

A Layered WEB Framework Model

Dengshan Tian, Jiwen Wen, Yin Liu, Ning Ma, Huiping Wei

School of Economics and Management, Beijing Forestry University, 100083, Beijing, China

Email: tds3218@163.com, wjwlinda@163.com, liuyin0404@126.com, maningmailbox@yahoo.com.cn, wei_11008@sina.com

Abstract: The requirements of Web application keep changing all along. To reduce the time-costing, loose-couple the modules, and adapt to the changing requirement, layering approach is used when developing Web application. After introducing four framework technologies: Module-View-Control(MVC); Inversion of Control (IOC); Aspect Oriented Programming (AOP); Object-relational mapping (ORM), Web application layering model based on framework technology is proposed at last. This model can improve the testability, scalability and maintainability of whole application, enhance the code reusability as well. With this model, programmer can ignore system-level coding when developing Web application, thereby devote to the business logic developing, and the developing efficiency is greatly improved.

Keywords: Layered approach; Module-View-Control; Inversion of Control; Aspect Oriented Programming; Object Relational Mapping; Web Development

WEB 分层框架模型研究

田登山, 温继文, 刘 音, 马 宁, 卫慧平

北京林业大学经济管理学院, 北京, 中国, 100083

Email: tds3218@163.com, wjwlinda@163.com, liuyin0404@126.com, maningmailbox@yahoo.com.cn, wei_11008@sina.com

摘 要: 为灵活应对 Web 应用的需求变化, 缩短开发周期, 降低程序模块间的耦合度, 将分层方法引入到 Web 应用开发中; 接着基于分层方法对四种框架技术(MVC、IOC、AOP 和 ORM)做了介绍; 最后提出了基于框架技术的 Web 分层模型。该模型可以提高程序的可测试性、可扩展性和可维护性, 提升了代码复用率。采用该模型进行 Web 应用的开发的程序员可以忽略系统级编码, 从而致力于业务逻辑开发, 大大提高了开发效率。

关键词: 分层方法; MVC; 控制反转; 面向方面的编程; 对象关系映射; Web 开发

1 引言

分层方法因为其可伸缩性、可配置性、灵活可靠性和体现软件集成的思想, 而成为广泛使用的软件应用技术^[1, 2], Web 工程也不例外。分层方法在 Web 工程中的广泛应用, 使得 Web 开发技术中出现了许多框架技术: MVC、ORM、IOC 以及 AOP。

MVC 是 Xerox PARC 在八十年代为编程语言 Smalltalk-80 发明的一种软件设计模式, 已被广泛应用于各种 Web 编程语言。孙凌燕等研究了 Struts1.1 的框架, 并对如何应用于开发做出了阐述^[3]。朱少敏等研究了 Webwork 框架, 并介绍了如何应用 Webwork 进行开发^[4]。作为数据持久层框架, ORM 为数据的存取

提出了面向对象的解决方案。吴卫平等研究了 ORM 原理, 并对流行的 ORM 框架做出了比较^[5], 李斌勇等对 ORM 映射原理进行了研究^[6]。为了降低 Web 各层的耦合度, Martin Fowler 提出 IOC 模式^[7], 姜锋等对 IOC 原理进行了研究, 并给出了轻量级 IOC 容器的设计^[8]。对于各层的公共性问题, AOP 的明确捕获和模块化关注点的机制, 可以有效的解决由于横切面属性引起的代码分散和代码混论等问题^[9], 赵艳妮研究了 AOP 框架技术并实现了该框架^[10]。但是, 以上研究只是局限于某一种框架技术。

李春红等利用分层方法改进了 MVC 框架^[11]; 孟凡新介绍了如何基于 AOP 技术和 Web 分层模型进行开发^[9]; 王非等介绍了如何使用 Struts、Spring 和 Hibernate 开发电子商务系统^[12]。但是综合应用分层方法和框架技术进行 Web 开发, 如何降低模块耦合度,

本文受北京林业大学科技创新计划项目(BLRW200922)资助。

国内研究较少。

为此, 本文将研究分层方法以及主流的 WEB 框架技术, 提出一个基于框架技术的 Web 分层模型, 为开发人员、项目管理人员提供参考。

2 分层方法与框架技术

分层方法因其可扩展性、灵活性而受到了广泛关注和应用。体系结构的分层方法把系统组织成一系列的层次，每一层提供一组服务。分层方法最好的例子就是 **OSI** 的七层模型。分层的方法支持系统的增量式开发。如一个层被开发完，该层提供的服务就可以被用户使用了。这个体系结构还是可改变的和可移植的。如果一层的接口被保留下来，这个层就能被另外的一个层替换。当一层的结构发生变化时，只有毗邻的层受影响。

虽然应用分层方法进行 Web 开发，会增加 Web 应用的复杂度。但是，WEB 应用程序的可扩展性、可移植性、可测试性都得到了很大的提升。因此，分层模式被广泛应用到 Web 应用开发中来，且已经成为主流。

随着 Web 应用技术的发展,涌现了许多 Web 开发框架,以提供系统的设计重用性和系统的可扩展性,从而缩短大型应用软件系统的开发周期,提高开发质量。常用的 Web 框架有: MVC、ORM、IOC 和 AOP。

2.1 MVC

MVC 是一个设计模式，它强制性的使应用程序的输入、处理和输出分开。MVC 主要由三个部分组成：模型(Module)、视图(View)和控制器(Control)。

模型、视图、控制器三者之间的关系图 1 所示。

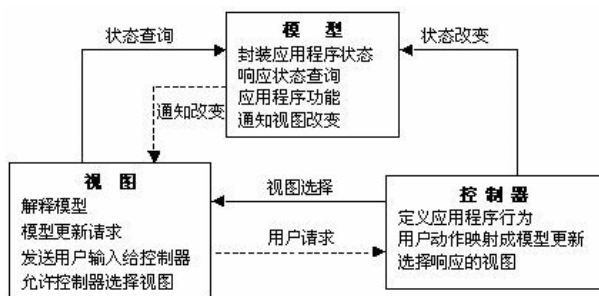


图 1 MVC 框架原理图

图 1 中, 视图代表用户交互界面, 对于 Web 应用来说, 可以概括为 HTML 界面, 但有可能为

XHTML、XML 和 Applet。随着应用的复杂性和规模性，界面的处理也变得具有挑战性。一个应用可能有很多不同的视图，MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交予模型(Model)处理。比如一个订单的视图只接受来自模型的数据并显示给用户，以及将用户界面的输入数据和请求传递给控制和模型。

模型(Model)就是业务流程/状态的处理以及业务规则的制定。业务流程的处理过程对其它层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是 MVC 最主要的核心。

控制(Controller)可以理解为从用户接收请求,将模型与视图匹配在一起,共同完成用户的请求。控制层的作用相当于一个分发器,选择什么样的模型,选择什么样的视图,可以完成什么样的用户请求。控制层并不做任何的数据处理。例如,用户点击一个连接,控制层接受请求后,并不处理业务信息,它只把用户的信息传递给模型,告诉模型做什么,选择符合要求的视图返回给用户。因此,一个模型可能对应多个视图,一个视图可能对应多个模型。

MVC 设计模式具有设计清晰，易于扩展，运用可分布的特点，因此在构建 Web 应用中具有显著的优势。可适用于多用户的、可扩展的、可维护的、具有很高的交互性的系统，如电子商务平台、CRM 系统和 ERP 系统等。

引入 MVC 框架后,开发人员避免了服务器端 Servlet 层的重复性工作。许多重复性且基本相似的工作都交由控制器完成,编码者可以致力于应用级别的开发(业务逻辑处理)。J2EE 平台 MVC 框架有 Struts、Webwork 和 Spring MVC;而 .Net 平台下常用的 MVC 框架为 WebForms。

2.2 ORM

ORM 是在关系型数据库和对象之间产生一个自动映射，访问数据库的代码就不需要再与复杂的 SQL 语句打交道。编码人员只需要关注业务逻辑，而不是底层的重复性的数据库 SQL 和 JDBC 代码。目前 ORM 已经成为面向对象开发中的主流开发模式。

ORM 框架的核心设计理念如图 2 所示:

从图 2 可以看出, 每张表由多个字段组成, 每张表都包含若干条记录。面向对象的编程思想中, 类是

对对象的抽象，类有多个属性组成，通过类可以构建多个对象。通过 ORM 框架，在类与表之间建立映射关系，确定表的字段与类的属性的一一对应关系。同时，在映射关系中明确类的那个属性对应数据表的那个字段，同时定义类的属性与数据库表的主码的对应关系。这样一来，ORM 框架可以自动维护普通值对象与数据库记录间的一致性。至于数据库表与表之间的一对一、一对多和多对多的关系，在逻辑模型上与类之间的关联刚好对应，也可以由 ORM 框架自动完成映射。

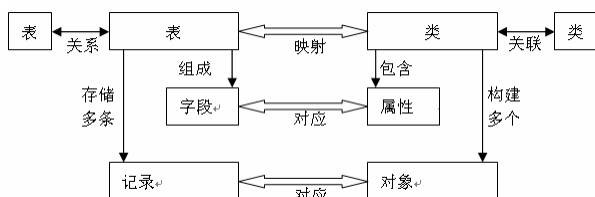


图 2 ORM 映射原理

ORM 框架分为自动化 ORM 框架和半自动化 ORM 框架两类。所谓的自动化是指对于开发者而言，直接操作类和对象，ORM 框架负责维护数据库，典型代表框架为 Hibernate。开发者使用半自动化 ORM 框架则需要告知该框架对应的 SQL 语句，其典型代表框架为 iBatis。

J2EE 平台常用的 ORM 框架有 Hibernate、iBATIS、EJB、Apache 的 Cayenne；.Net 平台下则可以使用 LINQ to SQL、iBATIS.NET 和 NHibernate。引入 ORM 框架后，只需要通过少量代码就可以完成数据库的存取访问，系统级别的控制代码再次得到了缩减。

2.3 IOC

所谓的对象的依赖关系，也就是被依赖的对象在依赖对象构建之前必须已经存在，否则依赖对象将无法构建。为了简化这种依赖关系，可以考虑把自定义类将交由容器去控制，而不是在类的内部控制，类与类之间的关系将交由容器处理，一个类在需要调用另一个类时，只要调用另一个类在容器中注册的名字就可以得到这个类的实例^[10]。这样一来，所有类的关系都交由容器来控制，这就是 IOC。IOC，即控制反转，又称依赖注射 DI(Dependency Injection)，其核心是“推，拉”模式中的“推”模式，编程方式与传统的也有很大的差异。

IOC 核心理念是设计一个容器，能够维护类之间

的关系。为了使类关系灵活度最大化，通常会使用 XML 类型配置文档来存储类之间的关系。主流的 J2EE 平台 IOC 容器为 Spring，对应 .Net 平台的 IOC 容器为 Spring.NET。

2.4 AOP

对于模块间的公共行为，传统的设计理念无法将这种公共行为分离出来，比如 Web 工程常用到的用户操作的日志记录、数据库事务处理、权限控制等问题。这种公共行为关注的是程序执行过程中的某一个时刻点。而 AOP 则刚好可以解决这个问题。

AOP 是由 Kiczales 等在 1997 年提出的，其基本思想是将横跨多个模块的行为分离出来，并封装为一个新的模块。可单独对它进行编程、修改，在需要时使用编织器把这个模块编织到原有的代码中。它克服了 OOP 不能处理跨越多个不相关模块的行为的缺陷。

AOP 的核心概念是关注点，指在程序执行过程中定义好的点，如方法调用、循环开始和对象构建等。具有横切多个模块的行为的一类特殊关注点称为横切关注点。AOP 提供了一种描述横切关注点的机制，能够自动将横切关注点织入到面向对象的软件系统中，从而实现模块化。被模块化的横切关注点就称为方面。开发者可以在编译时更改、插入或删除系统的方面，甚至重用系统的方面。

J2EE 平台下的 Spring 框架和 .Net 平台的 Spring.NET 都支持 AOP 功能。

3 基于框架技术的 Web 分层模型

综上，引入 MVC、ORM、IOC 和 AOP 框架技术后的 Web 分层模型如图 3 所示：

图 3 中，基于框架技术的 Web 分层模型由六层和横跨多层的一个 IOC 容器和一个 AOP 框架组成：

(1)表示层，即客户端浏览器显示的页面。在 J2EE 应用中，表示层可以使用 HTML 或者 Java Server Page(JSP)来实现。当然也可以利用框架来实现，比如 Adobe Flex、Struts 2 或者 JSF；而 .Net 应用中，可以使用 HTML 或者 Web Forms 页面(.aspx 后缀文件)来实现。

(2)MVC 框架层接收来自客户端的 HTTP 请求，并通过 HTTP 响应向客户端发送结果页面。该层即 MVC 框架的控制器部分，开发者无需编码。J2EE 平台下可以使用的 MVC 框架有：Struts、Webwork、Spring MVC，.NET 平台可以使用 ASP.NET MVC 框架。

(3)MVC 模型层,即按照 MVC 框架约束建立的模型(Module)对象,用于处理客户端的请求。模型层负责调度业务逻辑层对象,完成业务处理,最后告知 MVC 框架结果页面的名称。如果基于 Struts 构建 J2EE 应用,则 Module 对象的定义类需要继承 Action 类;而基于 ASP.NET MVC 框架构建的 .NET 应用,则 Module 对象的定义类需要继承 Controller 类。

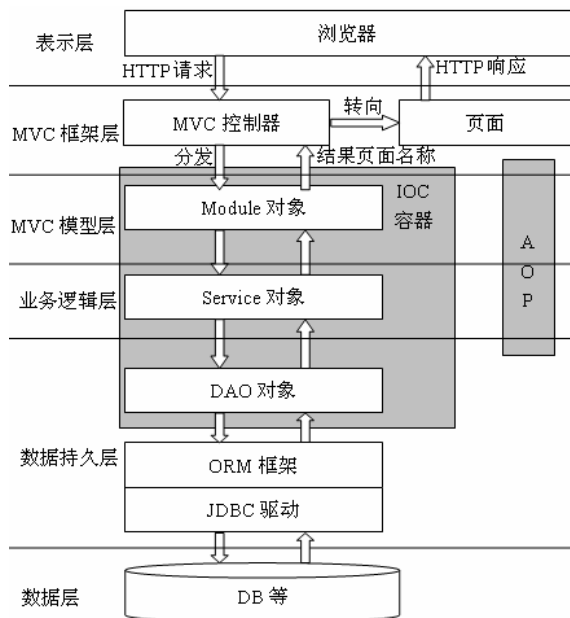


图 3 基于框架技术的 Web 分层模型

(4)业务逻辑层由 Service 对象组成,负责具体的业务逻辑处理,包括:指令处理、操作约束、异常处理以及业务策略。业务逻辑通常用于判断当前情形下执行那个处理动作。对于 J2EE 应用和 .Net 应用而言,Service 对象定义类使用普通类进行定义。

(5)数据持久层。数据可以存储在关系型数据库、面向对象的数据库、LDAP 数据库、文件等,而数据访问对象(Data Access Object,即 DAO)可以抽象并封装数据访问操作,并管理与数据源的连接以及存取操作。

DAO 大多用于访问数据库,为了减少系统级的编码,编码者会使用 ORM 框架,完成数据库访问操作。而与数据库的交互则由 JDBC 类库完成。对于 J2EE 应用和 .Net 应用而言,DAO 对象的定义类使用普通类进行定义。

(6)数据层,多数情况下数据存储在数据库服务器,如 Oracle、DB2、MySQL 等。少数情况下可以使

用 LDAP 服务器、文件甚至是底层的套接字。

(7)IOC 容器,负责维护 Module 层、Service 层和 DAO 层对象及其关系。也就是说,开发者自定义的类及其依赖关系通过配置文件由 IOC 容器进行维护,从而最大程度的增加系统代码的灵活性。

(8)AOP 框架,用于在 Module 层、Service 层和 DAO 层之间以及 Module 层之前设置关注点,灵活的设置关注行为,并实现了切面代码的复用性,从而实现了模块间行为的重用。

从图 3 可以看出,客户端浏览器通过 URI(Uniform Resource Identifier)发送请求,通过 HTTP 协议经网络传送到服务器端的 Web 服务器;经 MVC 框架转发至对应的模型对象,模型对象则负责把请求信息传送到 Service 对象,由 Service 对象调度基于 ORM 框架的 DAO 对象,完成业务的持久化存储。处理结果按照相反的次序进行传递,从 DAO 至 Service,再传递至模型对象当前的处理结果,由模型对象负责告知 MVC 框架结果页面名称,最后由 MVC 框架负责把结果页面通过 HTTP 响应传送到客户端浏览器。

对于开发人员而言,只需要对页面、Module 类、Service 和 DAO 进行编码,并根据 IOC 容器和 AOP 框架的要求进行配置即可,其他各层无需编码。

3.1 IOC 容器与可测试性

对象信息及其依赖关系都需要通过 IOC 进行配置。也就是说,原先需要手动编码控制构建对象的过程,都交由 IOC 容器来完成,甚至于对象之间的依赖关系都可以通过 IOC 来自动解决。这样一来,可以通过 IOC 容器随意构建任何一个对象,这使得对象的测试变得更加容易,程序的健壮性得到了保证。测试性能的提高,保证了系统的可维护性,也为极限编程的应用提供了技术支持。

3.2 需求变化应对

图 3 中的 Web 分层模型基于分层方法,自然可以灵活应对 Web 应用需求的变化。如果希望支持无线应用协议(WAP)协议,表示层修改为无线标记语言(WML),同时适当修改 Module 类,其他各层不受任何影响。

该模型基于 MVC 框架和 ORM 框架技术,编码人员关注的是业务逻辑而不是系统级编码,提高了开发效率。使用 IOC 容器维护对象及其关系,使得模块间

的耦合度降到了最低。即使需求发生了变化,编码人员可以在最短时间内快速作出调整,系统维护的成本也大大降低。

3.3 AOP 与模块间公共事务

引入 AOP 后,服务器端层与层之间的公共模块变得可以复用,编码的灵活性得到了很大的提升,业务模块开发者无需关注公共行为,从而提高开发效率。

如果需要把用户权限控制、用户行为日志记录、数据库事务这些公共行为单独独立出来,AOP 提供了良好的支持。开发人员只需要在 Module 类、Service 类和 DAO 类中寻找合适的关注点,配置处理逻辑对象。比如 Module 行为执行前,通过权限控制以此决定当前用户是否可以访问该也无模块;Module 行为执行结束后,记录执行结果,以此记录用户的行为日志。对于 Service 层,需要根据业务要求调用一个或者多个 DAO。对于分布式 WEB 应用,Service 层可能会调用不同数据库服务器的多个 DAO,因此,必须要在 Service 层加入数据库事务处理。通过 AOP,也可以在 Service 层之前,通过 IOC 容器选择性植入 AOP 事务处理模块。总之,AOP 框架使得各层之间的公共性交互变得更加容易处理,为应用程序的开发人员提高了更大的自由度。

结论

基于框架技术的 Web 分层模型因其可伸缩性能、可配置型、灵活可靠性,对于解决传统开发方法中存在的问题具有显著作用。该模型具有以下特点:

(1)框架技术的引入,可以使得编码人员忽略系统级别的编码,而致力于业务逻辑编码。采用基于框架技术的 Web 分层模型可以提高软件开发效率,增强系统的可移植性、可维护性以及互操作性。

(2)基于框架技术的 Web 分层模型可以灵活应对系统需求变化,实现的系统可扩展性强,易于升级和维护。

(3)框架技术 MVC、ORM、IOC 和 AOP 提高了基于框架技术的 Web 分层模型的可测试度,能够结合极限编程思想快速开发 Web 应用。

(4)基于框架技术的 Web 分层模型代码复用率高,在很大程度上降低了系统工作量,为程序的快速开发提供了保证。

但是,该模型基于分层模型和框架技术,系统复杂度稍高。此外,由于框架技术的引入,开发团队的人员素质要求较高。但是 Web 工程的需求持续变化,

该模型能快速作出调整,因此该模型得到了广泛认可并在许多大型 Web 工程中得到了广泛应用,并逐步成为 Web 开发的主流。

References(参考文献)

- [1] Y. L. ZUO, Z. Q. HUANG, P. GAO, B. Z. ZHANG, and D. W. ZUO, "Design and Implementation of Distributed Multilayer Application Systems," Computer Engineering, Vol. 27(3), pp. 176-177, 2001 (Ch).
左银龙, 黄志球, 高鹏等. 分布式多层应用系统的设计与实现[J]. 计算机工程. 2001, 27(3): 176-177.
- [2] S. Sarkar, G. Maskeri, and S. Ramachandran, "Discovery of architectural layers and measurement of layering violations in source code," JOURNAL OF SYSTEMS AND SOFTWARE, Vol. 82(11), pp.1891-1905, 2009.
- [3] L. Y. SUN, B. L. LU, and J. SUN, "Design and Study of Web Application Framework Based on Struts," Computer Engineering, Vol. 31(8), pp.57-59, 2005 (Ch).
孙凌燕, 陆保岚, 孙健. 基于 Struts 的 Web 应用框架设计与研究[J]. 计算机工程. 2005, 31(8): 57-59.
- [4] S. M. ZHU, J. M. LIU, "Design and realization of government procurement system based on lightweight J2EE framework," Computer Engineering and Design, Vol. 28(24), pp.5945-5948, 2007 (Ch).
朱少敏, 刘建明. 基于轻量级 J2EE 平台的政府采购系统的设计与实现[J]. 计算机工程与设计. 2007, 28(24):5945-5948.
- [5] W. P. WU, L. F. WANG, Z. J. JIANG, "Research of Data Persistent Layer Based on ORM," MICROELECTRONICS & COMPUTER, Vol. 25(7), pp.188-190, 2008 (Ch).
吴卫平, 王丽芳, 蒋泽军. 基于 ORM 的数据持久层框架研究[J]. 微电子学与计算机. 2008, 25(7): 188-190.
- [6] B. Y. LI, Q. LI, "Research on ORM Mapping Mechanism Based on NHibernate," COMPUTER TECHNOLOGY AND DEVELOPMENT, Vol. 19(7), pp.32-34, 2009 (Ch).
李斌勇, 李庆. 基于 Nhibernate 的 ORM 映射机制研究[J]. 计算机技术与应用. 2009, 19(7): 32-34.
- [7] F. Fowler, "Inversion of Control Containers and the Dependency Injection pattern," [EB / OL].
<http://martinfowler.com/articles/injection.html>. 2003-1-4.
- [8] F. LOU, Y. SUN, "Design and Implementation of Lightweight IoC Container," COMPUTER TECHNOLOGY AND DEVELOPMENT, Vol. 17(1), pp.91-93, 2007 (Ch).
娄锋, 孙涌. 轻量级 IOC 容器的研究与设计[J]. 计算机技术与应用. 2007, 17(1): 91-93.
- [9] F. X. MENG, J. J. ZHANG, G. Y. LIU, "Multi-tier Distributed System Based on AoP and Web Services," Computer Engineering, Vol. 36(1), pp.62-63, 2010 (Ch).
孟凡新, 张京军, 刘光远. 基于 AOP 和 Web 服务的多层分布式系统[J]. 计算机工程. 2010, 36(1): 62-63.
- [10] Y. N. ZHAO, Y. H. WANG, Y. LEI, "Research and implement of software framework based On IOC / AOP," Computer Engineering and Applications, Vol. 44(29), pp.92-95, 2008 (Ch).
赵艳妮, 王映辉, 雷宇. 一种基于 AOP / IOC 的软件框架研究与实现[J]. 计算机工程与应用. 2008, 44(29): 92-95.
- [11] C. H. LI, J. H. GAO, "Improved MVC based on layers model," Computer Engineering and Design, Vol. 28(4), pp.766-769, 2007 (Ch).
李春红, 高建华. 使用分层模型改进 MVC 设计架构[J]. 计算机工程与设计. 2007, 28(4): 766-769.
- [12] F. WANG, Y. DING, W. C. LIN, "Design and development of E-business system based on framework technique," Computer Engineering and Design, Vol. 28(22), pp.5507-5510, 2007 (Ch).
王非, 丁元, 林文昌. 应用架构技术设计与开发电子商务系统[J]. 计算机工程与设计. 2007, 28(22): 5507-5510.