

用 J2EE 技术实现虚拟软件企业构件库系统

龚永罡 石 峰 骆正华 (北京理工大学计算机系软件研究所 100081)

摘要:作为实现软件复用基础设施之一的构件库系统,在虚拟软件企业开发过程中扮演了极其重要的角色。本文结合虚拟软件企业支撑平台设计思想,通过对面向虚拟软件企业平台构件分类、组织模式、存储和检索算法等关键问题的分析,给出了基于 J2EE 技术的构件库管理系统的具体实现。

关键词:J2EE 构件库系统 虚拟软件企业 XML

1 引言

“虚拟企业”(Virtual Enterprise)概念的提出源于传统的制造行业^[1]。10 多年来,关于虚拟企业的理论和实践不断成熟,作为一种以核心能力为形式的、对企业外部资源进行优化整合的有效手段——虚拟企业,正成为各领域企业的现实选择。借鉴虚拟经营的思想,结合软件企业的特点,参与或建立虚拟软件企业,是国内软件企业提高软件工程化开发能力,快速聚集软件开发资源,把握市场机遇,参与国际软件业竞争的一条有效途径。

对于一个虚拟软件企业来说,软件构件库是其持续发展的核心竞争力之一。构件技术和基于构件的软件开发(Component-Based Software Development,简称 CBSD),是近几年来软件工程领域对软件复用研究的一项重要成果^[2]。要有效实现 CBSD,必须要有一个企业级的构件库管理系统来管理大批量的各种形式、分属不同领域、不同阶段的通用构件;另外,随着基于构件的软件复用技术的深入实践,虚拟软件企业内部软件开发过程中积累下来的软件产品资源,经过再工程之后,也要放入构件库中进行复用。因此,构建虚拟软件企业支撑平台基础设施之一的构件库管理系统,就变得十分重要。

本文通过对面向虚拟软件企业平台构件分类、组织模式、存储和检索算法等关键问题的分析,采用当今比较流行的 Web 技术,设计和实现了基于 J2EE 技术的构件库管理系统。

2 面向虚拟软件企业平台的构件库管理系统结构

面向虚拟软件企业平台的构件库管理系统按照浏览器/服务器(Browser/Server, B/S)方式,采用基于 J2EE 框架的 Web Service 技术实现,运行于互联网之上,其体系结构如图 1 所示。系统分为五层结构:



(1) 表示层:实现用户界面与应用程序调用接口;

(2) 控制层:通过一个全局的 ActionServlet(继承自 HttpServlet)截获用户请求和返回响应;通过 Action Beans 和 ActionForm Beans 实现业务控制和客户端的基本数据验证等;

(3) 业务层:通过 Java Bean、EJB Session Beans 实现恰当粒度的业务逻辑;

(4) 公共构件层:通用的基础操作,比如数据库 SQL 操作组件、文档上传下载组件等粒度最小的组件;

(5) 实体层:通过实现 EJB Entity Beans CMP 实现从对象到关系数据库的映射。

用户可以通过 Web 界面直接登录上来做相应操作,比如客户注册登录、发布构件、查找构件、下载、维护等;同时,也提供给虚拟软件企业支撑平台开发调用接口,通过此服务接口,可以实现平台内部基于构件的软件开发的完整性。因此,此系统不仅是面向虚拟软件企业平台基于构件的软件开发的基础,也是面向 Web 的构件开发组装平台和商务平台。

3 系统实现中关键问题和主要算法

构件库的本质问题是如何建立构件表示和分类模型,从而用精确的方法表达构件的语义,使 CBSD 过程中构件能够得到正确的理解和复用。下面就构件库管理系统设计中构件的分类、组织模式、存储和检索等关键问题和算法,提出了具体的设计方案。

3.1 构件分类策略——刻面分类

构件的分类有很多方法,信息科学方法是实际项目中应用较为成功的一类,包括枚举、刻面、属性值、关键词和正文检索几种常见方法,其中又以关键词分类(Keyword)和刻面分类(Faceted)两种应用最多。

关键词分类简单而且为人熟悉。每个构件以一组与之相关的关键词编目,查询者给出关键词来描述所需的构件并由此获得构件。由于关键词缺少上下文语境而不够精确,导致该方法所支持的查询效率不高。刻面分类方法是关键词(术语)置于一定的语境中,并从反映构件本质特性的不同视角(刻面)对构件进行分类。每个刻面中有一组术语,术语间由于有一般特殊关系和同义词关系而形成结构化的术语空间,允许术语之间有同义词关系。术语仅限在给定的刻面中取值,在术语空间中游历可以帮助复用者理解相关领域,术语空间可以演变。刻面分类方法能够提高检索效率,而且有助于复用和理解构件以及目标领域。与面对多个异质领域的构件库相比,刻面分类方法更适合于面向特定领域构件库^[3]。

本系统采取刻面分类为主,关键字描述为辅来描述构件。目前本构件库包含六个刻面,具体刻面与其简单术语的描述如表 1 所示。

3.2 构件的 XML Schema 描述

XML Schema^[4]是 W3C 的推荐标准,于 2001 年 5 月正式发布,XML Schema 一经确定下来,立刻成为全

球公认的首选 XML 环境下的建模工具。XML Schema 完全使用 XML 作为描述手段,具有很强的描述能力、扩展能力和处理维护能力。XML Schema 是用一套预先规定的 XML 元素和属性创建的,这些元素和属性定义了 XML 文档的结构和内容模式。它规定 XML 文档实例的结构和每个元素/属性的数据类型。

表 1 构件的刻面分类与术语描述

刻面分类	术语描述
构件平台	CORBRA, ActiveX, COM, .Net, Java Beans, EJB...
开发工具	VC, VB, C#, Java, eclipse, Delphi, C++ builder, PB...
功能	图象处理、文档上传、数据 Grid 显示...
计算机领域	网络、数据库、多媒体、虚拟现实、人工智能...
应用领域	企业管理、金融、电信、保险、工业控制...
运行操作系统类型	Dos, Windows95/98/me, WindowsNT/2000/XP, Unix/Linux, Mac OS...

这里给出一个 ComponentSpec.xml 实例文档示例,可以分析出这类文档的基本特征和属性,从而根据这些特点编写 XML Schema 文档,实现对此类文档的约束。

```
<? xml version = "1.0" encoding = "gb2312" ? >
<ComponentSpec>
<ComInfo>
<CID>1</CID>
<CName>Component for CLMS</CName>
<Chinese>构件库最小公共构件</Chinese>
<creator>Jessie</creator>
<Createtime>2005-5-17</Createtime>
<Introduction>用于电子商务系统的底层开发</Introduction>
<Version>1.0</Version>
<Is_Free>Y</Is_Free>
<Price>0</Price>
<Updatetimes>0</Updatetimes>
<Downloadtimes>10</Downloadtimes>
</ComInfo>
<ComFacets>
<Facet>
<fld>1</fld>
<fname>应用领域</fname>
<Glossary>
<gld>1</gld>
```

```

    <gName>计算机</gName>
  </Glossary>
</Facet>
<Facet>
  <fld>2</fld>
  <fName>开发工具</fName>
  <Glossary>
    <gld>3</gld>
    <gName>Java</gName>
  </Glossary>
</Facet>
</ComFacets>
<ERReports>
<ERReport>
  <eld>1</eld>
  <Text>集成时出现错误!</Text>
  <creator>weddu</creator>
  <createtime>2004-5-31</createtime>
</ERReport>
</ERReports>
<Feedbacks>
<Feedback>
  <fdid>1</fdid>
  <Text>这个构件不错!</Text>
  <creator>edith</creator>
  <createtime>2004-5-25</createtime>
</Feedback>
</Feedbacks>
</ComponentSpec>

```

构件详细信息 XML 文档有一个主元素 ComponentSpec 和四个子元素 ComInfo、ComFacets、ERReports 和 Feedbacks 组成。这些子元素也依次包含其他的子元素等等。比如说 ComInfo 还包括 CID、CNAME 等子元素; ComFacets 包含多个 Facet 的子元素。元素如果包含子元素或者带有属性则被称为复合类型, 比如 ComFacets, 反之元素如果仅仅包含数字、字符串或者其他数据, 但不包含任何子元素则称为简单类型, 比如 CID。在实例文档中复合类型和一些简单类型是在构件详细信息的模式文档中定义的。

3.3 基于关系数据库的存储管理

构件描述的 XML 数据存储方式大致可以分为三类:

- (1) 文件存储;
- (2) 数据库存储;

(3) Native - XML 数据库。关系数据库是迄今为止最为成熟的数据库管理系统, 使用数据库方式存储 XML 数据, 是一种比较成熟的方案, 通过 XML 模式与关系数据模型的映射, 将 XML 文档分成一些语义相同的、灵活的小单元进行处理, 从而将关系数据库的完整的访问控制和存储管理应用到 XML 数据的存储管理中。

3.4 构件查询算法

构件库系统除了对构件进行存储、增、删、改等日常维护外, 关键目标是支持用户高效、准确地发现他们所需要的构件。构件的检索一直是一个制约构件复用的关键因素^{[5][6]}。如果检索过程的效率太低, 或检索到的构件并不合适, 必然会使软件开发人员放弃使用可复用构件。

刻画从多个方面描述了构件, 每个刻画把构件集合分成一些子集, 各个刻画术语划分的子集形成了一些较小的交集。在检索一个构件时, 给出一组特征值, 将确定一组子集合, 构件应该在这些子集合的交集中, 我们称此交集为命中构件集, 其中的构件称为相似构件。

构件的 XML 描述数据, 存储在不同的数据库表中, 系统提供两种不同的检索方案, 即普通检索与高级检索。普通检索采用输入关键字的方式进行检索, 在进行高级检索时, 用户将面对相对复杂的查询条件的输入。除关键字外, 构件的任何属性, 例如作者、开发语言、入库时间、使用环境、版本号等都可能会作为查询输入条件, 构件查询的操作不再是简单的字符串匹配。对于不同特征的构件查询有着不同的算法描述, 以构件开发语言作为查询条件的构件查询的过程为例, 对构件查询算法加以说明:

算法: 构件普通检索算法

BEGIN

(1) 从构件基本信息表 component 中查询得到所有关键词描述 describe 字段包含“Java”的所有记录的主键值存在数组 componentID [] 中;

(2) 从构件刻画描述表 com_facet 中检索得到所有的刻画分类名称包含“Java”的纪录关联的 fk_com-

ponent 值;将其存入数组 componentID [] 中;

(3) componentID 为符合查询条件的构件 ID 数组;

(4) 遍历数组 componentID [], 从构件基本信息表 component 中查询主键为 componentID [i] 的记录, 将其返回给用户。

END

在进行普通检索时, 系统提示用户输入的若干个字符串, 然后在构件的描述信息当中以关键字匹配的方式进行比较, 当满足匹配时, 这个构件就会被收录到结果当中去并返回给用户。同时, 若用户输入的关键字也匹配构件的刻面分类命名时, 构件也会作为结果返回给查询用户。由于输入的查询描述语是若干个, 它们之间存在着逻辑与、或的关系, 这也和普通 web 搜索引擎相似。构件库查询模块就要对用户输入的字符串之间的逻辑关系进行相应的逻辑筛选。

高级检索采取的是关键字法与刻面分类法相结合的检索方式, 用户首先采用关键字对构件描述属性进行初次检索, 输入对需求构件的关键字描述, 根据模糊匹配的原理得出预检索结果, 然后在此基础上给出刻面相似度方法检索结果以供决策, 根据各构件的刻面特征进行选择, 通过二次检索得到更加精确的结果。

4 安全性的实现

本系统的用户分为普通用户、管理员和访客, 不同的用户有不同的权限, 登陆成功后才可以进行页面的浏览和操作。为防止未经注册的用户绕过登陆界面直接进入相关页面, 通常采用 Session 对象进行注册验证。Session 对象可以把用户登录后相关操作信息保留下来, 让后继的网页读取。使用 Session 对象进行注册验证, 每个 JSP 页面都应判断请求该页面的用户是否具有该页面的访问权限, 在系统页面少的情况下, 控制起来比较方便。但是一旦页面规模较多, 不同页面允许访问的用户角色类型和操作经常变化的情况下, 使用 Session 方法带来的工作量大、维护困难的缺点就暴露无遗。因此, 本构件库系统中, 设计了一种方便、简单、统一的权限校验方法——Filter 来实现页面访问的权限控制。

首先, 写一个配置文件 acl - config. xml, 在该文件中定义受保护页面以及可访问该页面的角色:

```
<protected - resource >
  <uri > EditAccount. do </uri >
  <desc > edit account </desc >
  <role > administrator, user </role >
</protected - resource >
```

由于 Filter 可对任何页面进行过滤, 在过滤过程中通过解析 acl - config. xml 配置文件来判断用户访问的是不是安全页面, 若是则首先判断用户登陆否, 若已登陆, 再从 session 中得到用户的 role, 并根据 acl - config. xml 判断用户是否有权限访问该页面。用 Filter 的好处是对权限进行了集中控制, 动态页面里的代码更加清晰, 不必写许多烦琐的权限判断。

5 结束语

作为实现软件复用基础设施之一的构件库系统, 在虚拟软件企业开发过程中扮演了极其重要的角色, 它不仅是基于构件软件开发的基础, 也是面向构件开发的组装平台和商务平台。本文结合“虚拟软件企业软件开发支撑与过程管理平台”课题, 使用 J2EE 技术, 设计完成了基于 XML 的分布式构件库管理系统的大部分功能, 随着课题的深入, 我们的构件库管理系统会越来越完善。

参考文献

- 1 Martin H, Richard B. The industrial virtual enterprise[J], Communications of ACM, 1997, 40(9): 59 - 61.
- 2 A. Brown. Large - Scale Component - Based Development. New Jersey: Prentice Hall, Inc., 2000.
- 3 R. P - Diazm, Implementing Facetted Classification for Software Reuse, Communications of the ACM 34 (5) 1991.
- 4 XML Schema Part 0: Primer (<http://www.w3.org/TR/xmlschemat-0/>).
- 5 Wang YF. Research on retrieving reusable components classified in facetted scheme [Ph. D. Thesis]. Shanghai: Fudan University, 2002 (in Chinese with English abstract).
- 6 常继传、李克勤、郭立峰、梅宏、杨英清, 青鸟系统中可复用软件构件的表示与查询, 电子学报, 2000, 28(8): 20 ~ 24.