structure information for an XML document in the storage process by the good index method provided in the relationship model; The attributes of elementprevectorencode and elementpostvectorencode in the element table respectively represent vector preorder and postorder coding of the current node. The attributes of parentprevectorencode and parentpostvectorencode in the element table respectively represent vector preorder and postorder coding of the father node of the node. According to the definition of encoding attribute of the node and its ancestors in the Element table, meanwhile, according to the judge discussion about the ancestors-seed relations for newdietz coding in subsection II (A), it can be determine the ancestors-seed relationship and the father-son relationship among the nodes, then realize document information storage (The relations view co de of XML document based on NewDietz coding in relationship in storage pattern shows in Fig. 3.) The concrete realization process of relation mode can be divided into four steps:

- Analysis the XML document with JDOM, construct the XML document tree, and obtain the attributes and values of the node
- Coding the XML document tree nodes by NewDietz, and return the NewDietz coding value of the nodes in the tree
- Storage node name, node coding, node attributes and data in accordance with the new relationship logic pattern
Update the storage method of the node in the new mode: It defines the new insert node's storage in this relationship model through the NewDietz coding of the new inserted nodes in the XML document tree in section II(B). It only needs to be provided the information of the new inserted node's coding and the new inserted node's father nodes, and then it can be able to execute the inserting operation directly in Element table. And it does not need to amend XML document information storage in the logic pattern. Instead of amendment. Taking the mode 2 in the Fig. 1 as an example, the key inserting method for new nodes is presented as shown below:

```
```

Fig. 2: Pub.xml example

Fig. 3: The example of xpath query under the model
### Table 1: The storage view of pub.xml under new relational model

<table>
<thead>
<tr>
<th>The current node markers</th>
<th>Father quarter node markers</th>
<th>The preorder newdietz coding values of the current node</th>
<th>The postorder newdietz coding values of the current node</th>
<th>The preorder newdietz coding values of the father node</th>
<th>The postorder newdietz coding values of the father node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Author</td>
<td>&lt;1, 4, 4&gt;</td>
<td>&lt;1, 2, 4&gt;</td>
<td>&lt;1, 3, 2&gt;</td>
<td>&lt;2, 5, 2&gt;</td>
</tr>
<tr>
<td>Title</td>
<td>Article</td>
<td>&lt;2, 5, 7&gt;</td>
<td>&lt;4, 7, 4&gt;</td>
<td>&lt;1, 2, 4&gt;</td>
<td>&lt;1, 3, 2&gt;</td>
</tr>
<tr>
<td>Author</td>
<td>Article</td>
<td>&lt;1, 3, 3&gt;</td>
<td>&lt;2, 5, 3&gt;</td>
<td>&lt;1, 2, 3&gt;</td>
<td>&lt;1, 3, 3&gt;</td>
</tr>
<tr>
<td>Name</td>
<td>Author</td>
<td>&lt;4, 7, 4&gt;</td>
<td>&lt;3, 4, 4&gt;</td>
<td>&lt;1, 2, 3&gt;</td>
<td>&lt;1, 3, 3&gt;</td>
</tr>
<tr>
<td>Article</td>
<td>Pub</td>
<td>&lt;1, 2, 7&gt;</td>
<td>&lt;1, 3, 4&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;0, 1, 0&gt;</td>
</tr>
<tr>
<td>Price</td>
<td>Book</td>
<td>&lt;2, 3, 5&gt;</td>
<td>&lt;1, 1, 5&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;2, 3, 5&gt;</td>
</tr>
<tr>
<td>Author</td>
<td>Book</td>
<td>&lt;3, 5, 5&gt;</td>
<td>&lt;2, 3, 5&gt;</td>
<td>&lt;1, 1, 5&gt;</td>
<td>&lt;2, 3, 5&gt;</td>
</tr>
<tr>
<td>Book</td>
<td>Pub</td>
<td>&lt;1, 1, 5&gt;</td>
<td>&lt;3, 5, 5&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;0, 1, 0&gt;</td>
</tr>
<tr>
<td>Title</td>
<td>Book</td>
<td>&lt;3, 4, 5&gt;</td>
<td>&lt;5, 4, 5&gt;</td>
<td>&lt;1, 1, 5&gt;</td>
<td>&lt;3, 5, 5&gt;</td>
</tr>
<tr>
<td>Name</td>
<td>Editor</td>
<td>&lt;0, 1, 4&gt;</td>
<td>&lt;1, 4, 5&gt;</td>
<td>&lt;1, 5, 5&gt;</td>
<td>&lt;1, 5, 5&gt;</td>
</tr>
<tr>
<td>Book</td>
<td>Pub</td>
<td>&lt;3, 1, 4&gt;</td>
<td>&lt;4, 3, 4&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;0, 1, 0&gt;</td>
</tr>
<tr>
<td>Editor</td>
<td>Pub</td>
<td>&lt;1, 5, 5&gt;</td>
<td>&lt;1, 5, 5&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;0, 1, 0&gt;</td>
</tr>
<tr>
<td>Email</td>
<td>Editor</td>
<td>&lt;0, 5, 2, 5&gt;</td>
<td>&lt;0, 5, 2, 5&gt;</td>
<td>&lt;1, 5, 5&gt;</td>
<td>&lt;1, 5, 5&gt;</td>
</tr>
<tr>
<td>Library</td>
<td>Pub</td>
<td>&lt;4, 1, 4&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;0, 1, 0&gt;</td>
</tr>
<tr>
<td>Pub</td>
<td>Null</td>
<td>&lt;1, 0, 0&gt;</td>
<td>&lt;0, 1, 0&gt;</td>
<td>&lt;10000, 10000&gt;</td>
<td>&lt;10000, 10000&gt;</td>
</tr>
<tr>
<td>Name</td>
<td>Author</td>
<td>&lt;5, 4, 5&gt;</td>
<td>&lt;5, 3, 4&gt;</td>
<td>&lt;4, 3, 2&gt;</td>
<td>&lt;3, 2, 2&gt;</td>
</tr>
<tr>
<td>Name</td>
<td>Author</td>
<td>&lt;5, 3, 3&gt;</td>
<td>&lt;5, 2, 3&gt;</td>
<td>&lt;2, 1, 2&gt;</td>
<td>&lt;2, 1, 2&gt;</td>
</tr>
<tr>
<td>Author</td>
<td>Book</td>
<td>&lt;2, 1, 3&gt;</td>
<td>&lt;2, 1, 3&gt;</td>
<td>&lt;3, 1, 3&gt;</td>
<td>&lt;4, 3, 3&gt;</td>
</tr>
<tr>
<td>Price</td>
<td>Book</td>
<td>&lt;7, 3, 5&gt;</td>
<td>&lt;3, 1, 3&gt;</td>
<td>&lt;3, 1, 3&gt;</td>
<td>&lt;4, 3, 3&gt;</td>
</tr>
<tr>
<td>Title</td>
<td>Book</td>
<td>&lt;5, 2, 3&gt;</td>
<td>&lt;4, 1, 3&gt;</td>
<td>&lt;3, 1, 3&gt;</td>
<td>&lt;4, 3, 3&gt;</td>
</tr>
<tr>
<td>Author</td>
<td>Book</td>
<td>&lt;4, 3, 3&gt;</td>
<td>&lt;3, 2, 3&gt;</td>
<td>&lt;3, 1, 3&gt;</td>
<td>&lt;4, 3, 3&gt;</td>
</tr>
<tr>
<td>email</td>
<td>Author</td>
<td>&lt;3, 2, 3&gt;</td>
<td>&lt;7, 3, 2&gt;</td>
<td>&lt;2, 1, 2&gt;</td>
<td>&lt;2, 1, 2&gt;</td>
</tr>
</tbody>
</table>

**Application examples:** Based on NewDietz coding it designed a module including an XML document storage and restructuring in relational databases using by the programming (Using Jdom (Elliot, 2002) and XQEngine (James et al., 2003) class package under the W3C standards). And aiming at the XML document pub.xml, it gave an example of a storage and XPath inquiring pub in this module (Fig. 2 is the pub.xml document, Table 1 is pub.xml document storage views under the relation mode, Fig. 3 is a Xpath query examples of pub.xml document under the module). From the design process and the query examples got from the module we can see: Compared with the original Dietz coding, NewDietz coding scheme can well support dynamic renewal of an XML document, and according to the comparison result of the relations logic storage pattern that NewDietz coding scheme designed and XRel relations logic storage mode, it not only supported the document storage and reconstruction in relation mode, but also gave attention to dynamic XML documents stored in the relationship mode.

**CONCLUSION**

In recent years, the native database based on XML document has a rapid development. But it still lacks of uniform industry standards and project practice test. So a relational database mode is still an important research direction for XML data storage. This study, aiming at low efficiency problem of XRel relationship storage pattern in dynamic XML document storage aspects, has designed a dynamic XML data storage mode, based on the New Dietz coding scheme, and has taken the concrete development practice. Through the concrete design development, it shows that, compared with XRel relations logic pattern, this approach has effectively supported the updates for XML document in storage pattern relationship, and has retained the former model advantages.

**REFERENCES**


James, M.G., B. Per, C. Kurt, L. James and N.Vaidyanathan, 2003. X Query Kick Start. Indianapolis, IN, Sams, USA.


