XML 文档中时态信息存储方法的研究与比较*)

汤 娜 刘瑞君 陈罗武 汤 庸 武 迪
(中山大学计算机科学系 广州 510275)

摘 要 XML 作为 Web 上新的数据发布语言,将成为 Web 下一代"数据表达"和"数据交换"的统一标准。然而 XML 文档很少是静止的,它经常会被修改。引入"时态表达"后,时态 XML 文档能够记录一系列的修改痕迹以及数据的变化过程。本文提出了将双时态 XML 数据模型映射到双时态 XML 文档的四种映射方法,最后通过实验对比了这些映射方法及其适用场合。

关键词 XML,时态数据库,时态扩展

Research of Temporal Information Storage in XML Document

TANG Na LIU Rui-jun CHEN Luo-wu TANG Yong WU Di
(Computer Science Department of Sun Yat-sen Universary, Guangzhou 510275, China)

Abstract XML which is a new language for data representing, is expected to become a universal format for data exchange on the Web. Generally, we make some changes on XML documents as time goes by. Of course, XML is more natural than RDB on representing the temporal information. So the topic of representing, querying and updating temporal information in XML has received some attention. In this paper, four different ways of mapping bitemporal XML data model into a (bitemporal) XML document are discussed. Finally, all methods are compared.

Keywords XML, Temporal database, Temporal extension

1 引言

XML 技术的应用越来越广泛,如在网络、无线通信、数据库等都有极大的应用。特别在数据传送方面,XML 为电子数据交换提供了新的思路。因为数据的传送要获得真正的独立性、跨平台性,XML 可能是满足这些要求的唯一选择[1]。同时 XML 具有很多优点,如开放性、简单性、高效可扩展性、自描述性等等,数据访问领域的技术的最佳选择便落在 XML 的身上。

一方面,由于 XML 是一种自我描述的标记性语言,即使用者可以定义标记来描述文件中的任何数据,因此随着 XML 的广泛应用和发展,XML 成为统一信息交换的数据格式是一种必然的趋势。另一方面,XML 文档的内容是经常随着时间而改变的。例如,我们创建一个新的 XML 文档,删除一个文档的内容,而且经常是更新文档的内容或数据。然而在很多情况下,我们常常需要对历史数据进行查询,或取出在一定时间内有效的文档数据,或查询文档的变化情况。要实现这种情况,有两个方案。

方案一:简单地将历史 XML 文档都备份保存下来。这个方案的优点是容易实现,但是这样会造成很大的冗余数据,造成资源的浪费。

方案二:将时态信息加入 XML 文档。这样可以大大节省存储空间而达到以上所述的目的。

本文是从方案二的角度上,提出几个时态信息的在 XML 文档中方法。通过对这几种的方法性能的比较,从而找出一种比较好的方法来在 XML 文档中表示时态信息。

2 研究现状

近几年,关于如何在 XML 中表达时态信息的研究[1-7]可以分为两大类。

第一类:对 XML 数据模型进行时态扩充。

Grandi 和 Mnadreoli 为每个 XML 文档增加一个〈valid〉标签以支持有效时间[1]。Amagasa 等人对 Xpath 数据模型进行时态扩充[2],该模型将 XML 文档表示成一个有向图,并为每条边增加有效时间戳。Vaisman 和 Mendelzon 等人提出的数据模型[3]与文献[2]类似,并增加了 ID,IDREF 属性,以及 version node 思想。与这两个模型相反,Shuohao Zhang 和 Cuitis E. Dyreson 是为每个结点增加了有效时间戳[4]。Dyreson 等人首次研究如何扩充 Xpath,以支持事务时间[5]。与上面的模型都不同,他们提出的模型为结点和边都增加了事务时间戳。这些研究通过为 XPath 增加一些 axes, node test 等,以及 built-in 时间函数,来支持时态查询。

第二类:不对 XML 数据模型进行任何扩充。

这一类研究主要以 Wang Fu-Sheng 为代表,他们只是通过为 XML 文档中的每个元素增加 vstart, vend 或 tstart, tend 属性来描述时态信息(称这些 XML 文档为 H-document),而不对数据模型进行任何的时态扩充。另外,他们利用 XQuery 支持用户自定义函数这一特性,定义一些时间函数来简化和方便"时态查询"。

Wang Fu-Sheng 认为 XML 和 XQuery 比传统的关系表和 SQL 能更好地支持时态信息,他主要研究如何用 XML 表示传统关系表[8]和源 XML 数据库的历史变迁,以及 Web 上文档的版本管理[6]。他在文献[7]提出了如何在 XML 中表示关系数据库的事务变迁,并且基于此思想,在传统关系数据库的基础上用 XML 实现了支持事务时间的时态数据库系统 ArchIS,文献[8]是该系统的技术报告。

在对 XML 进行时态扩展的研究上,Clifford 将时态数据模型分为两大类: temporally ungrouped 和 temporally grouped。前一种是数据的自然叠加,简单却较为冗余;而后一种模型具有强大的表达能力,并且由于是面向历史而显得

*)由国家自然科学基金(60673135,60373081)、重点项目(60736020)教育部新世纪优秀人才支持计划(NCET-04-0805)资助。

更为自然[8]。

在XML文档的时态表示上，Bela Stantic等人利用Fernandez等人在文献[9]中从关系数据库到XML的映射方法进行元素表示，并将其命名为DIRECT模型[10]；另外，在常用的属性表示法中，Wang Fu-Sheng在文献[11]中将temporally grouped模型引入XML，提出了XBit数据模型。虽然各种表示模型都被提了出来，且出现很多模型的定性分析，但性能上的定量分析对比、应用范围等还比较少。

3 时态信息表示

在关系数据库中，双时态的表达思想是在原始的关系数据快照数据中加上双时态标签。时间标签表示事务时间和有效时间的二元组(a, B)，其中a为事务时间，B是有效时间段或有效时间点的集合。用T来表达时态时间标签的集合，那么

$$T=\left\{\begin{array}{l}(a,B)\mid a\in[\text{事务时间起},\text{事务时间止}],\\ B\in[\text{有效时间起},\text{有效时间止}]\end{array}\right\}$$

采用时态二元组的办法进行有效时间和事务时间的储存，是为了减少冗余。表1是双时态数据的表示例子。

表1 双时态数据表示例

书名	有效时间	事务时间
XML数据库	["1998-01-01", "2000-01-01"]	["1998-02-01", "2000-02-01"]
XML数据库导论	["2000-01-02", "now"]	["2000-02-03", "UC"]

3.1 四种表示方案

·无重复映射，通过引进两个元素标记和四个时态属性：

① "属性"元素(attribute element)：对应标记为〈time:attribute〉；

② "文本"元素(text element)：对应标记为〈time:text〉；

时态属性(temporal attributes)：vstart，vend，tstart和tend。

用无重复映射表示表1，如下：

```
〈title vstart="1998-01-01" vend="now" tstart="1998-02-01" tend="UC"〉
  〈time:text vstart="1998-01-01" vend="2000-01-01" tstart="1998-02-01" tend="2000-02-01"〉
    XML数据库
  〈/time:text〉
  〈time:text vstart="2000-01-02" vend="now" tstart="2000-02-03"tend="UC"〉
  XML数据库导论
  〈/time:text〉
〈/title〉
```

·重复映射，通过为每个元素增加四个时态属性(vstart，vend，tstart和tend)从而体现它的变化过程。

用重复映射表示表1，如下：

```
〈title tstart="1998-02-01" tend="2000-02-01" vstart="1998-01-01" vend="2000-01-01"〉
  XML数据库
〈/title〉
〈title tstart="2000-02-03" tend="UC" vstart="2000-01-02" vend="now"〉
  XML数据库导论
〈/title〉
```

·元素表示法，在每个元素中增加四个子结点(vstart，vend，tstart和tend)来表示时态信息。它和重复映射的最大区别在于它将vstart，vend，tstart和tend看成元素而不是属性。

用元素表示法表示表1，如下：

```
〈title〉
  〈tstart〉1998-02-01〈/tstart〉
  〈tend〉2000-02-01〈/tend〉
  〈vstart〉1998-01-01〈/vstart〉
  〈vend〉2000-01-01〈/vend〉
  XML数据库
〈/title〉
〈title〉
  〈tstart〉2000-02-03〈/tstart〉
  〈tend〉UC〈/tend〉
  〈vstart〉2000-01-02〈/vstart〉
  〈vend〉now〈/vend〉
  XML数据库导论
〈/title〉
```

储存应该尽可能地减少空间。通过观察，我们可以改进重复映射的表示方法而提出第四种表示方法，在本文我们称它为"单属性表示方法"。顾名思义就是重复映射用两个属性来表示有效时间或事务时间。我们在这里通过一个属性来表示有效时间和事务时间，即将重复映射的两个属性合并为一个属性。

·单属性表示，通过为每个元素增加两个时态属性(valid和transaction)从而体现它的变化过程，其中valid = [vstart，vend]和transaction = [tstart，tend]。

用单属性法表示表1，如下：

```
〈title transaction = "[1998-02-01 2000-02-01]" valid = "[1998-01-01 2000-01-01]"〉
  XML数据库
〈/title〉
〈title transaction = "[2000-02-03 UC]" valid = "[2000-01-02 now]"〉
  XML数据库导论
〈/title〉
```

3.2 性能比较

实验数据：随机生成某单位员工的数据(ID，姓名，工资，职称，部门，其中ID当作员工的属性，实验中分别以员工数(50；50；500)、修改次数(5)、修改salary和ID进行性能比较。

性能参数：XML文档大小，DOM数据模型结点数。

实验结果：

(1)修改salary元素的值

① 四种方案的XML文档大小比较见图1。

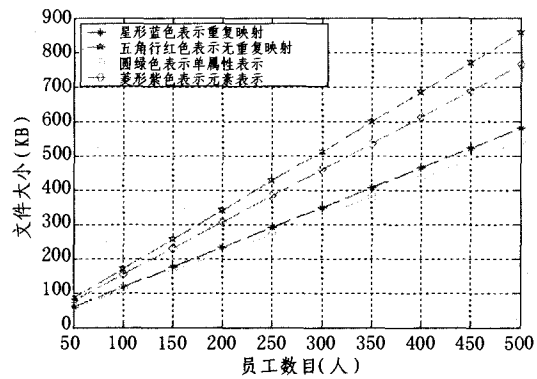

图1 修改salary值文件大小比较

从图1可以看出，在修改salary时的文档大小排序如下：

无重复映射>元素表示>重复映射>单属性表示

由于修改了salary的内容，它处于数据模型的最低层次，这四种方案都没有导致信息的冗余，但是无重复映射多了一些额外的〈time:attribute〉和〈time:text〉标记，因此导致要存储的数据量过大。同样，元素表示中表示一个元素要四个子结点(vstart，vend，tstart，vend)，这样也会导致要存储的数据量较大。而重复映射和单属性表示只是多了一些属性，相对以上两种表示方法而言，要存储的数据量就会小很多，同时单属性比重复映射少了一个属性，所以它要存储的数据必然是最少的。

② 四种方案的结点数比较见图2。

从图2可以看出，四种方案的DOM数据模型的结点数从大到小的排列顺序如下：

元素表示>无重复映射>重复映射>单属性表示

由于修改了salary的内容，元素表示要增加元素结点来表示修改后的文档树，每修改一个都要增加四个结点，因此它

的结点数必然是最多的。而无重复映射，它本来都要通过增加两个时态标示来表示时态信息，因此它是次少的。相比之下，重复映射和单元素表示就更少了。

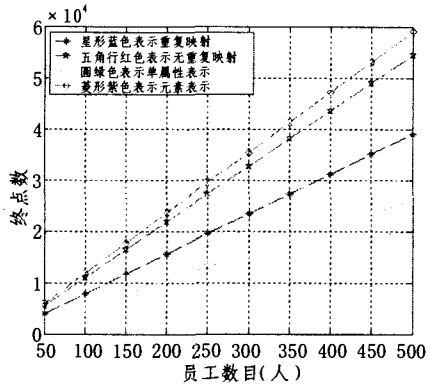

图 2　修改 salary 值结点数比较

(2)修改元素 ID 的值

① 四种方案的 XML 文档大小比较见图 3。

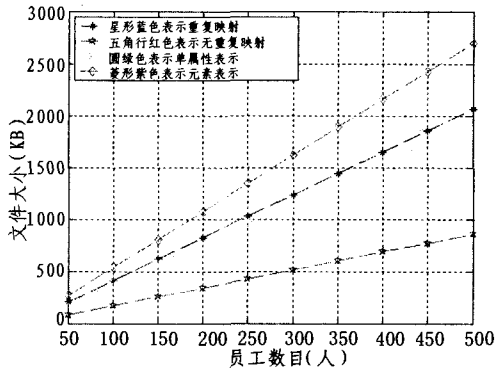

图 3　修改 ID 值文件大小比较

从图 3 可以看出，在修改 salary 时的文档大小排序如下：

元素表示＞重复映射＞单属性表示＞无重复映射

由于修改了 ID 的内容，它处于数据模型的较高层次，元素表示和重复映射都要重复整个元素的所有内容。元素表示除了重复元素的内容外，还要重复四个表示时态信息的结点，所以它是最多的，而重复映射次之。我们知道单属性表示是重复映射的改进，它占有的空间必然会比重复映射少。无重复映射除了表达上有一些冗余外，并不会造成数据的冗余，当修改高层次的元素内容时，它要用的存储空间是最少的。

② 四种方案的结点数比较见图 4。

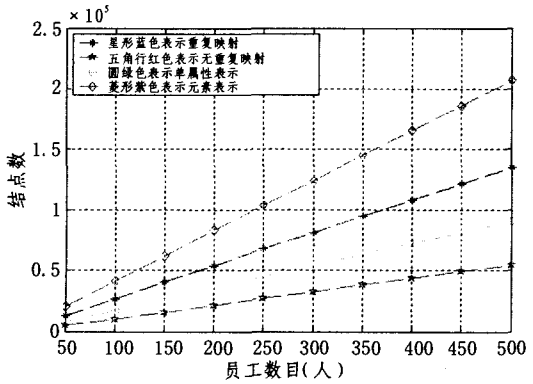

图 4　修改 ID 值结点数比较

从图 4 可以看出，四种方案的 DOM 数据模型的结点数从大到小的排列顺序如下：

元素表示＞重复映射＞单属性表示＞无重复映射

由于修改了 ID 的内容，元素表示要增加元素结点来表示修改后的文档树，而且要增加整个元素的内容，包括四个表示时态信息的结点，因此它的结点数是最多的。重复映射只是通过增加整个元素的内容，结点数次之。一样的道理，由于单属性减少了一个属性来表示，它的结点数必然会比重复映射少。无重复映射由于它并没有造成数据冗余，因此结点数是最少的。

3.3　结论

从以上的实验结果我们可以看出，首先元素表示并不是我们要选择的方案；重复映射和单属性相比较，无论是从修改低层次还是高层次的结点来说，从文件大小和结点来衡量，单属性的性能都比重复映射的好。这样，我们只要在单属性表示和无重复映射这两种方案中选取一种比较好的方案就可以了。

表 2

衡量因素	单属性	无重复映射
可读性	好，直观	不好，难读
修改低层次结点	存储空间小 DOM 结点数少	存储空间大 DOM 结点数多
修改高层次结点	存储空间大 DOM 结点数多	存储空间小 DOM 结点数少
数据冗余	是	否

从表 2 可以看出，这两种表示各适用于不同的场合。对于修改高层次的结点时，虽然无重复映射可读性较差，但是我们关注的是存储空间，应该选择无重复映射；而对于修改低层次的结点时，我们应该选择的是单属性映射。

结束语　本文讨论如何有效地在 XML 文档中存储时态信息，并提出了四种存储方法。通过实验对其性能进行比较，最终选出了两种方法单属性和无重复映射方法。这两种方法在不同的场合中各有各的优点。但现实应用中，经常修改低层次的结点，因此常常选择的是单属性表示方法。对于如何有效地在 XML 文档中对时态信息进行查询操作，还需要对 XML 的查询的语言进行扩展，这也是很重要的一方面，需要很多的学者对这一方面进行进一步的研究。

参 考 文 献

[1] Grandi F, Mandreoli F. The Valid Web: An XML/XSL Infrastructure for Temporal Managementof Web Documents. In ADVIS, 2000

[2] Amagasa T, Yoshikawa M, Uemura S. A Data Model for Temporal XML Documents. In DEXA, 2000

[3] Vaisman A O, Molinari M E, Tome P. Temporal XML: Datamodel, query language and implementation. http://www.cs.toronto.edu/? avaisman/papers.html, 2004

[4] Zhang S, Dyreson C. Adding Valid Time to XPath. In DNIS, 2002

[5] Dyreson C E. Observing Transaction-Time Semantics with T-TXPath. In WISE, 2001

[6] Wang F, Zaniolo C. Temporal Queries in XML Document Archives and Web Warehouses. InTIME-ICTL, 2003

[7] Wang F, Zaniolo C. XBiT: An XML-based Bitemporal Data Model. In ER, 2004

[8] Clifford J, Croker A, Grandi F, et al. On Temporal Grouping. In Recent Advances in Temporal Databases, Springer Verlag, 1995:194-213

[9] Fernandez M, Tan W, Suciu D. Silkroute: Trading between Relations and XML//Proceedings of the 9th Internationa WWW Conference on Computer Networks, 2000

[10] Stantic B, Governatori G, Sattar A. Handling of Current Time in Native XML Database //ACM Proceedings of the 17th Australasian Database Conference. Vol. 49, 2006

[11] Wang Fu-Sheng. XML-based Support for Database Histories andDocument Versions. http://citeseer.istpsu.edu/cache/papers/cs/30282/http:zSzzSzwww.cs.ucla.eduzSz%7Ewangfshz

[12] 汤娜，汤庸，蔡苗苗. 基于 XPath 数据模型的双时态扩展. 计算机研究与发展，2006，43(Supp.)：504-509