

# XML 及其在 MVC 模式中的应用

潘海兰, 吴翠红, 葛晓敏

(上海第二工业大学 计算机与信息学院, 上海 201209)

**摘 要:**通过图书馆管理系统来论述 XML 技术和 MVC 模式如何在 WEB 应用中结合。文章首先阐述了 XML 技术的相关知识及其优势,同时详细描述 MVC 模式的设计思想,并列举出 MVC 模式的优点和不足。其次论述了在 WEB 应用开发中如何结合 XML 技术和 MVC 模式来开发 WEB 应用程序,在此基础上结合图书馆管理系统实例来介绍如何使用 JDOM 开源框架来实现视图、控制层和数据模型的搭建,从而完成整个系统的开发。最后作者提出 XML 技术的发展前景,并对其与 MVC 模式的结合提出展望。

**关键词:**MVC 模式;可扩展标记语言;XSL 转化;文档对象模型

**中图分类号:**TP301

**文献标识码:**A

**文章编号:**1673-629X(2010)02-0202-04

## XML and Its Application in MVC Model

PAN Hai-lan, WU Cui-hong, GE Xiao-min

(School of Computer and Information, Shanghai Second Polytechnic University, Shanghai 201209, China)

**Abstract:** Discuss library management system to XML technology and MVC pattern combination in WEB applications. First of all introduce related knowledge about XML technology and its advantages, at the same time a detailed description of the MVC design pattern is described, and listed the advantages of MVC patterns and inadequate. Second, discuss how to combine the XML technologies and MVC pattern to develop WEB applications, and based on the examples of library management system introduce open-source framework for how to use JDOM to achieve view, control layer and data model structure in order to complete the entire system. Finally, propose the development prospects of XML technology and its combination with the MVC pattern to outlook.

**Key words:** MVC pattern; XML; XSLT; DOM

## 0 引 言

随着 Web 技术的快速发展,基于模型-视图-控制器(Model-View-Controller, MVC)的三层软件开发模式已在管理信息系统、电子商务等领域得到了广泛的应用。通过这种模式,可以开发一个具有伸缩性、便于扩展、便于整个流程维护的平台,可以很好地实现多用户系统中控制逻辑与表现逻辑的分离,在提高系统交互性的同时使代码的重复降到最低。可扩展标记语言(eXtensible Markup Language, XML)是用来定义文档标记语言的框架,它具有强大的数据描述能力和简便的数据交换能力。与现有的动态显示技术(如 JSP, PHP, .NET)相比,利用 XML 技术来实现 WEB 应用程序,可以提高系统的可扩展性和可维护性,此外还可以提高数据的重用性,减少网络负载<sup>[1]</sup>。在该文中,

通过图书馆管理实例描述了如何在 MVC 模式下利用 XML 来构建 WEB 应用程序。

## 1 XML 和 MVC

### 1.1 XML

XML 是一种脱胎于标准通用标记语言(Standard Generalized Markup Language, SGML)的可扩展元标记语言,它比 SGML 更容易操作,便于在 WWW 环境下实现;它支持 Internet 上有结构文档的交换,与 HTML 相比,XML 是面向内容的,它以一种开放、自我描述的方式定义数据结构,简单而易于掌握<sup>[2,3]</sup>。XML 正以其良好的格式、灵活的 WEB 应用、便捷的数据处理、面向对象的特点、开放的标准和数据的选择性更新等优越性受到业界的青睐。

XML 最大的一个优势就是利用 CSS, XSL(eXtensible Stylesheet Language)等技术实现数据内容与显示方式的分离,同时在不同的应用之间传输数据。其中 XSL 分为 XSL 转换(XSL Transformation, XSLT)和格式化对象(Formatting Object, FO),与 CSS 相比,它的

收稿日期:2009-05-18;修回日期:2009-08-07

基金项目:上海市教育委员会高校人才培养工程建设项目(200812)

作者简介:潘海兰(1979-),女,讲师,研究方向为电子商务、工作流。

使用更为简单,转换 XML 文档的功能更强。XSLT 能实现向输出文件中移动或添加新元素、重新排列或索引数据、对数据进行分组操作,并且能决定所要显示的元素;它运用 XPath 和 XQuery 技术将 XML 中的数据匹配到相应的样式。XSLT 的输出可以将 XML 数据格式转换为另外一种数据格式,比如 HTML 文档、PDF 文档,甚至是一段声音,也可以转换为需求程序中使用的 SQL 语句或 HTTP 信息等。FO 则提供了另一种格式化显示 XML 文档的方式,同时可以把样式应用到 XML 文档中。

对于存储在 XML 中的数据,可以利用文档对象模型(Document Object Model, DOM)与 XML 的简单 API(Simple API for XML, SAX)两种方式来解析,以达到读取、操作和修改 XML 数据的目的。DOM 为 XML 文档的已解析版本定义了一组接口,解析器读入整个文档,然后构建一个驻留内存的树结构,这样程序就可以使用 DOM 接口来进行各种操作,如遍历树、变换节点数据等。SAX 为任何一个 XML 解析器提供了基于事件的接口,程序员只要编写响应文档开始或结束、文本、元素开始或结束等这些事件的代码即可。DOM 的遍历能力较强,因而常用于 XML 文档需要频繁地改变的服务中。但是使用 DOM 时需要处理整个 XML 文档,所以对性能和内存的要求比较高,尤其是遇到很大的 XML 文件的时候。SAX 对内存的要求通常会比较低,因为它让开发人员自己来决定所要处理的 Tag,特别是当只需要处理文档中所包含的部分数据时,SAX 这种扩展能力得到了更好的体现。但用使用 SAX 编码工作会比较困难,而且很难同时访问同一个文档中的多处不同数据。当然目前也有许多结合这两种方法的优点而发展出来的简单易用的解析 XML 的方式,如 DOM4J、JDOM 等等,文中正是采用了 JDOM 来对 XML 文档进行解析<sup>[4]</sup>。

## 1.2 MVC 模式

### 1.2.1 MVC 设计思想

MVC 模式是 Xerox PARC 在 20 世纪八十年代为编程语言 Smalltalk-80 发明的一种软件设计模式,被用来构建用户界面,至今已广泛使用,众多 Web 开发框架都基于 MVC 模式进行设计。

在 MVC 设计模式中,MVC 的主要关系是由 Observer, Composite 和 Strategy 三个设计模式给出的。MVC 强制性地应用程序的输入、处理和输出分开。应用程序被分成三个核心模块:模型、视图、控制器,分

别承担不同的任务。图 1 显示了这三个模块各自的功能以及相互间的关系<sup>[5]</sup>。

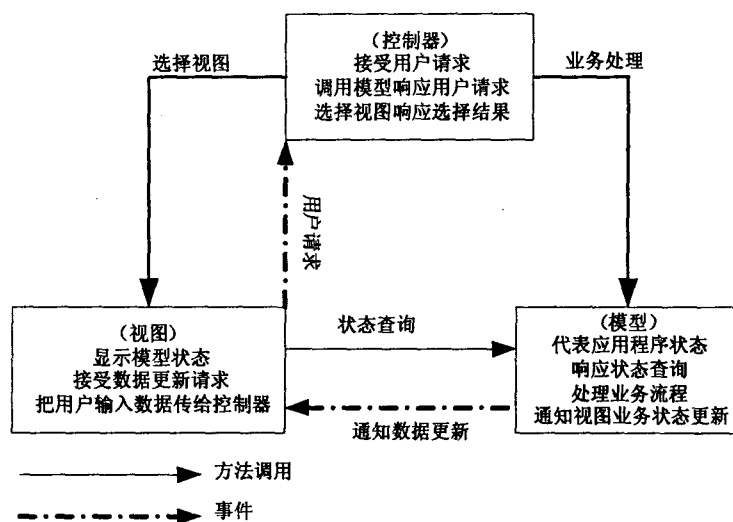


图 1 MVC 设计模式

从图 1 中简要地总结 MVC 的处理过程:首先控制器接收用户的请求,并决定调用哪个模型来进行处理;然后模型用业务逻辑来处理用户的请求并返回数据;最后控制器用相应的视图格式化模型返回的数据,并通过表示层呈现给用户。

### 1.2.2 MVC 的优点

MVC 的优点体现在以下几个方面:

(1)在目前用户需求的快速变化下,可能有多种方式访问应用的要求。按 MVC 设计模式,一个模型以及多个视图即可解决问题。这样减少了代码的复制,即减少了代码的维护量,一旦模型发生改变,也易于维护。其次,由于模型返回的数据不带任何显示格式,因而这些模型也可直接应用于接口的使用。

(2)由于一个应用被分离为三层,因此不需要改变所有,有时改变其中的一层就能满足应用的改变。一个应用的业务流程或者业务规则的改变只需改动 MVC 的模型层。而控制层把不同的模型和不同的视图组合在一起完成不同的请求,因此,控制层可以说是包含了用户请求权限的概念。

(3)MVC 思想还有利于软件工程化管理。虽然三个层次各司其职,但每一层不同的功能都具有某些相同的特征,这有利于通过工程化、工具化产生管理程序代码。

### 1.2.3 MVC 的不足

MVC 的不足体现在以下几个方面:

(1)增加了系统实现的复杂性。对于简单的应用,严格遵循 MVC,使模型、视图与控制器三者分离,会增加系统的复杂性,并可能产生过多的更新操作,降低系统运行效率。

(2)视图与控制器之间过于紧密的连接。视图与控制器虽相互分离,但视图如离开控制器,其应用是十分有限的,反之亦然,这样就妨碍了它们的独立重用。

(3)视图对模型数据的低效率访问。依据模型操作接口的不同,视图可能需要多次调用才能获得足够的显示数据。对未变化数据的不必要的频繁访问,也将损害操作性能。

## 2 XML 在 MVC 三个设计模式中的应用

以上简单介绍了 XML 和 MVC 这两个技术的重要知识点,那么 XML 可以在 MVC 的三层框架中起到怎样的作用?与传统的 MVC 实现不同,XML 作为一个数据流的载体贯穿整个系统的三个层次,有效地衔接了各层之间的数据交互。下面对每个层次分别说明 XML 技术的应用:

### (1) Model 部分。

业务模型的设计可以说是 MVC 最主要的核心。业务流程的处理过程对其它层来说是不可见操作,模型接受视图请求的数据,并返回最终的处理结果。目前流行的 EJB 模型就是一个典型的应用例子,它从应用技术实现的角度对模型做了进一步的划分,以便充分利用现有的组件,但它不能作为应用设计模型的框架。它仅仅告诉你按这种模型设计就可以利用某些技术组件,从而减少了技术上的困难。对一个开发者来说,就可以专注于业务模型的设计。

MVC 设计模式把应用的模型按一定的规则抽取出来,抽取的层次很重要,这也是判断开发人员是否优秀的设计依据。抽象与具体不能隔得太远,也不能太近<sup>[6]</sup>。MVC 并没有提供模型的设计方法,而只告诉你应该组织管理这些模型,以便于模型的重构和提高重用性。

采用 XML 作为数据存储的载体,让用户可以根据自己的需求定制 XML 文档,对应应用程序的实体。使得数据存储方便,易于维护。通过 Dom 或 Sax 模型对 XML 数据进行访问,结合 XPath 和 XQuery 技术来更好地拓展 XML 技术在 MVC 中的应用。

### (2) View 部分。

视图代表用户交互界面,对于 Web 应用来说,该项目结合 CSS/XSL 等技术,在 JSP 页面中使用 XML 技术,使得页面显示更加丰富生动。随着应用的复杂性和规模性,界面的处理也变得具有挑战性。一个应用可能有很多不同的视图,MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理,以及用户的请求,而不包括在视图上的业务流程的处理。业务流程的处理交予模型处理。比如图书馆借阅的视图只接受

来自模型的数据并显示给用户,以及将用户界面的输入数据和请求传递给控制和模型。

### (3) Control 部分。

一个模型可以对应多个视图,一个视图也可以对应多个模型。控制可以理解为从用户接收请求,将模型与视图匹配在一起,共同完成用户的请求。控制层就是一个分发器,选择什么样的模型,选择什么样的视图,可以完成什么样的用户请求,都是由控制器支配,但控制层并不做任何的数据处理。基于 XML 技术来实现 Servlet 在驱动 workflow 时对相应页面的选择<sup>[7]</sup>。

## 3 应用实例

以下结合一个图书馆管理程序来演示如何把 XML 与 MVC 相结合。在 Model 层主要采用 Javabeans 组件,分为 Action Beans 和实体 Beans 两部分,分别负责业务处理和数据处理,其中结合 JDOM 来对 XML 文档进行解析,并使用 XML 文件作为数据源;在 View 部分用 XML + XSL 来实现数据的发布;在 Controller 层,使用 Servlet 作为控制器,并结合 XML 技术实现了逻辑处理的封装和应用程序工作流的驱动。图 2 为本实例的体系结构示意图<sup>[8]</sup>。

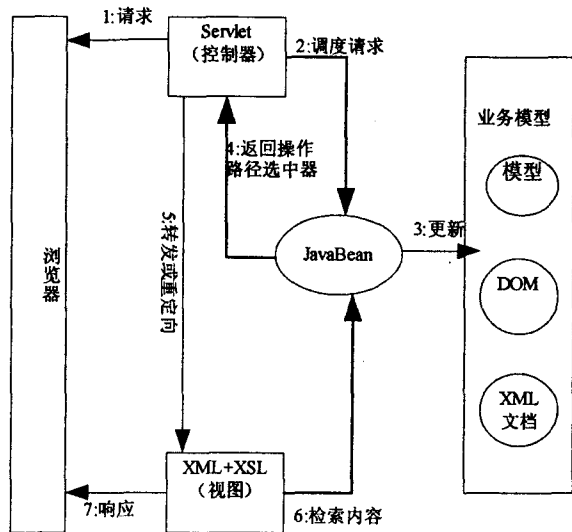


图 2 图书馆管理系统的体系结构示意图

### 3.1 用 XML + XSL 实现视图

用户视图显示方式是 HTML 文档,这里用 XML + XSL 来实现表现层,展示用户视图。可以通过 Castor XML mapping 把一个 XML 配置文件实例化为一个 Java 对象,通过访问 Java 文档,返回给客户端。用 XML 来展示视图,无需考虑用户使用的平台,可以提高页面的灵活性。

### 3.2 配置文件在控制层的作用

应用程序在执行某个逻辑时需要调用的若干

XML 文件称为配置文件。应用程序通过 JDOM 接口读取这些配置文件,通过获取的信息来进行相应的操作。例如库中的 Dispatch.xml 文件用来存放用户请求与相应处理模块之间的映射,控制器 Servlet 可以根据该映射将请求与模块相互对应来处理;另外还有用来描述页面显示格式的表单定义文件,应用程序可以根据所读取的表单描述格式,结合 XSL 文件,通过 XSL 转换器就可以输出 HTML 显示了;同样 XML 文档在生成或解析时也需要调用相应的 Schema/DTD 文件对其进行约束和验证<sup>[5]</sup>。

### 3.3 XML 数据访问使用的 JavaBean

#### (1) 模型类。

包括所有实体文件,如 BookXMLModel 为图书信息的实体文件。它们中的属性与 XML 文档中的节点元素相互对应,并都包含 setXXXX() 和 getXXXX() 方法。

#### (2) 模型抽象类。

即 BaseXMLModel,它是模型类的父类,便于所有模型类的统一管理。

#### (3) XML 访问接口抽象类。

即 BaseXMLAction,它定义了多个抽象方法:loadXML(...), newXMLNode(...), updateXMLNode(...), deleteXMLNode(...) 等对实体的操作。

#### (4) XML 访问接口类。

对每个实体的 ActionBean 分别实现父类 BaseXMLAction 的抽象方法,如图书信息实体为 BookXMLAction。

#### (5) XML 访问具体实现类。

即 XMLManager,在这个类中,结合 Jdom 实现所有类型的 XML 文档的访问,包括增加、修改、删除和查询等。

### 3.4 JDOM 对 XML 数据的解析示例

JDOM 包含来自广泛的 Java 开发人员的投稿、集中反馈及错误修复的系统,并致力于建立一个完整的基于 Java 平台的解决方案,通过 Java 代码来访问、操作并输出 XML 数据。JDOM 可以读入和输出 DOM 和 SAX 的内容,可以和很好地和已有的用 DOM 或 SAX 建立的系统之间整合。

对 XMLManager 类中的 XML 文档的处理,以下为其中删除操作的部分代码:

```
SAXBuilder sb = new SAXBuilder();
```

```
Document doc = sb.build(new FileInputStream(xmlFileName));
```

```
Element root = doc.getRootElement();
```

```
List list = root.getChildren();
for(int i=0; i< list.size(); i++)
{
    Element subElement = (Element) list.get(i);
    Element subSubElement = subElement.getChild("bookId");
    String text = subSubElement.getText();
    if (text.equals(bookId))
    {
        list.remove(i);
        saveXML(xmlFileName, doc);
        return true;
    }
}
```

## 4 结束语

目前,MVC 模式已经在软件设计中得到了极大的认同,同时已有许多优秀的开源项目(Struts, WebForm 等)来支持它,并且把 XML 作为它的数据存放的格式。当然 XML 技术也在迅速地发展和不断地完善中,特别是在 XQuery 技术的发展使得 XML 在许多领域内得到广泛的支持,拥有广阔的应用前景。对于电子数据交换、电子商务等更是把 XML 作为一种支柱性的技术来看待。因此如何更好地结合二者的优点,是今后研究的方向。

#### 参考文献:

- [1] 刘芳珠,潘亦,潘金贵. 基于三层网络架构及 DOM 的 XML 系统模型[J]. 小型微型计算机系统,2001,22(12): 1512-1515.
- [2] Bourret R. XML Database Products XML Content Management System[EB/OL]. 2007-10-10. <http://www.rp-bourret.com/xml/prodsCMS.htm>.
- [3] Rod J. J2EE development frameworks[J]. Computer, 2005(1):107-110.
- [4] 孙卫琴.精通 Struts:基于 MVC 的 Java Web 设计与开发[M]. 北京:电子工业出版社,2004:9-11.
- [5] Hu Tianlei, Chen Gang. Adaptive XML to relational mapping: an integrated approach[J]. 浙江大学学报: A 卷英文版,2008,9(6):758-769.
- [6] 卢菊平,郭江杰.精通 JSP+XML+CSS 网络开发混合编程[M]. 北京:电子工业出版社,2006:5-308.
- [7] 张杰,李秀娟,漆随平.基于 XML 的房地产管理信息系统设计[J]. 微型电脑应用,2007,23(3):30-31.
- [8] 黄凤爱.基于 XML 的图书信息管理系统的的设计及实现[J]. 现代情报,2008(1):114-115.