

# 基于 Struts2+Ajax+JDBC 的企业级 Java Web 架构<sup>①</sup>

欧阳宏基, 葛 萌

(咸阳师范学院 计算机学院, 咸阳 712000)

**摘 要:** 为提高 Java 技术在企业应用中的开发效率, 缩短开发周期, 减少开发成本, 通过研究 Struts2 框架、JDBC 协议以及 Ajax 异步刷新机制, 依据 MVC 设计模式的思想, 提出一种基于 Struts2+Ajax+JDBC 的企业级 Web 开发架构. 通过某企业生产统计管理系统的信息发布模块, 阐述了该架构中表示层、控制层、业务逻辑层和持久化层的实现过程. 实践表明该架构具有良好的移植性、扩展性和维护性, 降低了应用程序的复杂性和耦合度, 提高了开发效率和用户体验.

**关键词:** Struts2; Ajax; JDBC; Java Web 架构

引用格式: 欧阳宏基, 葛萌. 基于 Struts2+Ajax+JDBC 的企业级 Java Web 架构. 计算机系统应用, 2017, 26(8): 77-82. <http://www.c-s-a.org.cn/1003-3254/5883.html>

## Enterprise Java Web Architecture Based on Struts2+Ajax+JDBC

OUYANG Hong-Ji, GE Meng

(Computer College, Xianyang Normal University, Xianyang 712000, China)

**Abstract:** In order to improve the development efficiency, shorten the development cycle and reduce the development cost of Java technology in enterprise application, by researching Struts2 framework, JDBC protocol and Ajax asynchronous refresh mechanism, this paper proposes an enterprise Web development framework based on Struts2+Ajax+JDBC in accordance with the idea of MVC design pattern. According to the information release module of production statistics management system, it describes the implementation process of presentation layer, control layer, business logic layer and persistence layer in this architecture. Practice shows that the architecture has good portability, expansibility and maintainability, and it reduces the complexity and coupling of the application, and improves the development efficiency and user experience.

**Key words:** Struts2; Ajax; JDBC; Java Web architecture

随着互联网的发展以及在跨平台方面的优势, Java 已成为 Web 开发的一种主流技术. 目前的开发架构主要有面向小型规模应用的 Model2(JSP+Servlet+JavaBean)方案, 以及适合企业级应用的、以多种框架相结合为基础的方案. 在以框架为基础的企业级开发中, 以 Struts、Spring、Hibernate、Mybatis 等轻量级框架最具代表性, 是当前企业级开发的研究热点.

文献[1]提出一种基于 Struts1 与 Hibernate 框架集

成的开发方案, 主要是通过 Struts1 实现控制逻辑, 在 Action 组件中通过 DAO 模式调用 Hibernate API 来访问数据库. 文献[2,3]提出一种基于 Struts+Spring+Hibernate 框架的开发方法, 并应用到 EPON 性能监控系统 and 生产统计管理系统中, 通过系统实施证明了可靠性和稳定性. 文献[4]提出了 Spring+Mybatis 框架的开发方法, 并应用到信用卡业务系统的开发中, 证明了相关方法的稳定性和维护性. 虽然上述基于各种

① 基金项目: 咸阳师范学院专项科研计划项目(2014XSYK038)

收稿时间: 2016-12-02; 采用时间: 2016-12-19

框架的开发方法都具有很好的稳定性和扩展性,但都存在以下方面的缺点:(1)每个框架都只关注分层中的某一层或多层,相互之间需要整合来完成整体的开发,所以学习成本较高、初学者不易上手。(2)每个框架都需要配置文件或注解的支持才能运行,而且层与层之间需要域模型对象的支持才能传递数据,系统越复杂,配置文件和域模型对象越多,必然会导致开发效率的下降。(3)对于表示层数据的请求都是同步式的,用户体验较差。(4)Hibernate和MyBatis框架是建立在JDBC之上的,虽然使用它们能够使开发人员以面向对象思维访问关系型数据库,但是在执行效率、多数据表联合查询、支持存储过程等方面都不如JDBC。

本文在前者的研究基础上,提出一种基于Struts2+Ajax+JDBC的企业级Web开发方法,该方法具有以下优点:(1)满足MVC设计模式,实现了系统整体的松耦合。由Struts2担当控制逻辑与业务逻辑。(2)在传统JSP页面上融入Struts2标签和Ajax技术,能够实现页面异步刷新,从而节省网络数据传输,提高交互体验。(3)通过DAO封装JDBC实现数据的持久化,保留了JDBC访问数据库最快的优势。同时利用策略与模板设计模式对JDBC进行了改进,使得持久化逻辑便于进行单元与整体测试,最大限度降低了DAO实现类的代码量。

## 1 相关技术

### 1.1 Struts2 框架

Struts2是建立在WebWork与Struts1这两个框架基础上的新一代Web开发框架,架构如图1所示。核心组件包括:拦截器、Action控制器、标签库、配置文件和视图组件。其中核心拦截器StrutsPrepareAndExecuteFilter用于将请求定位到Action。Action执行业务逻辑,普通拦截器用于在Action执行之前或之后增加与业务逻辑“正交性”的功能,这些功能可以被重用,是AOP思想的具体实现。标签库以OGNL表达式为基础,提供了视图组件的数据处理、流程控制、国际化、Ajax支持等功能。组件之间、请求与响应的映射关系由配置文件关联起来。

### 1.2 Ajax 及其工作原理

Ajax是由JavaScript、XML、XHTML、DOM以及XMLHttpRequest等相互结合而成的一种页面异步刷新技术。它与传统Web的区别在于:浏览器与服务

器之间加了一个中间层-Ajax引擎,如图2所示。客户端表示层通过JavaScript将请求提交给Ajax引擎,数据编辑、页面导航等不需要重新载入整个页面的需求由Ajax引擎来执行<sup>[5]</sup>,只有确实需要从服务器读取数据时,再由Ajax引擎通过HTTP请求访问服务器,服务器端将响应的数据以XML或JSON格式传递给Ajax引擎,再通过HTML+CSS形式刷新客户端表示层。从用户提交请求到服务器响应的过程中,网络中传输的仅仅是异步交互的数据,页面只是局部更新。所以,Ajax极大减少了用户对服务器的冗余请求,减轻了服务器的访问负担,缩短了用户的实际等待时间,提高了交互体验<sup>[6,7]</sup>。

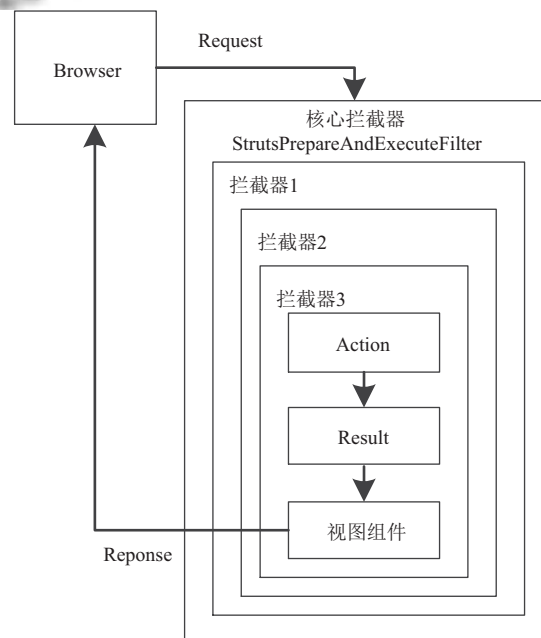


图1 Struts2 核心架构

JDBC是JavaEE规范中用来访问关系型数据库的一组协议,主要包括面向开发人员的JDBC API、驱动管理器、以及由数据库厂商提供的JDBC驱动程序。由于JDBC具有使用成本低、执行速度快、不依赖于业务容器、擅长多表联合查询等优点,在JavaEE的数据持久化操作中具有广泛的应用。开发人员使用JDBC,主要包括图3的5个核心接口<sup>[8]</sup>:(1)Connection:代表应用程序与数据库之间的一条连接,用来在上下文中创建其他的JDBC对象,并管理数据库事务。(2)Statement:用于执行不带参数的SQL语句并返回处理的结果。(3)PreparedStatement:Statement的子接口,由

于执行预编译的 SQL. (4) CallableStatement: 用于执行存储过程. (5) ResultSet: 由 SQL 查询语句生成的结果集对象, 并提供了遍历数据记录的方法. 进行 JDBC 开发都要经过 5 个步骤: ①加载驱动; ②创建 Connection; ③创建相关 Statement; ④执行 SQL 语句; ⑤释放资源.

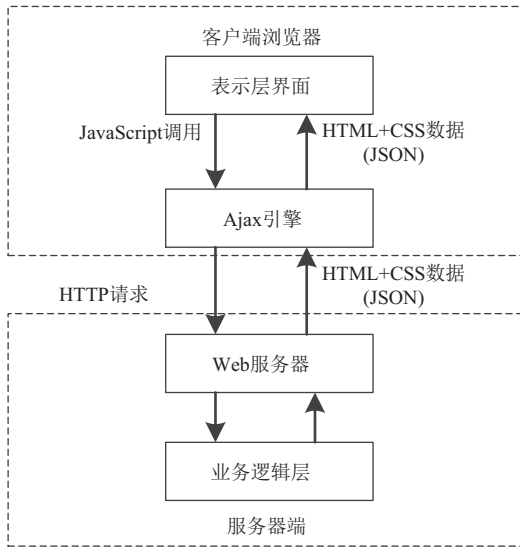


图 2 Ajax 网络应用模型

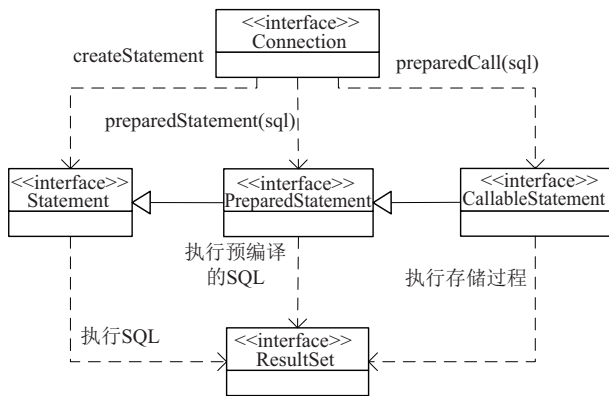


图 3 JDBC 核心接口

## 2 基于 Struts2+Ajax+JDBC 的 Web 应用架构

根据 MVC 模式, 构建的基于 Struts2+Ajax+JDBC 的 Web 架构如图 4 所示. 采用分层设计思想和 OOP 技术, 实现各层的通信和访问接口, 使该架构具有层与层之间松耦合、各层内部高内聚的特点. 在表示层采用 JSP 技术结合 Struts2 的标签库实现页面的构建和控制, 通过在 Struts2 中引入 Ajax 插件, 借助 Ajax 标签实现页面的局部刷新. 控制层由 Struts2 的核心过滤器和配置文件构成, 表示层提交的请求统一由

StrutsPrepareAndExecuteFilter 接收, 对于 Action 的请求由 struts.xml 配置文件确定映射关系. 业务层由 Struts2 的 Action 组件、业务对象和拦截器构成, 实现普通业务逻辑的管理. 通过拦截器实现系统业务逻辑的管理并完成流程的跳转. 数据持久层通过 DAO 等模式封装 JDBC 实现. 存储层是关系型数据库, 通过引入数据库厂商的 JDBC 驱动实现与数据持久层的交互.

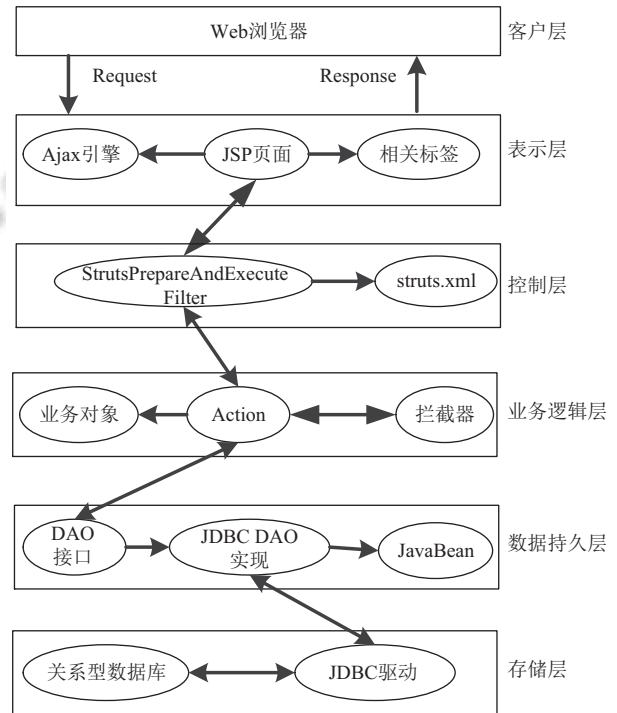


图 4 Struts2+Ajax+JDBC 的 Web 架构

## 3 企业信息发布模块的实现

某煤炭企业的生产统计管理系统是一个负责制定未来生产计划、统计以往各项生产数据, 合理安排煤炭的生产、运输、销售等工作的综合信息平台. 采用了笔者提出的 Struts2+Ajax+JDBC 的系统架构实现. 利用 Tomcat 6.0 作为 Web 服务器, 采用 MySQL5.0 作为数据库, JDK 为 1.6, 集成开发环境 MyEclipse9.1. 目前已在该煤炭企业发布应用, 并获得了较好的用户满意度和良好的经济效益.

根据业务需求系统分为: 基础数据维护、数据上报、矿级数据汇总、煤业公司级数据汇总、信息查询和信息发布等 6 个模块. 系统角色分为游客、各矿级科室业务员、矿级统计员、煤业公司业务员、煤业公司统计员和管理员. 其中, 游客是未注册的用户, 只具有浏览信息发布模块的权限. 矿级各科室业务员负责

矿级基础数据的维护和上报. 煤业公司业务员负责煤业公司基础数据的维护和上报. 矿级统计员将上传的各项基础数据分类汇总后形成统计报表和台帐. 煤业公司统计员接收矿统计科上传的矿级统计数据, 并检验数据的正确性, 依据上月累计数据生成本月累计数据以及年度累计数据. 管理员主要是对基础数据、统计数据等进行查询, 管理要发布的信息以及分配权限等. 下面以信息发布模块为例, 给出系统详细的实现过程.

### 3.1 表示层的实现

表示层主要与客户端进行交互, 进行数据的接收与显示, 由 JSP、Struts2 标签、CSS 和 JavaScript 构成. JSP 和 CSS 完成页面的布局; JavaScript 完成页面的部分动态效果以及数据格式的合法性校验; Struts2 标签负责呈现后台数据以及进行逻辑判断, 能够实现显示代码与 Java 代码的解耦合, 提高表示层的开发效率. 信息发布模块的表示层主要由三级页面构成, 一级页面由企业新闻、职工之家、下载中心、政策法规等四个主要栏目构成, 每个栏目按照相关信息的发布时间倒序显示若干条标题. 二级页面显示某一个栏目的信息标题, 并具有翻页功能. 三级页面显示信息的具体内容, 并具有上一条、下一条的快速导航功能. 以下是 Struts2 标签读取新闻条目的相关代码, 其中 list\_notice 是控制层 Action 写入到值栈中的集合对象.

```
<s:iterator value="list_notice" var="info">
  <tr>
    <td width="220px" align="left" ><a
href="/huanglin_web/frontArticleAction?idType=12&id=
<s:property value="#info.id"/>" class="myt"
target="blank">
      <s:property value="#info.title"/></a>
    </td>
    <td width="70px" align="right"
      <s:property value="#info.upDate"/>
    </td>
  </tr>
</s:iterator>
```

### 3.1 表示层的实现

控制层负责系统的流程, 将表示层和业务逻辑层“关联”起来. 首先, 在 web.xml 中配置核心过滤器, Web 容器在启动时会调用 init()方法会读取配置文件-struts.xml, 将其中定义的 Action 等对象以 JavaBean 形

式存放到内存, 完成初始化操作.

然后, 定义配置文件. Struts2 一共提供了 5 个配置文件, 多数情况下只需完成 struts.xml 核心文件的配置. 配置的内容按顺序依次如下<sup>[9]</sup>: ①文件声明及头描述信息; ②系统常量配置; ③包配置; ④拦截器配置; ⑤未处理的异常配置; ⑥Action 配置. 对于企业级的应用, 建议采用分模块的配置方式, 将各模块的配置文件统一包含到 struts.xml. 这样有利于各模块控制层的并行开发, 能够避免配置冲突并提高开发效率. 最后, 定义拦截器与过滤器.

拦截器主要以 AOP 方式对 Action 增加逻辑, 并对 Action 的调用起到控制作用. 凡是系统中与业务正交的功能都可以采用拦截器实现, 本系统的权限控制和日志记录功能是通过拦截器实现的. 以日志记录功能为例, 日志的功能是在数据库中记录当前用户通过哪个 IP, 在什么时候, 执行了什么样的操作.

步骤 1. 定义一个 property 类型的配置文件, 其中的 key 值为“Action 名称.Action 方法名”, value 值为 key 的描述字符串.

步骤 2. 通过继承 AbstractInterceptor 定义日志拦截器类, 重写 Interceptor 方法. 该方法的逻辑是: 通过 ActionInvocation 对象得到 HttpServletRequest 对象和 Session 对象, 获取当前登录用户和他对应的 IP 地址, 通过读取步骤 1 的配置文件得到当前操作的 Action 信息, 最后构造一个日志 Bean 对象, 调用对应的 DAO 对象执行插入操作.

步骤 3. 在 struts-information.xml 文件中配置步骤 2 定义的拦截器以及自定义的拦截器栈. 如下代码所示:

```
<interceptors>
  <interceptor name="loggerIntercepto"
class="com.huanglin.information.interceptor.LoggerInter
ceptor">
    </interceptor>
  <!--拦截器栈 -->
  <interceptor-stack name="infodefault">
    <interceptor-ref name="defaultStack" />
    <interceptor-ref name="loginInterceptor" />
    <interceptor-ref name="loggerInterceptor" />
  </interceptor-stack>
</interceptors>
```

步骤 4. 将需要记录日志的 Action 对象引用步骤 3 定义的拦截器栈. 只需在相关 Action 配置的<result> 标签前加入<interceptor-ref name="mydefault"/>即可.

通过上述步骤可以看出, 拦截器可以不改变原有 Action 的代码而增加新的逻辑, 很好的满足了“开闭原则”, 提高了系统的可扩展性.

在 Web 项目中, 资源都是用 URL 表示的, 不同的权限对应不同的 URL. 考虑到系统的安全性, 只有授权用户才能访问对应的 URL, 本系统使用自定义的 AccessFilter 过滤器来实现. 以管理员访问的资源为例, 首先判断用户是否已经登录, 然后获取登录用户的角色, 如果是管理员角色并且请求的 URI 中包含“/admin”子串, 那么该请求就过滤, 否则就拦截. 相关核心代码如下:

```
protected boolean doFilterWork(HttpServletRequest request) {
    //判断用户是否登录
    User user=LoginUtil.getCurrentUser(request);
    if(user!=null){
        //判断登录用户的角色
        if(user.getRole().equalsIgnoreCase("/admin/"))&&request.getRequestURI().indexOf("/admin/")==0)
            return true; }
}
```

然后在 web.xml 中对 AccessFilter 过滤器进行配置, 权限过滤功能才能生效. 经过如下配置, 一个非法的 URL 请求就会跳转到警告页面.

```
<filter>
    <filter-name>AccessFilter</filter-name>
    <filter-class>com.huanglin.info.filter.AccessFilter
</filter-class>
    <init-param>
    <param-name>RedirectPage<param-name>
    <param-value>/warning.jsp</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>AccessFilter</filter-name>
    <url-pattern>/admin/*</url-pattern>
</filter-mapping>
```

### 3.2 业务逻辑层的实现

业务逻辑层为用户提供业务服务, 包括业务逻辑

和数据计算. 通过提供业务对象与 Action 组件实现该层功能. 首先, 通过继承 ActionSupport 类定义 Action, 每个 Action 组件包括两大类功能: ①定义用于接收表示层数据的属性及其对应的 set 和 get 方法, 在数据量较大的情况下, 通过实现 ModelDriver 接口的方式接收. ②调用业务对象的方法. 然后, 定义业务对象类, 它是 JavaBean 类型. 最后, 在 Action 的调用业务对象的方法后加入调用数据持久层 DAO 的逻辑, 因为数据经过业务对象处理后, 将其中需要长期保存的部分进行持久化处理.

通过上述步骤可以得知: 业务逻辑层在显示层和持久层之间增加了灵活的机制, 避免了两者的紧耦合.

### 3.3 数据持久层的实现

数据持久层为整个系统提供动态数据存取服务, 所以它的性能一直是 Java EE 开发重点考虑的问题. 在目前现有的各种持久化技术中, JDBC 具有访问速度最快、擅长多表联合查询等优点, 所以本架构的持久层使用 JDBC. 为了克服它的编写繁琐、代码冗余度高等缺点, 在综合应用 DAO、模板和策略模式的基础上对传统 JDBC 操作进行了改进, 设计了如图 5 所示的持久层类图关系.

首先, JDBCStrategy 作为策略接口封装获取连接和释放资源两个操作, 具体策略类 DBPoolStrategy 以数据库连接池方式获取连接和回收资源, 因为连接池方式能够对 Connection 进行复用, 并且在高并发访问量情况下提高系统响应速度并对服务器起到保护作用. 具体策略类 TraditionalStrategy 通过 DriverManager 类获取连接和释放资源. 前者策略用于系统整体测试和项目部署阶段, 后者策略用于持久化相关代码的单元测试. JDBCStrategyFactory 是简单工厂模式, 用于创建不同的具体策略类. 然后, JDBCTemplate 作为模板模式中的抽象类角色定义了通用的 CRUD 方法, 通过接口 RowMapper 中的 mapRow 方法在 SQL 执行结果与实体对象之间进行映射. 最后, 利用 DAO 模式解耦合业务逻辑层与持久化逻辑的代码. DAO 通常是与实体对象一一对应的, DAO 接口定义相关实体对象的持久化方法, DAOImp 实现 DAO 与 RowMapper 接口, 通过关联的 JDBCTemplate 完成真正的数据库访问. 为了提高持久层代码的移植性以及减少编译次数, 将 JDBC 驱动类名称、数据库 URL、用户名和密码、以及 DAO 中所用到的 SQL 语句等信息封装到配置文件中.

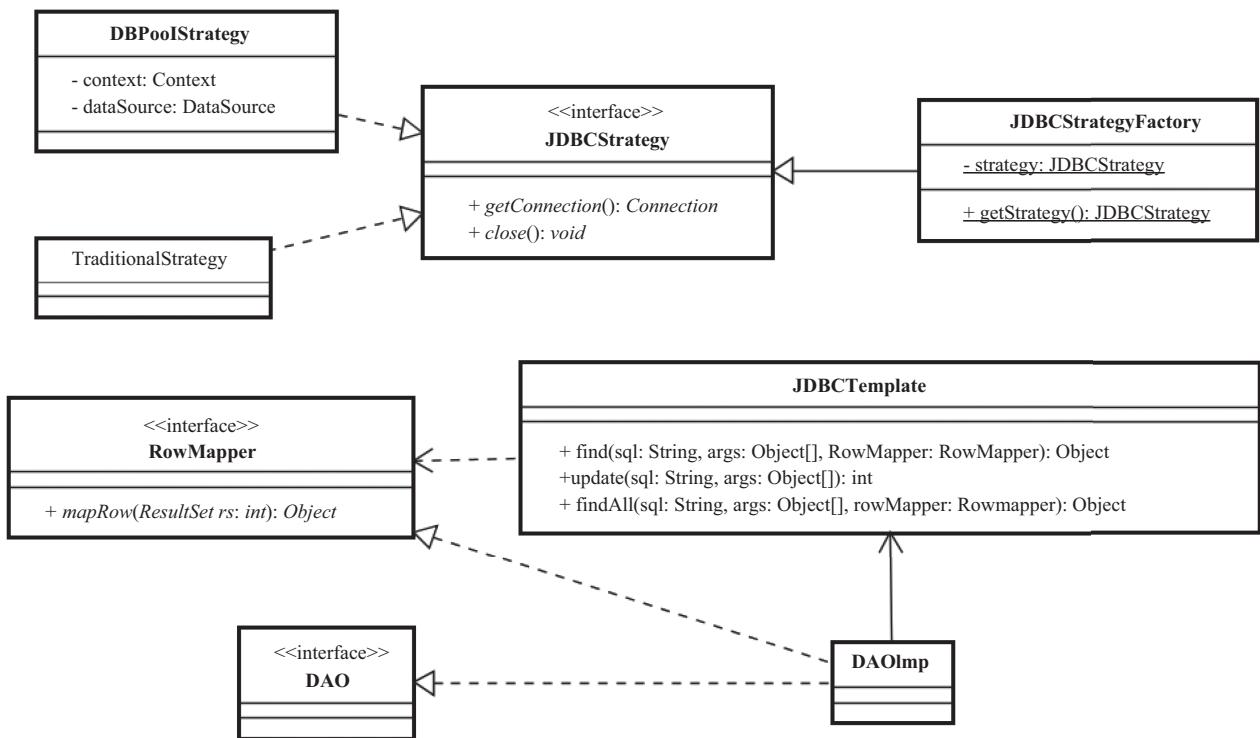


图5 持久层类图关系

通过上述设计后可以看出: 传统 JDBC 操作中的 ①②③⑤操作交由 JDBCStrategy、JDBCTemplate 和 RowMapper 完成, 开发人员只用专注于 DAO 的开发(操作④)即可. 从而降低了持久化代码的编写量, 达到了复用的目的<sup>[10]</sup>.

#### 4 结语

笔者研究了 Struts2 框架, Ajax 异步数据传输机制以及 JDBC 的核心接口, 通过整合上述 3 种技术构建了一个符合 Java EE 规范的 Web 开发架构. 其中, Struts2 框架为构建 Web 应用提供了稳定的 MVC 框架, 为视图、控制器和模型均提供了简化的实现方式; 在表示层引入 Ajax 引擎, 实现了页面异步通信, 减轻了服务器负担, 减少了冗余数据的传输; JDBC 在持久层解决方案中性能卓越, 通过 DAO、模板、策略等设计模式对 JDBC 进一步封装, 不但实现了业务逻辑层与持久层的松散耦合, 而且减少了 DAO 实现类的代码重复率. 将该架构应用到某企业的生产统计管理系统的开发中, 通过实践证明该架构有效降低了开发成本和复杂性, 提高了应用程序的开发效率, 并具有较好的扩展性和维护性.

#### 参考文献

- 1 王海涛, 贾宗璞. 基于 Struts 和 Hibernate 的 Web 应用开发. 计算机工程, 2011, 37(9): 112-114.
- 2 龚尚福, 龚琴, 冯健. 基于 SSH 框架的 EPON 性能管理系统的设计与实现. 计算机应用, 2013, 33(2): 571-574.
- 3 欧阳宏基, 葛萌. 基于 S2SH 框架的煤炭企业生产统计管理系统. 计算技术与自动化, 2015, 34(3): 118-122.
- 4 高秀慧, 高建华. 基于 J2EE 框架的 Web 应用可靠性研究. 计算机工程与设计, 2013, 34(4): 1270-1275, 1323.
- 5 甘文丽, 刘为超. 基于 Struts2 和 Ajax 的企业级 Web 应用开发. 工矿自动化, 2013, 39(2): 23-26.
- 6 李洋, 孙永维, 许冰, 等. 基于 Ajax, Struts, Hibernate 和 Spring 的 J2EE 架构. 吉林大学学报(信息科学版), 2011, 29(6): 576-583.
- 7 仰燕兰, 金晓雪, 叶桦. ASP.NET AJAX 框架研究及其在 Web 开发中的应用. 计算机应用与软件, 2011, 28(6): 195-198.
- 8 刘云玉, 段中兴, 原晋鹏. JDBC 数据库重连机制的研究与实现. 计算机应用与软件, 2011, 28(7): 38-40.
- 9 刘艳春, 洪晓慧. Struts2 框架核心配置文件的研究与应用. 计算机技术与发展, 2013, 23(2): 77-81.
- 10 欧阳宏基, 葛萌, 陈伟. 基于 JDBC 的数据持久化层性能优化研究. 网络新媒体技术, 2016, 5(5): 9-15.