

# ORM 技术在机场信息系统中的应用

沈洋

(南京航空航天大学经济与管理学院, 南京 210016)

**摘要:** 在描述对象关系映射(ORM)技术发展的起因及其主要优点的基础上, 介绍了基于 NHibernate 的 ORM 技术架构, 讨论了在机场信息系统的开发过程中全面使用 ORM 技术的方法, 分析了使用 NHibernate 实现 ORM 过程中影响效率的一些关键问题。

**关键词:** 对象关系映射; NHibernate; 机场信息系统

## Aoplication of ORM Technique in Airport Information System

SHEN Yang

(College of Economics & Management, Nanjing University of Aeronautics & Astronautics, Nanjing 210016)

**【Abstract】** Based on the discussion about advantage of object-relation mapping(ORM) this paper introduces the architecture of ORB based on NHibernate, gives the procedure of using NHibernate in the information system of airport, and analyzes several key factors which influence the efficiency of using ORM.

**【Key words】** object-relation mapping(ORB); NHibernate; airport information system

20 世纪 90 年代以来, 面向对象的分析设计技术逐渐成为系统开发方法的主流, 尤其是 G. Booch 等人于 1997 年提出的统一建模语言(unified modeling language, UML)标准<sup>[1]</sup>, 极大地促进了面向对象技术的发展。与此同时, 虽然对象数据库产品已经出现, 但由于技术成熟程度和商业推广等因素的影响, 在应用领域关系数据库仍是大多数系统的选择。随着对面向对象方法研究的深入, 人们逐渐意识到用面向对象方法设计的系统与按关系类型存放的数据库之间存在着一定的矛盾。对象关系映射(object-relation mapping, ORM)技术将对对象模型中的数据表示映射为相应的关系数据模型, 从而解决了这一矛盾。应用 ORM 技术的主要优点是: (1)减轻了开发人员编写数据访问代码的工作量, 使得他们可将更多的注意力放在商业逻辑上; (2)降低了系统对数据库产品的依赖, 有效地提高了系统的可移植性。

近年来出现了一些对 ORM 进行研究的文献: 文献[2]从概念上解释了使用 ORM 技术进行软件开发的原理; 文献[3~4]介绍了 Hibernate 在软件开发中过程的应用方法; 文献[5]中对如何避免在使用 Hibernate 装载对象时导入冗余数据的问题进行了讨论。但目前对在大型复杂信息系统中全面使用 ORM 技术和对更深层次的 ORM 技术应用的报道尚不多见。本文结合民航机场信息系统的开发实际, 讨论了基于 NHibernate 的 ORM 解决方案, 并对 ORM 应用过程中的一些关键问题进行了分析。

### 1 NHibernate 的体系结构

NHibernate 是 JBoss 公司支持下发展的一个开源 ORM 产品, 它的应用环境为 Microsoft Windows 下的 VS.NET 开发平台, 通过 NHibernate 不仅可以方便地完成从 .NET 类到各种关系数据库的映射, 而且还提供了灵活的查询机制, 可大幅度缩减开发人员处理 SQL 和 ADO.NET 所花费的时间<sup>[6]</sup>。图 1 描述了 NHibernate 的总体结构。

由图 1 可见, NHibernate 由 ORM 组件、数据库配置文

件和 XML 类映射文件组成。ORM 组件中封装了一系列用于进行对象关系映射的类, 它可由开发人员在程序中进行调用; 通过对数据库配置文件的设置可实现与不同的数据库产品进行交互; 在 XML 类映射文件中则定义了 .NET 类属性与数据库字段的映射关系。

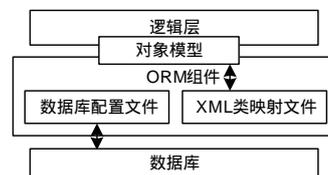


图 1 NHibernate 的总体结构

使用 NHibernate 实现 ORM 的步骤可概括如下: (1)创建反映问题领域数据模型的关系模式, 即设计出合理的保存持久化类属性信息的数据库表; (2)设计需持久化的 .NET 类; (3)对每个 .NET 类创建一个 XML 映射文件; (4)设计数据库配置文件, 以便 NHibernate 可选择适当的驱动程序和 Dialect 与相应的数据库产品进行连接; (5)在应用程序中使用 NHibernate 提供的 ORM API 对对象模型进行持久化操作。

## 2 NHibernate 在机场信息系统中的应用

### 2.1 机场信息系统的技术架构

在国内某大型机场的信息系统建设过程中, 笔者全面使用了 NHibernate 以实现对象模型与关系数据库的连接。该信息系统的技术架构如图 2 所示。

由图中可见, 该系统是一个基于 WebService 技术的多层结构系统, 以对象模型为核心的逻辑层通过 ORM 技术读写数据库中的信息, 客户端通过 WebService 与逻辑层进行数据交换。

**作者简介:** 沈洋(1973 - ), 男, 博士, 主研方向: 复杂系统, 最优化算法, 数据挖掘与知识管理, 管理信息系统

**收稿日期:** 2006-10-13 **E-mail:** shen.y@163.com

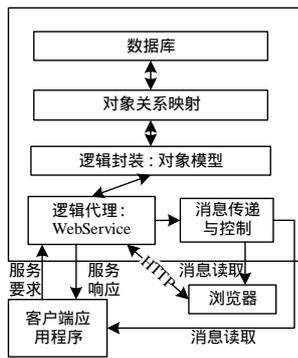


图2 机场信息系统的技术架构

## 2.2 航班类简介

在机场信息系统的对象模型中，最核心的是航班类 Flight，该类的主要属性见图3。其中，航班号、预计时间、预报时间、实际时间和运作日期5个属性为基本数据类型；航班属性、航班类型、航线、飞机、停机位、登机门和行李转盘7个属性为对象类型；而值机柜台属性较为特殊，其数据类型为列表(即一个航班可占用多个值机柜台)。

Flight	
._ HBH: String	//航班号
._ HBSX: HBSX	//航班属性
._ HBLX: HBLX	//航班类型
._ HKHX: HKHX	//航线
._ FEJI: FEJI_INFOR	//飞机
._ YJSJ: DateTime	//预计时间
._ YBSJ: DateTime	//预报时间
._ SJSJ: DateTime	//实际时间
._ YZRQ: DateTime	//运作日期
._ JW: JW_INFOR	//停机位
._ DJM: DJM_INFOR	//登机门
._ XLZP: XL_TQZP	//行李转盘
._ ZJGT: IList	//值机柜台列表

图3 航班类的主要属性

## 2.3 航班类的 XML 映射文件

如第1节所述，需要为每一个设计完成的.NET类建立一个XML映射文件，以便将类属性与数据库的关系模式对应起来，Flight类的映射文件内容如下：

```
<hibernate-mapping>
<class name="HAOS.Flight, MeiLanService" table="FLIGHT">
  <id name="ID" column="ID">
    <generator class="assigned" />
  </id>
  <property name="HBH" column="HBH" />
  ...
  <many-to-one name="FEJI" column="JH" class="HAOS.FEJI_INFOR,MeiLanService"/>
  ...
  <bag name="ZJGT">
    <key column="ID" />
    <one-to-many class="HAOS.ZJGTFP, MeiLanService"/>
  </bag>
  ...
</class>
</hibernate-mapping>
```

针对不同性质的属性应采用不同的映射方法：对于航班号等基本数据类型的属性可直接进行映射；对于对象类型的属性，应根据类之间的对应关系(一对一、一对多和多对多)进行不同的映射处理，在Flight类中，航班与飞机的关系为多对一，航班与值机柜台则是一对多的关系。

## 2.4 数据库配置文件

每一个使用NHibernate进行ORM操作的应用程序都必须有一个数据库配置文件，在该文件中需对与数据库连接的各种参数进行设置。本系统所使用配置文件的部分内容如下：

```
...
<property name="connection.provider">
NHibernate.Connection.DriverConnectionProvider</property>
<property name="connection.driver_class">
NHibernate.Driver.OracleClientDriver</property>
<property name="connection.connection_string">
Data Source=AODB;User ID=root;Password=root;
</property>
<property name="dialect">
NHibernate.Dialect.Oracle9Dialect</property>
</property>
<property name="show_sql">>false</property>
<mapping assembly="MeiLanService" />
...
```

在该配置文件中，设置了与名为AODB的Oracle9i数据库的连接，并且指定与该配置文件对应的XML映射文件被包含在程序集MeiLanService中。

## 2.5 使用ORM API进行对象持久化操作

在完成上述工作后，可以方便地使用NHibernate提供的ORM API进行对象的持久化操作。在进行持久化操作时，应首先通过以下语句生成应用程序与NHibernate的接口：

```
private static
ISessionFactory m_NHibernateSessionFactory
= new Configuration().AddAssembly("MeiLanService")
.BuildSessionFactory();
```

然后就可以使用该接口在Session中进行对象的持久化操作，包括装载、删除、改写和查询等。下面给出一个查询的示例代码：

```
ISession session=
m_NHibernateSessionFactory.OpenSession();
query=session.CreateQuery("from HAOS.Flight as f where
(f.YZRQ = :yzrq).SetDateTime("yzrq",System.DateTime.Today);
ArrayList flights = (ArrayList)query.List();
session.Close();
```

通过这样简单的代码就可以将当日的航班装载到航班对象数组中，而不必像传统方法那样对数组中每个元素的每个属性值逐一赋值。

## 3 ORM效率分析

与传统方法相比，使用ORM技术进行对象持久化操作有着明显的优势，但在系统开发，尤其是拥有众多客户端三层结构的大型信息系统开发过程中，随意地使用ORM技术也会带来严重的效率问题。本文结合机场信息系统开发实际，对其中的一些关键问题进行了总结：

(1)应将会话工厂SessionFactory设计为全局的和静态的，否则每一次对ORM的调用都将创建一个新的数据库连接，从而导致内存占用不断增长，而这种内存的增长是.NET CLR无法自动回收的。

(2)在进行一些查询时，常出现需将对象的部分属性进行装载的情况(类似于只查询某个表的部分字段)，而NHibernate只能实现对象的整体装载，这就会产生数据冗余。对于该问题，本文认为可对被多次部分查询的属性进行分类组合，把它们抽象为原类的超类以实现新的映射关系。另外，也可以

(下转第248页)