

● 计算机科学

高校业务系统中基于 MyBatis 实现可复用的 DAO 模型

姜娜, 孔浩, 张子锋

(昭通学院 计算机科学系, 云南 昭通 657000)

摘要: 分层体系架构已经是高校业务系统开发中必须要考虑的部分, 在分层架构中最为重要的一层就是 DAO 层, DAO 完成了业务对象到关系数据库的转换, 常规的 JDBC 的实现将会使得 DAO 层的开发复杂并且不具备重用性. 目前 MyBatis 框架有效的封装了 JDBC 为开发人员提供方便的接口, 合理的设计 MyBatis 可以实现可复用的 DAO 层的处理模型.

关键词: 持久化层; MyBatis; 高校业务系统; J2EE

中图分类号: TP312

文献标志码: A

文章编号: 1008-9322(2012)05-0031-04

随着各所高校的校园网络的硬件条件的逐渐完善和成熟, 高校内部需要大量的软件系统来帮助行政人员进行各种管理工作, 高校的业务系统往往都有相对复杂的功能需求, 并且大量的需求会在系统运用过程中发现其需要改进之处, 所以要求高校的业务系统必须具备良好的可扩展性. 分层体系结构是实现可扩展性的基础, 基于 J2EE 的架构对分层体系提供了原生态的支持, 目前四层体系结构是常用的分层体系结构, 在四层体系结构中将整个软件项目分为表示层、服务层、持久化层和数据层, 表示层负责将数据呈现给用户, 并且为用户提供输入和输出, 服务层主要负责接收用户的请求, 以此完成系统的业务逻辑, 持久层负责对对象进行持久化操作, 数据层专门负责存储各种业务对象信息.

1 持久化层分析

在基于面向对象的软件项目中, 必须将对象持久化到相应的文件系统中, 目前最为主要的存储方式是存储到关系型的数据库中, 要将对象存储到关系型的数据库中必须进行 ORM(Object-relational mapping 对象关系映射), 由于数据层的存储数据库可能会随着需求的改变而发生变化, 为了在变化时提供更好的灵活性, 为此就指定

了持久化层专门来完成数据对象的持久化操作, 当需要改变底层数据库时, 只用调整持久化层的代码, 而不用涉及修改表示层或者服务层的代码, 如此就大大提高了系统的灵活性.

在持久化层中通常都是通过相应的 DAO (Data Access Object 数据访问对象) 完成操作的, 为每一个需要存储的业务对象创建一个独立的 DAO 访问接口和 DAO 的实现类来完成业务对象的持久化操作, 将会涉及的持久化操作主要就是增删改查四大类. 在持久化层所涉及的对象有如下三类: 每个业务对象的 DAO 接口, 实现这个接口的实现类, 一个创建具体实现类的工厂类.

2 基于 J2EE 架构的持久化层的实现技术

2.1 基于 JDBC 的实现方案

在基于 J2EE 的架构中实现持久化层的技术很多, 最为简单的就是基于 JDBC 进行实现, JDBC 是 Java 为了访问数据库而提供的一组接口, 通过调用 Connection、Statement、ResultSet 这三个接口完成对数据库的所有操作, 使用该方式的优点是: 运行的所有技术都是 Java 最基本的技术, 不需要太多的学习时间即可上手, 而且由于 JDBC 直接访问数据库, 效率相对较高; 但是缺点也非常明显: 由于对数据库的操作步骤基本都是

收稿日期: 2012-09-08

基金项目: 云南省教育厅研基金项目(2012Y437).

作者简介: 姜娜(1984—), 女, 云南曲靖人, 助教, 硕士, 主要从事计算机软件研究.

类似的,在使用 JDBC 实现时不可避免的需要编写大量相似的代码完成开发,为此大大降低了项目的灵活性,所以该方式仅仅只会在相对小型的项目中使用。

2.2 基于 Hibernate 等 ORM 框架的实现

由于 JDBC 的特性使得目前大量的公司研究自己的 ORM 框架,这些框架为用户屏蔽了 JDBC 接口的调用,让用户可以直接面向对象进行操作,目前最为流行的就是 Hibernate 框架,该框架将对数据库的操作封装得非常好,仅仅只用调用简单的 save、update、select、delete 方法就可以完成对象的存储和查询,该框架最大的优点就是简单,初学者甚至不用去学习和数据库相关的 SQL 语言就可以进行开发;但是缺点就是由于该框架使用了大量的缓存,这些缓存虽然是为了提高效率,但是如果使用不好就会出现大量性能瓶颈,特别是在获取某个对象的关联对象时性能问题更加明显。因此性能问题基本是所有该类 ORM 框架的通病。

2.3 基于 MyBatis 的实现

MyBatis 的前身是著名的 IBATIS 框架,它也算是一个持久化层框架,但是和 Hibernate 这类重量级的框架相比,MyBatis 最明显的优点就是将 JDBC 的访问接口进行了封装,在操作数据库时,会自动调用开发人员预先编写在配置文件中的 SQL 来完成,开发人员就省去了大量调用 JDBC 接口的重复性工作。由于 MyBatis 是由开发人员自己编写 SQL,这样可以灵活的运用 SQL 语句来查询不同的对象,这是 Hibernate 等重量级框架所不具备的,所以 MyBatis 的性能比重量级的 ORM 框架要高得多,这也就是目前大量开发人员选择 MyBatis 的主要原因。

3 基于 MyBatis 的可复用的 DAO 模型的设计

3.1 常用的 DAO 模型

对于 DAO 对象的操作,常规设计模型如图 1 所示,需要为每一个业务对象创建该对象的 DAO 访问接口和 DAO 的实现类,创建接口主要有两个方面的考虑,第一是确定一种规约,让所有的实现类都必须实现接口中的各个方法;第二就是为了提高可扩展性,当将来需要修改不同的数据库实现时只用增加一个实现类就可以利用多态特性

轻松改变实现类的调用。对于所有业务对象的 DAO 接口而言无非都是 add, delete, update,

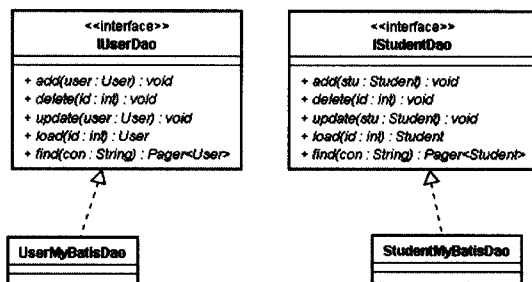


图 1 常规的 DAO 设计模型类图

load, find 等类似的方法,如果为每一个对象都添加一个 DAO 的接口和实现类在开发中需要编写大量的重复代码来实现,这样严重影响项目的灵活性和可扩展性。为此需要设计一种可复用的 DAO 模型,通过该模型来完成持久化层的设计会很大程度上减少代码的编写量。

3.2 可复用的 DAO 模型

MyBatis 框架封装了大量的基于 JDBC 的操作,让开发人员可以相对简化的完成业务对象对关系数据库的转换,开发人员仅仅只用专注于编写 SQL 语句,并且为每个 SQL 语句确定相应的命名空间和 ID,在调用时直接使用某个命名空间中 ID 可以访问到相关的 SQL,MyBatis 就会根据该 SQL 完成对数据库的调用。这种操作方式为开发人员提供了一种设计复用模型的方式,可复用

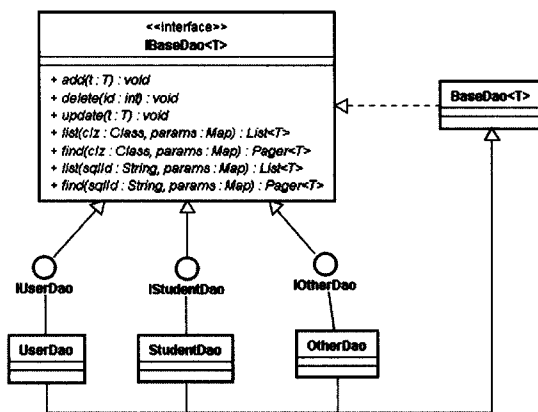


图 2 可复用的 DAO 设计模型类图

的 DAO 模型如图 2 所示,通过创建一个统一的 IBaseDao 接口,该接口中将常用的所有方法都添加进去,通过 Java 的泛型进行定义,之后为该 IBaseDao 创建相应的实现类,在该实现类中将一

些通用的增删改查方法一一实现。这时如果希望添加一个 User 的 DAO 接口,首先可以创建 IUserDao,这个接口实现类 IBaseDao,就拥有了这个接口中的所有方法,之后为 User 创建 DAO 实现类,并且让该实现类继承 BaseDao,如此就继承了 BaseDao 中所有已经实现的方法,就不用再为 UserDao 编写任何代码。同理,StudentDao 和其他 DAO 均可以采用该种方式,不管创建多少个 DAO 都可以直接使用 BaseDao 中的实现,如果某个特殊的业务对象需要 BaseDao 基本方法之外的其他方法,这样直接在该对象的 DAO 接口中创建该方法即可,此时也仅仅只用在在这个对象的 DAO 的实现类中单独实现该方法即可,这样将会很大程度上简化开发的工作量。

3.3 分页和排序的实现

在获取对象列表时必须考虑两个重要的功能:分页和排序。常用的分页功能就是为方法传递两个参数,访问的页码和每页显示的条数,之后通过相应的 SQL 语句完成数据的获取;排序功能也需要为获取列表的函数增加两个参数,排序的字段和排序的类型,之后拼接相应的 SQL 语言完成排序操作。按照以上常规的方法就需要为列表函数增加多个参数来处理,如果在 BaseDao 中进行处理的话会使得获取对象列表方法调用变得更加复杂,而且可能需要根据不同的需求创建多个获取对象列表的方法,这显然无法实现方法的复用。解决以上问题的最佳方法就是利用 Java 的 ThreadLocal 对象,在表示层通过 Filter 将分页和排序的信息存储到 ThreadLocal 中,在 BaseDao 中直接从 ThreadLocal 中获取相应的分页和排序的参数,这样就可以在不修改任何 BaseDao 的前提下完成列表对象的分页和排序处理。

3.4 DAO 工厂的设计

对于 DAO 而言,工厂是必须存在的,DAO 工厂使得代码具备更高的可扩展性,如果不使用工厂,创建 DAO 的实现类均是通过 new XxxDao() 创建的话,在将来变更实现类时需要修改所有创建了原有实现类的代码,这样完全违背了软件开发的 OCP 原则,使用 DAO 工厂可以有效的解决这个问题,DAO 工厂的设计模型如图 3 所示,首先创建一个工厂接口,在该接口中编写获取各个 DAO 业务对象的抽象方法,这些方法的返回

值都是各个业务对象 DAO 的接口,这样可以针对接口编程,当有不同的 DAO 实现时,只用创建该 DAO 工厂的实现类并且让其实现 IDAOFac-tory 然后覆盖各个 DAO 的创建方法即可。

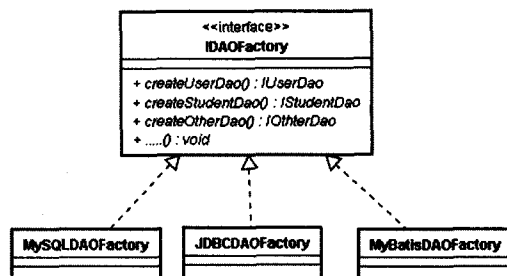


图 3 DAO 工厂的设计类图

由于使用了工厂模式,在代码中获取具体的实现类就可以直接通过工厂创建,项目中所使用的工厂实现类可以在 properties 配置文件中定义,之后通过 Java 的反射机制来读取具体使用的工厂,之后在调用到 DAO 的位置通过依赖注入的方式将具体的实现类进行注入。如果将来需要修改了工厂的实现类,只用简单的调整 properties 的配置文件就可以完成工厂实现类的替换。这将会很大程度上提高代码的灵活性和可扩展性。

4 结束语

高校业务管理系统是由多个复杂而且庞大的软件系统构成的,基于 J2EE 的分层体系架构可以方便的实现系统的设计,持久化层主要是通过 DAO 对象完成系统的业务对象到关系数据库的转换,利用 MyBatis 框架来实现持久化层即可以减少代码的工作量又可以得到较好的性能,合理设计 DAO 可复用的模型可以让开发人员相对快捷灵活的编写代码。

参考文献:

- [1]Bruce Eckel. Java 编程思想[M]. 侯捷,译. 第 2 版. 北京:机械工业出版社,2002.
- [2]Art Taylor. JDBC 数据库编程与 J2EE[M]. 李东升,译. 北京:电子工业出版社,2004.
- [3]William Crawford, Jonathan Kaplan. J2EE 设计模式[M]. 刘绍华,译. 北京:中国电力出版社,2005.
- [4]阎宏. Java 与模式[M]. 北京:电子工业出版社,2002.

[5]孔浩. 学生管理系统中 DAO 模式分析[J]. 昭通师范高等专科学校学报, 2008, (5): 42—43.

On the Implementation of the MyBatis-Based Reusable DAO Model in University's Management System

JIANG Na, KOMNG Hao, ZHANG Zi-feng

(Department of Computer Science, Zhaotong University, Zhaotong 657000, China)

Abstract: Layered Architecture has become an irreplaceable part in the development of university's management system in which one of the most important layers is DAO. DAO completes the transition from the business objects to relational database but ordinary JDBC may make the development of DAO layer more complicated and un-reusable. Now, MyBatis framework has sealed the disadvantage of JDBC and provides a more convenient interface for the developers. The reasonable and rational design of MyBatis can realize and establish the reusable DAO processing model.

Key words: sustained layer; MyBatis; University's Management System; J2EE

作者：[姜娜](#)，[孔浩](#)，[张子锋](#)
作者单位：[昭通学院计算机科学系, 云南昭通, 657000](#)
刊名：[昭通师范高等专科学校学报](#)
英文刊名：[Journal of Zhaotong Teacher's College](#)
年，卷(期)：2012, 34 (5)
被引用次数：1次

参考文献(5条)

1. [BruceEckel](#);侯捷 [Java编程思想](#) 2002
2. [ArtTaylor](#);李东升 [JDBC数据库编程与J2EE](#) 2004
3. [WilliamCrawford](#);Jonathan Kaplan;刘绍华 [J2EE 设计模式](#) 2005
4. [阎宏](#) [Java与模式](#) 2002
5. [孔浩](#) [学生管理系统中DAO模式分析](#) 2008 (05)

引证文献(1条)

1. [刘斌](#), [黄卫东](#), [肖仁辉](#) [基于远程医疗背景的服务接口集成设计](#)[期刊论文]-[福建电脑](#) 2013 (10)

引用本文格式：[姜娜](#), [孔浩](#), [张子锋](#) [高校业务系统中基于MyBatis实现可复用的DAO模型](#)[期刊论文]-[昭通师范高等专科学校学报](#) 2012 (5)