

# XML 数据库的研究与应用

周爱武, 李孙长, 程 博, 夏 松

(安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

**摘 要:** XML 是一种专门为 Internet 所设计的标记语言, 随着 XML 的大量应用, 它已经成为 Internet 上数据表示和数据交换的标准, 如何有效地管理大量的 XML 数据显得非常重要。使用数据库技术对 XML 进行管理, 这已经成为一种必然的发展趋势。XML 与数据库相结合已经成为一个新的研究领域, 也是 XML 和数据库发展的趋势。文中探讨和分析了 XML 数据库的相关知识, 介绍了 Native XML 数据库及一个 NXD 系统 eXist, 最后表明 XML 数据库将成为一个新的研究领域。

**关键词:** XML; 数据库; Native XML 数据库; eXist

**中图分类号:** TP311.131

**文献标识码:** A

**文章编号:** 1673-629X(2009)09-0218-04

## Research on XML Database and Its Application

ZHOU Ai-wu, LI Sun-chang, CHENG Bo, XIA Song

(School of Computer Science & Technology, Anhui University, Hefei 230039, China)

**Abstract:** XML is a special markup language for Internet. With the emergence of a lot of XML documents, it has been become the standard for data representation and exchange on Internet. How to manage XML documents effectively has been become very important. It has become an inevitable trend to manage XML documents with the technology of database. The combination between XML and database has become a trend and a new field of research. Explores and analyzes XML and database, focusing on Native XML database and eXist system. Finally, XML database will be a new field of research.

**Key words:** XML; database; Native XML database; eXist

## 0 引 言

XML<sup>[1,2]</sup>(eXtensible Markup Language)即可扩展标记语言,它与 HTML 一样,都是 SGML(Standard Generalized Markup Language,标准通用标记语言)的派生。XML 实际上是 Web 上表示结构化信息的一种标准文本格式,它没有复杂的语法和包罗万象的数据定义。XML 的简单使其易于在任何应用程序中读写数据,这使 XML 很快成为数据交换的唯一公共语言,虽然不同的应用软件也支持其它的数据交换格式,但不久之后它们都将支持 XML,那就意味着程序可以更容易地与 Windows、Mac OS、Linux 以及其它平台下产生的信息结合,然后可以很容易加载 XML 数据到程序中加以分析,并以 XML 格式输出结果。

目前,处理 XML 文档的方式主要有 SAX 与 DOM 两种。SAX(Simple API for XML)是一种基于流的、以事件处理方式工作的接口。SAX 2.0 在 2000 年 5 月

发布,增强了许多功能,包括对名字空间的支持。DOM(Document Object Model)则是在对 XML 文档进行分析后,在内存中建立起一个完整的树结构,然后在此基础上进行各种操作。简单地比较来看,SAX 对系统资源要求低、速度快,但对文档的操作是只读的;DOM 的处理能力强大,但要求大量的系统资源,尤其是对于大的文档。

此后还出现了 Xpath 和 Xpointer,用以完成 XML 的搜索和转换;XSL、XSLT 和 SOAP 用以完成 XML 的远程对象访问,XML Query Languages 的出现使 XML 查询语言可用于任何 XML 文档。

随着 Web 技术的不断发展,信息共享和数据交换的范围不断扩大,传统的关系数据库也面临着挑战。数据库技术的应用是建立在数据库管理系统基础上的,各数据库管理系统之间的异构性及其所依赖操作系统的异构性,严重限制了信息共享和数据交换范围。同时,XML 已经成为数据表示和交换的标准,伴随着各种 XML 应用的快速发展,XML 数据大量出现,以数据库方式实现 XML 数据的有效管理和快速查询已经成为趋势。

收稿日期:2009-01-13;修回日期:2009-04-01

基金项目:安徽省自然科学基金项目(kj2008B092)

作者简介:周爱武(1965-),女,安徽舒城人,副教授,研究方向为数据库与 WEB 技术。

## 1 XML数据库的类型

XML数据库<sup>[3~5]</sup>是当今数据库领域和XML领域的一个新的研究方向。随着时间的发展,目前XML数据库已经基本上可以分为三种类型:XML Enabled Database(XEDB),即支持XML的数据库;Native XML Database(NXD),即纯XML数据库(或者原生XML数据库);Hybrid XML Database(HXD),即混合XML数据库。

### 1.1 XML Enabled Database(XEDB)

XML Enabled Database(XEDB)<sup>[6]</sup>,即支持XML的数据库。其特点是:不考虑底层数据的存储模式,只要能存入和取出XML数据,并符合数据库的基本特性。也就是说在原有的数据库系统上扩充对XML数据的处理功能,使之能适应XML数据存储和查询的需要,如Oracle Oracle9i, IBM DB2, MS SQL Server2005等。XML的半结构化特性并不符合传统的关系模块的结构化特性,强制转换会造成数据信息的丢失和系统性能的减弱。这样的数据库可能使原始的XML元数据和结构丢失,而且数据检索的结果也不能保证是原始的XML形式。XEDB把XML数据存在关系表中,在访问相应的表之前,XML数据模式必须被翻译为关系模式,同样地,XML查询语言也必须翻译为SQL语言以访问这些表,这些翻译操作可能会消耗大量的CPU资源,造成系统性能降低。图1显示了支持XML的关系数据库的体系结构。

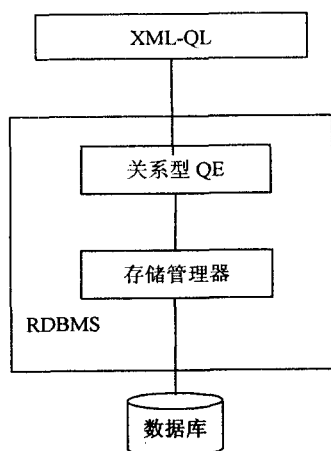


图1 支持XML的数据库的体系结构

### 1.2 Native XML Database(NXD)

Native XML Database(NXD)<sup>[7~9]</sup>,即纯XML数据库(或者原生XML数据库)。纯XML数据库直接存储XML数据,数据库引擎在访问XML数据时不需要执行任何转换工作。这是支持XML的数据库和纯XML数据库之间的主要差别。其特点是:以XML文档作为基本的逻辑存储单位,针对XML的数据存储

和查询特点专门设计适用的数据模型和处理方法。图2显示了通过XML引擎存储和获取XML数据的过程。

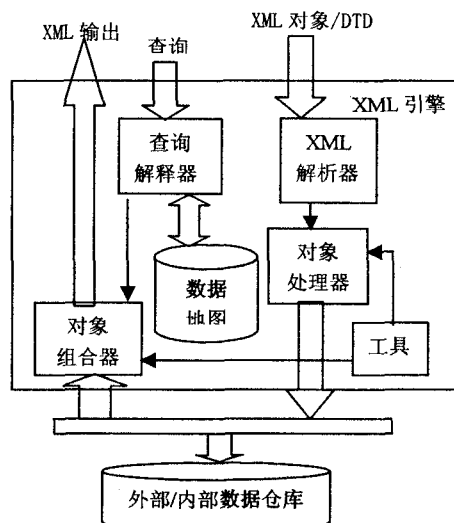


图2 纯XML数据库的XML引擎的体系结构

### 1.3 Hybrid XML Database(HXD)

Hybrid XML Database(HXD),即混合XML数据库。根据应用的需求,可以视其为XEDB或NXD的数据库,典型的例子是Ozone。

## 2 Native XML Database(NXD)的研究

在XML数据库中,纯XML数据库相对其他的数据库更具有发展的趋势,虽然现在的纯XML数据库技术相对当前流行的关系数据库(或对象数据库)还不是很成熟,但随着Internet信息的大量交互,纯XML数据库以其独有优势一定会超越现行的数据库。这就使得研究纯XML数据库显得更加有意义。

### 2.1 纯XML数据库的基本概念

关于Native XML数据库目前还没有一致公认的定义,Staken K<sup>[10]</sup>对其作了一个说明,学术界比较认同,介绍如下:

1) Native XML数据库为XML文档(而不是文档中的数据)定义了一个逻辑模型,并且根据该模型存取文档。模型包括元素、属性、PCDATA和文件的次序。目前已经采用的模型有XPath数据模型、XML - Infoset、DOM模型和SAX事件模型等。

2) Native XML数据库以XML文档作为其基本(逻辑)存储单元,正如关系数据库以表中的行作为基本逻辑存储单位一样。

3) Native XML数据库对底层的物理存储模型没有特殊要求,即它不一定必须建立在关系、层次或面向对象的数据库基础上,也不一定必须规定存储格式,如

索引或文件压缩。

从以上描述中得出,纯 XML 数据库是专门用于存储文档并且保持其完整性,存储 XML 均以文档为基本单位。纯 XML 数据库不是一种全新的数据库底层处理模式,它不是用于取代现存的数据库,它仅仅用于帮助人们更好地处理 XML 文档。

根据 XML 数据不同的存储形式,可以将纯 XML 数据库体系结构分为:基于文本的 NXD 和基于模型的 NXD。这两者在数据存储层之上的部分没有本质的区别,主要区别在于基于文本的 NXD 将 XML 文件视为一种文本,强调文本之间的层次关系;基于模型的 NXD 将 XML 文件视为一种数据模型,强调数据之间的逻辑结构。

基于文本的 Native XML 数据库将整个 XML 文档作为文本存储,文档可以是文件系统中的文件、关系数据库中的 BLOB 字段或其他特定的文件格式。基于文本的 Native XML 数据库与层次模型的数据库很相似,当存取预定义好层次的数据时,其效率胜过关系数据库,而存取任意数据元素的组合时,其效率不理想。

基于模型的 Native XML 数据库根据 XML 文档构造一个内部模型并存储这个模型。有些数据库将该模型存储在关系型或面向对象数据库中,也有的采用专为模型作了优化的存储方式,如果按照数据存储的顺序读取文件,则定义了物理存储格式的基于模型的 NXD 可能有类似于基于文本的数据库的效率。基于文本的系统明显地在返回结果为文本时比较快,而基于模型的系统多数情况在传回的结果是 DOM 树时较快。

## 2.2 纯 XML 数据库的研究方向

存储大量 XML 数据,高效的查询能力对 NXD 是非常重要的。NXD 的查询技术还处于起步阶段,目前的 NXD 查询的一个重要的特点是它的查询语言的设计与它的数据模型是紧密相关的,实际上每一种 XML 数据模型都隐式地决定了查询语言的结构和语义描述。XML 查询语言,如 XPath<sup>[11]</sup>和 XQuery<sup>[12]</sup>都是将路径表达式作为核心内容。这种方法简单直接,但执行效率不能得到保证,尤其是在大数据量的情况下。对于 XML 数据的更新操作,无论在语言还是在操作方法上目前都没有一个统一的标准。更新操作从逻辑上是指元素的插入、删除和更新。关于 XML 数据的更新语言,W3C 目前还没有这方面的工作计划,XQuery 中也没有更新 XML 数据的描述,但民间团体 XML:DB Initiative(参阅 <http://www.xmldb.Org/>)则

给出了更新语言 XUpdate 的规范,但这是不是能被 W3C 所接受,目前还不清楚,因此 XML 数据更新也是一个重要的研究方向。

## 3 纯 XML 数据库系统 eXist

当前已经出现了不少的 Native XML 数据库产品,如 dbXml(它能够索引和存储 XML 文档集合)、eXist(它有依据索引的 XQuery 处理程序,可以自动进行索引,扩展的全文本搜索,XUpdate 支持并且它与现存的 XML 开发工具可以紧密地结合在一起)、OrientX(Native XML 数据库管理系统,以 Native 方式存储 XML 数据,保留 XML 数据的树状模型,并支持 XPath 和 XQuery 等 XML 查询以读取数据)等。

下面简单介绍一下 eXist 系统。eXist 是一个开放源码的 Native XML 数据库系统,它与现在流行的 XML 开发工具(比如 Apache 的 Cocoon 系统)紧密结合。eXist 覆盖了一个 Native XML 数据库应该具有的大多数基本功能,而且提供了其他一些先进的技术,比如对文本进行关键词检索,模糊查询和基于规则表达式的检索模式。它是一个轻量级的,完全用 Java 语言实现的,容易部署的数据库系统。eXist 提供了可插拔的存储后端(见图 3),可以把文档存储在内部的 XML 数据库或外部的关系数据库系统(如 MySQL)中,但 eXist 本身的设计目标是一个纯 XML 数据库系统。

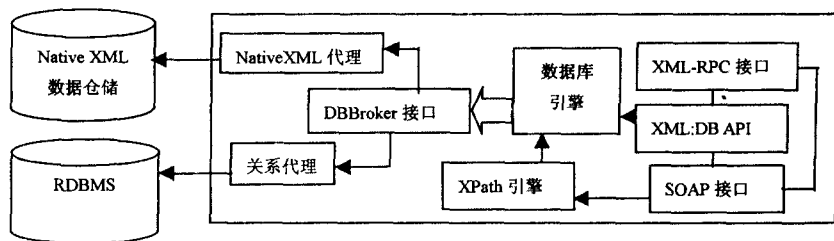


图 3 eXist 体系结构

eXist 具有以下几点特性:

1) 无模式的 XML 数据存储:系统中的文档并不需要与它们关联的模式或者文档类型定义。这样可以允许 XML 开发者在完成文档后再设计它的 DTD 或者模式。

2) 数据集合:数据集合可以随意嵌套,它们不和一个预先定义的模式相关联,所以一个数据集合中的文档类型的数目不受任何限制,在一个数据集合中,任意类型的文档都可以混合存储。可以使用 XPath 语法扩展直接查询数据集合层次结构中的一个指定内容,甚至是数据库中所有文档。

3) 基于索引的查询处理:eXist 使用了一个数字

索引模式,以识别所有的 XML 结点。这种索引模式不仅把 XML 存储中实际的 DOM 结点与索引入口连接起来,而且提供了对文档结点树中的结点之间可能关系(包括父-子结点关系或者祖先-子孙关系等)的快速识别功能。

4) 用于全文检索扩展: eXist 提供了一系列的扩展以有效地支持全文查询。一个附加的索引结构对文本内容中每个词的出现情况进行记录,帮助用户实现对文本内容的查询。系统还提供了特殊的全文查询操作符和函数,用于对文本结点以及属性值进行查询操作。

下面给出一个基本 eXist 系统查询 XML 的实例。首先,在 eXist 中创建一个新的集合(如 StuInfo),其次,把已经写好的 StuInfo.xml 文件导入到 StuInfo 集合中,StuInfo.xml 文件如图 4 所示。现查询年龄大于 25 的学生的学号、姓名、年龄并以 XML 文档的形式输出。

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE StuInfo [
3    <ELEMENT StuInfo (Student*)>
4    <ELEMENT Student (Id, Name, Age, Gender)>
5    <ELEMENT Id (#PCDATA)>
6    <ELEMENT Name (#PCDATA)>
7    <ELEMENT Age (#PCDATA)>
8    <ELEMENT Gender (#PCDATA)>
9  ]>
10
11  <StuInfo>
12    <Student>
13      <Id>10001</Id>
14      <Name>Millie</Name>
15      <Age>25</Age>
16      <Gender>Female</Gender>
17    </Student>
18
19    <Student>
20      <Id>10002</Id>
21      <Name>John</Name>
22      <Age>26</Age>
23      <Gender>Female</Gender>
24    </Student>
25
26    <Student>
27      <Id>10003</Id>
28      <Name>Jim</Name>
29      <Age>27</Age>
30      <Gender>Male</Gender>
31    </Student>
32
33    <Student>
34      <Id>10004</Id>
35      <Name>Tom</Name>
36      <Age>20</Age>
37      <Gender>Female</Gender>
38    </Student>
39
40    <Student>
41      <Id>10005</Id>
42      <Name>Sanny</Name>
43      <Age>23</Age>
44      <Gender>Male</Gender>
45    </Student>
46  </StuInfo>

```

图 4 StuInfo.xml 文件内容

查询语句如下。

```

for $ Student in //Student where (data($ Student/
Age)>25) return <StuInfo> { $ Student/Id } { $ Stu-
dent/Name } { $ Student/Age } </StuInfo>

```

查询结果如图 5 所示。

eXist 系统是一个开放源码的 Native XML 数据库

系统,系统还有许多有待改进的地方。在 eXist 系统中,包括多个文档资源对象的复杂的查询必须手动地分解为数个 XPath 表达,而且,某些类型的应用需要一个更加复杂的查询结果转换系统。eXist 目前并不支持对 DOM 中单独结点的修改工作,这对使用大文档的应用来说很明显是一个主要约束。

```

Results:
XML Trace
<StuInfo>
  <Id>10002</Id>
  <Name>John</Name>
  <Age>26</Age>
</StuInfo>
<StuInfo>
  <Id>10003</Id>
  <Name>Jim</Name>
  <Age>27</Age>
</StuInfo>

```

图 5 查询结果

#### 4 结束语

XML 数据库是当前研究的热点。文中结合 XML 技术和数据库技术探讨了 XML 数据库成为发展趋势的必然性,简单介绍了三种类型的 XML 数据库,并分析它们之间的特点,最后简单介绍了 eXist 系统。

#### 参考文献:

- [1] World Wide Web Consortium. Extensible Markup Language (XML) 1.0[EB/OL]. 1998-02. <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [2] Peter G A. 微软 XML 技术指南[M]. 谢君英, 译. 北京: 中国电力出版社, 2003.
- [3] 秦 杰, 杨树强, 窦文华. XML 数据库技术研究[J]. 计算机科学, 2003, 30(8): 6-9.
- [4] 龚红炎, 刘奕明, 陈涵生. XML 与数据库结合技术的探讨[J]. 计算机工程, 2005, 31(4): 114-116.
- [5] 董 东, 马 丽, 苏国斌. XML 数据库和关系数据库之比较[J]. 计算机工程与设计, 2005, 26(8): 2092-2099.
- [6] Chaudhri A B, Rashid A, Zicari R. 纯 XML 和支持 XML 的数据库系统[M]. 邢春晓, 张志强, 李骅竞等译. 北京: 清华大学出版社, 2006.
- [7] 逯 鹏, 良 双, 庆 一. Native XML 数据库技术综述[J]. 计算机科学, 2004, 31(4): 84-88.
- [8] 李亚伟, 段会川, 姚 进, 等. Native XML 数据库及其应用

(下转第 224 页)

钮,每个按钮分别对应第 3 和 4 帧<sup>[7]</sup>。

## 2.2 VC++ 软件模块

此软件模块主要含是基于 MFC 的表单视图多文档结构<sup>[8]</sup>。CPage1View 的关键在于利用 Flash 的多帧控制功能,数据的读写接口是 Flash 中的变量,并控制帧间的跳转。在 CPage1View 类中关键的函数是 OnTimer(),下面就是定时器函数程序简介。图 4 为其流程图。

```
void CPage1View::OnTimer(UINT nIDEvent)
{
    int CurrentFrame = m_swfPage1.CurrentFrame();
    if(CurrentFrame > 1)
        ReadCommonData(); // 读取公共数据
    if(CurrentFrame == 2) {
        OnButton1(); // 响应按钮 1 的函数
        m_swfPage1.GotoFrame(1); // 挪开此帧到转换停留帧
    }
    if(CurrentFrame == 3) {
        OnButton2(); // 响应按钮 2 的函数
        m_swfPage1.GotoFrame(1); // 挪开此帧到转换停留帧
    }
}
```

注意在此 CurrentFrame() 函数取到的 FrameNumber 帧计数从 0 计数,而 Flash 中的帧计数是从 1 计数,所以要清楚它们之间的这种对应关系。另一方面,语句 GotoFrame(1); 则是将 Flash 动画帧设置到第二帧。图 4 所示的帧控制层的第一帧称为初始化帧,第二帧称为转换停留帧,其他称为按钮对应帧。

下面一段程序可以用在 OnTimer 函数中,来实现另外一种按钮功能的方法。这种方法用于帧的序号位置不变时,即点击按钮后到定时器响应帧的序号位置不变。否则在 Flash 帧间循环播放动画时出现紊乱,假设 m\_FileOpenFlag 的初始值为 1,按钮的功能使 m\_FileOpenFlag 置成 0,这样就造成 VC++ 程序还没读取 0 值就又变回 1。

```
str = m_swfPage1.GetVariable("m_FileOpenFlag");
if(str == "0") {
    m_swfPage1.SetVariable("m_FileOpenFlag", "1");
    ..... // 相应的处理代码
```

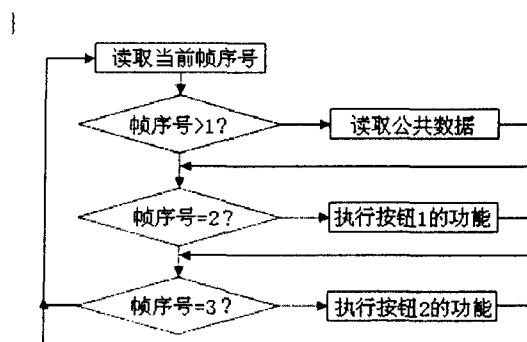


图 4 VC++ 程序定时器函数流程

## 3 结束语

探讨了基于 Flash 页面显示的 VC++ 编程在数据监控系统中的应用,及其相关的 Flash 制作、数据传递交互性和动态性的具体实现,通过实际的应用和测试,成功实现了通过 Flash 页面对监视数据的显示和对设备的控制。

文中也为类似的软件项目开发提供了一种思路和参考模板。

## 参考文献:

- [1] Penner R. Programming Macromedia Flash MX[M]. [s.l.]: McGraw-Hill, 2002.
- [2] Chien C C. Professional Software Development with Visual C++ + 6.0 & MFC[M]//Delmar Thomson Learning. [s.l.]: [s.n.], 2001.
- [3] Chun R, Robertson H P. Macromedia Flash 8 Advanced for Windows and Macintosh[M]. [s.l.]: Peachpit Press, 2005.
- [4] 马晓辉. 在 VC 中实现 Flash 动画播放[J]. 河北工程技术高等专科学校学报, 2005, 6(2): 48-50.
- [5] 李康满, 刘朝晖. 在 VC++ 中使用 Flash 动画技术[J]. 衡阳师范学院学报, 2005, 26(6): 86-88.
- [6] 杨勇. 调用 Flash 美化 VC 应用程序界面的实现[J]. 电脑知识与技术, 2006, 8(23): 168-194.
- [7] 边国栋, 谢矿生, 周小燕. 利用 FLASH MX 开发竞赛用计时器软件[J]. 微机发展(现更名: 计算机技术与发展), 2003, 13(1): 17-19.
- [8] 赵剑秋, 朱明. 用 VC 实现控制面板应用程序[J]. 计算机技术与发展, 2006, 16(6): 110-112.

(上接第 221 页)

- 研究[J]. 计算机应用与软件, 2004, 21(4): 29-31.
- [9] 刘刚, 喻成. Native XML 数据库的研究与应用[J]. 微机发展(现更名: 计算机技术与发展), 2005, 15(8): 65-67.
  - [10] Staken K. Introduction to Native XML Databases[EB/OL]. 2001-10-31. <http://www.xml.com/pub/a/2001/10/31/nativexmldb.html>.
  - [11] Clark J, DeRose S. XML Path Language(XPath)[EB/OL]. 1999-11. <http://www.w3.org/TR/xpath>.
  - [12] Chamberlin D. XQuery: A query language for XML W3C working draft[EB/OL]. 2003-12. <http://www.w3.org/TR/xquery/>.