

基于 J2EE 的通用 Web 信息系统框架设计与实现

Design and Implementation of Universal Web Information System Frame Based On J2EE

陈 军 (中南大学网络中心 湖南长沙 410083)

段辉良 (中南大学现代教育技术中心 湖南长沙 410083)

摘要:本文描述了一个通用的 Web 信息系统开发框架在 J2EE 平台上的设计与实现过程。此通用框架符合 MVC 结构的特点,能改善和优化信息系统开发过程;提高系统性能;使系统具有良好的可扩充性、可重用性和可维护性。

关键词:通用框架 web 开发 j2ee mis

1 引言

随着 Internet 技术的日益成熟,网络已成为新的信息系统平台。由于服从统一的网络协议和标准,统一的浏览器界面,基于 Web 技术的信息系统应用越来越广泛。目前,Web 应用程序开发一般采用微软的 .NET 架构和 SUN 的 J2EE 架构,这两种架构各有其优缺点。综合考虑系统结构、开发周期、可扩充性、可重用性和可维护性等方面的因素,采用符合 MVC (Model - View - Controller) 的开发模式,用 J2EE 构建通用 Web 信息系统框架是较为合适的选择。

(Object Management Group) 的公用对象请求代理程序结构 CORBA (Common? Object Request Broker Architecture) 和 SUN 的 J2EE。DCOM 的致命缺点是跨平台性能差,从部署到运行只能采用微软的平台;CORBA 的缺点是庞大而复杂,标准的更新相对较慢;而 J2EE 简化和规范了多层分布式系统的开发和部署,得到业界的广泛支持,常常是企业级分布式应用程序的首选平台。J2EE 提供一个多层结构的分布式应用程序模型,该模型拥有方便存取数据库的 JDBC 技术,基于 XML 的数据交换技术,组件重用能力,统一的安全模式和灵活的事务控制机制,还提供对 EJB, Servlets 和 JSP 的全面支持^[2]。

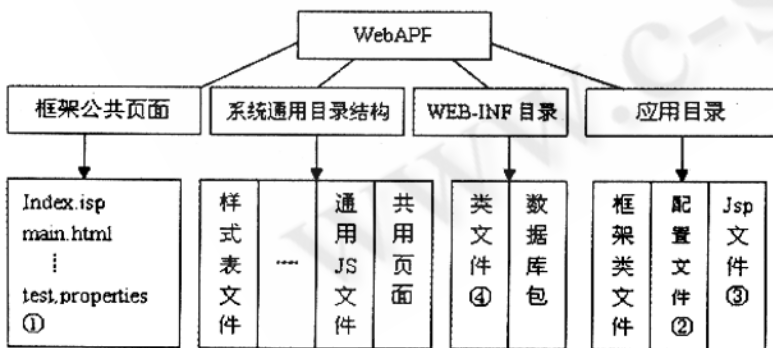


图 1 通用 Web 信息系统框架

2 J2EE 的技术优势

分布式对象技术有 3 种工业标准^[1]: MS 的分布式组件对象模型 DCOM (Componet Object Model), OMG

3 通用 Web 信息系统框架结构模型

框架可使编程人员在一个通用功能已经实现的基础上开始具体的系统开发,它提供了可重用的抽象算法及高层设计,并能将大系统分解成更小的构件,而且能描述构件间的内部接口,这些标准接口在已有的构件基础上通过组装建立各种各样的系统成为可能。

通用 Web 信息系统框架^[4](图 1)是一个符合 MVC 设计模式的通用系统开发模型,通过此模型,将大大简化系统开发复杂程度,使程序层次结构清晰,调试简单。

图 1 几点说明:

(1) 框架公共页面: 包含与框架有关的所有静态页面。

(2) 系统通用目录结构: Images 目录存放框架相关图片, Js 目录放置常用的 Web 控件, Css 目录存放样式表文件。

(3) WEB-INF 目录: 包含所有的 Java 类文件, 针对不同数据库的包文件。

通用 Web 信息系统框架数据处理逻辑如图 2 所示。

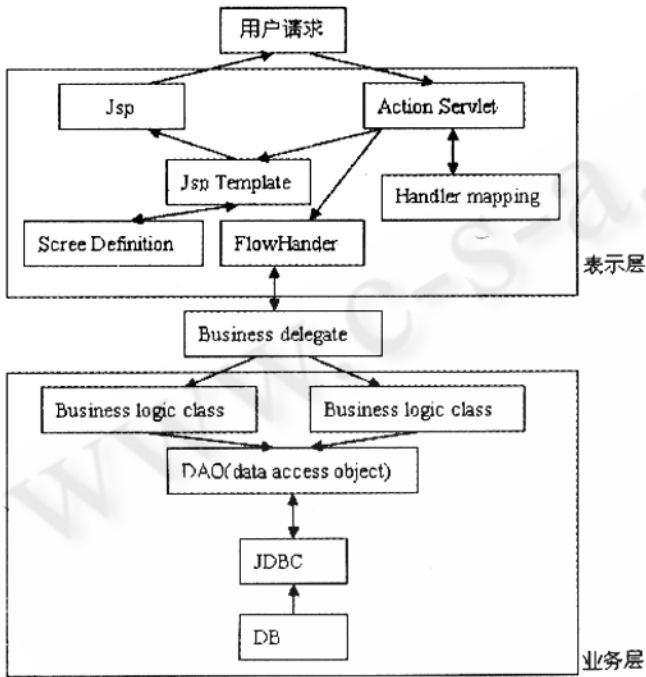


图 2 通用 Web 信息系统框架数据处理逻辑图

图 2 几点说明:

(1) JSP 是应用的 View, 处理表示逻辑。采用 JSP 模板技术, 动态组合最终显示页面。

(2) Action Servlet 是应用的 Controller, 是所有请求的入口和分发中心, 它根据 HTTP 请求的 URL 决定将要进行的操用, 并从 Handler mapping 中找到相应的 handler 处理类, handler 负责处理 HTTP 请求, 收集参数, 调用相应的 Businesslogic 类, 将处理结果放入 HttpRequest/Http Session 对象中, 同时格式化处理信息。通过解释 Screedefinition.xml 文件决定下一页 URL。

(3) Business Delegate Classes 提供应用业务的接口, 对 Web 层隐藏业务逻辑的实现, 以便将来与 AE (Application Envelope) 集成, 不同层之间交换的数据

保存在 model 类中。

(4) Business logic classes 封装应用的业务逻辑, DAO (Data Access Object) 则完成相关数据库操作, DAO 层使用 modellist 组件。

本通用 Web 信息系统框架的特点是: 只要修改图 1 中①②处就可以配置任何 Web 应用程序; 修改图 1 中③④处就可以对系统进行功能扩充; 可以方便地进行快速开发。类文件的开发是任何 Web 应用开发的重点, 我们以所处理的业务不同开发不同的功能模块, 并组织成一个包。为避免出现命名冲突, 任何功能模块都采用: Action + Dao + Model 的结构。

4 通用 Web 信息系统框架的实现

4.1 实现模型

4.1.1 Action、Dao、Model 的协作工作

我们以数据库查询为例来说明 Action、Dao、Model 的协作工作

(1) Action 实现数据库记录的增加, 删除, 查询, 更新四种操作。关键代码如下:

```

if ( request.getParameter ( Screen.JSP_FORM_ACTION ). equals ( Screen.JSP_ACTION_VALUE_QUERY ) )
    { model.setSTUDENTID ( getInPutValue ( "STUDENTID" ) );
      try
        { ml = db.InitializeQuery ( model, 0, Screen.SES_PerPage );
          request.getSession ( ). setAttribute ( WebKeys.SES_WillInputModel_KEY, model );
          request.setAttribute ( WebKeys.SES_Query_STUDENT_WILL, ml );
          this.forwardAction ( );
          return; }
        catch ( DAOException e ) { throw new HTMLActionException ( error ); }

```

关键代码说明: 通过获取页面动作来选择相应的 Action, 同时利用会话 Session 与 Model 完成页面间数据的传递。

(2) Dao 主要完成 SQL 语句的拼凑工作, 关键代码如下:

```

private String QuerySql ( WillInputModel model )
{ sql.append ( "select * from " ). append ( Data-

```

```

baseNames. STUDENT_WILL). append(" where " );
    str = model. getSTUDENTID( );
    if ( str != null && ! str. equals(" "))
        { sql. append(" STUDENTID = "). append
(model. getSTUDENTID( )). append(" and " ); }
    if ( sql. toString( ). substring( sql. length( ) -
7). equals(" where " ))
        { //如果没有给定条件,则查询全部记录
        sql. delete( sql. length( ) - 7, sql. length
( ) ); //去掉 sql 语句后的 where
        }
    else
        { sql. delete( sql. length( ) - 5, sql. length
( ) ); //有指定条件时,去掉 sql 语句后的 and }
        return sql. toString( );
    }
}

```

(3) Model 定义一个数据模型,针对每一个数据表或视图均建立一个 Model。关键代码如下:

```

public WillInputModel( ) { }
//定义字段变量
private String WILL_CODE;
public void setWILL_CODE( String WILL_CODE ) {
this. WILL_CODE = WILL_CODE; }
public String getWILL_CODE( ) { return this. WILL_
CODE; }

```

采用 Action + Dao + Model 这种结构可以使得整个开发和调试过程变得简单。

4.1.2 数据库设计

为实现此框架,我们设计了如下四个表。其中 CUS_SYSUSER、CUS_ROLE 和 CUS_ROLE_PURVIEW 分别为用户表、角色表和角色权限表,通过这三个表实现了系统的授权控制。本框架采用用户角色组来实现分级授权,不同的用户权限分配不同的使用菜单,同时上级用户可以向下级用户授权。CUS_ACTIONS 是系统功能表,在开发信息系统时进行相应的更改。

4.1.3 数据库访问

我们在 Proxool 连接池^[5]组件的基础上撰写了一个类来封装一些基本的方法。这些方法是在我们使用 JPool 时用过的方法,可以从结果集中取记录而不必重新执行 executeCount() 来取得。

Jpool 类关键代码实现的了以下方法^[6]:

表 CUS_SYSUSER

字段	数据类型
USER_ID	varchar(14)
ROLE_ID	varchar(14)
USER_NAME	varchar(25)
PASSWORD	varchar(14)
USER_DESC	varchar(50)
USER_STATUS	varchar(14)
CREATOR - ROLE	varchar(14)

表 CUS_ACTIONS

字段	数据类型
FUNCTIONID	varchar(14)
FUNC_PATH_ID	varchar(50)
FUNCTIONDESC	varchar(100)
FUNCTIONSCCLASS	varchar(500)
URL	varchar(300)
URL_DESC	varchar(100)

表 CSU_ROLE_PURVIEW

字段	数据类型
PURVIEWID	varchar(14)
STATUS	varchar(14)
FUNCTIONID	varchar(14)
ROLE_ID	varchar(14)
FLAG	varchar(14)

表 CUS_ROLE

字段	数据类型
ROLE_ID	varchar(14)
ROLE_NAME	varchar(50)
ROLE_DESC	varchar(100)
CREATE_ROLE	varchar(14)

```

(1) public static int executeCount( String source,
String sql) throws SQLException //执行 SQL,得到结果集中记录条数

```

```

(2) public static void executeTransaction( String

```

source, String sql) throws SQLException//执行事务^[6]

(3) public static void rollback (UserTransaction userTransaction)// 回滚事务^[6]

(4) public static ResultSet getResultSet (String source, String sql) throws SQLException//得到一个结果集对象

(5) public static void returnResultSet (ResultSet resultSet) throws SQLException//释放结果集对象

(6) public static Connection getConnection (String source, int connectionId) throws IllegalArgumentException, SQLException//取得一个数据库连接对象

(7) public static void close (Connection connection) 关闭 connection 对象

4.2 实现实例

4.2.1 系统配置文件 test. properties

应用配置文件是通用 Web 信息系统的一个重要文件,通过此文件来映射 Action 与页面文件的动作。以下是此框架配置文件的部分内容:

SCREEN_NoAction = /pub/NoAction. jsp#用户请求的 Action 不存在时返回页面

SCREEN_NoPurview = /pub/NoPurview. jsp#NO 权限返回页面

SCREEN_SessionInvalid = /pub/SessoInInvalid. jsp# session 失效时返回界面

channel_logon = components. admin. logon. actions. LogOnAction#系统登录处理

//以下是系统用户模块的配置文件的部分内容

SystemUser_add = components. admin. systemuser. actions. SystemUserAddAction#系统用户添加

SystemUser_quer = components. admin. systemuser. actions. SystemUserQueryAction#系统用户查询

SystemUser_modify = components. admin. systemuser. actions. SystemUserModifyAction#系统用户修改

SystemUser_detail = components. admin. systemuser. actions. SystemUserQueryAction#系统用户详细信息查看

4.2.2 以用户管理为例说明此框架的工用流程

用户登录页面 index. jsp 有如下关键代码:

```
<form action = "/channel_logon. do" method = "post" name = "logonForm" >
```

```
<input type = hidden name = " action" value = "
```

query" >

此段程序将被 Web. xml 文件解释成查询 test. properties 属性文件,执行相应类文件。

web. xml 是系统启动时自动加载的文件,通过配置 web. xml 文件可以设置系统初始化参数,自动加载程序等。在使用数据库操作池时,还要设置数据库名,数据库驱动程序,连接数据库用户,密码等作为系统的全局初始化参数。除此之外,还要指定系统启动时自动加载的初始化程序。

任何 Web 应用程序的开发都可以按配置 web. xml, proxool. xml, test. properties, 再实现相应 Model, Dao, Action, 最后开发相应 JSP 页面的步骤来进行。

5 结束语

本文详细论述了基于 J2EE 平台下通用 Web 信息系统框架的结构及实现过程。本框架符合 MVC 结构的特点,采用了 Action + Model + Dao 的设计模式,结合 JSP、Servlet、会话 Bean 及 JDBC 技术,在 J2EE 平台上实现了一个结构清晰,易扩充,高可靠性,跨平台,安全的通用 Web 信息系统的应用开发框架。利用此框架开发 Web 应用程序,只需通过修改小量的配置文件即可新建一个 Web 应用。可以极大提高开发效率,使系统具有良好的可维护性,可扩充性和重用性。

参考文献

- 1 赵东、周明天,分布对象评述[J],计算机工程与应用,2000,12:7-10.
- 2 [美]Martin Bond, Dan Haywood, Debbie Law 等,图书馆目录[M],北京:人民邮电出版社,2002. 12 134-140.
- 3 黄承慧,J2EE 平台的通用资源管理池的分析和实现[J],计算机工程与设计,2005,26(1):237-240.
- 4 樊会峰、李剑等,EWAF:一种 WEB 应用程序框架[J],计算机工程与设计,2005,26(3):562-564.
- 5 郭银章、徐玉斌、曾建潮,基于 CORBA/WEB 技术的企业级信息系统集成[J],计算机工程与设计,2004,25(3):350-389.
- 6 徐学东、孙延明等,JDBC 事务处理在 WEB 信息系统中的应用[J],计算机工程与应用,2005,01:171-173.