

Spring 在实现 MVC 构架中的应用

符培炯,杜忠军

(四川大学 计算机学院,四川 成都 610065)

摘 要: Spring 框架是一种在 J2EE 的基础上构建起来的一个轻量级的框架实现。叙述了基于 MVC 模式的 Spring 框架的应用,讲解了在 Spring 框架中各个模块的划分和功能,阐述了各个模块的实现逻辑,特别针对 MVC 模式中的控制器、模型和视图三个方面进行了详细的分析,从而说明了该框架的特点和优势。文中还通过对于该框架中一个典型的页面处理流程进行了分析和解释,阐明了将 Spring 框架用于 Web 应用的实现方式,并展望了 Spring 框架与其他框架集合的运用前景。

关键词: Spring; MVC; Web 框架

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2006)06-0236-03

Application of Spring in Realizing MVC Framework

FU Pei-jiong, DU Zhong-jun

(Institute of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: Spring is one kind of layered Java/J2EE application frameworks. This article mainly describes the application of Spring in realizing MVC framework. It introduces in detail the architecture of whole framework and function of each module, describes implementation logic of each module, especially analyzes three aspects(controller, model and view) of MVC framework, showing the characteristic and advantage of the framework. By analyzing a typical process flow for page, the paper illuminates a mechanism applying Spring to Web, and the prospect to associate Spring with other frameworks.

Key words: Spring; MVC; Web framework

1 Web 应用开发模型与 MVC 架构

随着 J2EE 平台的日趋成熟和广泛应用,基于 Java 的 Web 应用开始进入框架时代。由于为 Web 应用提供了现成的通用的框架,即预备的软件架构和相关的软件包,因此使用 Web 应用框架可以大大地提高开发 Web 应用的速度和效率,规范开发的过程。同时基于 MVC 的 Web 框架成为了其中的主流框架,如 Struts, Webworks, jsf 和文中要讲到的 Spring。

MVC 是 Model - View - Controller 的简称,即模型 - 视图 - 控制器^[1]。它是一种设计模式,强制性把应用程序的输入、处理和输出分开。对于界面设计可变性的需求, MVC 把交互系统的组成分解成模型、视图、控制三种部件。它们分别担负不同的任务。

1.1 视图

提供应用的表示层,由 JSP 页面产生,在 JSP 页内没有处理逻辑,它仅负责检索原来先由 Servlet 创建的 JavaBean 对象。

1.2 模型

表示企业数据和业务规则。MVC 系统中的 Model 部分从概念上可以分为两类——系统的内部状态和改变系统状态的动作,通过 JavaBean 和 EJB 组件实现,同一个模型可以被多个视图重用。

1.3 控制器

控制器的作用是从客户端接收请求,并且选择执行相应的业务逻辑,然后把响应结果送回到客户端。

MVC 模式把应用逻辑,处理过程和显示逻辑分成不同的组件来实现,即将内容的显示和产生分离,使三个模块相互独立,提高了应用程序的灵活性和可配置性,也使软件在健壮性、代码重用和结构方面上了一个新的台阶。

2 Spring

2.1 Spring 的总体框架

Spring 框架是一种在 J2EE 的基础上构建起来的一个轻量级的框架实现^[2],相对于 J2EE,具有维护容易、分层清楚、速度快、代码少、支持最新的 ORM 和 AOP 概念等优点,同时在设计上非常强调简单性和一致性,简约的配置文件格式,各个组件的实现一致性,小到包名、类名等命名都体现一种统一的风格。

一方面, Spring 可以成为一种一站式解决方案(即在一个框架完成所有的服务),但同时, Spring 也是组件化

收稿日期:2005-09-07

作者简介:符培炯(1978-),女,河南洛阳人,硕士研究生,研究方向为数据库与管理信息系统;杜忠军,硕士研究生导师,研究方向为数据库与管理信息系统。

的、轻量级的,允许你使用它的部分组件而不需牵涉其他部分^[3]。可以使用 bean 容器,在前台表示层使用 Struts,还可以只使用 Hibernate 集成部分或是 JDBC 抽象层。因此 Spring 是无侵入性的,用户可以根据实际使用的要求和范围,自行选择恰当的框架组合。

图 1 就是 Spring 最主要的 7 个模块,分别是 Spring AOP, Spring ORM, Spring DAO, Spring Web, Spring Context, Spring Web MVC 和 Spring core。而用 Spring 实现 MVC 主要就是基于 Spring Web MVC 模块。

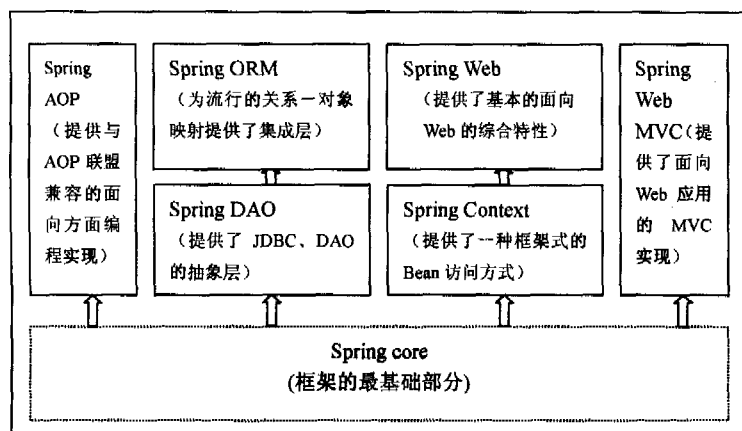


图 1 Spring 总体框架图

2.2 Spring 的 Web MVC 框架介绍

Spring 的 Web 框架是一个请求驱动的 Web 框架,核心是基于分发器 (DispatcherServlet) 设计的^[3]。DispatcherServlet 将从 Web page 收集到的不同请求分发到不同的控制器,控制器 Controller 接收 request 和 response 参数,干脆利落地返回 ModelAndView。而其他的框架控制器一般只返回 View name, Model 还需要通过其他的途径返回^[4]。Spring 还提供了一个非常灵活的 MVC 框架的实现和扩展方式,能够与许多流行的 Web 框架集成,如 Struts, 与其他现有的 Web 框架比较,在依赖注射、AOP 方面也更加出色。

(1) 控制器: Spring 中的控制器概念相当于 Struts 框架中的 action, 实现 MVC 框架中的模型。控制器定义了应用的行为,至少能使用户访问到这些行为。控制器解释用户输入,并将其转换成合理的模型数据,从而可以进一步地由视图展示给用户。

Spring 的缺省处理器是一个简单的控制器 (Controller) 接口,这个接口仅仅定义了 ModelAndView handleRequest(request, response) 方法。你可以自己实现这个接口生成应用的控制器,但使用 Spring 提供的一系列控制器实现会更好一些,比如 AbstractController, AbstractCommandController, 和 SimpleFormController, 应用控制器一般从它们继承。

Spring 还以一种抽象的方式实现了控制器的概念,这样使不同类型的控制器可以被创造。Spring 的控制器包含表单控制器、命令控制器、执行向导逻辑的控制器等。

(2) 视图: Spring 的视图解析极为灵活。Spring 提供

了视图解析器,这使得你在浏览器显示模型数据时不需要指定具体的视图技术,允许使用 Java Server Page, Velocity 模版和 XSLT 视图等多种方式来实现视图。

Spring 处理视图的两个重要的类是 ViewResolver 和 View。View 接口为请求作准备,并将请求传递给某个视图技术。ViewResolver 则提供了一个视图名和实际视图之间的映射。Controller 的实现甚至可以直接将视图写到相应视图名中去,返回 ModelAndView 为 Null。ModelAndView 实例由视图名称和模型 Map 组成,包括 Bean 名或者通过属性文件,也可以通过自己的 ViewResolver 实现抽象的模型。

(3) 分发器 (DispatcherServlet): Spring 中的分发器实现 MVC 模式中的 Controller。

DispatcherServlet 是 Spring MVC 框架的中心,它是一个将请求分发到控制器的 Servlet,并与 Spring 的 ApplicationContext 相集成,实现更多的功能,如绑定主题解析器到请求上,绑定本地信息解析器到请求上,使用在配置在 WebApplicationContext 中的视图解析器,显示视图,利用 WebApplicationContext 处理异常等。

在 Web 应用的 Web.XML 中声明 Servlet, 即 DispatcherServlet 后,必须对 DispatcherServlet 所要处理的请求进行映射,在同一 Web.XML 文件中使用 url-mapping 来配置,定义映射。

3 用 Spring 实现 MVC 流程

Spring 与其他的 Web 框架一样,要解决的问题主要如下:

- (1) 将 Web page 的输入元素封装为一个请求数据对象。
- (2) 根据请求的不同,调度相应的逻辑处理单元,并将请求数据对象作为参数传入到处理器单元。
- (3) 逻辑处理单元完成运算后,返回一个结果数据对象。
- (4) 将结果数据对象中的数据与预先设计的表示层相融合并展现给用户。

图 2 就为 Spring 单独实现 MVC Web 框架的时序图。

对于一个典型的页面处理流程如下:

(1) 收集 Web page 页面的请求传给 DispatcherServlet。

```
<? xml version="1.0" encoding="ISO-8859-1"? >
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
version="2.4">
<servlet>
<servlet-name>Dispatcher</servlet-name>
<servlet-class>
```

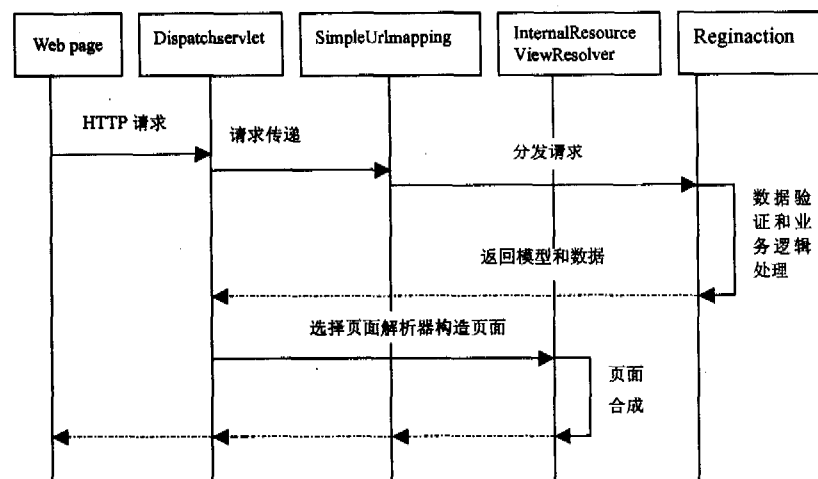


图 2 Spring Web 时序图

```

<!DOCTYPE beans PUBLIC "-//
SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/
spring-beans.dtd">
.....
<!-- Request Mapping -->
<bean id="urlMapping" (5)
class="org.springframework.web.
servlet.handler.SimpleUrlHandlerMapping">
<property name="mappings">
<props>
<prop key="/login.do">LoginAction
</prop>
</props>
</property>
.....
</bean>

```

</bean>

表示 DispatchServlet 将以 .do 结尾的请求分发给 loginAction 控制器来进行业务逻辑处理。

(3) 如果模型数据被返还,就使用配置在 WebApplicationContext 中的视图解析器进行视图解析。

```

.....
<!-- Definition of View Resolver -->
<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalRe-
sourceViewResolver">
<property name="viewClass">
<value>
org.springframework.web.servlet.view.JstlView
</value>
</property>
.....

```

定义视图解析器。

(4) 最后将解析好的视图返回到 Web page,显示给用户。

由于 Spring 框架是一个轻量级,无入侵性的框架实现方式,MVC 实现的可扩展性很强、可以很容易地将你选择的 Web 框架(如 Struts 和 WebWork)和 Spring 结合起来^[5]。只要通过 Spring 的 ContextLoadListener 启动一个 Spring 的根应用上下文,并且通过它的 ServletContext 属性(或者 Spring 的各种帮助方法)在 Struts 或 WebWork 的 Action 中访问。

4 用 Spring 实现 MVC 前景

作为一个新型的 Web 框架, Spring MVC 的优点主要在于明确的角色分工,框架和应用程序类可以作为 JavaBean 进行强大和直接的配置。对于给定的应用,可以使用任何需要的 Controller 子类(普通对象, Command, Form, Wizard, Multi Action, 或定制对象),而非所有事情

(下转第 241 页)

```

org.springframework.web.servlet.DispatcherServlet
</servlet-class>
<init-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/Config.xml</param-value>
>
</init-param>
</servlet>
<servlet-mapping><servlet-name>Dispatcher</servlet-
name>
<url-pattern>*.do</url-pattern>
</servlet-mapping>
</web-app>
.....

```

在 Web.XML 中定义以 .do 结尾的页面由 Dispatcher 来分发。

(2) DispatchServlet 将这些请求通过处理器映射(如 simpleUrlmapping)分发给相应的 Controller,进行数据绑定和验证并进行业务逻辑处理,将结果返回到 DispatchServlet。

```

.....
<!-- Request Mapping -->
<bean id="urlMapping"
class="org.springframework.web.servlet.handler.SimpleUr
lHandlerMapping">
<property name="mappings">
<props>
<prop key="/login.do">LoginAction</prop>
</props>
</property>
</bean>
.....

```

定义处理器映射为 simpleUrlmapping,即通过 simpleUrlmapping, DispatchServlet 来实现“请求/处理单元”关系映射。

```
<? xml version="1.0" encoding="UTF-8"? >
```

架。

对于功能框架而言,其复杂性及可复用度都有相应的变化。一般来说,功能框架的复杂性高而可复用度降低,当然这不是反比的关系。当功能框架在实现某一功能时能够涵盖所有可能出现的问题时,其复杂性更高,而可复用度也会提高。

因此,结构框架适合于通用领域,而功能框架针对特定领域,通用性较低。

3 实例应用

在教学评估系统的开发过程中,从评估领域的分析入手定义结构框架和功能框架,建立了框架模型(如图3所示),并设计了框架工具对框架进行定制以适应不同的应用系统。

这种机制对于系统的构建奠定了良好的基础,使其更

有利于采用积木搭建方法生成应用系统。例如:评估问卷管理框架,它是一个结构框架,其中有4个功能框架:New Questionnaire, Modify Questionnaire, Change State 和 Delete Questionnaire。功能框架又调用相应构件,构成了框架和构件的有机结合。

4 结论

框架的大粒度的重用^[5]使得系统平均开发和维护费用降低,开发速度加快,开发人员减少;而框架分层不仅提高了框架的复用程度,而且其适应性、灵活性和可扩展性也增强了。应用框架的设计本身是一个循环迭代的过程,它在不断应用于实践的过程中,得到更新和提高。下一步工作要在框架的层次划分上不断进行改进,对于功能框架的动态扩展方面还需要进一步探讨。

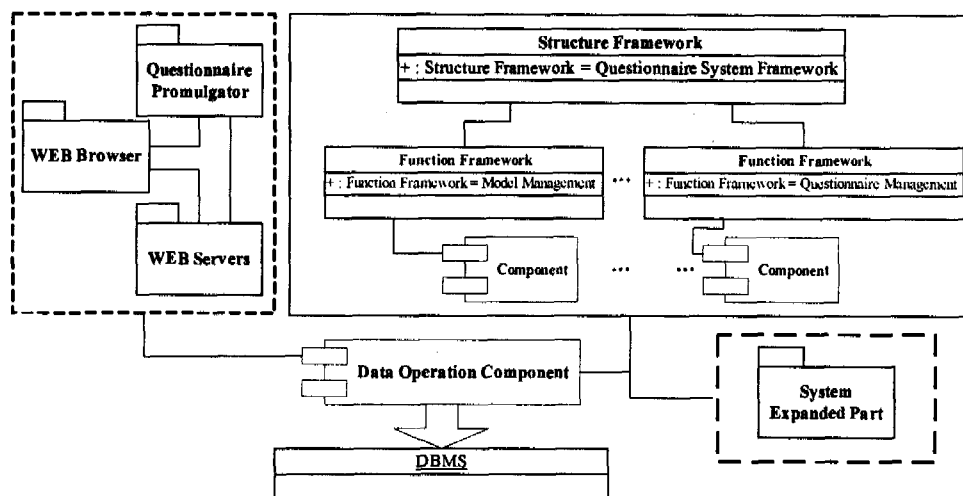


图3 教学评估系统框架结构

参考文献:

- [1] Johnson R E. Frameworks = (Components + Patterns)[J]. Communications of the ACM, 1997, 40(10): 39-42.
- [2] Rumbaugh J, Jacobson I, Booch G. The UML Reference Manual[M]. New York: Addison-Wesley, 1999. 27-29.
- [3] Yin Sheng Bin, Bian Xiao Fan. Construct the Component of the Immature Realm Using Data Mining[A]. Proceedings of 2004 International Conference on Machine Learning and Cybernetics[C]. Shanghai: [s. n.], 2004. 1509-1514.
- [4] Cohen S, Northrop L M. Object-Oriented Technology and Domain Analysis[A]. Proceedings of the Fifth International Conference On Software Reuse[C]. [s. l.]: IEEE Computer Society Press, 1998. 86-93.
- [5] 张世琨, 张文娟, 常欣, 等. 基于软件体系结构的可复用构件制作和组装[J]. 软件学报, 2001(9): 1351-1359.

(上接第238页)

都要从 Action/ActionForm 继承。另外可定制的绑定和验证,可定制的处理映射和视图解析,可定制的本地化和主题解析和支持 AOP 和 IOC 都使 Spring MVC 框架越来越被认可和采用。

参考文献:

- [1] 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.
- [2] 夏昕. Spring 开发指南 8.0[EB/OL]. http://www.xiaxin.net/Spring-Dev-Guide.rar. 2004-09.
- [3] 黄华. 框架技术在 Web 系统开发中的应用[J]. 微机发展, 2005, 15(5): 77-79.
- [4] 张娅, 杜友福, 张南平. 用 Struts 建立 MVC 模式的 Web 应用[J]. 微机发展, 2003, 13(12): 105-107.
- [5] Totodo. 使用 struts + spring + hibernate 组装 web 应用[EB/OL]. http://www.kissjava.com/doc/j2ee/webservice/2005-06-07/19451118128953.html. 2005.