

双时态 XML 查询中 now 语义失真的研究与扩展

汤娜 陈罗武 刘瑞君 汤庸

(中山大学计算机科学系 广州 510275)

摘要 本文研究如何有效地查询基于“单属性表示时态信息”的 XML 文档的时态信息操作,并选择 XML 功能较为强大的查询语言 XQuery 进行时态查询扩展。文中先讨论如何解决时态数据库中特殊元素 now 的语义失真的问题,然后对 XQuery 进行双时态扩展,最后给出扩展后的查询实例。

关键词 XML, 时态数据库, 时态扩展, XQuery

Research of Temporal XML and Querying

TANG Na CHEN Luo-wu LIU Rui-jun TANG Yong

(Computer Science Department of Sun Yat-sen University, Guangzhou 510275, China)

Abstract We will study how to query temporal information in XML documents. The temporal information is expressed in the xml documents through “Single Attribute Expression Method”. In this paper, we will first discuss how to solve the semantic distortion problem of “now”, and then make the XQuery to support the temporal querying in the XML document and show the cases of querying the temporal information in the xml documents.

Keywords XML, Temporal database, Temporal extension, XQuery

1 引言

作为互连网上数据表达和数据交换的下一代标记语言, XML 在新一代电子政务、电子商务等 Web 应用中正发挥越来越重要的作用, XML 数据管理成为当前信息系统研究热点^[1,2]。目前 Web 应用的支撑技术大多局限于静态和快照式,但是 XML 文档很少是静态的,文档的结构和内容随着时间的流逝不断变化;新的文档创建,文档删除,最重要的是文档不断被修改。因而在很多应用中还需要查询 XML 文档的历史信息,取出某一时间合法的数据信息,查询文档作了哪些变动和修改等。这些应用需求,对 Web 信息系统提出了新的要求——时态信息的处理。

而且时间是自然界无所不在的客观属性,所有信息都具有相应的时态属性^[3,4],几乎所有的应用都要涉及到时间问题。XML 文档的版本管理和内容的时态查询正在成为下一代 Web 信息系统中重要的应用,特别是在电子政务、电子商务、数据仓库、数字图书馆、协同编辑等信息系统中扮演着日益重要的角色。

时态 XML 信息处理技术的本质是借助 XML 作为实现途径对时态信息技术进行研究。

2 研究现状

随着大量的 XML 数据的出现,如何有效地存储、管理和查询这些数据,这就引出了 XML 数据库的研究问题,目前 XML 数据库研究主要有两种途径:一是纯(native)XML 数据库系统;二是 XML 使能(XML-enabled)数据库系统。在针对 XML 的查询语言问题上,目前大多数的研究都集中在 XQuery 上。XQuery 起源于 Quilt,而 Quilt 是集 ODMG, XPath1.0, XQL, XML-QL 等诸多特性于一身的。可以说, XQuery 是目前最好的 XML 查询语言。

在对 XML 查询语言的时态扩展上, Gao, Snodgrass 等人通过定义新的 BNF 来将 XQuery 扩展为 τ XQuery 语言^[5];而 Fusheng Wang 在对 XQuery 进行时态查询处理时则是直接利用 XQuery 本身的函数支持特性,以增加时态查询函数的方法来达到扩展目的^[6]。而不少数据库研究者和开发商,还有 SQL 标准化组织也都在积极研究通过扩展 SQL 来支持 XML 查询。

双时态扩展中 now 的处理是较为麻烦的。目前,由于 XQuery 查询语言还不是十分的完善,在其上进行双时态扩展后的 now 的处理的研究也就非常少了。文献[7,8]对 now 语义的不确定性做了理论的研究,其中文献[7]主要从语义和关系代数的角度来讨论 now 的不确定性,而文献[8]中通过实验的方法来说明把 now 用极大值 max 代替后产生的查询不确定性,并且比较了由 now 的处理引起的不同的查询速度,但并没有对 now 进行一个明确的理论分析。目前还没有针对 now 语义失真的具体实现方案。本文首先讨论了 now 的语义失真问题,接着介绍了如何对 XQuery 进行双时态扩展以解决特殊元素 now 的语义失真问题,最后给出扩展后的查询实例。

3 now 的语义失真分析

时态数据库中有两个特殊元素 UC 和 now,而实际应用中,对 UC 的处理是很简单的,它是事务时间的一个特殊元素,在逻辑上代表着记录在数据库中没有被删除,当我们要在数据库中删除记录时,通常是将记录的 UC 改为删除时的当前时间。相反,元素 now 是有效时间的一个特殊元素,它具有的语义很复杂,常常引起语义的失真。下面来讨论 now 的语义分析和处理。

(1)当前时间语义

now 的基本语义:表明数据所反映客观事实直到“当前”

都是有效的,如表 1 所示。

表 1

teachername	title	tstart	tend	vstart	vend
Bob	Professor	2002-06-18	UC	2000-04-01	now

(2) 过去时间语义

① 事务时间的滞后性

在实际的应用中很多时候 now 并不很简单地表示当前的时间语义,因为客观事实实际发生的时间和数据录入的时间并不总是同步的,由于同一记录有效时间和事务时间是正交的,在现实中往往事务时间是滞后于有效时间的,这就产生了 now 语义的失真问题。

例如:

假设 Bob 在 2001-04-01 调入计算机系(CS),而数据库的滞后时间为 3 天,在调入计算系之前 Bob 是在数学系的,我们在 2001-04-02 日查询 Bob 所在的系列时,由于数据库的记录还没有更新,它返回的 Bob 还是在数学系,这就是由于事务时间滞后性而引起的不正确。

② “过去数据”中的 now

由于事务时间记录的是数据库的各种操作的历史,我们在查询过去数据库的状态时,元组中的“now”是表示过去的语义的。而这些数据是数据库的过去的历史,这样这个元组就在逻辑上从数据库中被删除,这个元组也随之失效,如表 2 所示。

表 2

teachername	title	tstart	tend	vstart	vend
Bob	Professor	2002-06-18	2003-01-01	2000-04-01	now

这个元组在逻辑上已经在数据库中被删除了,因此 now 在这里表示的应该是过去的语义。

(3) 将来时间语义

① 数据在“生效”前超前录入

如果一个元组的有效起始时间大于当前时间,则该元组是在数据生效前超前录入数据库,now 在这里表示的是将来时间语义。

例如,假设今天是 2007-04-18,如表 3 所示。

表 3

teachername	title	tstart	tend	vstart	vend
Bob	Professor	2007-04-18	UC	2007-07-01	now

② now 本质上就是将来的语义

如果 now 表示的有效时间的将来时间并不是确定的,其本意就是包括将来一段时间也是有效的,但是这一段时间并不能确定,这样 now 表示的就是将来的语义,如表 4 所示。

表 4

teachername	title	tstart	tend	vstart	vend
Bob	Lecturer	2007-04-18	UC	2007-04-25	now

Bob 表示在 2007-04-25 后就是 Lecturer,如果我们查询在 2007-08-25 的 Lecturer 的名单,那应该包括 Bob,因为这里的 now 表示的是将来的语义,包含了 2007-08-25 这一时间。

4 now 处理算法

now 的处理算法

算法符号说明:

tstart:表示事务时间开始

tend:表示事务时间终止

vstart:表示有效时间开始

vend:表示有效时间终止

current_date:当前系统时间

offdate:事务滞后时间

qd:要查询的时间

为了处理 now,先做以下假设。

假设:

对于在事实上没有发生改变的记录(也就是说将来的不确定情况),我们假设数据库的记录是正确的。

针对以上的 now 的语义分析和假设,我们总结了 now 的处理方法。

if $tend \neq 'UC'$

则表示查询的记录数据是“过去状态”中的数据,将 now 确定为 tend。

此时,now 是“过去时间”语义;查询的信息是确定的。

if $tend = 'UC'$

suredate = current_date - offdate

If $vstart \leq current_date$

If $qd \leq suredate$

则 now 为确定的当前时间 current_date,信息是确定的。

if $if\ suredate < qd \leq currentdate + offdate - 1$

则 now 为不确定的,可能是“过去时间”,语义也可能是“当前时间语义”,查询的信息是不确定的。

if $qd > currentdate + offdate - 1$

则 now 为将来时间语义,查询信息是确定的。

else (即 $vstart > current_date$)

则 now 是将来时间,查询信息为确定。

5 双时态查询的扩展

5.1 实现方案讨论

XML 有多种查询语言,如 XPath, Quilt, XQL, XQuery 等等,其中 XQuery 的功能最为强大。因此我们选择的扩展语言是 XQuery。如何让 XQuery 更好地支持时态信息的查询呢?一般来说有以下两种方案可以考虑:

- 中间层扩展方案^[9];
- 函数扩展方案^[6]。

由于到目前为止,XQuery 还是 W3C 的工作草案,时时在变化,采用中间层扩展方案并不是一种很好的选择,相反 XQuery 具有比较强的时态表达能力,如它支持 datetime, time, duration 等等时态数据类型,而且它支持用户自定义函数功能。因此采取函数扩展方案是一个比较灵活和简便的方法。

在这里,我们根据时态数据库的知识分别对事务时间维、有效时间维和双时态特殊的操作关键字写成 XQuery 的函数形式供用户直接调用,从而达到对 XML 文档中时态信息的查询操作,这些函数的实现对用户来说是透明的,大大方便了用户。通过以上的三个方面时态函数扩展,我们组成一个时态函数库 temporal.xq,这样用户就可引用 temporal.xq 的库

函数进行时态方面的查询操作。temporal.xq 的一些函数如表 3 所示。

表 3 temporal.xq 的函数说明

函数名	函数说明
tstart(\$e)	求出 \$e 事务时间的开始时间
toverlaps(\$e1, \$e2)	判断 \$e1, \$e2 两个元素的事务时间间隔是否相交, 其中 e1 的事务时间的始点在 e2 之前
vstart(\$e)	求出 \$e 有效时间的开始时间
voverlaps(\$e1, \$e2)	判断 \$e1, \$e2 两个元素的有效时间间隔是否相交, 其中 e1 的有效时间的始点在 e2 之前
bitelement(\$be)	函数返回一个元素 \$be 的事务时间和有效时间
newbitelement(\$ts, \$te, \$vs, \$ve)	函数由四个事务时间, 有效时间来形成一个新的双时态元素 bitelement 返回

```
xquery version "1.0";
import module namespace temporal="C:\Program Files\Altova
\XMLSpy2006\Examples\XQuery\temporal" at "temporal.xq";
{
  results
  {
    for $s in doc("teachers.xml")/teachers/teacher[name="John"]/salary
    where
      temporal:tstart($s) <= xs:date("2007-04-28")
      and temporal:tend($s) >= xs:date("2007-04-28")
      and temporal:vstart($s) <= xs:date("2007-04-28")
      and temporal:vend($s) >= xs:date("2007-04-28")
    return ($s, temporal:makesure($s, "2007-04-28"))
  }
}
结果:
<results>
<salary transaction="[2007-01-03 UC]" valid="[2007-01-01 now]">5700</salary>
</results>
```

由于今天是 2007-04-29 日, 我们在时态查询的时候假设事务滞后时间是 3 天, 因此查询的结果应该是不确定的(即带 <unsure>? </unsure> 子元素)。

结束语 本文主要讨论 XML 时态查询扩展方案、时态数据库特殊元素 now 的语义分析和处理算法, 在此基础上我们对 XML 查询语言 XQuery 进行了双时态的查询扩展, 使得能更好地支持时态查询操作, 并且对查询方法进行了实现与演示。但是本文对 XQuery 进行扩展的方案并不是很完整的, 首先在查询的过程中, 并没有在完整性和约束性上做太多的考虑。同时由于 XQuery 也有它自身的缺陷, 例如, 它并没有支持 Create, Update, Delete 和 Insert 操作, 本文也只讨论了 XQuery 的查询操作的扩展。随着 XML 的应用越来越广泛, Create, Update, Delete 和 Insert 操作也越来越重要, 这也是 XQuery 需要完善的地方。

参考文献

- [1] Gray, Jim (Microsoft). The next database revolution // Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France. SIGMOD 2004, June 2004; 1-4
- [2] 孟小峰, 周龙骧, 王珊. 数据库技术发展趋势. 软件学报, 2004, 15(12): 1822-1836
- [3] Grandi F. An annotated bibliography on temporal and evolution aspects in the WWW. a TIMECENTER Technical Report. <http://www.cs.auc.dk/research/DP/tdb/TimeCenter/TimeCenterPublications/TR-75.pdf> 2003
- [4] 汤庸. 时态数据库导论. 北京大学出版社, 2004
- [5] Gao D, Snodgrass R T. Syntax, Semantics, and Query Evaluation in the TXQuery. Temporal XML Query Language. TR-72 A TIMECENTER Technical Report. 2003
- [6] Wang FuSheng. XML - Based Support for Database Histories and Document Versions. <http://citeseer.ist.psu.edu/cache/papers/cs/30282/http://zSzzSzwww.cs.ucla.edu/zSzzSz%7Ewangfshz>
- [7] Stantic B, Governatori G, Sattar A. Handling of Current Time in Native XML Database // Proceedings of the 9th International WWW Conference on Computer Networks, 2000
- [8] 叶小平, 汤庸. 时态变量 "Now" 语义及相应时态关系运算. 软件学报, 2005(5)
- [9] 陈子聪, 汤庸, 刘海, 汤娜. 时态数据库中间件的设计与实现. 计算机研究与发展, 2006, 43(11). 增刊)

5.2 查询实现

查询实现环境为 XMLSPY2006, 数据源是 teachers.xml, 将 temporal 时态函数库路径设为: "module namespace temporal="C:\Program Files\Altova\XMLSpy2006\Examples\XQuery\temporal";

有效时间查询: 查询教师名字为 "Bob" 在有效时间区间为 [1998-01-01, 2000-01-01] 的职称情况。

查询例子 1:

```
xquery version "1.0";
import module namespace temporal="C:\Program Files\Altova
\XMLSpy2006\Examples\XQuery\temporal" at "temporal.xq";
{
  results
  {
    for $s in doc("teachers.xml")/teachers/teacher
    [name="Bob"]/title
    [temporal:voverlaps(temporal:velement("1998-01-01",
    "2000-01-01"), ., .)]
    return $s
  }
}
结果:
<results>
<title transaction="[1998-12-18 1999-06-18]" valid="[1998-09-01 1999-03-01]">lecturer</title>
<title transaction="[1999-06-18 2000-09-18]" valid="[1999-04-01 2001-03-01]">lecturer</title>
</results>
```

关于 now 和 UC 的查询: 查询教师名字为 "Sam" 的现在的工资情况。

查询例子 2:

```
xquery version "1.0";
import module namespace temporal="C:\Program Files\Altova
\XMLSpy2006\Examples\XQuery\temporal" at "temporal.xq";
{
  results
  {
    for $s in doc("teachers.xml")/teachers/teacher[name="Sam"]/salary
    where temporal:vend($s)=temporal:now()
    and temporal:tend($s)=temporal:UC()
    return $s
  }
}
结果:
<results>
<salary transaction="[2002-06-18 UC]" valid="[2000-04-01 now]">2100</salary>
</results>
```

now 的语义的处理: 查询教师 John 在 2007-04-28 的工资情况。

查询例子 3: