

文章编号: 1005-8451 (2006) 11-0008-03

基于MVC的企业级应用开发

张翼, 揭金良

(成都理工大学 信息工程学院, 成都 610059)

摘要: 阐述MVC模型的基本理论以及各层间的关系, 详述MVC模式及MVC的一种具体实现框架, 即 Struts 的组成及工作原理, 并基于该设计模式, 实现一个网络医疗信息管理系统, 证明MVC在企业级应用系统中的应用及优越性。

关键词: Struts; 模型—视图—控制器; 数据传输对象; 开发

中图分类号: TP39

文献标识码: A

Enterprise application based in MVC

ZHANG Yi, JIE Jin-liang

(College of Information Engineering, Chengdu University of Technology, Chengdu 610059, China)

Abstract: The basic theories of MVC model and the relationships of each tier were presented. MVC model as well as the architecture and working principle of struts which was one specific implemented framework of MVC were described in details. Furthermore, the Intranet Health Information System based on this design pattern was implemented successfully, meanwhile, the application and the advantages of MVC in Enterprise Application System were also analyzed.

Key words: Struts; model-view-controller; data transfer object; development

MVC 是模型—视图—控制器 (Model-View-Controller) 的简称, 它是一种设计模式, 它强制性地 把应用程序的输入、处理和输出分开^[1]。图1显示了这几个模块的功能及相互关系。

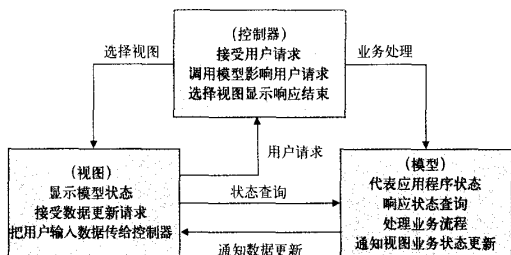


图1 MVC模型结构

1 Struts实现MVC的机制

MVC 只是一种理论, 现在已有一些关于这种理论的具体框架, Struts 就是一种不错的选择。在 Struts 框架中, 模型由实现业务逻辑的 JavaBean 或 EJB 组件构成, 控制器由 ActionServlet 和 Action 来实现, 视图由一组 JSP 文件构成。基于 Struts 实现

的 MVC 框架如图2所示。

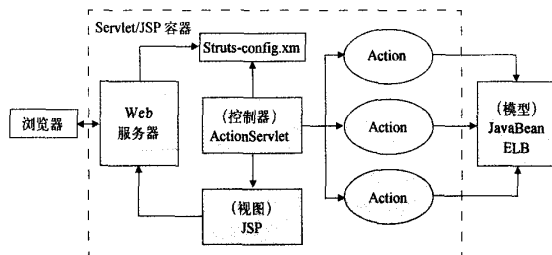


图2 Struts实现的MVC框架

1.1 视图

在 Struts 中, 视图就是一组 JSP 文件以及 ActionForm Bean。在 JSP 文件中没有业务逻辑, 也没有模型信息, 只有标签, 这些标签可以是标准的 JSP 标签或客户化标签。ActionForm Bean 也是一种 JavaBean, 除了具有一些 JavaBean 的常规方法外, 还包含一些特殊的方法, 用于验证 HTML 表单数据以及将其属性重新设置为默认值。Struts 利用 ActionForm Bean 来进行视图和控制器之间数据的传递。Struts 把用户输入的表单数据保存在 ActionForm Bean 中, 把它传递给控制器, 控制器可以对 ActionForm Bean 中的数据进行修改, JSP 文件使用 Struts 标签读取修改后的 ActionForm Bean 的信息, 重新设

收稿日期: 2006-08-13

作者简介: 张翼, 在读硕士研究生, 揭金良, 教授。

置 HTML 表单。

1.2 控制器

控制器由 `ActionServlet` 类和 `Action` 类来实现。`ActionServlet` 类是 Struts 中的核心组件，它在 MVC 模型中扮演中央控制器的角色。`ActionServlet` 主要负责接收 Http 请求信息，根据配置文件 `Struts-config.xml` 的配置信息，把请求转发给适当的 `Action` 对象，如果该 `Action` 对象不存在，`ActionServlet` 会先创建这个 `Action` 对象。

`Action` 类负责调用模型的方法，更新模型的状态，并帮助控制应用程序的流程。对于小型简单的应用，`Action` 类本身也可以完成一些实际的业务逻辑。

对于大型应用，`Action` 充当用户请求和业务逻辑处理之间的适配器 (Adaptor)，其功能就是将请求与业务逻辑分开，`Action` 根据用户请求调用相关的业务逻辑组件。业务逻辑由 `JavaBean` 或 `EJB` 来完成，`Action` 则侧重于控制应用程序的流程，而不是实现应用程序的逻辑。通过将业务逻辑放在单独的 `Java` 包或 `EJB` 中，可以提高应用程序的灵活性和可重用性。

当 `ActionServlet` 控制器收到用户请求后，把请求转发到一个 `Action` 实例。如果这个实例不存在，控制器会首先创建它，然后调用这个 `Action` 实例的 `execute()` 方法。`Action` 的 `execute()` 方法返回 `ActionForward` 对象，它封装了把用户请求再转发给其他 Web 组件的信息。用户定义自己的 `Action` 类，即 `Action` 基类的子类时，必须覆盖 `execute()` 方法。在 `Action` 基类中该方法返回 `null`。

1.3 模型

在 Struts 中，模型表示应用程序的状态和业务逻辑。对于一般的中小型应用，一般可以用 `JavaBean` 实现业务逻辑。对于大型的分布式应用系统，一般采用 `EJB` 封装业务逻辑。

1.4 Struts 工作流程

Struts 的工作流程如图 3 所示。

(1) 检索与用户请求匹配的 `ActionMapping` 实例，如果不存在，就返回用户请求路径无效的信息；(2) 如果 `ActionForm` 实例不存在，就创建一个 `ActionForm` 对象，把客户提交的表单数据保存到 `ActionForm` 对象中；(3) 根据配置信息决定是否需要进行表单验证，如果需要验证，就调用 `ActionForm` 的 `Validate()` 方法；(4) 如果 `ActionForm` 的 `validate()` 方法返回 `null` 或返回一个不包含 `ActionMessage` 的 `ActionErrors` 对象，就表示表单验证成功；(5)

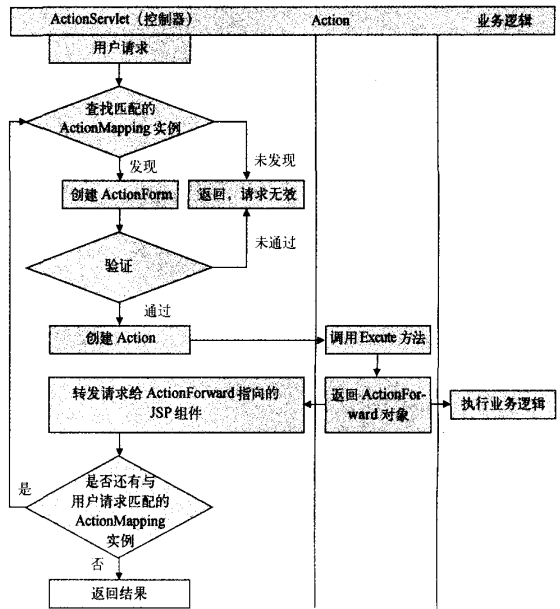


图 3 Struts 工作流程图

`ActionServlet` 根据 `ActionMapping` 实例包含的映射信息决定将请求转发给哪个 `Action`。如果相应的 `Action` 实例不存在，就先创建这个实例，然后调用 `Action` 的 `execute()` 方法；(6) `Action` 的 `execute()` 方法返回一个 `ActionForward` 对象，`ActionServlet` 再把客户请求转发给 `Action-Forward` 对象指向的 JSP 组件；(7) `ActionForward` 对象指向的 JSP 组件生成动态网页，返回给客户。对于以上的流程 (4)，如果 `ActionForm` 的 `validate()` 方法返回一个包含一个或多个 `ActionMessage` 的 `ActionErrors` 对象，就表示表单验证失败，此时 `ActionServlet` 将直接把请求转发给包含用户提交表单的 JSP 组件。在这种情况下，不会再创建 `Action` 对象。

1.5 Struts 的配置文件 struts-config.xml

一个用户请求是通过 `ActionServlet` 来处理和转发的。那么，`ActionServlet` 如何决定把用户请求转发给哪个 `Action` 对象呢？这就需要一些描述用户请求路径和 `Action` 映射关系的配置信息了。

在 Struts 中，配置映射信息都存储在特定的 XML 文件 `Struts-config.xml` 中。在该配置文件中，每一个 `Action` 的映射信息都通过一个 `<action>` 元素来配置。这些配置信息在系统启动的时候被读进内存，供 Struts 在运行期间使用。在内存中，每一个 `<action>` 元素都对应一个 `ActionMapping` 类的实例。

2 开发实例

2.1 网络医疗信息管理系统功能

我们开发的网络医疗信息管理系统就是基于 Struts 的应用系统。系统功能模块如图 4 所示。

| | | | |
|--------|--------|------|------|
| 住院病人管理 | 出院病人管理 | 药房管理 | 住院管理 |
| 医嘱管理 | | | |
| 物流管理 | 病人帐务管理 | | 电子病历 |

图4 网络医疗系统功能图

2.2 网络医疗信息系统软件架构

系统由 Presentation Tier (呈现层)、Bussniss Tier (业务逻辑层) 和 DataTier (数据层) 组成。

2.2.1 呈现层

呈现层包括浏览器、前台的 JSP 以及 Action 组件 (ActionForm 和 Action)。用户通过浏览器把请求传送给控制器, 控制器调用相应的 Action, Action 再把客户请求转发给 Jsp 组件, 或是通过 Action 转向代理, 向 SessionBean 传输数据。

2.2.2 业务逻辑层

业务逻辑层包括 SessionBean 和 EntityBean (BMP)。Action 与 SessionBean 之间不直接通信, 而是通过代理连接。通常, 所有的界面上的逻辑在 Action 中, 而 Session 提供业务逻辑。业务代理为 Action 与 SessionBean 中业务逻辑应用的连接提供一种透明的机制。此外, 业务代理在需要时还提供缓冲, 标记以及处理异常的逻辑。数据传输对象被用于在 Web 与 EJB 之间传送数据, 在此项目中应用了 SearchModel 与 InfoModel 做为 DTO。SearchModel 用于从页面传输指定的数据到 SessionBean, InfoModel 用于返回数据到 Web。

2.2.3 数据层

数据层 SessionBean 通过数据库管理器操纵数据库连接池。此外本系统还特别采用了一种 Merlin 组件, 包括 Merlin Libraries、Merlin EJB Module 和 MerlinWeb, 这是专为为本系统设计的一种封装的组件, 这里就不作介绍了。

2.3 MVC 在网络医疗信息系统中的体现

总结以上分析, 整个系统构架是典型的 MVC 模式。

2.3.1 视图部分

视图包括 JSP 以及 ActionForm。JSP 是与用户交

互的组件, 用户通过 JSP 页面把信息输入系统, 同时 JSP 还可以进行一些页面上的逻辑验证, 例如用户名、密码和类型等的验证。ActionForm 实质上是一种 JavaBean, 它接受从 JSP 来的用户信息。通过一系列的 get/set 方法获得从 JSP 来的用户信息。表单验证也可以放到 ActionForm 中。

2.3.2 控制器部分

控制器组件由 ActionServlet、Action 以及 Struts-config.xml 文件构成。Action 的主要功能是调用相应的模型组件来完成业务逻辑, 并决定把客户请求转发给哪个视图组件。用户每一个具体操作都对应一个 Action。如按一个回车, 点一个查询按钮或者添加、确认按钮等等。Action 的调用都写在 Struts-config.xml 中, 每一个 Action 的映射信息都通过一个 <action> 元素来配置。

在此项目中, 把视图部分与控制器部分都划分到了呈现层。

2.3.3 模型部分

模型部分由 EntityBean (BMP) 和 SessionBean 构成, 用以处理具体的业务逻辑。最后通过 SessionBean 与数据库连接。具体实现由 DBManager (Merlin 中的一个组件) 执行数据库连接池并操纵数据。

3 结束语

网络医疗信息管理系统采用 MVC 软件设计模式, 充分利用了 MVC 的现有框架 (Struts), 结合 J2EE 技术, 开发出了一个功能全面的医院信息系统。该系统已投入运行, 并获得客户好评。但同时也显露出了一些问题, 较为突出的是速度问题, 目前正在解决之中。方案是采用负载均衡技术或移植到 Unix 系统上。

参考文献:

- [1] James Holmes. Struts 程序员查询词典[M]. 北京: 中国铁道出版社, 2005.
- [2] 丰伟. Struts 教程[EB/OL]. <http://www.xpbook.com/soft/2779.html>, 2004: 12—16.
- [3] 孙卫琴. 精通 Struts [M]. 北京: 电子工业出版社, 2004: 10—25.
- [4] 孙卫琴. 创建 Web 应用和 Struts 框架的配置文件[EB/OL]. <http://linux.cddnet.com/pub/article/c322-a181127.p1.html>, 2004: 11—24.