第32卷 第20期 计 算 机 工 程 2006年10月
Vol.32 № 20 Computer Engineering October 2006
·软件技术与数据库· 文章编号：1000—3428(2006)20—0053—03 文献标识码：A 中图分类号：TP393

基于 XML 的多层分布式系统

刘兴伟[1]，崔 霄[1,2]
（1. 西华大学数学与计算机学院，成都 610039；2. 西安电子科技大学通信工程学院，西安 710071）

摘 要：针对传统的紧耦合 Web 系统模式存在的问题，该文提出了一种符合 J2EE 规范的、松散耦合的多层分布式系统，利用 XML 来分离业务逻辑和数据库。系统配置、数据传输以及层次之间的接口部分或全部由 XML 来实现，研究并实现了 XML 与异构数据源的转化机制，并将其应用到了 Web 应用系统——MEDLOG 中。

关键词：XML；分布式系统；Web 应用

Multi-storey Distributed System Based on XML

LIU Xingwei[1], CUI Xiao[1,2]
(1. School of Mathematics & Computer, Xihua University, Chengdu 610039;
2. College of Communications Engineering, Xidian University, Xi'an 710071)

【Abstract】 Aiming at the disadvantage of traditional, this paper presents close coupling Web application system model based on a J2EE and loose coupling multi-storey distributed system, which separates business logic layer from database layer. It implements system configuration, data transmission, inter-layer interface with XML technology. A data transfer strategies between XML documents and heterogeneous data source is discussed and implemented. The architecture is applied to the Web application system——MEDLOG.

【Key words】 XML; Distributed system; Web application

随着Web应用需求日益增多，Web应用系统也越来越复杂[1]，传统的系统模式即紧耦合对象系统越来越不能适应当前的需要。存在的主要问题是系统层次结构不清晰、应用开发需要大量编码、组件之间的紧密耦合以及数据存储的异构性导致应用系统的开发和维护变得异常困难[2]。

XML是W3C定义的一组规则，以无格式文本来描述结构化数据[3]。自 1998 年被W3C确定为标准以后，XML已广泛应用于数据表，从行业专用词汇、行业应用程序到描述Web服务中交换数据的协议如SOAP等，XML为构建松散耦合、语言和平台无关的多层分布式系统提供了良好的技术支持。

1 多层分布式系统的结构

作为当前基于组件的多层体系结构的主流规范，J2EE 把应用程序分为用户层、表现逻辑、业务逻辑和数据层，其应用开发一般采用 MVC(模型-视图-控制器)设计模式。

实际中如何很好地实现这种模式需要解决好以下几个关键问题[1,2]：

(1)用户层和表现逻辑之间存在用户访问安全性问题。虽然多数企业应用系统在设计中不同程度地实现了访问安全控制的功能，但这个功能其实是分散在各个应用程序之间，需要把安全访问部分抽象出来，作为系统结构的一个安全过滤子层，以解决包括用户校验、会话跟踪、数据访问控制等安全性问题。

(2)通过编写大量代码方式实现的控制层效率低下、代码冗余、不能很好地贯彻组件化设计的原则，造成系统维护困难。

(3)企业数据存放的异构性导致应用系统需要访问不同的数据源，需要根据业务类型接口抽象出一个数据访问子层而不管具体的数据源是平面数据文件、XML 数据文件还是关系型数据库，以实现应用程序对数据源的透明访问。

(4)EJB 提供了支持多种客户程序类型的事务管理与业务处理服务，但是服务器端的业务逻辑修改不可避免地会引起应用程序服务接口的改变。因此需要抽象出一个服务发布层，按照不同的用户请求提供不同的业务类型调用，以避免本地和远程的大量服务请求直接对业务组件的访问。

通过对 J2EE 规范的 MVC 设计模式进行适当的扩充，提出了一种多层分布式系统结构，如图 1 所示。

图 1 扩充的多层系统结构

从 MVC 的角度看，安全过滤子层在功能上属于控制层，服务发布层从属于业务逻辑层，数据访问层相当于数据库中间件的接口定义部分。但是这种结构最大的特点是充分利用了 XML 语言的特性，在组件之间以及各个层次之间使用

基金项目：四川省教育厅自然科学基金资助项目(2004B012)
作者简介：刘兴伟(1969－)，男，博士、副教授，主研方向：计算机网络及应用；崔 霄，博士生
收稿日期：2005-11-20 **E-mail：**xingweiliu@sohu.com

XML 配置描述，所有同数据库交互的业务逻辑都通过 XML 描述文件和 XML 数据文件方式实现，如图 2 所示。

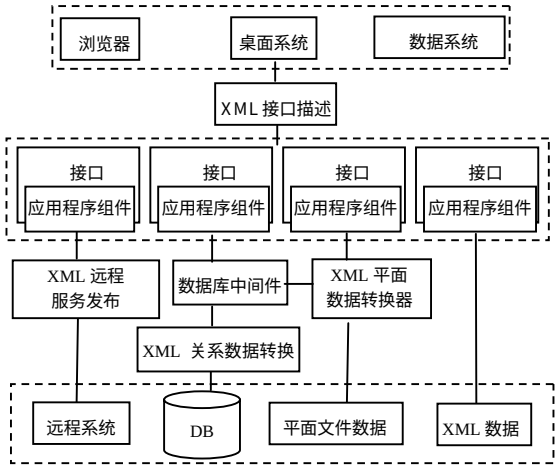

图 2 松散耦合结构

1.1 安全过滤子层

安全过滤子层主要通过一个过滤器来实现。过滤器对象由 J2EE 容器实例化和管理，并且有一个类似于 Servlet 的存活期，包括实例化、初始化、服务和销毁过程。主要功能如下：

(1)验证 HTTP 请求：如果一个 HTTP 请求是无效的(包含有无效的参数或参数值以及缺少必要的参数等)，那么过滤器可以终止处理，并向容器报告 HTTP 出错。这样可以避免在 Servlet 或者 JSP 网页中实现任何验证逻辑。

(2)记录 HTTP 请求：一般情况下，日志记录是由 Web 服务器来完成，但是 Web 服务器没有提供如何完成日志记录的控制机制。安全过滤子层可以根据 HTTP 请求的内容，实现自定义的全部 Web 资源访问日志记录，对请求进行扫描以决定相关的事件。

(3)用户认证：J2EE Servlet API 提供 4 种不同的认证用户的机制，包括基本认证、数字认证、基于表单的认证和 HTTP 客户认证。在安全过滤子层中，可以使用 J2EE 提供的标准 API 定义用户认证类型。

(4)HTTP 请求授权：典型情况下，Web 应用程序在每个 Servlet 和 JSP 网页中都要实现自定义的授权代码。在安全过滤子层中，可以在一个过滤器中实现自定义的授权，所有 HTTP 请求都包含这个过滤器。

1.2 服务发布层

根据 Java 的业务委派设计模式，可以抽象出一个服务发布层，它可以直接与各业务组件协同工作，为客户提供一个本地接口。服务发布层也可以作为客户方业务信息缓存的逻辑位置。如用来处理值对象、重试失败的调用以及对不同服务器的故障恢复等。客户程序通过 JNDI 找到服务逻辑，每个服务逻辑的名字和服务对象通过 Hash 表映射，把一个服务对象和一个名字关联起来。

1.3 数据访问层

J2EE 提供的实体 EJB 具有强大的并发数据访问能力以及持久性机制，但是用来列出数据库表的数据时却显得效率低下。假设要求列出一个分类目录中的数据，通常只需要获得只读数据用于浏览，系统并不严格要求数据的任何快照(瞬态值)都必须反映出相关数据的绝对最新状态。但是通过定义 EJB 的方法进行操作时，发现生成了多个 EJB 来完成这种简单的工作，这种情况对处理速度有较高要求的应用程序显得不是很合适。

因此数据访问层提供了一个自定义的数据访问对象 DAO。DAO 封装了定位和访问数据源所需要的代码，所有的业务或者表示逻辑都使用 DAO 来检索和存储数据。这样可以直接地访问数据而不需要通过与数据关联的实体 EJB。数据访问层的实现首先需要定义一些数据访问对象，这些对象用来将数据源记录转变为实体对象以及逆向转变。同时封装数据库调用并把 SQL 语句与应用程序其它部分进行隔离。实体对象和数据访问对象之间是一对一的关系，每一个实体对象有一个相应的数据访问对象，作为该实体封装数据库调用。数据访问对象使用数据库事务映射规则文件，定义数据库事务处理方法。数据访问层根据数据源形式决定调用方式，如果是关系数据库数据，则采用基于模板的 XML 文档与关系数据库映射实现数据库操作。如果是平面数据文件，则调用文件管理器校验输入/输出格式，同时引入/引出数据。

2 XML 数据转换机制

2.1 XML 数据与平面文件数据的转换

XML 数据与平面文件数据转换的模型如图 3 所示。

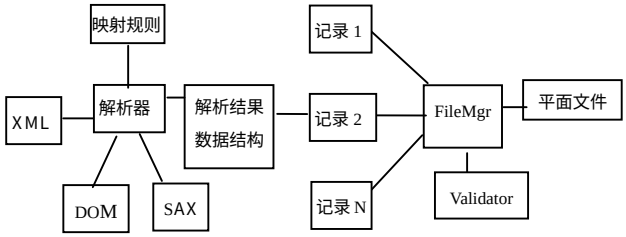

图 3 XML数据与平面文件数据转换模型

(1)确定数据结构。平面数据文件一般采用符号来分隔数据，定义了记录结束符、文件头和文件尾的数据指示符等。通过对文件格式的具体分析，可以确定数据结构，一般采用 List 或 Map 结构来存放结果对象。

(2)确定映射规则。即在平面文件对象模式和 XML Schema/DTD 之间建立双向映射，把一个 XML 文档表示为由数据对象构成的树，其中的每一个元素类型与平面文件对象模式中的对象一一对应。具体的映射规则如下：简单元素和单值属性直接映射到字段；复杂元素形成对象，对复杂元素包含的内容分类处理，如果是单值属性和简单子元素则映射规则同简单元素。若是复杂子元素，形成对复杂子元素对象的引用，当该元素形成对象后，把这个引用改成“对象——属于该对象的属性”关系来联结这个复杂子元素与其父元素；还可以采用 XML 文件中的根节点来唯一对应一个平面文件，多次重复的复杂元素对应一条记录，复杂元素的子元素对应字段，元素属性对应另一条记录中的字段的映射模式。上述映射规则也可适用于对关系数据库的转化。

(3)格式校验。对文件的长度、格式等进行有效性校验。

(4)文件操作管理。FileMgr 对象负责校验平面文件数据格式即读空文件错误处理、文件序列化失败等。同时负责管理数据转换方向即是把这些记录写入平面文件还是从平面文件中读出。

2.2 XML 数据与关系数据库的转换

XML 数据与关系数据库的转换方法主要有以下 2 类：

(1)基于模板的转换，直接将 SQL 语句嵌入到 XML 模板中，由专门的数据传输中间件来执行该模板。

(2)基于模型的转换，使用某种数据结构在 XML 文档模式和数据库模式(即 DTD/Schema 与 E-R 图)之间建立某种对应关系。

由于在关系数据库模型中，实体和属性之间有着截然不同的严格区分(这在 E-R 图中可以很明显地表示出来)，因此建立合适的映射规则尤其重要。关系数据库中的主键/外键关系可以用 XML 中的 ID/IDREF 来等价处理。另一方面从文本和 ID/IDREF 中可以获得 XML 文档中元素的嵌套结构，因此可以从单一表格中选取数据生成 XML 文档。同时对于多个表格关联的情况，只要把主键/外键关系映射成相应的 XML 文档元素或属性，就可以实现从多个关联的表格中获取数据，然后加以优化组合生成一个 XML 文档。关系数据库的数据转换为 XML 文档具体步骤如下：

(1)对应每一个表创建一个元素；

(2)对表中的每一列创建一个属性或者是一个只有 CDATA 内容的子元素；

(3)根据表中的每一主键/外键关系创建该表元素的子元素，通过关系表的主键、外键将各元素、子元素串接成树状结构的 XML 文档，然后合并属性；

(4)删除关系数据库中因规范化而导致的冗余，优化 XML 文档。

XML 文档转换成关系数据库数据的规则定义如下：

(1)简单元素和单值属性直接映射到表的列；

(2)复杂元素形成表，对复杂元素包含的内容分类处理。若是复杂子元素，则形成对复杂子元素对象的引用，当该元素形成表后，把这个引用改成 PK-FK(在用指针连接起来的数据关系中，将关系数据库中的 Primary keys 和 Foreign keys 之间的等值连接联系起来)；若是用“*”或者“+”形成的多个子元素，则该元素形成一个表，在父元素和子元素的表之间添加 PK-FK；

(3)如有多值属性，则需要创建一个单独的表来存储这些值，包含 FK 以及它的父元素所在表的 PK 形成的连接。

2.3 XML 数据与面向对象数据库数据的转换

在 XML 与面向对象数据库的双向映射中，由于 XML 文档和面向对象数据库的结构都是树型结构，因此这二者之间的映射比 XML 与关系数据库的映射要简单。MEDLOG 系统结构如图 4 所示。

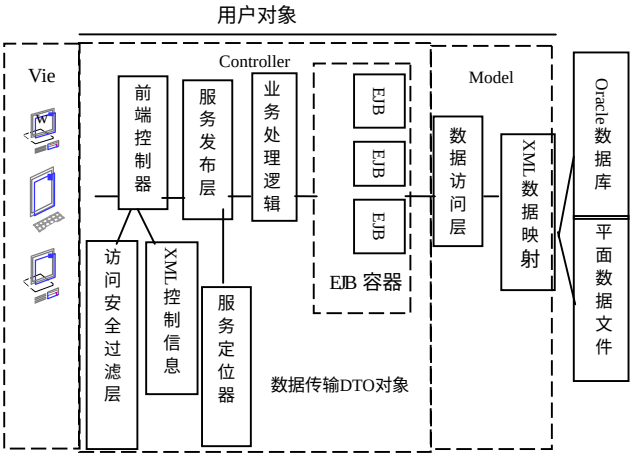

图 4 MEDLOG 系统结构

具体方法是把一个 XML 文档表示为由数据对象构成的树，通过解析 DOM 树，把每个节点作为一个对象，每个元素类型和对象模式中的对象一一对应。

3 应用示例

应用上述基于 XML 的多层分布式系统结构设计了 MEDLOG(Medical Log System)系统，其主要目的是实现就医人员的医疗资料备份，医疗设备维护和网上个人身体健康资料查询，医疗中心资产管理等，最终目标是实现全市乃至全国医院资料和资源共享，并每天自动备份所有就医人员的病历，定时备份医疗设备使用及维护信息等。如图 4 所示，整个系统具有良好的伸缩性和灵活性，各部分简介如表 1 所示。

表 1 MEDLOG 主要功能模块

序号	模块名称	序号	模块名称
1	参数设置	8	营业管理
2	组织机构	9	补给供应管理
3	预算分配	10	事故处理
4	主目录结构	11	设备维护
5	固定资产管理	12	明细管理
6	病历管理	13	借出/入管理
7	出入库管理		

(1)表现层用来处理用户与应用程序的交互。用户分为普通 HTTP 用户、本地用户和上级主管部门用户。不同用户可以分别通过 HTTP 浏览器和桌面应用程序登录。

(2)使用前端控制器调用安全控制层过滤器，对合法请求产生一个用户对象，同时记录用户信息；然后产生一个有状态会话 Bean 来跟踪用户执行记录。

(3)控制器根据 XML 控制信息描述，决定如何调用、处理逻辑和转发客户请求。

(4)业务逻辑使用相应的 EJB 产生数据传输对象 DTO；然后调用数据访问层 DAO，操作并返回数据资源。

(5)数据层实际上也就是资源管理层。分别使用文件服务器和数据库服务器存储数据。

根据客户需求和系统功能的要求，在确定了系统的体系结构之后，MEDLOG 系统可以划分为以下 13 个模块，见表 1 所示。MEDLOG 系统共设计了 200 多个数据库表和视图，其运行的硬件环境为 PIV 2.8GHz/256MB，软件环境为 Win-dows 2000 Server 版、WebLogic8.1、Oracle9i、Jsdk1.4、Ant1.6.1 等。系统经过严格的测试最终完成并交付使用，目前运行稳定，达到了设计要求和预期目的。

4 结束语

本文提出了一种基于 XML 的多层分布式系统结构，实现了 XML 数据与平面文件数据、XML 数据与关系数据库以及面向对象数据库之间的转换算法。实验表明，松散耦合的多层分布式系统可以降低应用系统开发和维护的难度，对于企业信息集成具有较高的应用和推广价值。如何进一步在 Web 应用体系结构中简化设计模式，降低组件之间的耦合度、提高数据转换的效率以及传输安全性都是进一步研究的重点。

参考文献

1 韦银星, 张申生, 周晓俊等. 企业应用集成技术研究[J]. 计算机集成制造系统, 2002, 8(8): 593-596.

2 李军怀, 周明全, 耿国华等. XML 在异构数据集成中的应用研究[J]. 计算机应用, 2002, 22(9): 10-12.

3 Bray T, Paoli J, Sperberg C M. Extensible Markup Language(3th Edition)[EB/OL]. http://www.w3.org/TR/ 2004/REC-xml-20040204, 2004.

4 淡猛刚, 李 剑, 樊会峰等. XTRANS: 一个 XML 与关系数据相互转化系统[J]. 计算机工程与应用, 2004, 40(19): 168-171.

5 Lee D, Mani M, Chiu F. Constraints-Preserving Inlining Algorithm for Mapping XML DTD to Relational Schema[J]. Data & Knowledge Engineering, 2001, 39(1): 3-25.