**Struts——MVC 的一种开放源码实现**

本文介绍Struts，它是使用 servlet 和 JavaServer Pages 技术的一种 Model-View-Controller 实现。Struts 可帮助您控制 Web 项目中的变化并提高专业化水平。尽管您可能永远不会用 Struts 实现一个系统，但您可以将其中的一些思想用于您以后的 servlet 和 JSP 网页的实现中。

**简介**

小学生也可以在因特网上发布 HTML 网页。但是，小学生的网页和专业开发的网站有质的区别。网页设计人员（或者 HTML 开发人员）必须理解颜色、用户、生产流程、网页布局、浏览器兼容性、图像创建和 JavaScript 等等。设计漂亮的网站需要做大量的工作，大多数 Java 开发人员更注重创建优美的对象接口，而不是用户界面。JavaServer Pages (JSP) 技术为网页设计人员和 Java 开发人员提供了一种联系钮带。

如果您开发过大型 Web 应用程序，您就理解 *变化* 这个词的含义。“模型-视图-控制器”(MVC) 就是用来帮助您控制变化的一种设计模式。MVC 减弱了业务逻辑接口和数据接口之间的耦合。Struts 是一种 MVC 实现，它将 Servlet 2.2 和 JSP 1.1 标记（属于 J2EE 规范）用作实现的一部分。尽管您可能永远不会用 Struts 实现一个系统，但了解一下 Struts 或许使您能将其中的一些思想用于您以后的 Servlet 的 JSP 实现中。

**模型-视图-控制器 (MVC)**

JSP 标记只解决了部分问题。我们还得处理验证、流程控制和更新应用程序的状态等问题。这正是 MVC 发挥作用的地方。MVC 通过将问题分为三个类别来帮助解决单一模块方法所遇到的某些问题：

* **Model（模型）**

模型包含应用程序的核心功能。模型封装了应用程序的状态。有时它包含的唯一功能就是状态。它对视图或控制器一无所知。

* **View（视图）**

视图提供模型的表示。它是应用程序的 *外观*。视图可以访问模型的读方法，但不能访问写方法。此外，它对控制器一无所知。当更改模型时，视图应得到通知。

* **Controller（控制器）**

控制器对用户的输入作出反应。它创建并设置模型。

**MVC Model 2**

Web 向软件开发人员提出了一些特有的挑战，最明显的就是客户机和服务器的无状态连接。这种无状态行为使得模型很难将更改通知视图。在 Web 上，为了发现对应用程序状态的修改，浏览器必须重新查询服务器。

另一个重大变化是实现视图所用的技术与实现模型或控制器的技术不同。当然，我们可以使用 Java（或者 PERL、C/C++ 或别的语言）代码生成 HTML。这种方法有几个缺点：

* Java 程序员应该开发服务，而不是 HTML。
* 更改布局时需要更改代码。
* 服务的用户应该能够创建网页来满足它们的特定需要。
* 网页设计人员不能直接参与网页开发。
* 嵌在代码中的 HTML 很难看。

对于 Web，需要修改标准的 MVC 形式。图1显示了 MVC 的 Web 改写版，通常也称为 MVC Model 2 或 MVC 2。

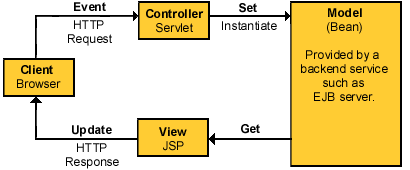


图1. MVC Model 2

**Struts，MVC 2 的一种实现**

Struts 是一组相互协作的类、servlet 和 JSP 标记，它们组成一个可重用的 MVC 2 设计。这个定义表示 Struts 是一个框架，而不是一个库，但 Struts 也包含了丰富的标记库和独立于该框架工作的实用程序类。图2显示了 Struts 的一个概览。

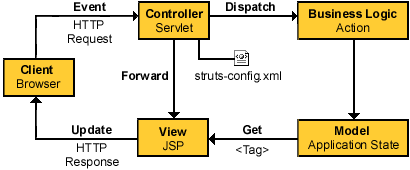


图2. Struts 概览

**Struts 概览**

* Client browser（客户浏览器）

来自客户浏览器的每个 HTTP 请求创建一个事件。Web 容器将用一个 HTTP 响应作出响应。

* Controller（控制器）

控制器接收来自浏览器的请求，并决定将这个请求发往何处。就 Struts 而言，控制器是以 servlet 实现的一个命令设计模式。 struts-config.xml 文件配置控制器。

* 业务逻辑

业务逻辑更新模型的状态，并帮助控制应用程序的流程。就 Struts 而言，这是通过作为实际业务逻辑“瘦”包装的 Action 类完成的。

* Model（模型）的状态

模型表示应用程序的状态。业务对象更新应用程序的状态。ActionForm bean 在会话级或请求级表示模型的状态，而不是在持久级。JSP 文件使用 JSP 标记读取来自 ActionForm bean 的信息。

* View（视图）

视图就是一个 JSP 文件。其中没有流程逻辑，没有业务逻辑，也没有模型信息 -- 只有标记。标记是使 Struts 有别于其他框架（如 Velocity）的因素之一。

**详细分析 Struts**

图3示的是 org.apache.struts.action 包的一个最简 UML 图。图3显示了 ActionServlet (Controller)、 ActionForm (Form State) 和 Action (Model Wrapper) 之间的最简关系。

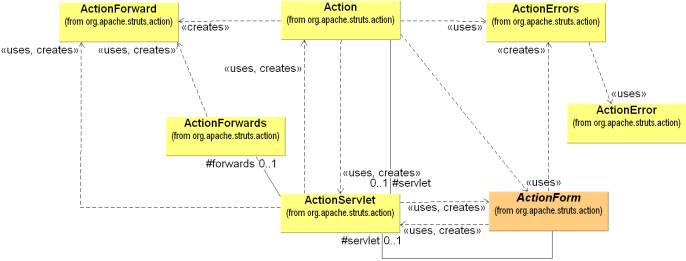


图3. Command (ActionServlet) 与 Model (Action & ActionForm) 之间的关系的 UML 图

**ActionServlet 类**

您还记得函数映射的日子吗？在那时，您会将某些输入事件映射到一个函数指针上。如果您对此比较熟悉，您会将配置信息放入一个文件，并在运行时加载这个文件。函数指针数组曾经是用 C 语言进行结构化编程的很好方法。

现在好多了，我们有了 Java 技术、XML、J2EE，等等。Struts 的控制器是将事件（事件通常是 HTTP post）映射到类的一个 servlet。正如您所料 -- 控制器使用配置文件以使您不必对这些值进行硬编码。时代变了，但方法依旧。

ActionServlet 是该 MVC 实现的 Command 部分，它是这一框架的核心。 ActionServlet (Command) 创建并使用 Action 、 ActionForm 和 ActionForward 。如前所述， struts-config.xml 文件配置该 Command。在创建 Web 项目时，您将扩展 Action 和 ActionForm 来解决特定的问题。文件 struts-config.xml 指示 ActionServlet 如何使用这些扩展的类。这种方法有几个优点：

* 应用程序的整个逻辑流程都存储在一个分层的文本文件中。这使得人们更容易查看和理解它，尤其是对于大型应用程序而言。
* 网页设计人员不必费力地阅读 Java 代码来理解应用程序的流程。
* Java 开发人员也不必在更改流程以后重新编译代码。

可以通过扩展 ActionServlet 来添加 Command 功能。

**ActionForm 类**

ActionForm 维护 Web 应用程序的会话状态。 ActionForm 是一个抽象类，必须为每个输入表单模型创建该类的子类。当我说 *输入表单模型* 时,是指 ActionForm 表示的是由 HTML 表单设置或更新的一般意义上的数据。例如，您可能有一个由 HTML 表单设置的 UserActionForm 。Struts 框架将执行以下操作：

* 检查 UserActionForm 是否存在；如果不存在，它将创建该类的一个实例。
* Struts 将使用 HttpServletRequest 中相应的域设置 UserActionForm 的状态。没有太多讨厌的 request.getParameter() 调用。例如，Struts 框架将从请求流中提取 fname ，并调用 UserActionForm.setFname() 。
* Struts 框架在将 UserActionForm 传递给业务包装 UserAction 之前将更新它的状态。
* 在将它传递给 Action 类之前，Struts 还会对 UserActionForm 调用 validation() 方法进行表单状态验证。 **注：** 这并不总是明智之举。别的网页或业务可能使用 UserActionForm ，在这些地方，验证可能有所不同。在 UserAction 类中进行状态验证可能更好。
* 可在会话级维护 UserActionForm 。

注：

* struts-config.xml 文件控制 HTML 表单请求与 ActionForm 之间的映射关系。
* 可将多个请求映射到 UserActionForm 。
* UserActionForm 可跨多页进行映射，以执行诸如向导之类的操作。

**Action 类**

Action 类是业务逻辑的一个包装。 Action 类的用途是将 HttpServletRequest 转换为业务逻辑。要使用 Action ，请创建它的子类并覆盖 process() 方法。

ActionServlet (Command) 使用 perform() 方法将参数化的类传递给 ActionForm 。仍然没有太多讨厌的 request.getParameter() 调用。当事件进展到这一步时，输入表单数据（或 HTML 表单数据）已被从请求流中提取出来并转移到 ActionForm 类中。

注：扩展 Action 类时请注意简洁。 Action 类应该控制应用程序的流程，而不应该控制应用程序的逻辑。通过将业务逻辑放在单独的包或 EJB 中，我们就可以提供更大的灵活性和可重用性。

考虑 Action 类的另一种方式是 Adapter 设计模式。 Action 的用途是“将类的接口转换为客户机所需的另一个接口。Adapter 使类能够协同工作，如果没有 Adapter，则这些类会因为不兼容的接口而无法协同工作。”（摘自 Gof 所著的 *Design Patterns - Elements of Reusable OO Software* ）。本例中的客户机是 ActionServlet ，它对我们的具体业务类接口一无所知。因此，Struts 提供了它能够理解的一个业务接口，即 Action 。通过扩展 Action ，我们使得我们的业务接口与 Struts 业务接口保持兼容。（一个有趣的发现是， Action 是类而不是接口）。 Action 开始为一个接口，后来却变成了一个类。真是金无足赤。）

**Error 类**

UML 图（图 3）还包括 ActionError 和 ActionErrors 。 ActionError 封装了单个错误消息。 ActionErrors 是 ActionError 类的容器，View 可以使用标记访问这些类。 ActionError 是 Struts 保持错误列表的方式。

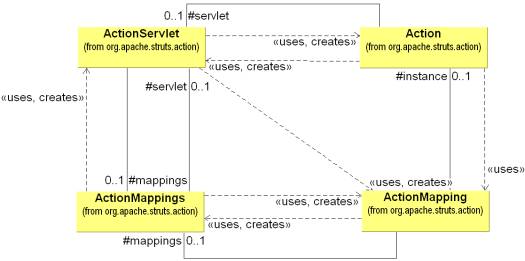


图4. Command (ActionServlet) 与 Model (Action) 之间的关系的 UML 图

**ActionMapping 类**

输入事件通常是在 HTTP 请求表单中发生的，servlet 容器将 HTTP 请求转换为 HttpServletRequest 。控制器查看输入事件并将请求分派给某个 Action 类。 struts-config.xml 确定 Controller 调用哪个 Action 类。 struts-config.xml 配置信息被转换为一组 ActionMapping ，而后者又被放入 ActionMappings 容器中。（您可能尚未注意到这一点，以 *s*结尾的类就是容器）

ActionMapping 包含有关特定事件如何映射到特定 Action 的信息。 ActionServlet (Command) 通过 perform() 方法将 ActionMapping 传递给 Action 类。这样就使 Action 可访问用于控制流程的信息。

**ActionMappings**

ActionMappings 是 ActionMapping 对象的一个集合。

**Struts 的优点**

* JSP 标记机制的使用

标记特性从 JSP 文件获得可重用代码和抽象 Java 代码。这个特性能很好地集成到基于 JSP 的开发工具中，这些工具允许用标记编写代码。

* 标记库

为什么要另发明一种轮子，或标记库呢？如果您在库中找不到您所要的标记，那就自己定义吧。此外，如果您正在学习 JSP 标记技术，则 Struts 为您提供了一个起点。

* 开放源码

您可以获得开放源码的全部优点，比如可以查看代码并让使用库的每个人检查代码。许多人都可以进行很好的代码检查。

* MVC 实现样例

如果您希望创建您自己的 MVC 实现，则 Struts 可增加您的见识。

* 管理问题空间

分治是解决问题并使问题可管理的极好方法。当然，这是一把双刃剑。问题越来越复杂，并且需要越来越多的管理。

**Struts 的缺点**

* 仍处于发展初期

Struts 开发仍处于初级阶段。他们正在向着发行版本 1.0 而努力，但与任何 1.0 版本一样，它不可能尽善尽美。

* 仍在变化中

这个框架仍在快速变化。Struts 1.0 与 Struts 0.5 相比变化极大。为了避免使用不赞成使用的方法，您可能隔一天就需要下载最新的 Struts。在过去的 6 个月中，我目睹 Struts 库从 90K 增大到 270K 以上。由于 Struts 中的变化，我不得不数次修改我的示例，但我不保证我的示例能与您下载的 Struts 协同工作。

* 正确的抽象级别

Struts 是否提供了正确的抽象级别？对于网页设计人员而言，什么是正确的抽象级别呢？这是一个用 $64K 的文字才能解释清楚的问题。在开发网页的过程中，我们是否应该让网页设计人员访问 Java 代码？某些框架（如 Velocity）说不应该，但它提供了另一种 Web 开发语言让我们学习。在 UI 开发中限制访问 Java 有一定的合理性。最重要的是，如果让网页设计人员使用一点 Java，他将使用大量的 Java。在 Microsoft ASP 的开发中，我总是看到这样的情况。在 ASP 开发中，您应该创建 COM 对象，然后编写少量的 ASP 脚本将这些 COM 对象联系起来。但是，ASP 开发人员会疯狂地使用 ASP 脚本。我会听到这样的话，“既然我可以用 VBScript 直接编写 COM 对象，为什么还要等 COM 开发人员来创建它呢？”通过使用标记库，Struts 有助于限制 JSP 文件中所需的 Java 代码的数量。Logic Tag 就是这样的一种库，它对有条件地生成输出进行管理，但这并不能阻止 UI 开发人员对 Java 代码的狂热。无论您决定使用哪种类型的框架，您都应该了解您要在其中部署和维护该框架的环境。当然，这项任务真是说起来容易做起来难。

* 有限的适用范围

Struts 是一种基于 Web 的 MVC 解决方案，所以必须用 HTML、JSP 文件和 servlet 来实现它。

* J2EE 应用程序支持

Struts 需要支持 JSP 1.1 和 Servlet 2.2 规范的 servlet 容器。仅凭这一点远不能解决您的全部安装问题，除非使用 Tomcat 3.2。我用 Netscape iPlanet 6.0 安装这个库时遇到一大堆问题，按理说它是第一种符合 J2EE 的应用程序服务器。我建议您在遇到问题时访问 Struts 用户邮件列表的归档资料。

* 复杂性

在将问题分为几个部分的同时也引入了复杂性。毫无疑问，要理解 Struts 必须接受一定的培训。随着变化的不断加入，这有时会令人很沮丧。欢迎访问本网站。

**Struts 的前景**

在这个软件开发的新时代，一切都变得很快。在不到 5 年的时间内，我们已经目睹了从 cgi/perl 到 ISAPI/NSAPI、再到使用 VB 的 ASP、一直到现在的 Java 和 J2EE 的变迁。Sun 正在尽力将新的变化反映到 JSP/servlet 体系结构中，正如他们对 Java 语言和 API 所作的更改一样。您可以从 Sun 的网站获得新的 JSP 1.2 和 Servlet 2.3 规范的草案。此外，一个标准 JSP 标记库即将出现。