

# 持久层实现研究

徐晓辉, 贾倩, 刘夏, 陈昂

(武汉大学 计算机学院, 湖北 武汉 430079)

**摘要:**开发一个同时使用面向对象技术和关系型数据库技术的项目是一件困难的事情。为了解决面向对象技术和非面向对象技术之间存在阻抗不匹配问题,在企业应用开发过程中需要使用持久层来减少不匹配带来的影响。持久层是根据分层思想的指导,通过建立逻辑门户,采取一定的对象-关系映射策略,隐藏所有数据库访问代码细节,向开发人员提供的透明的对象持久化机制。在对现今流行的持久层设计技术、DAO模式和CMP模式的实现,进行深入分析的基础上,讨论了一个良好的持久层所应具备的功能,提出了一个持久层实现模型。

**关键词:**对象持久化;持久层;对象-关系映射;数据对象

**中图分类号:**TP311.5

**文献标识码:**A

**文章编号:**1673-629X(2008)03-0092-04

## Research on Persistence Layer Model

XU Xiao-hui, JIA Qian, LIU Xia, CHEN Ang

(School of Computer, Wuhan University, Wuhan 430079, China)

**Abstract:** Designing software to connect an object-oriented business system with a relational database is a tedious task. In order to resolve inharmonious problem of impedance between object-oriented business system and non-object-oriented business system, the persistence layer is used to reduce effects that inharmonious impedance brings about while enterprises apply and exploit the system. The persistence layer is a transparent object-enduring-mechanism system provided to the persons who develop according to the directions of the thoughts with layers, by building up a logic door, by adopting the certain object-relation mapping tactics to conceal all code details which all data bases access. With the analysis of popular technology, the realization of DAO pattern and CMP pattern, discusses the necessary function of a well-designed persistence layer. At last, describe a realization model of persistence layer.

**Key words:** persistent objects; persistence layer; object/relation mapping; data object

### 1 持久层简介

数据是一个应用系统的核心,应用系统的所有功能的运行都依赖于数据的支持。对企业级的应用来说,数据不仅可访问,还必须持久(即数据不因系统重新启动、系统崩溃等事件而消失)。因此企业级的应用系统大多使用数据库技术来持久存储应用系统的数据,持久化则是应用系统中运行数据和数据库中的数据之间的交互,主要包括数据的查询、添加、修改、删除等操作。

当前主流的面向对象开发技术和关系数据库之间存在着阻抗不匹配(impedance mismatch),对象不能自然地与关系数据库中的数据交互。在实际应用系统的开发过程中开发人员需要花费大量精力来完成对象的持久化工作。如果应用系统的对象持久化工作是由开

发人员而不是设计人员完成,那么开发人员必须考虑诸如连接管理、数据访问效率、多线程并发、事务处理、不同系统之间的互操作问题,而且在应用系统的生命周期中,持久存储的类型有可能会改变,业务模型也有可能发生变更。为了解决以上问题,设计人员提出了持久层概念<sup>[1]</sup>,目的是隐藏持久化细节,让开发人员专注于问题域而不是对象持久化。

持久层是根据分层思想的指导,通过建立逻辑门户,采取一定的对象-关系映射策略,隐藏所有数据库访问代码细节,向开发人员提供的透明的对象持久化机制。

在设计中使用持久层可以为项目的开发带来如下好处:

- \* 分开数据持久逻辑与业务逻辑,降低了它们之间的耦合。

- \* 通过对象-关系映射向业务逻辑提供了面向对象的数据访问。

- \* 抽象数据库中存储数据的物理细节和数据库

收稿日期:2007-06-16

作者简介:徐晓辉(1977-),男,湖北咸宁人,硕士,研究方向为数据库系统的研究和开发、计算机网络、信息安全等。



\* 扩展性。对业务逻辑屏蔽持久机制的变更。

\* 运行时配置。

### 3.2 持久层设计框架

持久层的职责是实现对业务对象的透明存取。对于应用系统开发人员来说,他们只需要使用数据对象的相关方法,持久层便可以完成数据对象的持久化工作。图 3 是文中提出的持久层设计框架。

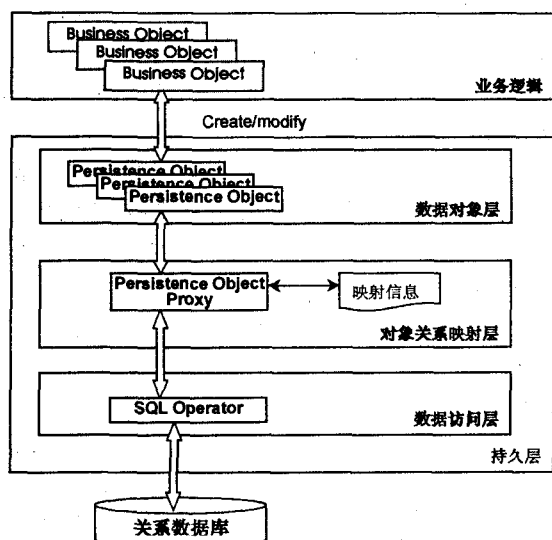


图 3 持久层框架

文中提出的持久层设计框架中的持久层由任务不同的三层组成:

\* 数据对象层。此层由数据对象组成,数据对象保存了应用系统的所有数据信息,并提供统一的访问接口供应用逻辑使用。

\* 对象-关系映射层。此层定义了数据对象层中的对象与关系数据库之间的映射,包括对象和表或视图之间的映射、对象属性与表或视图字段之间的映射、数据库连接管理、SQL 代码描述、字段类型与对象属性类型转换、事务管理等。

\* 数据访问层。根据映射信息完成数据对象的具体持久化工作:执行 SQL 语句,完成相关操作。

数据对象层、对象-关系映射层和数据访问层三者密不可分、有机地结合起来共同完成对象持久任务。应用逻辑的开发人员只能访问数据对象层,所有的数据访问操作都是简单的、透明的、面向对象的。应用逻辑调用数据对象相关接口以后,数据对象并不直接与数据库交互,而是将相应操作转移给对象-关系映射层的 PersistenceObjectProxy, PersistenceObjectProxy 根据映射信息生成相应操作的 SQL 语句和有关参数,最终数据访问层的 SQLOperator 添加参数,执行 SQL 语句,并将返回值转化为数据对象返回。

下面是三层的详细分析以及相关实现。

#### 3.2.1 数据对象层

数据对象保存了应用系统中的各种需要持久的信息。持久层任务便是完成这些数据对象的持久化工作,并向应用系统开发人员提供统一的接口。通过数据对象层提供的访问接口,应用开发人员拥有了对应用系统数据真正的透明存取的能力。根据持久层功能的要求,这一层的接口需要提供持久化对象的所有操作,包括单个数据对象的 CRUD、查询,基于条件的多对象操作等。

在图 4 所示的模型中, PersistenceObject 类是所有数据对象中的父类,它定义了所有需要持久的数据对象共同的方法。Student、Teacher、Course 是一个简单的应用系统中的三个数据对象(在对象-关系映射策略<sup>[4~6]</sup>上使用单表模式,即一个数据对象类与一个关系数据库的表或者视图对应)。现在,开发人员可以通过 PersistenceObject 提供的接口来新建一个学生的信息,查询指定职工号的员工信息,删除某一课程。对于学生和课程之间的多对多的关系则采用 Association Table 模式<sup>[3]</sup>,利用各个数据对象的主键建立新的关系表和与关系表对应的关系类,然后把对关系类的操作转嫁到主键对应的数据对象的相应的操作上去。

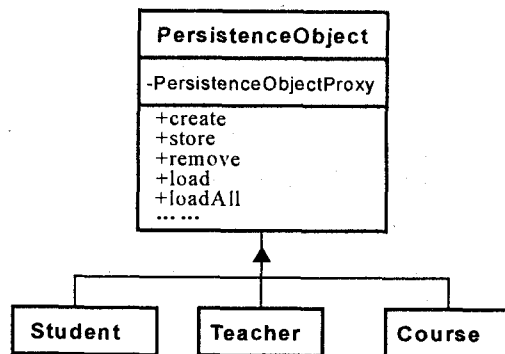


图 4 一种模型

根据降低各层之间耦合的设计原则,开发人员只操作持久层的数据对象层。开发人员需要的是透明的持久化,即调用了对数据对象的相应操作之后,数据对象的信息立即保存在关系数据库中或者立即从关系数据库中获取。至于持久层如何完成,开发人员并不关心。

#### 3.2.2 对象-关系映射层

对象-关系映射层是持久层的核心。将对象映射到关系型数据库,最终需要将对象值映射到数据库表中。映射层需要实现多种映射,包括数据对象与数据库表的映射、对象属性与数据库表字段之间的映射。映射层实现的重点在于从各种映射策略中选择合适的映射策略。除了上面所述的映射以外还有数据库连接

