第18卷 第4期
2008年4月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.18 No.4
Apr. 2008

邮政11185业务系统持久层的Hibernate解决方案

闫智敏,王 力,杜军朝,张立勇
(西安电子科技大学 软件工程研究所,陕西 西安 710071)

摘 要:对象-关系映射就是把对象映射到关系数据库中的记录,它是完全从面向对象的角度来设计和开发程序的。文中针对基于J2EE的邮政11185系统,在比较分析了几种目前比较流行的持久化解决方案后,提出了应用Hibernate实现邮政11185系统持久层的解决方案,并利用一个实例来描述其实现过程。通过使用ORM映射框架,可以大大减少程序的代码量并大大减少出错的机会,提高了程序的开发效率。

关键词:对象-关系映射;邮政11185系统;Hibernate;持久化
中图分类号:TP399 **文献标识码**:A **文章编号**:1673-629X(2008)04-0178-04

Persistence Solution for Post 11185 System Based on Hibernate

YAN Zhi-min, WANG Li, DU Jun-zhao, ZHANG Li-yong
(*Software Engineering Institute, Xidian University, Xi'an 710071, China*)

Abstract: Object-relation mapping (ORM) handles persisting objects to and from the records of underlying relation database, it is an object-oriented method to develop application. Bring out Hibernate persistence solution after comparing and analyzing the popular solutions for data persistence in Post 11185 system, and describe the implementation of Hibernate solution by an example. The ORM framework decreases the quantity of the source code and the possibility of mistake. Therefore the developing efficiency is greatly improving.

Key words: ORM; Post 11185 system; Hibernate; persistence

0 引 言

在基于J2EE平台的邮政11185系统中,数据持久层实现对象和关系数据库之间的映射,处理关系数据库中各个表之间的错综复杂的关联关系,负责管理整个应用系统的数据操作,包括数据的大量、频繁的增加、删除、更新和检索,因此持久层的解决方案的选择便是整个11185系统性能好坏的关键所在。

1 几种持久化解决方案的比较

目前,对于持久层的实现,有着多种流行的解决方案,例如:JDBC,EJB/CMP,JDO等,Hibernate给关系数据库的开发带了新的血液,为开发者提供了一个强大而易用的O/R映射机制,将对象和存放于数据库中的数据进行映射,避免了开发人员使用大量的SQL语句直接对数据库进行操作。

下面简要介绍几种持久层技术及其相互间比较:

(1) JDBC的技术。

用JDBC进行数据库程序的开发时,多数情况下使用DAO模式,采用SQL语句进行操作。用此种方式的访问方法,可以使应用程序与具体的数据库厂商和数据库位置无关,但是,JDBC是低级别的数据库访问方式,并不支持面向对象的数据库表示。在大型的数据库应用程序中,需要的代码的维护量是非常巨大的,并且不利于程序功能的扩展[1]。

(2) EJB/CMP的技术。

在采用J2EE的应用中,EJB/CMP方式得到了广泛应用。CMP不需要将SQL语句加入到代码中。目前,更加令人注意的是,随着EJB规范的发展,CMP也包含了一些高级关系的内容。CMP方式作为一种重大改进,减少了程序员直接写SQL语句操纵数据库的麻烦,由容器自动生成访问数据库的SQL代码,不过其方案不仅造价高,整体运行速度慢,而且开发难度也较大。它们都不利于实现Web应用系统的快速开发[2]。

(3) JDO的技术。

JDO是一个存储Java对象的规范。JDO规范1.0的提出可以将精力集中在设计Java对象模型,然后在企业应用软件架构的不同层面中存储传统的Java对

收稿日期:2007-07-12

作者简介:闫智敏(1981-),女,河南新乡人,硕士研究生,研究方向为软件工程、面向对象技术;王 力,教授,硕士研究生导师,研究方向为软件工程、虚拟现实技术等。

象(Plain Old Java Objects,POJOS),采用JDOQL语言进行SQL操作。一些公司(包括Sun)企图根据JDO规范进行设计并实现JDO产品,但它们都不能很好地实现,主要原因在于其性能优化上比较差[3]。

(4) Hibernate的技术。

Hibernate是一种实现对象/关系之间映射的开源框架,对JDBC进行了轻量级的对象封装,可以实现关系型数据库和对象之间的映射,使得程序员可以使用面向对象的编程思想来操作关系数据库,使得开发数据库系统非常方便,用于Java的超高性能的对象/关系持久性和查询服务。Hibernate可以应用在任何使用JDBC的场合,既可以在Java的客户端程序实用,也可以在Servlet/JSP的Web应用中使用,并且,Hibernate可以在应用EJB的J2EE架构中取代CMP,完成数据持久化的任务[4]。

2 Hibernate框架的原理和特点

2.1 Hibernate框架的原理

Hibernate是一种开放源代码的对象关系映射框架,使开发者能够充分使用对象编程思想来操纵数据库。Hibernate的体系结构如图1所示。

整个Hibernate的体系结构可以分为3层:应用层、数据持久层和数据库层。应用层主要对业务中所涉及的对象进行操作;数据库层主要是对底层的关系数据库进行操作;业务中的对象和数据库中的表之间的关系主要是通过基于Hibernate技术的对象-关系映射技术来实现,这样,程序员就不需要关心数据库的具体操作问题了,降低了学习和开发的成本。

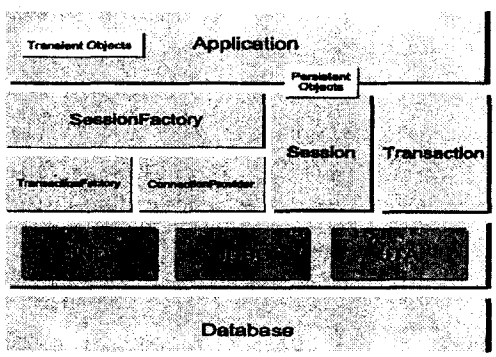

图1　Hibernate体系结构

在实际的应用系统的开发过程中,用户所关心的是其业务的具体功能,并不关心底层代码的具体实现,而实际的情况确是开发人员需要花费大量的时间和精力在编写底层数据库的访问和数据处理上。使用Hibernate的ORM技术,通过简单的设置配置文件hibernate.properties和创建xml映射,可以把数据库中的数据存储和对象建立关系,这样,开发人员只需关心对象的操作,不需要关心底层的数据库,即使数据库发生变化时,只需要简单地修改配置文件即可,并不需要修改程序的代码,大大减少了程序的代码量,也减少了出错的机会[5]。

2.2 Hibernate框架的特点

(1) 透明的持久化机制。无需在编译时手工生成持久机制代码(运行时自动生成代码)。

(2) 面向对象的查询语言,相对结构化查询语言具有更大的便利性。

(3) 高性能。基于普通Java对象和动态代理的轻量级O/R映射框架使其与其他持久化机制相比有着显著的性能优势。

(4) 简单易于操作的API接口,丰富的资源配置文档,使得应用程序的开发简单、快捷。由于它是开放源代码的,允许开发人员在需要的时候研究源代码,改写源代码,定制客户化功能。

(5) Hibernate既适用于独立的Java程序,也适用于Java Web应用,而且还可以在J2EE架构中取代CMP,完成对象持久化的重任。并且Hibernate可以和多种Web服务器、应用服务器良好集成,几乎支持所有流行的数据库服务器。

(6) 减少维护代价。由于数据库的许多复杂逻辑提到了持久层,当数据库表发生变化或数据库移植时,只需要修改持久层中的配置文件、映射文件及少量的代码,其他层无需修改。

(7) 减少代码编写。POJO可以根据Mapping文件产生,减少代码编写。

3 基于Hibernate的邮政11185系统持久层解决方案的实现

实施邮政11185系统的目的在于将复杂的邮政系统完全电子化,从而实现了整个过程的可控制性,进而提高了邮政工作人员的工作效率,方便了广大人民的生活。一个优秀的邮政11185系统应该具备以下几个特点:

(1) 高效性:邮政11185系统性能的好坏直接决定了操作人员的工作效率、用户对邮政业务的满意度,因此,高效的性能是非常重要的;

(2) 可移植性:邮政11185系统对应用服务器和各种各样的数据库都应该具备良好的可移植性,以便于系统在各个平台之间移植,或者在需要的时候进行升级;

(3) 稳定性:邮政11185系统应该具备很长时间的稳定性,尽量减少系统宕机时间;

(4) 可扩充性:随着目前国家邮政业务的不断增

强,新的业务不断出现,所以,邮政 11185 系统的功能需要不断的完善,因此要求该系统必须具备良好的可扩充性,能够方面快捷地增加新的功能模块,以便于适应市场的变化。

基于以上几点性能的考虑,在开发这套邮政 11185 系统的时候,对数据持久层采用了 Hibernate 技术。由于订单管理是邮政 11185 系统的核心功能,所以,下面就以订单管理功能中的 EMS 揽收的订单管理为例。详细叙述该项目的持久层的 Hibernate 解决方案。

3.1 数据库的建立与 Hibernate 的配置

在 Oracle 里建立名为 fz185 的数据库,Hibernate 的配置文件同时支持传统 properties 文件配置方式和 XML 格式的配置文件进行配置。由于 XML 格式的配置文件提供了更易读的结构和更强的配置能力,可以直接对映射文件加以配置并由 Hibernate 自动加载。而在 properties 文件中则无法做到这点,必须在程序中通过编码进行映射文件的加载,所以采用 XML 格式的配置文件。

配置文件的名称默认为:"hibernate. cfg. xml",Hibernate 初始化期间会自动在 classpath 中寻找这个文件,并读取其中的配置信息,为后期的数据库操作做好准备。在这个配置文件中主要是配置数据库连接的各种参数和根据项目的进展逐步添加涉及的映射文件,以供 Hibernate 管理事务、产生 SQL 和管理 JDBC 连接等。

使用 Hibernate 技术的好处在于当后台的数据库发生更改时,比如采用了 SQL SERVER、MYSQL 等数据库时,并不会导致后台应用程序的修改。而采用传统的 SQL 开发方式时,由于不同的数据库在数据库具体操作方面存在差异,从而会导致由于后台数据库的这种变更而导致程序代码的修改。Hibernate 技术的引入只需要简单对配置文件进行修改,从而降低了应用程序维护和更新的成本和代价。

在本实例中,配置文件 Hibernate. cfg. xml 的部分代码如下:

```
<? xml version="1.0" encoding="UTF-8"? >
<hibernate-configuration>
  <session-factory>
    <property name="hibernate. connection. driver_class">
    oracle. jdbc. driver. OracleDriver
    </property>
    <property name="hibernate. connection. url">
    jdbc:oracle:thin:@202. 117. 118. 120:1521:fz185
    </property>
    <property name="hibernate. connection. username">
    fzdemo
    </property>
    <property name="hibernate. connection. password">
    fzdemo
    </property>
    <property name="dialect">
    org. hibernate. dialect. OracleDialect
    </property>
    <property name="hibernate. c3p0. min_size">
    20
    </property>
    <property name="hibernate. c3p0. max_size">
    20
    </property>
    <mapping resource="service/S_ PostalPreOrder. hbm. xml"/
>
  </session-factory>
</hibernate-configuration>
```

3.2 持久化类的建立

持久化类(POJO,文件名:S_ PostalPreOrder. java)是指其实例需要被 Hibernate 持久化到数据库中的类,通常都是域模型中实体域类。持久化类符合 JavaBean 的规范,包含了一些属性,以及与之对应的 getXXX()和 setXXX()方法。getXXX()和 setXXX()方法必须符合特定的命名规则,否则,会导致 Hibernate 在运行时抛出异常。

3.3 映射文件的配置

映射文件(文件名:S_ PostalPreOrder. hbm. xml)是用来定义持久数据和在需要时保存关于对象的持久域、关联、子类的 XML 文档,映射文件的建立,使数据库表和对象类之间建立了某种映射关系。对于每个持久层对象和以名字 class name. hbm. xml 保存的文件来说,都要创建一个映射文档。在 class name. hbm. xml 中 class name 就是对象的类名。

以 EMS 揽收的订单 S_ PostalPreOrder 表和对象为例,在该文件中,定义了数据存储到哪个数据库表中,哪个字段映射到数据表中的哪个列字段,不同的对象之间如何关联。图 2 显示了 S_ PostalPreOrder. hbm. xml 配置的对象-关系映射:

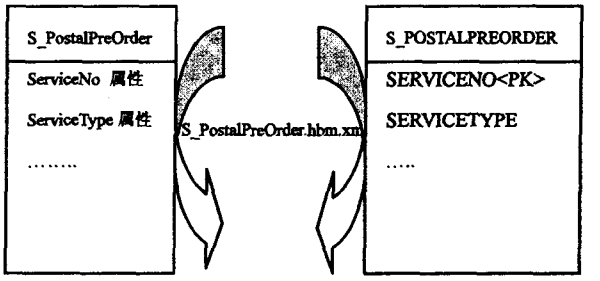

图 2 S_ PostalPreOrder. hbm. xml

映射文档在应用程序启动时编译,它可为Hibernate提供关于持久对象的相应类、它们的结构、它们应该映射到哪个数据库表格、哪个字段以及如何映射的信息。

3.4 进行DAO的定义

通过使用接口的编程方法,可以在方法的调用者和方法的具体实现之间直接建立一个屏障,即他们不存在任何的关联,这样,当方法的调用者发生改变时,不影响方法的具体实现;反之,当方法的具体实现发生改变时,不影响方法的调用。

本功能模块中的DAO为:

```
public interface S_PostalPreOrderDao extends Dao {
public Serializable save(S_PostalPreOrder s_PostalPreOrderObject)
throws DaoException;
public S_PostalPreOrder load(S_PostalPreOrderPK pk) throws
DaoException;
public void saveOrUpdate (S_PostalPreOrder s_PostalPre-
OrderObject)throws DaoException;
.....
}
```

3.5 业务逻辑层和表示层的配置

在本系统的各个功能模块中,逻辑层和表示层因为使用了Hibernate实现数据持久层而轻松了许多,对数据库的操作变成了对某个对象的操作,对于对单个对象的增加、删除、修改、更新、查找操作,基本不涉及SQL语句,而对于多个对象之间的级联操作,也可以方便地解决。

如下代码所示,简洁方便地完成了插入数据的操作,并且不用关心底层数据库的结构。

```
S_PostalPreOrder preorder=new S_PostalPreOrder ();
  preorder.setServiceNo ("9999");
  preorder.setCustomerName ("wanglin");
  S_PostalPreOrderDao preOrderDao=DaoFactory.getDaoFactory
().getS_PostalPreOrderDao();
  preOrderDao.save(preorder)
```

基于Hibernate的邮政11185系统的实现,由于使用了对象-关系映射技术,缩短了程序开发的周期,并且程序的可靠性得到了提高。

4 结束语

对象-关系映射技术是采用面向对象的方法技术来设计应用系统,通过将对象和数据库中的表之间建立的映射关系,来关注数据的实际存储,即使后台数据库发生变化,也不会导致对应用程序的修改。

文中通过研究持久层的各种解决方案,对邮政11185系统提出了基于Hibernate的持久层解决方案。在介绍Hibernate技术基础上,结合邮政11185系统的例子,具体说明了如何运用Hibernate实现系统持久层的设计。

参考文献:

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

[2] Adatia R,Arni F,Gabhart K.EJB编程指南[M].北京:电子工业出版社,2002.

[3] Tyagi S,McCammon K.JDO核心技术[M].北京:清华大学出版社,2005.

[4] 陈天河.Hibernate项目开发宝典[M].北京:电子工业出版社,2006.

[5] King G, Andersen M R, Bernard E, et al. Hibernate Core for Java[EB/OL].2007-04-02.http://www.hibernate.org/344.html.

(上接第120页)

为{a},与T的核和约简相同,可见新简化决策表定义和NSDT算法没有改变决策表表达的信息,所以也没有改变决策表的核和约简。

4 结　语

文献[5]提出的简化决策表,从减小决策表的空间入手,有效地降低了求决策表核和约简算法的空间、时间复杂度。但是由于在化简过程中对不一致问题考虑不完全,造成了丢失信息并且导致简化决策表的核和约简与决策表不同。文中分析了产生错误的原因,并且在保留简化决策表优点的前提下,修正了错误,给出了新简化决策表定义及建立算法。使用新简化决策表定义或NSDT算法,不用对原来的基于决策表或基于简化决策表的算法进行改动即可在新简化决策表下使用并得到正确结果。

参考文献:

[1] 王希雷,王　磊,马　涛,等.二进制区分的简化方法[J].微机发展,2003,13(6):101-103.

[2] 王希雷,王　磊,马　涛,等.一种高效属性约简算法[J].微机发展,2002,12(增刊):12-15.

[3] 王希雷,王　磊.粗集中区分矩阵对不一致问题处理的研究[J].微机发展,2003,13(2):119-120.

[4] 刘少辉,盛秋戬,吴　斌,等.Rough集高效算法的研究[J].计算机学报,2003,26(5):524-529.

[5] 徐章艳,刘作鹏,杨炳儒,等.一个复杂度为max(O(|C||U|),O(|C|²|U/C|))的快速属性约简算法[J].计算机学报,2006,29(3):391-399.