

J2EE 数据持久层的应用研究

方 巍,孙 涌,崔志明

(苏州大学 计算机科学与技术学院 江苏 苏州 215006)

摘 要 关系范例与对象范例之间存在“阻抗不匹配问题”。文中讨论了基于持久层的对象关系的数据库技术的设计与实现,针对 J2EE 开发过程中数据持久层的设计问题,介绍了目前比较流行的一种持久层解决方案 Hibernate 的应用开发过程及其不足之处。

关键词 J2EE;对象—关系型映射;数据持久层;Hibernate;Java 数据库对象

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2007)02-0068-04

Research and Application of J2EE's Data Persistence Layer

FANG Wei,SUN Yong,CUI Zhi-ming

(College of Computer Science & Technology,Suzhou University,Suzhou 215006,China)

Abstract :The impedance matching problem exists in the relation paradigm and the object paradigm. The several common solutions are analyzed. Aiming at the difficulties of design in data persistence layer under the J2EE development environment, this paper introduces the processes of the development of Hibernate, a popular solution for data persistence layer, as well as its defects.

Key words :J2EE;ORM;data persistence layer;Hibernate;JDO

0 引言

面向对象的 Java 和非面向对象的数据库之间差别的问题称为阻抗不匹配问题^[1,2],也就是对象范例和关系范例这两大领域之间“阻抗不匹配”。对象范例基于软件工程的一些原理,例如耦合、聚合和封装;而关系范例则基于数学原理,特别是集合论的原理。两种不同的理论基础导致各自有不同的优缺点。而且,对象范例侧重于从包含数据和行为的对象中构建应用程序,而关系范例则主要针对数据的存储。当为访问而寻找一种合适的方法时,“阻抗不匹配”就成了主要矛盾。数据持久层就是要在对象—关系数据库之间提供一个成功的企业级别的映射解决方案^[2],尽最大可能弥补这两种范例之间的差异,改善系统的可维护性和重用性,二者是面向对象设计的关键目的。

数据持久性是企业开发中最棘手的一个方面。一个企业数据持久性解决方案必须提供迅速的客户端事务,随着时间的过去确保数据完整性,以及在如系统崩

溃和网络故障之类的日常灾祸发生时使数据继续存在。文中通过持久层的技术研究来解决这个问题。

1 数据持久层简介

1.1 持久层常用表示方法

最普通的一种持久层表示方法^[3]如图 1 所示,将 SQL 语句嵌入到类中来实现持久性。它的优点是对于小型应用系统或原型时,开发人员能很快地进行编码。缺点在于关系数据库与业务类之间的直接耦合,这样只要数据库中一点小小的改动,就得重新进行编码。

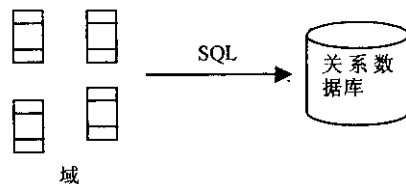


图 1 传统持久性表示

另外一种较好的轻量级持久层表示方法即 ADO (ActiveX Data Object),如图 2 所示,它将 SQL 语句和业务类封装成一个或多个数据类中。这种方法一般适合不超过 40~50 个类的小型系统中,但它仍然具有小的改动就需重新进行编码的缺点。

还有一种就是文中要详细介绍的健壮持久层表示

收稿日期:2006-05-18

作者简介:方 巍(1975-),男,安徽黄山人,博士研究生,研究方向为智能信息处理技术、分布式计算;孙 涌,副教授,博士,研究方向为现代软件工程、计算机网络与数据库;崔志明,教授,博士生导师,研究方向为智能化信息处理、计算机网络应用与数据库应用。

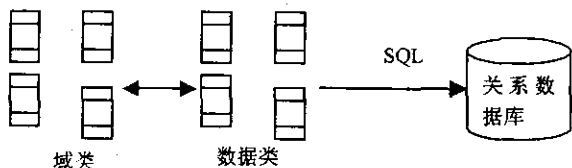


图2 ADO表示

方法,如图3所示,通过持久层来映射关系数据库,这样关系模式的改变不会影响到开发人员面向对象的代码。这种方法的优点还在于应用程序开发人员不需要了解关系数据库模式的细节,适合于大型系统的开发。

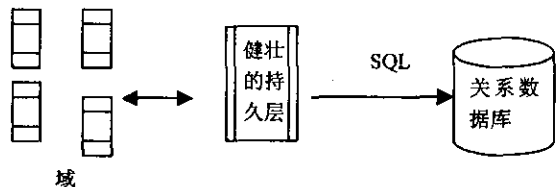


图3 健壮持久层表示

1.2 J2EE 持久层介绍

一个典型的J2EE的应用,至少包括以下3部分:表现层、业务逻辑层和数据持久层^[4]。为了更加容易地创建企业应用程序,许许多多的Framework涌现出来,表现层可以选择 Struts、JSF、Tapestry、WebWork、Velocity等,数据持久层可以选择原始的JDBC、ORMapping tools(Hibernate、toplink等)、SQLMapper tool(Ibatis)、JDO、EJB(Entity Bean)等,业务逻辑层可以用普通的Java Bean,也可以用Spring、EJB(Session Bean)。每种技术都有它的优缺点,各有自己的适用范围。例如EJB可以很好地进行分布式处理和Object Cache等,但EJB的运行需要EJB容器,开发调试起来很不方便,特别在需求不确定性很大、模型不稳定的情况下,实在是一种重量级别的开发;而Java Bean则是一种很轻量级的方式,开发调试容易,但又很难实现分布式处理。

所以,在构建企业应用时,应该有个好的技术框架,应考虑到各种主流的实现技术,既可以根据实际情况进行取舍,同时在从一种实现方式变更为另一种实现方式时,又可以进行平滑过渡,让多种技术实现并存,发挥最大技术优势,降低项目成本,提高开发效率。

2 J2EE 中持久层常用解决方案

2.1 实体 Bean

实体 Bean 提供健壮的数据持久性。Bean 容器处理大部分的数据完整性、资源管理和并发性功能,从而使开发人员关注业务逻辑和数据处理,而不是这些低级细节。使用 Bean 管理的持久性(Bean Managed Persistence, BMP)实体 Bean 时,开发人员编写持久性代码

而容器确定何时执行该代码。使用容器管理的持久性(Container Managed Persistence, CMP)实体 Bean 时,容器生成持久性代码并管理持久性逻辑^[5]。

2.2 JDBC

当与会话 Bean 结合时,它可提供简便的 EJB 开发和与平台无关的部署,而没有象 EJB 技术那样的资源使用和内存开销。象 BMP 实体 Bean 一样,该解决方案要求开发人员编写持久性代码。与 BMP Bean 不同的是,它还要求开发人员编写持久性逻辑。因而,开发人员负责确定何时将数据持久保留在数据存储中以及何时从数据存储装入数据^[5]。

2.3 JDO

Java 数据库对象(Java Database Object, JDO)是使 Java 对象具有持久性的标准,本质并不是对象关系映射技术,而是不考虑持久性机制,通过指定 API 接口管理对象的持久性,它可以是基于文件的系统或者是真正的对象数据库等任何对象。JDO 的优点在于 Java 开发人员不必担心这些细节,JDO 会提供透明的持久性。通过 JDO,可以在对象模型中通过标准对象设计技术,包括继承和组合技术来设计任何对象,包括持久性类。

2.4 O-R Mapping

O-R Mapping 具有自我存储到关系数据库的能力,对对象的改变能够直接得以存储,而不考虑数据库存取代码。这样,把全部精力集中到对对象和类进行编程,解决业务问题。在整个系统中除了这一个层次,没有一句数据库存取代码。其中,Hibernate 作为 O-R Mapping 中最好的开源工具,受到数量众多的程序员的拥护。

2.5 几种技术对比

在比较任何两种技术时,重要的是不仅要考虑它们的差别,还要考虑它们的相似点。作为 J2EE 技术,实体 Bean 和 Java 数据对象有几个共同的特性。

1)传统的架构:

Session Bean <-> Entity Bean <-> DB

2)为了解决性能障碍的替代架构:

Session Bean <-> DAO <-> JDBC <-> DB

3)使用 Hibernate 来提高上面架构的开发效率的架构:

Session Bean <-> DAO <-> Hibernate <-> DB

就上面3个架构对以下内容进行分析:

(1)内存消耗:采用 JDBC 的架构2无疑是最省内存的,Hibernate 的架构次之,EJB 的架构最差。

(2)运行效率:如果 JDBC 的代码写的非常优化,

那么 JDBC 架构运行效率最高,但是实际项目中,这一点几乎做不到,这需要程序员非常精通 JDBC,运用 Batch 语句,调整 Prepared Statement 的 Batch Size 和 Fetch Size 等参数,以及在必要的情况下采用结果集 cache 等等。而一般情况下程序员是做不到这一点的。因此 Hibernate 架构表现出最快的运行效率。EJB 的架构效率会差的很远。

(3)开发效率:在有 JBuilder 的支持以及简单的项目,EJB 架构开发效率最高,JDBC 次之,Hibernate 最差。但是在大的项目,特别是持久层关系映射很复杂的情况下,Hibernate 效率高的惊人,JDBC 次之,而 EJB 架构很可能会失败。

(4)在分布式、安全检查、集群、负载均衡的支持等方面,由于有 SessionBean 做为 Facade,3 个架构没有区别。

3 应用开发实例

3.1 Hibernate 简介

对象、关系的映射(ORM)是一种耗时的工作,在 Java 环境下,有几种框架表示持久数据,如实体 Bean,JDO 和 Hibernate 等。Hibernate 从本质上来讲是一种轻量级“对象-关系型数据映射”(Object Relational Mapping,简称 ORM)^[6]工具。它不仅提供了从 Java 类到数据表的映射,也提供了数据查询和恢复等机制。Hibernate 推动了基于普通 Java 对象模型,用于映射底层数据结构的持久对象的开发。EJB3.0 中也引入了 Hibernate 的思想。

对于 J2EE 的 3 层结构来说,Hibernate 主要能够协助完成第 3 层也就是持久层(persistence layer)的功能。图 4 是一张 Hibernate 的体系结构图,其中 Persistent Object 是简单的业务实体对象(要被持久化的对象)。通过 Hibernate 被透明地持久化到数据库中,从而减少了繁琐而且容易出错的 JDBC 的操作。下面通过实例来研究 Hibernate 持久层的表现形式。

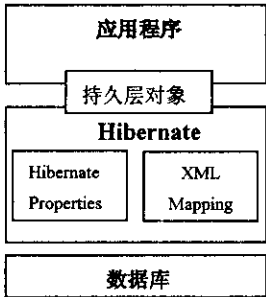


图 4 Hibernate 体系结构

3.2 整合 Struts 与 Hibernate 的开发实例

(1)建立关系数据库中备件表 Emos-Specialty 与

类 Specialty 的映射文件,名字为 Specialty.hbm.xml。

```
.....
<hibernate-mapping>
    <class name="Specialty" table="EMOS_SPECIALTY">
        <id name="ID" column="SpecialtyID">
            <generator class="increment"/>
        </id>
        <property name="PartName" column="PARTNAME"/>
        <property name="InUnit" column="INUNIT"/>
    </class>
</hibernate-mapping>
```

(2)使用 Hibernate 提供的工具来生成相应的 Javabeans 类 Specialty.java 和建表的 SQL 语句。用 Hibernate 对数据对象进行持久化后,然后通过创建系统 SessionFactory(通常只创建一次)操纵数据,然后实例一个 Session,通常直接对数据进行操作都是通过 Session 来完成。采用 hibernate.cfg.xml 来创建 SessionFactory 以及数据库连接。确定了创建方法以后就可以应用了,因 SessionFactory 通常只创建一次,所以推荐在系统初始化时建立,那么可以利用 Struts 的 plugin 机制来完成这一任务。

(3)整合 Struts,运行实例。由于访问通过使用 Hibernate 有一些共性的操作,在这里把这些共性的操作封装在一个专门的类中,这样其它的类就可以继承它:

```
Configuration ds = new Configuration().configure();
SessionFactory sessionFactory = ds.buildSessionFactory();//获得 SessionFactory
Session session = sessionFactory.openSession();//使用 SessionFactory 建立 session
private String configFilePath = "/hibernate.cfg.xml"//使用 session 存储
private SessionFactory factory = null;
static String contextFactory = "com.sun.jndi.rmi.registry.RegistryContextFactory";
Specialty specialty = new Specialty();
specialty.set(...);
Transaction tx = null;
try {
    tx = session.beginTransaction();
    session.saveOrUpdate(obj);
    session.flush();
    tx.commit();
} catch (Exception e) {
    try {
        if (tx != null) tx.rollback();
    } catch (Exception ex) {
```

```
.....}  
}
```

在创建 SessionFactory 后 ,就可以通过 JNDI 来获得 :

```
SessionFactory sf =( SessionFactory ) inttex. lookup(" hibernate-  
connection_ factory ");
```

3.3 Hibernate 的不足

目前 Hibernate 只支持对关系数据库表中数据的操作和对 sequence 的有限支持 ,其他的数据库对象全部都不支持。比如说 view ,store procedure ,function ,struct 等 ,这些数据库的高级功能都不支持。另外它也不支持 DDL ,比如说 create table ,drop table ,truncate table ,alter table add index 等 ,它只支持 DML。由于 Hibernate 只是封装了对表的操作 ,因此它只支持 select ,insert update delete。不过值得一提的是 select 中几乎所有的语法 Hibernate 都支持 ,包含子查询、连接查询、统计函数等。此外 Hibernate 映射复杂的表关联做得很出色。但是如果你要使用数据库 view ,调用 store procedur(存储过程) ,这些不是表的数据库对象 ,只能直接写 JDBC 代码^[2]。

4 小 结

文中针对两种范例的“ 阻抗不匹配问题 ” ,深入分析了数据持久层的设计困难的原因 ,通过 J2EE 几种数据映射方案的分析和比较 ,以及在项目开发过程中的使用特点 ,介绍了目前在 O - R Mapping 领域中最

受欢迎的开源项目 Hibernate ,阐明了采用 Hibernate 开发架构的技术优势。利用 Hibernate 的对象持久化服务 ,可以有效地进行数据库到业务对象的 O/R 映射 ,满足不同系统中的对象持久化要求。可以说 ,Hibernate 技术有着广泛的应用前景 ,现在已经在 Spring 轻量级容器构架中起到了不可动摇的作用。现在开发的工作流系统中 ,Hibernate 同样承担了 ORM 映射的重任。在对象数据库技术还不成熟的今天 ,Hibernate 将有可能取代 CMP ,帮助 Spring 撑起未来服务器构架的一片新天地。

参考文献 :

[1] Gallardo D J. Java Oracle 数据库开发指南[M]. 北京 :清华大学出版社 2003.
[2] 田 珂. J2EE 数据持久层的解决方案[J]. 计算机工程 , 2003 29(22) 93 - 95.
[3] King G. The Design of a Robust persistence Layer For Relational Databases[EB/OL]. 2004. [http //www. ambysoft. com/mappingObjects. html](http://www. ambysoft. com/mappingObjects. html).
[4] 安立学. 运用 J2EE 创建灵活易扩展的企业应用程序探讨 [EB/OL]. 2005. [http //gceclub. sun. com. cn/yuanchuang/ week - 15/minisoa. html# about](http://gceclub. sun. com. cn/yuanchuang/ week - 15/minisoa. html# about).
[5] Gabbar K. Java Data Objects and EJB[EB/OL]. 2005. [http //www - 900. ibm. com/developerWorks/cn/java/j - pj2ee4/index..shtml](http://www - 900. ibm. com/developerWorks/cn/java/j - pj2ee4/index..shtml).
[6] King G. Hibernate2 Reference Documentation[EB/OL]. 2004. [http //prdownloads. sourceforge. net/hibernate](http://prdownloads. sourceforge. net/hibernate).

(上接第 67 页)

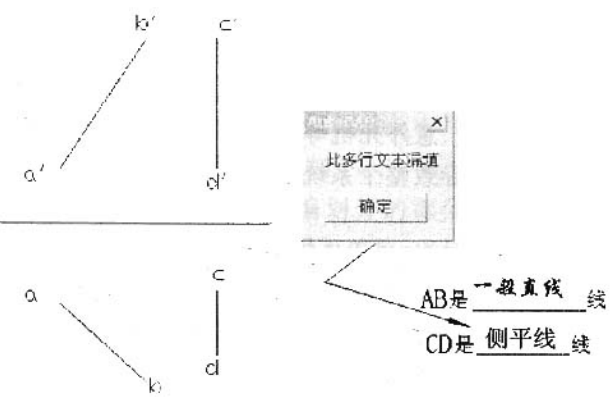


图 4 文本漏填

影和换面法) ,这就要求线的端点标注要一一对应 ,且字符要在端点处一定范围内 ,第三种是选择题 ,即将正确的选项填到规定的空圈内。这里对各种题型中文本判别可能出现的“ 漏填 ”、“ 错填 ”、“ 多填 ”、“ 位置不对 ” 等情况作了深入的研究和程序实现 ,有助于试题库的

完善 ,并可以广泛应用于基于 AutoCAD 的工程图学的作业和试卷的智能化评阅中。

参考文献 :

[1] 张晋西. Visual Basic 与 AutoCAD 二次开发[M]. 北京 :清华大学出版社 2002.
[2] 谭浩强. Visual Basic 程序设计[M]. 北京 :清华大学出版社 2001.
[3] 张 强 ,张园林. 基于 AutoCAD 的工程制图主观试题的自动判卷技术[J]. 微机发展 2004 ,14(12) :142 - 144.
[4] 翼 利 ,胡青泥. 多媒体教学习题部分智能批改的研究 [J]. 计算机与现代化 2004 ,1X 4) 99 - 101.
[5] 李绍彬. AutoCAD2000 基于 VBA 的二次开发[J]. 重庆师范学院学报 2003 20(1) 35 - 38.
[6] 张应中 ,罗晓芳 ,胡宜鸣 ,等. 工程制图智能电子习题集软件系统的设计与实现[J]. 工程图学学报 ,2004 25(1) :104 - 108.