第32卷 第22期 计 算 机 工 程 2006年11月
Vol.32 No.22 Computer Engineering November 2006
·开发研究与设计技术· 文章编号：1000—3428(2006)22—0274—03 文献标识码：A 中图分类号：TP311.11

Web应用分层与开发框架设计研究

李守振[1]，张南平[2]，常国锋[3]
(1. 武汉理工大学计算机科学与技术学院，武汉 430070；2. 武汉菲旺软件技术有限责任公司，武汉 430070；
3. 河南新乡师范高等专科学校计科系，新乡 453000)

摘 要：对Web应用进行了合理的分层，介绍了3种主流开源框架Struts、Spring和Hibernate，应用它们作为Web应用各层的相应实现，进行了有效整合，设计了一个优秀的、完整的开发框架，给出了该框架在实践中的成功应用。
关键词：Web应用；开发框架；Struts；Spring；Hibernate

Research on Layering of Web Application and Design of Development Architecture

LI Shouzhen[1], ZHANG Nanping[2], CHANG Guofeng[3]
(1. Institute of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070; 2. Philwong Software Tech. Co. Ltd., Wuhan 430070; 3. Computer Science Department, Xinxiang Teacher College, Xinxiang 453000)

【Abstract】 This paper rationally separates a Web application into different layers, then introduces three main open source Java Struts, Spring and Hibernate which are applied to corresponding layers of the Web application and are conformed effectively, sequentially. It designs an excellent and complete development architecture which is applied to the practice successfully.
【Key words】 Web applications; Development architecture; Struts; Spring; Hibernate

面向对象的最大优势之一就是软件复用，复用本身被分为代码级的复用和框架的复用。框架不同于设计模式，它是通过一组紧密关联的类彼此配合以完成某种可以重复运用的设计概念，框架规划了应用程序的骨干及整体风貌，使应用程序遵循特定的控制流程和机制来实现特定的需求[1]。

软件系统发展到今天已相当复杂，特别是服务器端软件，开发这类软件所需关注的技术点有许多，但是，某些方面的需求却是共性的，这就为使用框架提供了前提。本文所要讨论的是：如何对Web应用进行合理的分层及每层选用技术框架时的策略，以及如何整合每层的框架以使各应用层保持一致并以一种松散耦合的方式彼此作用而不用管低层的技术细节，进而构建出一个完整的Web开发框架，该框架应能指导代码如何分布，并把开发者从重复的编码中解放出来，以使他们能专注于业务本身的分析与研究。

1 应用程序的层次结构

基于Web的应用通常被分为4层，分别是：表示层，业务层，持久层和域模型层。通过分层，可以降低系统各部分之间的耦合程度，有利于开发人员的分工，增加系统的可维护性及可扩展性。

在进行具体框架介绍之前，有必要首先明确各层的职责，以便更好地理解相关框架的机制，为框架的选择和整合做好准备。各层的职责及简要说明如下：

(1)表示层：基于Web的应用面向用户的部分。职责：管理用户的请求，作出相应的响应；提供一个控制器，将页面的请求委派给其它层进行处理；为显示提供业务模型数据； 验证UI界面的输入内容。

(2)持久层：Web应用服务器端的最终部位，它是与数据库直接发生关系的层。职责：建立持久化类及其属性与数据库中表及其字段的对应关系；提供简化SQL语句的机制；实现数据的CRUD操作(创建、读取、更新及删除)；数据库连接的建立与管理。

(3)业务层：一个典型Web应用的中间部分，该层最容易被忽视。职责：建立持久化类及其属性与数据库中表及其字段的对应关系；提供简化SQL语句的机制；实现数据的CRUD操作(创建、读取、更新及删除)；数据库连接的建立与管理。

(4)域模型层：域模型层的Java类由实际需求中的业务对象组成，如用户、产品等。职责：业务领域相关对象的OO(面向对象)表现；在不同层之间传递数据，实现粗粒度的传递方式，提高系统的性能；为表示层提供表现所需要的数据源；为持久层提供被持久的对象。

2 开源框架介绍

针对表示层、持久层、业务层及整个应用程序，Java社区中都有相应的多种框架解决方案。其中，我们选择3种主流的开源框架Struts、Hibernate和Spring分别作为各层的相应实现。下面将对它们进行简要介绍，并给出我们选择相应框架时的策略。

2.1 Struts

勿庸置疑，Struts是目前Java Web MVC框架中不争的王者。经过长达5年的发展，Struts已经逐渐成长为一个稳定、成熟的框架，并且占有了MVC框架中最大的市场份额。但是Struts在某些技术特性上已经落后于新兴的MVC框架。面对Spring MVC、WebWork2这些设计更精密、扩展性更强的框架，Struts受到了前所未有的挑战。但站在产品开发的角度，Struts仍然是最稳妥的选择，因为目前市场上有大批精通

作者简介：李守振(1981－)，男，硕士生，主研方向：网络数据库，Web应用开发；张南平，教授；常国锋，助教
收稿日期：2006-02-05 **E-mail**：lishouzhen@126.com

Struts 的程序员，只需对他们进行相应的业务培训即可。这是 Struts 的战略性优势，其他 MVC 框架目前还无法在这点上与之并驾齐驱。

故选用 Struts 并结合 JSP 技术来实现我们的表示层。

2.2 Hibernate

Hibernate 是一个强悍的、成熟的、高性能的对象/关系数据库映射框架，它对 JDBC 进行了轻量级的对象封装，使得 Java 程序员可以使用面向对象编程思维来操作关系数据库。它不仅管理 Java 类到数据库表的映射，还提供数据查询和获取的方法，可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间。它还拥有一种功能非常强大的查询语言 HQL，这种语言与 SQL 非常相似，便于开发人员掌握，它是完全面向对象的，查询的是持久对象。更重要的是，Hibernate 支持大部分主流数据库。

由于在目前的应用系统中，大多数情况下，必须同时面向对象和关系型数据库进行设计和开发，因此将 Hibernate 作为持久层的实现是一种不错的选择。

2.3 Spring

Spring 是一个从实际项目开发经验中抽取的、可高度重用的应用框架。该框架立足于“依赖注入”的设计思想。将 Spring 作为业务层的实现可以带来如下好处：

(1)Spring 是一个非强制性框架，允许我们选用任何独立的部分，且不需要实现特定框架指定的接口。

(2)组件间的依赖关系减少，极大改善了代码的可重用性。Spring 的依赖注入机制，可以在运行期为组件配置所需资源，而无需在编写组件代码时就加以指定，从而在相当程度上降低了组件之间的耦合，实现了组件真正意义上的即插即用。这也是 Spring 最具价值的特性之一。

(3)面向接口编程。Spring 对于面向接口设计的意义在于，它为面向接口编程提供了一个更加自然的平台。基于 Spring 开发，程序员会更自然地倾向于使用接口来定义不同层次之间的关联关系。

3 框架的整合与实践

本文通过一个 User 对象的 Login 动作，展示各层是如何建立的，并通过分析一个请求如何贯穿于各层来揭示整个框架的概貌。

3.1 项目的目录结构

良好的目录结构能反映出应用的分层，并易于代码维护。以下是本示例的目录结构：

· com.mycompany.system.action.LoginAction—Strust Action 类；

· com.mycompany.system.form.LoginForm—Struts Form 类；

·com.mycompany.system.bean.User 和 User.hbm.xml—域对象和 Hibernate 映射文件；

· com.mycompany.system.biz.ILoginBiz—Spring 业务类接口；

· com.mycompany.system.biz.imp.LoginBiz—Spring 业务类接口的实现；

· com.mycompany.system.dao.ILoginDao—Hibernate 持久化类接口；

· com.mycompany.system.dao.imp.LoginDao—Hibernate 持久化类接口的实现。

3.2 域模型层

我们把 Oracle 数据库作为数据持久化的方式，并建立表 USER。通过 Hibernate 官方工具 Middlegen-Hibernate 和 Hibernate-Extensions 生成 USER 表的映射文件 User.hbm.xml 和域对象 User.java。

Hibernate 通过映射文件 User.hbm.xml 建立对象 User 与数据库表 USER 的关联，并把对持久化对象的操作转化为对数据库表的操作。另外，把对象 User 作为 LoginForm 的一个属性，以接收客户端的数据录入，并作为一个整体在 action、biz、dao 各层传递数据。

3.3 表示层的建立

结合 JSP 技术和 Struts 标签来建立与用户的交互界面，其内容相对简单，它只是提供一个包含“用户名”和“密码”的 Form，让用户输入登录信息。包含用户登录信息的 URL 请求“/LoginAction.do?operate=login”将被 Struts 的控制器 ActionServlet 接收，其通过查找 struts-config.xml 配置文件来决定该请求将被哪个 action 处理。Struts 将根据配置文件中的 parameter 参数直接查找 LoginAction 中 login 方法进行相应的处理。

所有的 Struts Action 类都继承自 BaseAction，基类 BaseAction 完成 Spring 上下文 applicationContext.xml 的加载，并提供一个公共的服务定位器方法 getBean()，它能根据传入的参数查找 Spring 的 Bean 资源，返回相应的接口，而接口的实现由 Spring 根据上下文动态注入。这里是表示层与 Spring 业务层的接口，利用 Spring 的“依赖注入”和“面向接口编程”的特性，保证了它们之间的松散耦合。

LoginAction 的 login 方法接收页面数据，通过服务定位器查找名为“myLoginBizProxy”的 Bean 资源，返回业务类的接口 ILoginBiz，并以域模型 User 对象为参数调用业务接口的 loadUser 方法，完成登录的业务逻辑验证。

3.4 业务层的构建

构建 Spring 业务层主要完成以下两方面的任务：

(1)配置 Spring 上下文 applicationContext.xml。该上下文将应用开发所需资源搭配起来，并指定它们之间的依赖关系。其中，我们配置了一个的名为“myLoginBizProxy”业务代理 Bean 节点，该节点的 parent 参数指定事务管理的策略，target 属性指定要代理的对象为“myLoginBiz”，并紧接着配置名为“myLoginBiz”的 Bean 节点，该节点有一个 loginDao 属性，它是业务层与 Hibernate 持久层的接口，该属性引用“myLoginDao”的 Bean 节点，最后配置“myLoginDao”节点。业务层通过调用 loginDao 属性的 getter 方法获得对 ILoginDao 接口的引用，进而以域对象 User 为参数调用该接口的方法 loadUser，以完成最终的持久化任务。而接口 ILoginDao 的实现由 Spring 通过“设值注入”的方式动态加载，这样就保证了业务层与持久层之间的松耦合。

(2)书写业务逻辑代码，包括 ILoginBiz 接口与其实现 LoginBiz。

3.5 持久层的搭建

我们已选定用 Hibernate 框架作为持久层实现，该框架对持久层的实现是相当全面和高效的。我们所要做的只是建立数据库表与 Java 对象的映射关系并书写持久化类接口 ILoginDao 及其实现 LoginDao。幸运的是，Hibernate 提供的官方工具能帮助我们自动完成前一任务。同样幸运的是，Spring 对 Hibernate 提供了良好支持，它对 Hibernate 的 SessionFactory 进行了整合，无需再通过 hibernate.cfg.xmll 对 SessionFactory 进行设定，Spring 上下文 mySessionFactory 节点包含了对 SessionFactory 的所有配置。另外，Spring 采用面向 Hibernate 的事务管理器实现 HibernateTransactionManager 来对 Hibernate 会话进行基于容器的、统一的事务管理。更为重要的是，Spring 提供了 HibernateTemplate，其对 Hibernate Session 操作进行了良好的封装，借助 HibernateTemplate，只需书写 HQL 语句并调用相应的模版方法，从而可以脱离每次数据操作必须首先获得 Session 实例、启动事务、提交/回滚事务以及烦杂的 try/catch/finally 的繁琐操作，从而获得精干

的代码呈现。通过以上的示例和分析，我们的 Web 开发框架已经成型，图 1 是该框架的总体结构示意图。

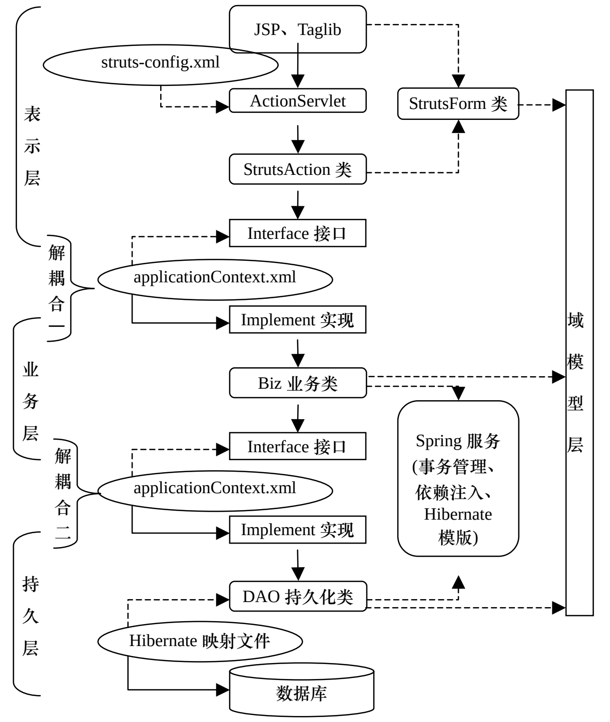

图 1 基于 Struts、Spring 和 Hibernate 的 Web 开发框架

4 结束语

本文所设计的 Web 开发框架基于良好的应用程序分层和成熟的开源项目，具有结构清晰、松散耦合、可扩展和可维护性好的特点，已在菲旺公司的办公自动化和人力资源管理系统中得到了成功应用。该框架表示层采用最为成熟的 MVC 实现 Struts；持久层则选用非常优秀的 ORM 产品 Hibernate，以完成对象和关系数据库之间的映射，它还提供高效的数据存储和获取能力；而业务层选用 Spring，它对 J2EE Web 应用开发的几乎所有方面都进行了良好的封装与抽象，同时它又是一个非强制性的框架，允许我们只选用任何一个独立的部分。的确，在实际的框架设计和应用开发中，Spring 的更为自然的面向接口编程、业务对象的依赖注入、对 Hibernate 的模版封装、声明式的事务处理、统一的配置文件等给我们带来了诸多便利。结合 Struts、Spring 和 Hibernate 的框架组合被认为是目前开发 J2EE Web 应用最为理想和成熟的框架，这种开发模式将会逐渐被广大程序员所接受，并在实际开发中得到广泛应用。

参考文献

1 赵增建. 应用系统框架分类及其应用情况概述[J]. 世纪杂志, 2004, 8 (3).

2 黄烟波, 张红宇, 李建华. 基于 Struts 和 Hibernate 的 J2EE 架构[J]. 计算机时代, 2004, 10(4).

3 夏　昕, 曹晓钢, 唐　勇. 深入浅出 Hibernate[M]. 北京：电子工业出版社, 2005.

4 Eagle M. Wiring Your Web Application with Open Source Java [EB/OL]. http://www.onjava.com/pub/a/onjava/2004/04/07/wiringwebapps.html?page=1.

~~~~~~

(上接第 256 页)

定义 7 数据资源属性定义

<数据资源属性>∷=〈文本数据资源属性>|〈视频数据资源属性>|〈音频数据资源属性>|<图像数据资源〉

<文本数据资源属性>∷=<A><文本数据资源特性>

<文本数据资源特性>∷=A

<视频数据资源属性>∷=<B><视频数据资源特性>

<视频数据资源特性>∷=A

<音频数据资源属性>∷=<C><音频数据资源特性>

<音频数据资源特性>∷=A

<图像数据资源属性>∷=<D><图像数据资源特性>

<图像数据资源特性>∷=A

其中：A 表示不区分特性。

3.3 几种典型资源类型的分类例子

(1)硬件资源

1)处理器资源。AAA：表示普通单 CPU 的 PC；AAB：表示多 CPU 的 PC；AAC：Workstation；AAD：小型机；AAE：集群；AAF：巨型机。

2)存储资源。ABD：内存 512MB；ACA：硬盘 20GB；ACB：硬盘 40GB。

(2)软件资源

1)系统软件。BAA：Windows 操作系统；BAE：Oracle 数据库系统。

2)工具软件。BBA：字处理软件；BBB：数据库开发工具。

3)应用软件。BCA：维修技术保障指挥控制软件；BCB：维修资源优化软件；BCC：三维虚拟战场软件；BCD：远程维修支持软件；BCE：备件优化选择软件。

(3)通信资源

CAA：(0-10Mbps)；CAB：(10-100Mbps)；CAC：(100-1 000Mbps)；CBA：(1 000Mbps-∞)。

(4)数据资源

DAA：电子书籍；DBA：视频数据；DCA：音频数据。

4 结束语

资源组织模型是网格资源管理的重要研究内容，对资源管理效率具有重要的意义。本文结合军事应用网格需要，提出了一种具有较高管理效率的基于自治资源域的逻辑资源树模型并将其应用到装备维修保障网格中。同时也对网格资源的描述方式进行了探讨，给出了网格资源的形式化表示方式，使资源描述简洁清楚。本文的后续工作是围绕资源模型开展维护和查找算法的研究。

参考文献

1 杨文广, 武永卫, 朱　晶. 一种全局统一的层次化网格资源模型[J]. 计算机研究与发展, 2003, 40(12): 1763-1769.

2 Foster I, Kesselman C. 金　海译. 网格计算(第 2 版)[M]. 北京：电子工业出版社, 2004.
~~~~~~