文章编号：1007-757X(2013)9-0012-03

数据库开发框架 NHibernate 应用研究

崔玉连，杨新锋

摘 要：NHibernate 是.NET 平台的一个对象持久化工具，它所提供的对象-关系映射（ORM）机制能够很好地解决对象与关系模型不匹配的问题。介绍了 NHibernate 的特性，分析了 NHibernate 的体系结构和映射机制，结合全程办税管理系统对 NHibernate 的实际应用进行了研究，结果表明利用该工具能够提高系统的扩展性、可维护性和应用系统的开发效率。

关键词：NHibernate；对象关系映射；数据持久化

中图分类号：TP311 **文献标志码**：A

Research on Application of Database Development Framework NHibernate

Cui Yulian, Yang Xinfeng

（School of Computer and Information Engineering,Nanyang Institute of Technology, Nanyang 473004,China）

Abstract: NHibernate is an object persistence tool of .NET platform, the object relational mapping (ORM) mechanism it provided can solve the problem of object and relational model matching. This paper introduces the characteristics of NHibernate, Analysis the architecture and the mapping mechanism of NHibernate, combined with the full tax management system on NHibernate practical application were studied, the results show that using the tool can improve the system scalability, maintainability and development efficiency of application system.

Keywords: NHibernate; Object Relational Mapping; Data Persistence

0 引言

面向数据集的开发模式和面向对象的领域模型是企业应用业务逻辑组织的两种主要模式。

在面向数据集的开发模式中，系统开发框架提供了大量的数据感知组件使得开发者可以使用 RAD 组件快速开发基于 DataSet 的企业应用。该模式非常适合于中小型的企业应用程序开发，但是应对复杂的应用程序，随着业务逻辑复杂度的增加其缺点就会逐步显现。

首先，使用 RAD 和数据感知组件，就意味数据表现层同数据库表的紧偶合，任何对数据模型的改变都会导致对所有绑定到改动的表或字段的数据感知组件的修改。其次，使用数据感知组件，意味着同数据库的特有特性的耦合，当向不同数据库平台移植时，需要重新编写大量的业务逻辑[1]。

因此，使用面向对象的企业应用开发框架来进行系统的开发越来越得到开发者的重视。但是由于对象和关系模型之间存在"阻抗不匹配"问题[2]，因此把面向对象的一些操作映射到关系数据库时，需要编写繁琐的数据访问代码，而这些代码总是有大量重复内容。一旦数据层发生变化，就需要修改业务层的代码来适应数据层的变化，导致系统难以维护。

开发人员试图采用对象—关系映射 （ORM，Object Relational Mapping）组件来解决上述问题，即在业务层和数据层之间添加一个组件，将面向对象编程所建立的对象在数据库中做一个映射，使之和数据库中的表建立对应关系，把对表的直接操作变为对类的属性和方法的操作[3]。用户不需要为每个类编写数据访问代码，把开发人员从重复的劳动中解脱出来，使之有更多时间和精力关注实际业务需求，从而提高系统的开发效率。

1 NHibernate

NHibernate 是基于.NET 的 ORM 开源框架，是来源于非常优秀的基于 Java 的 Hibernate 关系型持久化工具。NHibernate 从数据库底层持久化.NET 对象到关系型数据库[4]。它不仅管理.NET 类到数据库表的映射，还提供数据查询、获取数据的方法和代码自动生成机制，从而大幅度地减少开发人员直接使用 SQL 和 ADO.NET 处理数据的时间，摆脱了 SQL、ADO.NET 和事务、缓存等底层。

1.1 NHibernate 的特性

NHibernate 具有以下特性：

对象持续性：能够管理.Net 类到数据库表的映射，以对象的方式存取数据，支持复杂对象、复合对象，支持对象之间的关联。OR Mapping 的定义都是基于 XML，具有很好的扩展性和通用性。可以支持现有的数据库定义，很好地保护用户投资。

支持对象查询：提供了面向对象的查询语言（HQL 和条件查询），可以根据条件查询复合对象以及对象集合。

支持事务：创建还必须支持悲观锁的事务，并提供了乐观锁的并发支持。

性能优化：允许用户使用定制的 Sql 来提高查询的性能，提供了多种 SQL 自动策略开关，使得框架生成的 Sql 语句具有非常优化的性能。提供了灵活的 Cache 缓冲机制，以及

基金项目：河南省科技重点攻关项目（072102210061）

作者简介：崔玉连（1980-），女（汉），河南省唐河县人，南阳理工学院计算机与信息工程学院，硕士，助理实验师，研究方向：计算机实验教学与研究，南阳，473004

杨新锋（1979-），男（汉），河南省内乡县人，南阳理工学院计算机与信息工程学院，硕士，副教授，研究方向：计算机教学与研究，南阳，473004

延迟加载，批量更新的策略，保证一般应用的性能不会低于相应的数据集应用。

数据库平台无关性：使用 OR Mapping 技术实现了数据库平台无关性，可以随时切换开发及数据库发布平台，方便移植。

1.2 NHibernate 的体系结构

NHibernate 的体系结构，如图 1 所示：

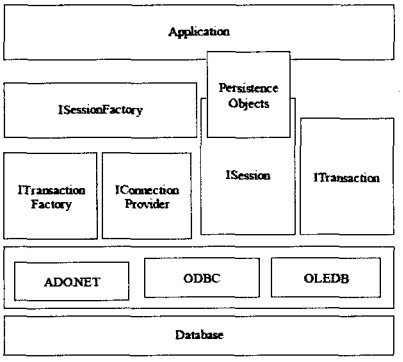

图 1 NHibernate 体系结构

采用 NHibernate 后，.NET 的系统架构仍然采用 N 层架构，界面层提供用户的操作接口，业务逻辑层同传统的 N 层架构一样通过调用持久层接口实现了所有对数据库的操作[5]。NHibernate 持久化层实现了应用程序与数据库的隔离，NHibernate 会生成相应的 SQL 语句实现与数据库的数据交互。持久化层封装了使对象持久化的行为，即从持久化存储对象中读取、写入、删除对象。

1.3 NHibernate 的映射机制

由于关系模型主要组成成分是数据库中的表和表与表之间的关系，而对象模型的主要组成成分是对象以及对象之间的联系，因此为了消除对象模型与关系模型的不匹配问题，有必要在它们的组成成分间建立一个映射，这个映射包括结构映射和关系映射。NHibernate 通过 XML 文件实现对象和关系数据库表的结构映射和关系映射，用户在 XML 文件中定义实体类和数据库表的映射，XML 文件必须以类名+.hbm.xml 作为文件名[6][7]。NHibernate 通过 XML 文件完成的映射关系，如图 2 所示：

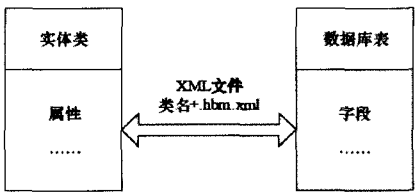

图 2 NHibernate 通过 XML 文件完成的映射关系

2 在全程办税系统中的应用

2.1 系统概述

全程办税及纳税评估系统是根据全程办税服务的实际情况，针对全程办税、指标管理、台帐管理、评估性管理等方面进行信息化建设,初步建立起一个高集成度、高共享性的现代全程办税管理信息系统,为全程办税质量和效率的提高、评估管理的规范，提供了一个强大、持久、稳定的技术平台。本系统共包括和全程办税服务的记录与监督、科室与个人的指标性管理、纳税台帐管理、评估性管理、管理五大模块。本系统采用C#语言开发基于C/S 架构的分布式信息管理系统。其中，NHibernate 作为数据持久层来封装对数据库的操作。

2.2 系统总体框架

依据系统设计强内聚、弱耦合的原则，以及划分功能模块要求设计简单、权限分配方便，便于用户理解的原则，本系统采用了 4 层架构，如图 3 所示：

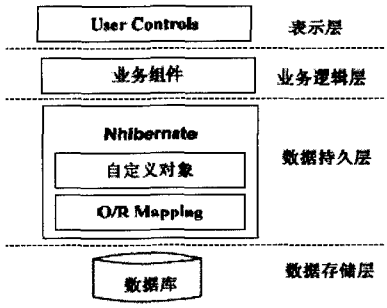

图 3 基于 Nhibernate 的全程办税及纳税评估系统框架

本系统的 4 层框架分别为：表示层、业务逻辑层、数据持久层和数据存储层。其中，表示层利用 C#进行编程，实现用户界面；业务逻辑层封装了全程办税的业务逻辑，通过定制业务访问组件 NHibernate，实现各业务逻辑模块；数据持久层封装了操作业务对象的持久化方法，利用 Nhibernate 进行对象－关系的映射，实现数据的持久操作。数据存储层即数据库服务器，负责数据的存储、组织和管理。

2.3 NHibernate 实施步骤

（1）创建 NHibernate 数据库连接

NHibernate 数据库连接由配置文件 NHibernat.cfg.xml 决定，文件中指定 NHibernate 所使用的数据库以及用户名、密码及其他相关配置。本系统配置文件内容如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<configSections>
    <section name="nhibernate" type="System.
    Configuration.NameValueSectionHandler,
    System,Version=1.2.0.4000,Culture=neutral,
    PublicKeyToken=b77a5c561934e089" />
</configSections>
<nhibernate>
    <add key="hibernate.show_sql" value="true" />
    <add
      key="hibernate.connection.provider"
      val-
ue="NHibernate.Connection.DriverConnectionProvider"
    />
    <add
      key="hibernate.dialect"
      value="NHibernate.Dialect.MsSql2000Dialect"
    />
    <add
      key="hibernate.connection.driver_class"
      value="NHibernate.Driver.SqlClientDriver"
     />
    <add
```

```
key="hibernate.connection.connection_string"
value="Server=localhost;initial
catalog=gxds_Data;uid=sa;pwd=admin"
/>
</nhibernate>
</configuration>
```

配置文件中参数的含义，如表 1 所示：

表 1 配置文件中参数含义

参数	值
hibernate.show_sql	True表示向控制台输出运行中产生的 Sql，用于调试目的
hiber-nate.connection.provider	表示使用指定的类来提供数据库连接缓冲池。
hibernate.dialect	表示 NHibernate 方言的类名，可以让 NHibernate 使用某些特定数据库平台的特性，目前 NHibernate 支持 Sql Server、Oracle、Mysql、Firebird、Sybase、PostgreSql 等数据库方言。
hiber-nate.connection.driver_class	Ado.Net 的驱动类，支持 SqlServer、Oracle、Mysql、OleDb、ODBC、Firebird 等驱动。
hibernate.connection. connection_string	对应于 ADO.net 的连接串。

（2）创建对象的相应持久化类

持久化类是指其实例需要持久化到数据库的类。在持久化类中只定义属性和属性相对应的 get、set 方法，属性和数据库表的字段一一对应。

（3）创建映射文件

每个数据库表对应一个 NHibernate 映射文件（命名格式为*.hbm.xml），用于生成数据模型。在映射文件中，需要定义数据存储到哪个数据库表，哪个属性映射到数据库表中的哪个列字段，不同的对象如何相互关联。

具体来说，xml 文件中的 Class 元素的 name 属性指定了要进行映射的类的名称， table 属性指定了要进行映射的类在数据库中进行持久存储的库表名称。Id 元素用于声明类标识属性对应于数据库表的主键字段的映射关系，其中 name 属性标识类的标识属性名称，column 属性标识持久存储的库表的主键字段的名称,type 是标识字段的数据类型，而内嵌的 generator 元素则指明主键字段唯一值的生成方法，若其值为 identity 则表示 NHibernate 使用 Sql Server 数据库本身提供的自增加字段的特性来保证键值唯一。

在运行过程中，NHibernate 根据映射文件生成各种 SQL 语句。

（4）持久化操作的实现

建立映射文件和持久化类之后，可以很容易地实现针对单个持久类对象所有的基本操作，包括添加、删除、更新和查找等。其基本过程为：

①加入 NHibernate.dll 的引用：

```
using NHibernate;
using Nhibernate.Cfg;
```

②创建 configuration 对象：

```
Configuration cfg=new Configuration();
cfg.AddAssembly("MyAssembly");
```

③创建会话对象通过调用 Configuration 对象的 BuildSession－Factory 方法创建 SessionFactory 对象，将 Configuration 对象包含的配置信息和 SessionFactory 发生关联；然后通过调用 SessionFactory 对象的 OpenSession 方法创建 Session 对象，Session 对象提供到后台数据库的链接；接着再通过调用 Session 对象的 BeginTransaction 方法创建可以被 NHibernate 管理的事务。

```
ISessionFactory factory=cfg.BuildSessionFactory();
ISession session=factory.OpenSession();
ITransaction tran=session.BeginTransaction();
```

④加载、保存、查询对象。通过调用 ISessionFactory 实例的 openSession 方法创建 Session 实例，最后通过 Session 接口的 save()、update()、delete()、load()和 find()等方法完成对数据库对应表的操作，完成对持久对象类数据的存取。

⑤提交事务并关闭会话。

2.4 实施效果

实践表明，由于本系统采用了 HNibernate 实现对象－关系的映射，在对象检索方式上节约了大量的资源，尤其当系统中进行大量数据查询时尤为明显。同时由于引入对象－关系映射技术，开发人员可以采用面向对象的方法来设计应用系统，用户更改后台数据库也不会导致程序的修改，只需要对配置文件进行简单的修改，从而降低程序维护和更新的成本，大大缩短了开发时间，提高了程序的可靠性。而且由于采用 NHibernate 实现了数据存储和业务逻辑的分离，使得对各层进行独立开发、跟踪及优化成为了可能。

3 总结

由于 NHibernate 的设计集成了很多操作数据库方面的有效经验，对生成的 SQL 语句进行最大限度的优化，可以高效地利用 ADO.NET 操作数据库。但是，NHibernate 也存在一些明显的缺点与不足：为了实现对象与关系模型之间的映射，需要编写复杂的 XML 映射文件且容易出错；需要学习 HQL 语言，增加学习成本；NHibernate 具有较大的灵活性，体系结构比较复杂，使用难度大；不支持存储过程、不具备事务处理等数据库高级功能[8]。这些都需要我们进一步研究和探讨。

参考文献：

[1] 陈省. NHibernate 之 Hello NHibernate[J].电脑编程技巧与维护, 2004, 10:42-45.

[2] 刘金, 徐苏, 冯豫华. 基于 Hibernate 的 J2EE 数据持久层的设计与实现[J]. 计算机与现代化, 2007, 23(4): 56-58.

[3] 赵广利. 基于 NHibernate 的数据持久化方案[J]. 计算机工程, 2009, 35（20）:53-55.

[4] 秦泽叶, 高改梅. NHibernate 在实验室信息管理系统中的应用研究[J]. 科学之友, 2010, 10:40,43.

[5] 李昕. NHibernate 在.NET 中的应用[J].福建电脑, 2009, 4:28,20.

[6] 刘伟, 严晖. 利用 NHibernate 开发与数据库无关的系统[J]. 计算机技术与发展, 2007,17(7):105-107.

[7] 陈龙. 基于.NET 平台 ORM 技术-NHibernate 的研究与应用[D].长春:长春理工大学,2006.

[8] 徐长盛, 戴超, 谢立. J2EE 数据持久化技术的研究[J]. 计算机应用与软件, 2006, 23(4): 56-58.

(收稿日期：2013.07.09)