

编者按: 现代农业信息技术是利用多媒体和计算机技术, 汇集各类农业专家的知识经验和各种信息, 具体指导良种选育、合理施肥、病虫害综合防治和综合栽培控制等。因此, 计算机软件系统的开发利用, 直接关系到农业信息技术的发展。选登此类论文, 供农业科技工作者参考。

## 基于JSF 技术的 WEB 系统开发与研究

冉崇善, 赵少龙, 郑东 ( 陕西科技大学电气与信息工程学院, 陕西西安 710021)

摘要 JSF 是真正的 MVC 模式, 它的侧重点是 Web 表现层, 提供了丰富的页面开发控件, 而且支持控件的开发方便、灵活, 弥补了 Java 在 Web 开发方面的不足, 非常适用于中小型 Web 系统的快速应用开发。

关键词 JSF; Web; Java; MVC; 生命周期

中图分类号 TP31 文献标识码 A 文章编号 0517- 6611(2007)07 - 02184 - 02

### Development and Research of Web System Based on JSF Technology

RAN Chong shan et al ( Shaanxi University of Science & Technology, Xian, Shaanxi 710021)

Abstract JSF is the real MVC mode, it focuses on the express layer of web. It provides abundant controls to develop the page. At the same time, it supports the development of the controls. It is convenient and vivid, and can make up the shortage of java in the web development. It is suit to the fast development of the middle and small-sized web systems.

Key words JSF; Web; Java; MVC; Life cycle

随着社会信息化的深入, Web 应用的需求在不断增加。Java 在开发 Web 应用中有跨平台和强大的可伸缩性的优点, 但是在中小型系统开发中, 按照一般的方法, 在技术和速度上远远不如微软的 NET 平台。同时, Web 应用程序面对网站上不同的使用者同时的存取, 其执行线程安全问题以及数据验证、转换处理等问题, 又是复杂且难以解决的。JSF (JavaServer Faces) 的提出正是为了解决这个问题, 它试图在不同的角度上提供解决方案给网页设计人员、应用程序设计人员、组件开发人员, 让不同技术的人员可以彼此合作又不互相干扰。

### 1 JSF 简介

JSF 是由 Java Community Process (JCP) 团队研拟出来的一套标准, 并在 2004 年 3 月发表了 JavaServer Faces 1.0 实现成果。JSF 技术包括一系列 API 和 2 个自定义标签库, API 用来描述用户界面组件, 管理其状态, 处理事件、数据的转换与验证, 定义页面导航, 支持国际化和可访问性等。自定义标签库用来在 JSP 标签中表示用户组件。JSF 在其参考规范中定义了 HTML 标签库 (简称 h) 和核心库 (Core Library, 简称 f), 应用程序完全由组件组成, 并且可以根据不同的用户作出相应的调整, 如: HTML、XML、XMLForm、WML 等, 使 Web 在最大程度上满足不同用户的需要。JSF 定义了功能强大的组件库, JSF API 还提供了扩展和创建 U 组件库的功能, 提高了组件的完善性和可扩展性。JSF 具有可重用、可扩展性, 并且在开发工具的支持下, 能够以拖放组件的方式编辑 Web 用户界面, 让服务器端来处理用户界面的管理, 弥补了 Java 在中小型 Web 应用开发中难度大、效率低的缺点。

1.1 MVC 模式 JSF 做到了应用程序逻辑和表示的完全分离, 是真正的 MVC 模式, 其架构如下图 1 所示。 Model 层。

是进行业务操作部分, 由它来实现业务逻辑。在 JSF 之中, Model 是一个符合 JavaBean 技术规格的 Java 类别。 View 层。即用户使用界面, 它是与用户交互的部分。JSF 中一般使用展示套件 (Render Kit) 来开发。 Controller 层。用来控制 Web 应用程序的运作流程, 在 JSF 之中, 是通过 java.faces. Webapp.FacesServlet 类配合 facesconfig.xml 控制文件来实现。

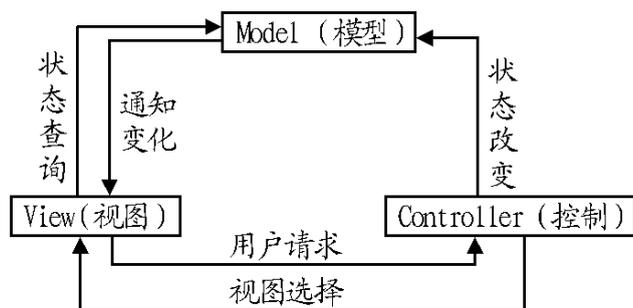


图1 JSF 的 MVC 模式

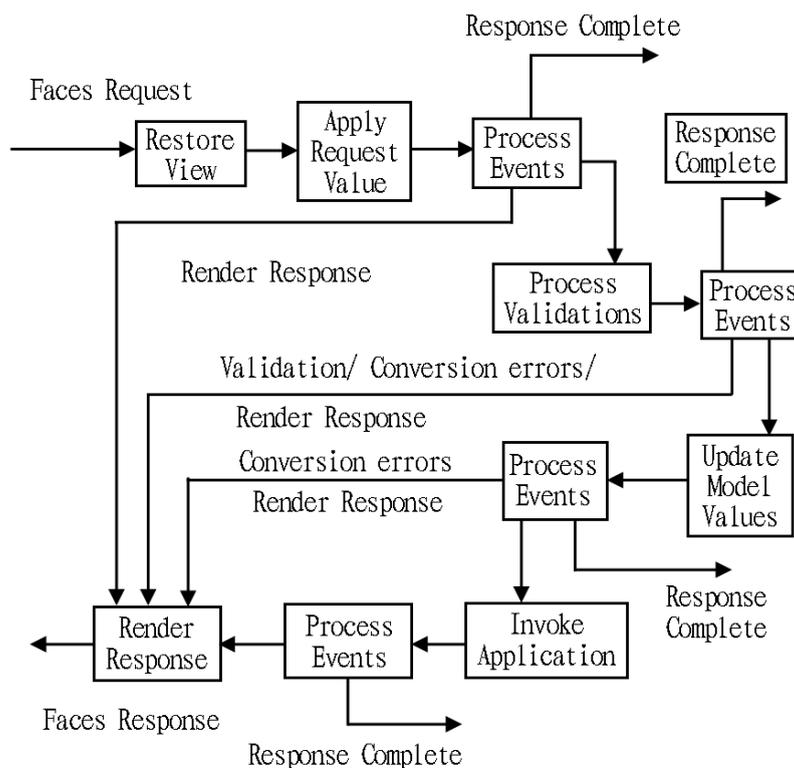


图2 JSF 的生命周期

1.2 JSF 页面的生命周期 由图 2 可知, JSF 的生命周期可以分为几个阶段。重建请求树 (Reconstitute Request Tree) 阶段。当用户单击某个链接或者按钮时 JSF 实现就开始了“重

基金项目 国家自然科学基金资助项目(60672001)

作者简介 冉崇善(1956 - ), 男, 陕西富平人, 副教授, 硕士生导师, 从事网络与数据库研究。

收稿日期 2006-11-17

建请求树”阶段。在此阶段中,JSF 实现将尝试完成以下任务:构建JSF 页面的组件树;为组件连接事件处理程序和验证器;把该树保存在FacesContext 中。应用请求值(ApplyRequest Values)阶段。生命周期中的下一个阶段叫作“应用请求值”阶段,在这个阶段中,每个组件基于嵌入在当前请求中的信息(request/post 参数、头部、cookie,等)更新其当前值。如果提取出来的值无法转换为组件所能接受的类型,就会产生一个错误并在FacesContext 中将该错误排入队列。

处理验证(Process Validation)阶段。在此阶段,JSF 框架处理已注册到树中的组件上的所有验证。验证器将检查每个组件的属性,并根据这些属性执行验证。如果值无效,就会向FacesContext 添加一条错误消息,而生命周期将跳到最后一个阶段,即“呈现响应”阶段。更新模型值(Update Model-Values)阶段。当到达此阶段时,与树中的组件相关联的所有值应该都是有效的。在此阶段中,具有valueRef 表达式的输入组件将被更新。调用应用程序(Invoke Application)阶段。JavaServerFaces 实现在此阶段处理应用程序级事件。在处理事件时,根据导航规则确定接下来要显示的页面,然后框架把控制传递到下一阶段。呈现响应(RenderResponse)阶段。是JSF 生命周期中的最后一个阶段。在此阶段中,该框架将呈现保存在FacesContext 中的组件树,输出结果通常是客户端能够理解的标记语言。

**1.3 JSF 的两种事件与属性** JSF 的目标是利用现有的模型和范例,以便开发人员能在他们的应用程序中快速采用这项技术。JSF 中有两种不同的事件:值更改(value-changed)事件和动作(action)事件。值更改事件将在组件的本地值被更改时发生。动作事件将在按钮或超链接被单击时发生。要设置应用程序对某个事件做出反应,要执行以下两个步骤:

实现一个事件侦听器(动作侦听器或者值更改侦听器)。

在组件上注册事件处理程序。可以使用valuechanged\_listener 和action\_listener 标签在组件上注册事件侦听器。下面的代码是应用程序登录页面(welcome.jsp)的部分源代码,它将一个动作侦听器注册到一个命令按钮上:<h:command\_button id = “denglu” label = “login” ;commandName = “login”action = “success” > ; <f:action\_listener type = “facing.jsf.DengluEventHandler” / > ; </h:command\_button > 。

**2 简单JSF 示例程序** 以登录页面简单介绍JSF 的工作流程,主要步骤及操作代码如下。首先,在SessionBean 中定义UserBean 的变量User、Password 及验证函数Verify。函数Verify 的作用是验证用户输入的用户名和密码是否正确,判断其时要进入下一个欢迎页面还是提示错误信息。主要代码:Public class UserBean() { ...public String getName() { return name ;} public void setName(String newValue) { name = newValue ;}

```
..public String verify() {if (! name.equals(“zsl”) || ! password.equals(“3564411”)) errorMessage = “name or password is error !” ;return “failure” ;else return “success” ;} } 。 登录页面(index.jsp)的主要代码为: ... <f:view > ; <h:form > 。 名称: <h:inputText value = “#{ user.name}” / > </p > 。 密码: <h:inputText value = “#{ user.password}” / > </p > ; ... </h:form > ; </f:view > 。 其中,f 和h 是JSF 特有的标签库,sessionBean 中的变量必须放在#{ } 中才能被关连起来。 欢迎页面(welcome.jsp)的主要代码为: <f:view > ; ... Welcome to JavaServer Faces 。 名称: <h:outputText value = “#{ user.name}” / > </p > ... </f:view > 。 对managed-bean 进行配置,设置Bean 名称为user。同时设置Bean 的生命周期为session。 <managed-bean > ; <managed-bean-name > user </managed-bean-name > ; <managed-bean-class > ; com.corejsf.UserBean; </managed-bean-class > ; <managed-bean-scope > session </managed-bean-scope > ; </managed-bean > 。 页面导航设置。在JSF 中是根据faces-config.xml 中<navigation-rule > 设定,以决定在符合的条件成立时,该连结至哪一个页面,一个基本的设定: <navigation-rule > ; <from-view-id > / pages/index.jsp </from-view-id > ; <navigation-case > ; <from-outcome > success </from-outcome > ; <to-view-id > / pages/welcome.jsp </to-view-id > ; </navigation-case > ; <navigation-case > ; <from-outcome > failure </from-outcome > ; <to-view-id > / pages/index.jsp </to-view-id > ; </navigation-case > ; </navigation-rule > ...
```

这样,程序从<from-view-id > 后的index.jsp 开始,根据<from-outcome > 后的返回值进行判断,从而进入不同的页面。

### 3 结语

JSF 的出现,很大程度上提高了Java 中小型Web 系统开发的速度。它具有强大的组件体系和事件处理系统,非常适用于Web 页面的开发,同时它也是完全实现了MVC 模式的应用架构,这使得它非常适合开发中小型Web 系统。不过,现在,JSF 在技术上还不是很成熟,它还有很多缺点,例如在数据验证上扩展性不够高、组件ID 命名怪异、不利于JavaScript 访问等。但经过发展,JSF 以后会更成熟,性能更强大,会有更多的中小型Web 系统用它来开发。

### 参考文献

- [1] 杜德尼.Msteing JavaServer Faces 中文版[M].北京:电子工业出版社,2004.
- [2] 郝晓静,熊前兴,吴艳丽.JSF 技术在Web 程序开发中的应用研究[J].计算机与数字工程,2006,34(7):52-57.
- [3] 王森.JavaServer Faces 基本架构与程序设计[M].北京:电子工业出版社,2004.