

# 基于 J2EE 和 XML 的构件库系统的设计

杨 森,曹宝香

(曲阜师范大学 计算机科学学院,山东 曲阜 273165)

**摘 要:**为了解决构件存储在不同数据库中存在的访问和控制不便的问题,在研究构件描述、分类、表示的基础上,选择合适的开发技术,设计了基于 J2EE 和 XML 的构件库系统实现对构件的有效管理。采用刻面分类模式描述构件,用 J2EE 的分层体系结构来搭建,用 XML 语言来表示和存储构件,使构件的存储和使用脱离了数据库,完全基于 Web 实现,增强了系统的灵活性和跨平台性,为项目中其它模块的快速开发提供了保障,具有一定的理论和实践意义。

**关键词:**构件;J2EE;XML;刻面分类;构件库

**中图分类号:**TP302.1

**文献标识码:**A

**文章编号:**1673-629X(2009)08-0100-04

## Design of Component Library System Based on J2EE and XML

YANG Sen, CAO Bao-xiang

(College of Computer Science, Qufu Normal University, Qufu 273165, China)

**Abstract:** In order to resolve the problem of the components which are stored in different databases that exist the inconvenience in access and control, on the basis of the research of component's description, classification and representation, choosed the right technology and designed a component library system based on J2EE and XML to manage the component effectively. Faceted classification model used to describe components, with J2EE structure of the system to set up, XML language used to express and store components, introduces a method fully based on web which can release the storage and use of components from the database, this will enhance the flexibility and cross-platform of the system and other projects fro the rapid development of the module provides protection. It also has a certain theoretical and practical significance.

**Key words:** component; J2EE; XML; faceted classification; component library

## 0 引言

软件行业的工业化趋势导致了软构件的产生。为了方便构件的管理和复用,应设计专门的构件库管理系统来有效地组织和管理大量可复用的构件,使基于构件的软件开发成为现实<sup>[1]</sup>。

文中主要研究构件的描述方法,构件的分类模型,构件信息的表示,构件在构件库中的存储、检索、删除和更新等内容,并在此基础上采用基于 J2EE 的分层体系结构来搭建,用 XML 语言表示构件的基本信息并进行存储,增强了系统的灵活性和跨平台性,具有一定的理论和实践意义。

## 1 关键技术

### 1.1 J2EE 技术

J2EE 是由 Sun 公司推出的一项中间件技术,提供了对 Java、Servlets、JSP、EJB 和 XML 技术的支持<sup>[2]</sup>。J2EE 使用多层的分布式应用模型,应用逻辑按功能划分为组件,各个应用组件根据它们所在的层分布在不同的机器上,其典型的四层结构为:客户层、Web 组件层、业务逻辑层和企业信息系统层。

由于 J2EE 的这种特点,笔者用 J2EE 标准的分层体系结构来搭建构件库管理系统,开发的构件也都基于 J2EE 的构件模型开发,符合 Java 语言语法的规范。

### 1.2 XML 技术

XML 是由 W3C 提出的一种可扩展标记语言<sup>[2]</sup>。由于 XML 具有独立于平台、可扩展性、自描述性等特性,故它已成为目前数据交换、电子商务和 Internet 等领域内应用程序数据表示和交换的事实上的标准。

由于 XML 可用于存储、交换和共享数据,因此,

收稿日期:2008-10-28;修回日期:2009-03-01

基金项目:国家自然科学基金资助项目(60072014);山东省自然科学基金资助项目(Y2003G01)

作者简介:杨 森(1982-),男,山东济宁人,硕士研究生,研究方向为网络数据库、构件和系统集成、面向服务的架构;曹宝香,教授,研究方向为网络数据库、中间件、CAD、数据库和系统集成。

采用 XML 来描述和存储构件信息,增强了系统和数据的互操作性,并且可以快速配置组件以适应新的系统和应用。

1.3 J2EE 与 XML 技术的结合

在 J2EE 架构中访问 XML,即强大系统架构和高度灵活的数据管理方案相结合,形成了一个强大的数据共享和处理的平台,并将其用在构件库管理系统中,进而增强了系统的开放性、可移植性、可维护性。

Java 有四种 XML 解析技术,即 DOM、SAX、JDOM 和 DOM4J。DOM 解析器把 XML 文档转化为一个包含其内容的树,但它需要处理整个 XML 文档,在处理较大的 XML 文件时,对计算机性能和内存的要求比较高;SAX 解析器采用了基于事件的模型,分析能够立即开始,而不是等待所有的数据被处理,应用程序只是在读取数据时检查数据,这对于读取大型文档来说是个巨大的优点;JDOM 可以对程序行为进行相当广泛检查以防止用户做任何在 XML 中无意义的事,简化了与 XML 的交互并且比使用 DOM 实现更快;DOM4J 的目标与 JDOM 是一样的,但 DOM4J 大量使用了 API 中的 Collections 类,付出了更复杂的 API 的代价,并且比 JDOM 更少强调防止不正确的应用程序行为。

综合比较了这四种解析器的特点,为了使构件库管理系统在运行时获得更好的性能,笔者采用更灵活、更易使用的 JDOM 实现对构件的添加、删除、修改、集成,用基于事件模型的 SAX 技术实现对构件的检索。

2 构件的描述、分类和表示

2.1 构件的描述

构件描述方法是指运用某种表示法描绘构件的抽象特征,它是基于构件的软件开发和构件复用过程中的一项关键技术,它规定了一个软件构件所需的结构和内容<sup>[3]</sup>。笔者采用了 BNF 文法<sup>[4]</sup>来对构件进行描述,描述构件的语法如下:

$\langle \text{Comp} \rangle ::= \langle \text{Concept} \rangle \langle \text{Context} \rangle [ \langle \text{Content} \rangle ]$

一个构件由概念、上下文和内容三部分组成。Concept 描述软件完成什么功能,描述构件的接口和语义,概念和内容(Content)能够进行通信;Context 是构件详细描述,描述上下文、组件在其适用领域内的配置,通过对概念、操作和实现特征的详细说明,上下文能够使软件寻找到满足应用需求的组件;Content 描述如何实现这个构件,一般情况下,构件的内容信息对临时用户是隐藏的,只有需要对构件作修改的用户才是可知的。

$\langle \text{Concept} \rangle ::= \langle \text{Comp\_name} \rangle \langle \text{Comp\_function} \rangle \langle \text{Comp\_interface} \rangle$

构件名按一定的命名规则命名,并能体现领域特色;构件功能部分描述构件能完成什么功能;一个构件可能有多个接口,有的构件有属性,描述构件接口的语法是:

$\langle \text{Comp\_interface} \rangle ::= [ \langle \text{Attribute\_set} \rangle ] \langle \text{Operation\_set} \rangle$

其中构件属性集和操作集的语法如下:

$\langle \text{Attribute\_set} \rangle ::= \langle \text{Attribute\_type} \rangle \langle \text{Attribute\_name} \rangle$

$\langle \text{Operation\_set} \rangle ::= \langle \text{Operation\_name} \rangle \langle \text{Param\_set} \rangle [ \langle \text{Operation\_type} \rangle ]$

操作由操作名、操作参数串和操作返回类型三部分组成。其中操作参数串可以包含多个参数,也可以一个参数也没有,其语法如下:

$\langle \text{Param\_set} \rangle ::= \langle \text{Param\_type} \rangle \langle \text{Param\_name} \rangle$

2.2 构件的分类

对大量构件进行分类既便于组织管理又方便查询和辅助理解。在众多的分类方法中,剖面分类模式具有准确的分类能力,能够解决构件分类的二义性和冗余问题,又具有可扩充性和灵活性,易于维护,是较为理想的构件分类方法。因此,在本构件库系统中采用了剖面分类方法。

如图 1 所示,每个构件(C)有一个描述(dc),是根据每个剖面(Fi)构成的有序术语集。每一个剖面与一个术语空间相关联,能充分并明确地描述构件库中全体构件,即每一个构件都可以用该剖面来分类。一个剖面的术语空间为有限的不定空间,即可以随应用的需要而动态地增加和删除术语。

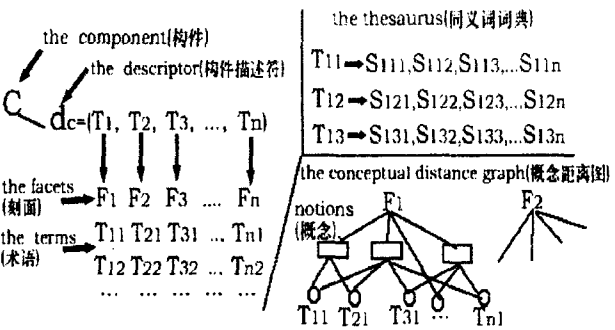


图 1 剖面分类模式的 3 个基本元素

笔者结合 XML 语言自身的特点,为系统设计了如下的剖面与子剖面(见表 1)。

表中五个剖面彼此间相互隔离,而且比较充分地体现了构件与复用相关的特性,能较好地适应构件库

今后的发展。五个刻画与其子刻画分别对应 XML 文件的元素和子元素。同时,设计中尽可能多地对元素术语指定枚举值,并在具体实现的系统界面中指定下拉列表选择,可在一定程度上减少软件术语差异产生的影响。

表 1 刻画分类表

刻画	子刻画	举例
构件形态 (CF)	表示媒体(RT)	源码、图形等
	抽象层次(LA)	分析、设计、编码等
应用环境 (AE)	操作系统(OS)	WindowsXP、Linux 等
	开发语言(CL)	Java、VC++ 等
构件功能 (CA)	应用领域(AD)	通用、商业、政府、制造、银行、办公、安全等
	功能分类(FC)	数据库、用户界面、业务逻辑、文件操作等
文件属性 (FP)	文件信息(FI)	文件名称,文件大小,制作日期,版本,作者等
	构件材料(CD)	构件实体、构件描述文件、构件级别等

对上述的刻画描述方案,将其中的刻画、子刻画分别映射为树中的父节点、子节点,对采用某个刻画描述方案描述的构件,将其刻画描述术语映射为对应的叶子节点,就得到一棵刻画树(见图 2)。

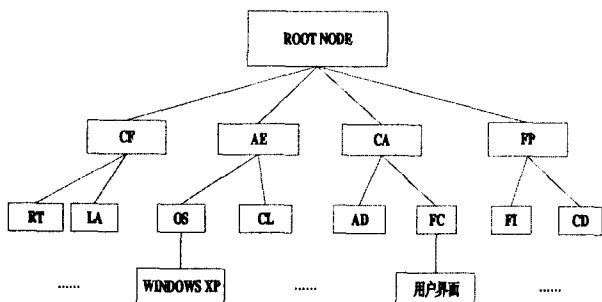


图 2 刻画描述树

### 2.3 构件的表示

构件描述可以展开为一棵刻画描述树,而 XML 语言本身的树结构特征,使得 XML 语言成为构件刻画描述的理想工具<sup>[5]</sup>。此外,对 XML 文件的验证技术有 DTD 和 XML Schema 两种。由于 DTD 不能很好实现应用程序不同模块间的相互协调,缺乏对文档结构、属性、数据类型等约束的足够描述,而 XML Schema 提供了一种模式定义语言,利用该语言来约束 XML 文档,这利于使用计算机对 XML 进行更准确的处理,并且它还是一种内容开放的模型,用户可以使用自己定义的元素和属性。因此,笔者采用 XML Schema 来实现对 XML 文档的验证。以下是用 XML 表示的构件基本信息:

```

<Component>
<ComponentForm>

```

<- 用来描述构件内容的语言形式或媒体 ->

```

<Represstation>源码</Represstation>

```

```

<LevelofAbstraction>Coding</LevelAbstraction>

```

```

</ComponentForm>

```

<- 使用、组装、修改构件时必须提供的硬件和软件平台及应用环境 ->

```

<ApplicationEnvironment>

```

```

<Operation System> Windows XP</Operation System>

```

```

< CompLanguage > Java</ CompLanguage >

```

```

</ApplicationEnvironment>

```

```

<ComponentFunction>

```

<- 构件的应用领域的名称 ->

```

<ApplicationDomain>通用</ApplicationDomain>

```

```

.....

```

```

</ComponentFunction>

```

<- 文件属性 ->

```

<FileProperty>

```

```

<FileInformation>

```

```

<FileName>标签</FileName>

```

```

.....

```

```

</FileProperty>

```

```

</Component>

```

## 3 构件库系统的设计与实现

笔者采用基于 J2EE 和 XML 的构件库管理框架模型(见图 3),其中用 XML 对全部构件进行描述,用相应的 XML Schema 文档对构件进行形式化验证。如果构件提供者所提供的构件通过验证,则可以加入或更新相应的构件文档。构件使用者可检索构件库中的构件,符合用户要求的构件可下载使用。为了避免构件过多导致 XML 文档过大,对构件分级进行存储。

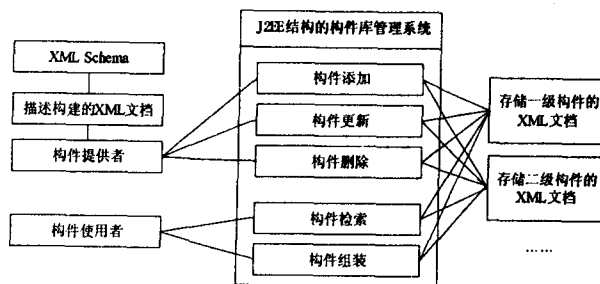


图 3 构件库管理系统架构图

### 3.1 构件的增加

用户在构件输入界面输入构件的信息,提交后把输入的信息生成 XML 文档,并验证 XML 文档的有效性,验证通过后进行构件添加操作,这部分功能封装在类 addComponent 中,具体实现如下:

```

Document doc = saxBuilder.build(new File("Components.xml"));
//解析文件
Element eltCom = new Element("Component"); //按照构件的表

```

示方法生成节点元素

```
Element eltComF = new Element("ComponentForm");
Element eltRep = new Element("Represstation");
.....
eltRep.setText(sourceCode); //把从构件输入界面得到的数据写
人相对应的元素中
.....
eltCom.addContent(eltComF); //把各个元素按照构件的表示方
法关联起来
.....
Element root = doc.getRootElement(); //得到 XML 文件的根节
点
root.addContent(eltCom); //把新增加的构件节点加入到根节点
下
.....
SchemaValidation validateCom = new SchemaValidation(); //验证
新增加的节点
InputStream xmlString = validateCom. getClass(). getResource-
AsStream("Components.xml"); //获取 XML 文件
InputStream schemaStr = validateCom. getClass(). getResource-
AsStream("Components.xsd"); //获取 XML Schema 文件
SAXReader reader = createSAXReader(schemaStr); //对验证文档
进行解析
Document document = reader.read(xmlString); //对 XML 文件进
行验证
.....
XMLOutputter xmlOut = new XMLOutputter(); //将验证后的文
档进行输出
.....
```

### 3.2 构件的检索

在 2.2 中采用刻面的方法对构件进行了分类,在文中笔者采用基于刻面的检索方法。用户在检索前先获得构件分类模式的详细报告,包括刻面和子刻面的数目,每个刻面的描述及刻面中的术语结构,找出自己需要检索的刻面。在检索构件时,用户根据查询需求从各个刻面中选择术语,组成合法的构件描述符。查询条件中有多个刻面时,各刻面检索到的构件集合之交构成最终结果集合。

一个结构良好的 XML 文件,其嵌套结构是完备的,每一个开始标签总会对应一个结束标签,而且不会出现标签嵌套之间的错位,因此可以使用栈来实现对文档结构的纪录。栈的特点是先进先出,笔者在 startElemnt() 中用 push 方法将刻面的名字添加到栈中,在 endElement() 中再用 pop 方法将其弹出。每一次 startElement() 方法的调用,必然会对应一个 endElement() 方法的调用,这样 push 和 pop 是成对出现的,只需分析栈的结构,用 characters 方法对匹配的刻面术

语做相应的处理,就可以知道当前标签所处在文档结构中的位置了。

### 3.3 构件的更新和删除

构件的提供者可以对构件进行更新或删除,首先检索出所有该用户提供的构件并显示出来,用户可选择相应的构件进行更新或删除。更新时,在更新界面修改构件的基本信息,然后对相对应的节点进行更新即可;删除时,调用 removeContent 方法把该构件节点删掉并存入回收站。

### 3.4 构件的组装

构件的组装采用可视化的组装环境,构件可以在此动态组装成应用程序,该环境分为三个部分,首先是存在一个构件树,用户可以从构件树上选择合适的构件进行组装。第二部分是构件组装区,用户从构件树中拖动想要的构件进行显示的连接进行组装,后台提能够自动组装功能,具体实现详见文献[6],此处不再赘述。第三部分是构件属性编辑区,在此区域可以动态地修改构件属性。

## 4 结束语

在研究构件的描述、分类、表示以及构件库体系结构的基础上,通过对构件库中构件模型和分类模型的研究,设计并实现了基于 J2EE 和 XML 的构件库系统。

和传统的构件库相比,本构件库采用 XML 语言表示构件的基本信息并进行存储,可以被更多的用户、更多的设备所利用;采用基于 J2EE 的分层体系结构来搭建,进而提高了系统的开放性、灵活性和可维护性,为项目中其它模块的快速开发做出了贡献。

### 参考文献:

- [1] 谷今杰,莫继红. 基于构件的软件复用技术研究[J]. 科学技术与工程,2005,5(12):824-827.
- [2] Knutson J, Kreger H. Web Services for J2EE, Version 1.0 [EB/OL]. 2002. <http://www.huihoo.org/openweb/web-services-for-j2ee/index-eng.shtml>. htm.
- [3] 姚全珠,李献令,孟 丽. 基于 XML 的构件库管理框架的研究与实现[J]. 计算机工程与应用,2006(21):78-80.
- [4] Prockick. RFC2234 - Augmented BNF for Syntax Specifications: ABNF [EB/OL]. 2002-01. <http://www.china-pub.com/computers>.
- [5] 郝春辉,邹 静. 基于 XML Schema 的 XML 存储[J]. 计算机工程与应用,2006(11):173-175.
- [6] 谢晓芹,李渭子,王 沛,等. 基于 XML 的构件自动组装系统的设计与实现[J]. 计算机工程与应用,2005(4):7-11.