基于 XML 的数据交换与存取技术研究

王　仲　陈晓鸥
（北京大学文字信息处理技术国家重点实验室　北京大学计算机研究所，北京　100871）
E-mail：wzh@icst.pku.edu.cn

摘　要　可扩展置标语言 XML(eXtensible Markup Language)是由 W3C(World Wide Web Consortium)组织于 1998 年 2 月制定的一种面向 Internet 应用的置标语言。作为对 SGML(标准通用置标语言)的一种改良，XML 具有良好的扩展性和自描述性、形式与内容分离、遵循严格的语法要求以及提供对多语种的支持等特点，使其成为 Internet 网上发布与数据交换的一门新兴技术，并有望在跨平台跨地域异构应用间的协同工作、基于语义的智能数据搜索等领域发挥重要作用。所有这些应用都又和 XML 的数据存取机制分不开的。基于上述种种原因，近年来，基于 XML 的数据交换与存取技术成为数据交换和存取领域的一项重要课题，并引起广泛关注。文章将此技术展开分析，详细探讨了 XML 数据存取机制，并结合关系型数据库和面向对象数据库，重点分析了 XML 在数据库中的存储模式和应用模式，最后对 XML 数据存取技术的发展趋势进行了展望。

关键词　XML　数据交换与存取　关系型数据库管理系统　面向对象数据库管理系统

文章编号　1002-8331-(2001)24-0108-04　**文献标识码**　A　**中图分类号**　TP311

Study of Data Exchange and Access Based on XML

Wang Zhong　Chen Xiaoou
(State Key Laboratory of Word Processing Technology, Institute of Computer Science and Technology, Peking University, Beijing 100871)

Abstract: XML (eXtensible Markup Language), a new Internet-oriented markup language, was put forward by W3C (World Wide Web Consortium) in February 1998. As an improvement to SGML (Standard Generalized Markup Language), XML brings some characters with itself, such as good extensibility and self-description, separation of content from presentation, conforming to rigid grammar definition and supporting multi-encoding character, which make it a novel technology in the fields of Web publish and data exchange on the Internet. Due to these advantages, XML-based data exchange and data access has become an important subject in relative fields in recent years. According to the technology, the theory and method of data access based on XML are studied in detail in this paper. Emphasized on the RDBMS (Relational Database Management System) and the OODBMS (Object-Oriented Database Management System), the model of access and application in database are analyzed. In addition, the tendency of data access technology based on XML is also predicted in the paper.

Keywords: eXtensible Markup Language (XML), Data exchange and access, Relational Database Management System (RDBMS), Object-Oriented Database Management System (OODBMS)

1　引言

可扩展置标语言 XML(eXtensible Markup Language)是一门新兴的面向 Internet 应用的置标语言，它是由 W3C(World Wide Web Consortium)组织于 1998 年 2 月制定的一种通用语言规范。XML 是 SGML(标准通用置标语言)的一个子集，其最大优点在于适合网上发布和数据交换。另外，作为对 SGML 语言标准的一种改良，XML 凭借其良好的扩展性和自描述性、形式与内容分离、遵循严格的语法要求以及对多语种的支持等特点，给跨平台跨地域异构应用间的协同工作、基于语义的智能数据搜索等重要领域带来重大突破。

从整体上讲，XML 定义了应用间所传递数据的结构，而且这种结构的描述不是基于二进制的、只能由程序去判读的代码，而是一种简单的、能够用任何编辑器读取得文本。利用这种机制，程序员可以制定底层数据交换的规范，然后在此基础上开发整个系统的各个模块，而各模块之间传输的数据将是规范的符合既定规则的数据。另外，XML 还允许为特定的应用制定特殊的数据格式，使其非常适合于在服务器与服务器之间传送结构化数据。

2　XML 数据存取机制

XML 数据源多种多样，根据具体的应用，大概可分为下面三种：一种是 XML 纯文本文档，第二种是数据库，第三种则来源于其他各种带有一定格式的应用数据，如邮件、目录清单、商务报告等等。其中，第一种来源，即 XML 纯文本文档是最基本的也是最为简单的，将数据存储于文件中，其最大的优点在于可以直接方便地读取，或者加以样式信息在浏览器中显示，或者通过 DOM 或 SAX 接口编程同其他应用相连。第二种数据来源是对第一种来源的扩展，其目的是便于开发各种动态应用，其优点则在于通过数据库系统对数据进行管理，然后在利用服务器端语言(如 ASP、JSP、PHP、Java Servlet 等)进行动态存取。

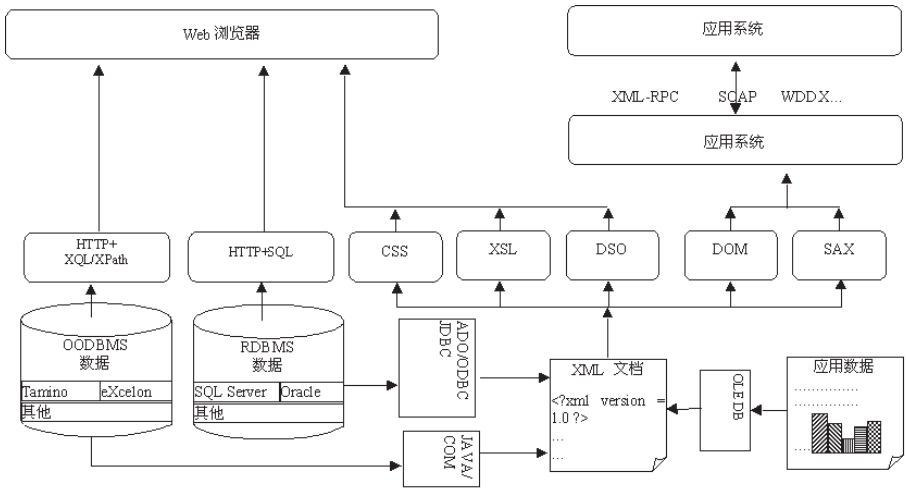

图 1　XML 数据存取机制

这种方式最适合于当前最为流行的基于三层结构的应用开发。第三种数据由于来源广泛，微软为此提供了基于 OLE DB 的解决方案，使得各种桌面应用系统可以通过 OLE DB 直接导出 XML 数据文档。该文的分析主要针对前两种数据来源进行分析。图 1 是 XML 数据存取机制示意图。

对于 XML 文档，可以通过 DOM（即文档对象模型）读取 XML 文档中的节点，这是最基本也是最底层的 XML 存取技术。DOM 是 W3C 推荐的一种技术标准，实际上是提供一组 API 来存取 XML 数据。它可以通过 JavaScript、JScript、VBScript 等脚本程序来调用，也可通过 C++、Java 等高级语言来调用。DOM 为开发基于 XML 的应用系统提供了便利。它通过一种随机访问机制，使得应用程序利用该接口可以在任何时候访问 XML 文档中的任何一部分数据，也可以对 XML 文档中的数据进行插入、删除、修改、移动等操作。另外一种 XML 编程接口是 SAX，它是由著名的 XML 讨论组 XML-DEV 开发的。SAX 提供了一种对 XML 文档进行顺序访问的模式，这是一种快速读写 XML 数据的方式。SAX 接口是事件驱动的，当使用 SAX 分析器对 XML 文档进行分析时，就会触发一系列事件，并激活相应的事件处理函数，从而完成对 XML 文档的访问。

另外，XML 文档可以通过赋予一定的样式信息进而在 Web 浏览器中显示。这一工作可以由 CSS 和 XSL 来完成。CSS 技术早在 HTML 3.2 中就得以实现，其关键是将 HTML 中的元素同预先定义好和一组样式类相关联以达到样式化的目的，而 XML 同样也支持这种技术。XSL 同 CSS 有些类似，不同之处在于它是通过定义一组样式模板将 XML 源文档转换成 HTML 文档或其它 XML 文档。XSL 实际上也是符合 XML 规范的，它提供了一套完整的类似控制语言的元素和属性，可以进行文本置换、排序、根据内容决定显示方式等等，最终可完成丰富多彩的样式描述。

其次，通过 DSO（Data Source Object）进行 XML 的数据绑定也可以方便地将 XML 节点同 HTML 标记捆绑，从 XML 文档中读取或写入数据，就象访问 Microsoft Access 或 Microsoft SQL Server 一样，最后将结果在 Web 浏览器中显示。DSO 的工作方式有几种，一种是同 DOM 类似，即通过对 XML 节点树进行遍历来搜索节点，每次仅将节点数据同 HTML 的一个元素（如 SPAN 元素）相关联；第二种方式同第一种的不同之处在于将节点数据同一个 HTML 多值元素（如 TR 元素）相关联。

基于 HTTP 协议通过在 URL 中嵌入 SQL 语句是关系型数据库存取技术上的一大进步，也是微软新近提出的 XML 数据库解决方案的核心，其基本原理是通过基于 HTTP 协议的 URL 方式直接访问 SQL SERVER 数据库，并返回以 XML 或 HTML 数据格式的文档，最后加以样式化或直接在浏览器中输出。目前，大多数数据库厂商均提供了对这一 XML 数据存取机制的支持。对于面向对象数据库来说，这一过程是通过在 URL 中嵌入 XQL/XPath 语句来完成的，而这又进一步体现了面向对象技术在 XML 数据存取中的优势，因为此时数据是被视为对象并按层次结构进行操作的[10,11]。

另外，关系型数据库中的数据也可以通过编程来输出 XML 文档。同 HTTP+SQL 机制相比，虽然在实现上较为复杂，但它提供了一定的可操作性。对于一个 C++程序员来说，编写一套访问数据库的 XML 应用程序可能需要利用 ODBC 和 C++ XML 语法分析器；而对于一个 Java 程序员来说，可能只需要 JDBC 和 Java XML 语法分析器就够了。利用 ASP 在页面文档中嵌入 ADO 对象从数据库中提取 XML 数据是微软对其 ASP 技术的一种扩展，ADO 取得数据后，可以调用 DOM 提供的 API 来动态生成 XML 文档，并进而同其他应用交换数据，或者直接在浏览器中显示。对于基于 XML 的面向对象数据库来说，大多都提供一套相应的开发机制或开发包，帮助开发者创建各种不同的应用系统，例如一组基于 Java 或 COM 组件的 API 作为对 Client 或 Server 端应用的扩展[10,11]。

然而，从现实的角度来看，一个完整的应用系统往往需要由许多小的子系统相互协同合作，才能最大限度地处理各种信息，这也就是所谓的分布式应用系统。在这种情况下，必须存在一种平台级的数据存取机制来保证数据的协调一致，否则，各个子系统就只能是一个个的“数字孤岛”。我们知道，作为 Web 应用的“灵魂”的 HTTP 协议，使得 Web 服务器和浏览器之间可以传输各种各样的内容，不论是简单的文本，还是形式丰富的多媒体信息。但是，从另一方面来看，越来越多的互联网应用采用 RPC（远程过程调用）进行数据交换，而 HTTP 本身显然是并不适合的。相反，在这种应用中，一些分布式对象协议，如 DCOM、IIOP/CORBA 则大行其道，但它们又会面临另外的难题——防火墙。作为互联网上的一种安全策略，防火墙一般会根据协议的端口号对来访的数据请求进行控制，而大多数的分布式对象协议通常使用动态生成的端口号，因此实际上会造成

同防火墙的冲突。为了克服这一问题,一种全新然而也是非常直观的解决方案–SOAP–诞生了。SOAP(即简单对象访问协议)的核心是在 HTTP 消息体的请求和应答数据中引入 XML 结构,不过依然以 HTTP 作为数据的载体,通过 POST 命令发送数据,利用 GET 命令接收消息。SOAP 最初由微软提出,后经 IBM 及 Lotus 参与修改,于 2001 年 5 月初提交 W3C 组织,并受到众多知名厂商的支持。除了 HTTP 协议之外,SOAP 也可以同其他协议配合使用[9]。

WDDX(即 Web 分布式数据交换)是由 Allaire 发布的旨在解决 Web 应用间传输关键数据的一项技术,它完全基于 XML 1.0 标准,在数据传输方面,广泛支持基于原文数据传输的协议,如 HTTP、SMTP、POP、FTP 等等。一般来说,任何需要通过 Internet 同其他应用共享数据的应用都可以利用 WDDX 来构建。比较典型的应用是 Web 网络联盟和企业对企业内部网和外部网应用,因为这些应用都需要进行数据发布,包括产品信息、供求信息、客户数据、订单数据等等。另外,WDDX 对于那些连接传统 Windows 桌面系统和 Web 系统的应用也是适合的[6]。

3 XML 与数据库技术

作为一种数据存储与交换的模式,长期以来文件系统在信息领域占据主导地位。但现在,越来越多的行业都在逐渐将关键数据放置于数据库中进行管理,一来目前数据库技术已经相当成熟,二来其管理功能的确非常强大。以往的数据库应用,基本上都是基于 C/S 模式,数据底层结构一般来说都是相对固定,也就是说,开发出来的应用程序是针对具体的数据结构,开放性较差,应用范畴也受到一定限制。而 XML 作为一种可扩展性置标语言,其自描述性使其非常适用于不同应用间的数据交换,而且这种交换是不以预先规定一组数据结构定义为前提的,因此具备很强的开放性,具有广阔的应用前景。为了使基于 XML 的业务数据交换成为可能,就必须实现数据库的 XML 数据存取,并且将 XML 数据同应用程序集成,进而使之同现有的业务规则相结合。正是由于这一原因,基于数据库的 XML 存储模式越来越受到数据库厂商以及相关研究人员的重视,而一些所谓的支持 XML 的或基于 XML 的数据库系统也相继推出。但总的来说,不外乎两种类型:一是以关系型结构为核心的“转型”关系型数据库(或称做对象关系型数据库),二是以面向对象技术为核心的“Native”面向对象数据库。

3.1 XML 与关系型数据库

从体系结构上看,数据库技术的发展历经了网状数据库、层次型数据库、关系型数据库、面向对象数型据库。到目前为止,在各个领域使用最广的还是要数关系型数据库。关系型数据库管理系统(RDBMS)采用二维表格作为存储数据的基本模型,表格由行和列组成,一般情况下,列被称作“字段”用于表示组成数据有效信息的属性,而行则用于表示一条完整的数据记录。由于数据间的相关性可以通过表与表之间关键字(外键)来关联,由此产生了“关系”类型数据库的由来。

随着技术的进步,关系型数据库在底层模型中引入了“对象”概念,并以此作为支持 XML 的基础。在这种“转型”的数据库中,“关系”的概念仍然起着核心地位,并且在 XML 对象与外部应用系统之间扮演着“接口/转换层”的角色。SQL(Structured Query Language)语言在关系型数据库中所占据的查询语言地位也没有改变,只不过为了使其适应 XML 存取以及面向 Web 应用的需要,又添加了一些 XML 的语法,并且直接嵌入到 URL 之中(形如下面的语句(Microsoft SQL Server 2000 提供支持),该语句将查询表 Customers 中的 CustomerID 和 ContactName 字段,并将返回的结果集直接以 XML 格式输出在浏览器中)。

http://IISServer/northwind?sql=SELECT+'<ROOT>';SELECT+CustomerID,+ContactName+FROM+Customers+FOR+XML+RAW';SELECT+'</ROOT>'[7]

3.2 XML 与面向对象数据库

面向对象数据库源于计算机编程语言中的面向对象技术。同以往的结构化编程语言相比,面向对象技术提供了一种同现实世界更加贴切的表达方式,它利用封装技术将属性和方法集成与对象之中,并且借助继承和派生的概念将对象及其子对象紧紧联系在一起。图 2 示出面向对象数据库管理系统概念。

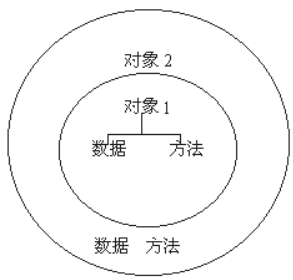

图 2 面向对象数据库概念示意图

面向对象数据库技术因为其技术的复杂性以及工业化成熟度不够曾一度陷入困境,但 XML 的诞生则又为其注入了新的生机和活力。面向对象数据库管理系统(OODBMS)使得文本、图像、视频和空间数据可以存储在数据库中,不过与关系型数据库不同:在关系型数据库中,数据仅仅是数据,它不包含层次结构信息;而面向对象数据库可以将数据视为对象,数据是作为一个整体,包含了属性和方法,并能体现数据间的继承关系。另外,面向对象数据库技术不使用 SQL 查询语言,在基于 XML 的数据存取应用中,多采用 XPath、XQL 进行查询和定位。对于一组给定的文档,用户可以指定其中每一个文档在分层结构中的位置,就象它们存储于文件系统之中一样。文档之间是相互独立的,尽管每个文档在数据库中可以仅仅存储一次,但是可以通过其存储路径来并借助 XPointer 和 XLink 机制对其进行关联。面向对象数据库系统也提供了基于 HTTP 的 XML 数据存取机制,典型的形如下面的语句(Software AG Tamino Server 提供支持),该语句将以 XPath 语法按 patient/name/surname 层次结构查询所有的 surname 数据。

http://wz2ksrv/tamino/mydb?_xql=patient/name/surname[10]

3.3 XML 在数据库中的存储模式

一般来说,XML 在数据库中的存储模式有三种类型,即:按结构层次拆分存入相应字段、作为 LOB 类型整存整取、作为数据对象存入数据库。

第一种类型是关系型数据库通常采用的存储模式。但是这样做的直接后果是在数据库的检索、索引方面会增加许多额外的工作,因为关系型数据库并不能很好地支持层次结构、顺序、包含等在结构化置标语言中十分本质的关系。尤其是随着 XML 文档节点层次的扩大,势必带来数据库中表与表之间更为复杂的关联关系,从而造成数据库执行效率的下降。另外,这样一来,XML 文档的整体性将受到破坏,除非有一个预先设定

的程序对数据库中数据进行整合，否则 XML 数据将变成一团糟。当然，如果将数据库字段作为元素的属性看待，也许这种做法到是可行的。但是，这种假设的前提是，该 XML 文档将只能表达简单的结构，对于复杂的文档就不太适用了。

第二种类型也是关系型数据库的一种存储模式。这种方式看似简化了操作，但在实际应用中，其应用环境将受到一定限制，因为关系型数据库不能很好地处理大容量的混杂以结构化信息和文本的数据，而且所能存储文档的大小要受数据库系统指标的限制，而不能无限制地扩大。

第三种类型是面向对象数据库所采用的存储模式。此时，XML 将不再被拆分而是被描述成一个对象存入数据库，其优点显而易见，XML 数据的结构和语义信息可以完整地保留下来。另外，由于 XML 数据在数据库中是作为对象来存储的，而面向对象数据库的特性又很好地保证了数据的可伸缩性。在这种情况下，数据可以任意地扩展和收缩，诸如节点元素和属性的增加或删除此类的修改都可轻松完成。相比之下，关系型数据库在处理时就复杂得多了，因为这种变化往往意味着数据库结构的调整，例如重新设计表格、追加记录等等。从这个意义上讲，XML 数据的结构化性使其更加适合于采用面向对象技术来处理，因此可以预言，基于 XML 的面向对象数据库系统必将成为未来的发展方向。

3.4 XML 在数据库中的应用模式

通常，XML 在数据库中的应用模型需要借助三层架构来实现。在这种模式下，一般会有一个代理程序运行于中间层，通过它来访问数据库系统中的数据并输出 XML 文档。代理程序实际上是一种在客户端桌面应用层与底层数据层之间传递数据的工具。利用 CSS 或 XSL 技术，XML 可以实现基于 Web 浏览器的多样式可视化显示。另外，这种代理程序还应该可以进行双向的基于事件的数据更新，也就是说，客户端的数据变化（如数据的插入、删除、修改等）可以通过代理程序反映到底层数据库，而数据库的更新也能够通知到客户端。表面上看，这种机制同传统的三层架构没有什么区别，但实际上是不同的，因为此时在传输过程中的数据都是已经 XML 化了的。

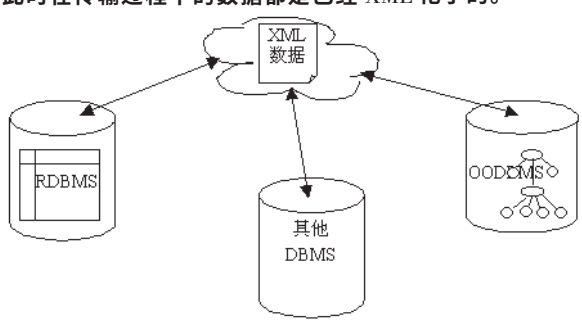

图 3 XML 与数据库交互示意图

XML 提供了一种连接关系型数据库和面向对象数据库以及其他数据库管理系统之间的纽带（见图 3）。XML 文档本身节点是一种有若干节点组成的属性结构，这种特点使得数据更适宜于用面向对象格式来存储，同时也有利于面向对象语言（C++、Java 等）调用 XML 编程接口访问 XML 节点。不同的数据库系统（包括关系型数据库和面向对象数据库）在进行数据交换之前首先需要将数据从数据库中提取出来，经过转换为或直接以 XML 数据形式发布到网上（局域网或 Internet 网），然后相互交换数据，经应用层系统处理后再转存入库。同其它面向应用的交换机制相比，它不仅对于同构数据库系统之间的数据交换是适用的，而且在异构数据库系统的处理上则具有更大的优势。

开发一个访问数据库的 XML 应用系统需要同时借助 XML 编程接口和数据库编程接口，前者用于对 XML 文档的解析、定位和查询，所需技术包括 XML DOM 和 SAX；后者则是用于访问数据库，如数据库中数据的更新和检索等等，需要利用的技术有 ODBC、JDBC、ADO 等。

4 支持 XML 的数据库

开发基于 XML 的动态应用，如动态信息发布、动态数据交换等，前提是必须有支持 XML 的数据库支持。在这一方面，Oracle 和 Microsoft 走在其他厂商的前面。号称全球第一大数据库及数据库应用解决方案提供厂商的 Oracle，早在 1999 年就率先推出支持 XML 的数据库产品－Oracle 8i。 而作为微软.Net 战略的一部分的 Microsoft SQL Server 2000 也正式提供对 XML 的支持，Web 开发人员无需进行复杂的数据库编程，只要在 Web 浏览器下输入一个 URL 地址，即可访问 SQL Server 数据库，而返回的结果可以是一个 XML 文档。另外，它还允许通过输入样式参数，指定模板文档，或借助 T-SQL 和存储过程进行更高层次上的数据处理。另一方面，作为面向对象数据库技术的代表，Software AG 和 eXcelon（原 Object Design）公司也各自推出了基于 XML 的面向对象数据库系统和应用开发平台。这些系统通过对 XML 语法分析器和 XML 引擎的封装从而高效地处理 XML 数据，通过同其它数据库系统的接口扩展系统的应用范围，通过对 XQL、XPath、XSLT 等规范的支持保证应用系统同业界标准的接轨，进而大大方便了跨平台、跨系统间异构数据的交换与存取。

XML 数据交换和存取技术为人们提供了广阔的开发天地，在实际开发过程中，应结合具体情况具体分析，采用最适合的存取模式。文章只是针对 XML 存取技术作一概述性分析，今后将结合具体的应用对这几种模式分别做更为细致深入的研究。（收稿日期：2001 年 8 月）

参考文献

1.Tim Bray，Jean Paoli，C M Sperberg-McQueen.Extensible Markup Language（XML）1.0 Specification[EB].http ://www.w3.org/TR/REC-xml，1998.2

2.DOM（Core）Level 1 Specification Recommendation[EB].http ://www.w3c.org/TR/REC-DOM-Level-1

3.CSS1(Cascading Style Sheet 1)[EB].http ://www.w3c.org/TR/css1.html

4.XSL1.0 Specification（Candidate Recommendation Apr.27，2000）[EB].http ://www.w3.org/TR/xsl/

5.Tim Bray.Using XML to Build the Annotated XML Specification [EB].http ://www.xml.com/xml/ xmlannotation.html

6.WDDX[EB].http ://www.wddx.org/

7.Microsoft MSDN online XML Developer Center[EB].http ://msdn.microsoft.com/xml/

8.Oracle Technology Network XML[EB].http://technet.oracle.com/tech/xml

9.Simple Object Access Protocol（SOAP）1.1.Don Box，etc[EB].http ://www.w3.org/TR/SOAP/，2000.5

10.Software AG Tamino Server[EB].http ://www.softwareag.com/xenon

11.eXcelon Server[EB].http ://www.exceloncorp.com/