

数据访问模型和数据访问层构造器的设计与实现

王合闯^{1,2}, 卢涵宇^{1,3}, 王 森⁴, 刘会芳¹, 王华军¹

(1 成都理工大学 信息工程学院, 四川 成都 610059; 2 濮阳职业技术学院 数学与信息工程系, 河南 濮阳 457000;
3 贵州大学 计算机与信息工程系, 贵州 贵阳 550003; 4 北京理工大学 机电学院, 北京 100081)

摘 要: 通过研究 MVC 等开发模型, 提出了一种在分层开发模式下引入数据访问模型的设计方法, 该方法用于屏蔽表示层开发者对底层数据库的依赖性, 设计了数据访问模型构造器, 加快了程序开发的进度, 节约了程序开发过程中的人力、物力和时间, 且能够快速响应用户的需求。应用实例表明该方法简单有效, 具有较强的实用性。

关键词: 数据访问模型; 数据访问模型构造器; RAD; MVC 模型

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2011)05-0001-03

Design and Implementation of the Data Access Model and Data Access Layer Constructor

WANG He-chuang^{1,2}, LU Han-yu^{1,3}, WANG Sen⁴, LIU Hui-fang¹, WANG Hua-jun¹

(1 College of Information Engineering, Chengdu University of Technology, Chengdu 610059, China;

2 Department of Mathematics and Information Engineering,

Puyang Vocational and Technical College, Puyang 457000, China;

3 Department of Computer and Information Engineering, Guizhou University, Guiyan 550003, China;

4 College of Mechanical and Electrical Engineering, Beijing University of Technology, Beijing 100081, China)

Abstract: Through the research on MVC model for development, this paper put forward a preliminary ideology about Data Access Model in the hierarchical development mode, Data Access Model used in shielding that of dependence for the developer in presentation layer. It expedites progress and saves man-power, material and time in the application development, and able to respond quickly to the needs of users. The experiments demonstrate that the proposed algorithm is simple, efficient and practicable.

Key words: Data Access Model; Data Access Model Constructor; RAD; MVC Mode

1 引言

在以往的应用程序开发过程中,许多大型软件的开发往往采用增量模型或螺旋模型^[1-2],系统需要经过 OOA、OOD、OOP 等过程^[3],虽然这种开发模式使得软件开发能够达到 CMM3^[4]或者更高级,但却延长了系统的开发周期和增大了系统开发的复杂度。同时客户往往在开发中不断改变他们的需求,并期望开发人员能接受他们的需求,这些往往带来应用程序架构的改变。于是快速程序开发(RAD)^[5]和

敏捷程序开发(XP)^[6]这些编程策略常常被系统开发者所采用,来解决如何快速实现 MVC 模型^[7]中的 Model 和 Data Access Layer,并能够快速响应客户的需求改变的问题。

鉴于上述原因,文中在研究快速应用程序开发的三层模型和 MVC 模型的基础上,提出了一种在分层开发模式下引入数据访问模型的设计方法,设计了数据访问模型构造器,本层对数据层做出映射,并对中间业务逻辑层提供操作接口,尽可能地屏蔽用户需求改变给上层带来的影响,采用代码构造器,

收稿日期: 2010-06-16; 修回日期: 2010-09-16

基金项目: 贵州省教育厅自然科学基金项目(2007068)

完成系统需求改变到代码层的快速映射. 应用实例表明该方法简单有效, 具有较强的实用性.

2 数据访问模型研究

MVC(Model-View-Controller), 即把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离, 这样一个应用被分成三个层——模型层、视图层、控制层. 视图(View)代表用户交互界面, 模型(Model): 就是业务流程/状态的处理以及业务规则的制定. 控制(Controller)可以理解为从用户接收请求, 将模型与视图匹配在一起, 共同完成用户的请求. 本模型研究在三层结构和快速应用程序开发中, 如何快速地为团队开发的表示层提供一个实现这种经典设计模式的类似环境. 表示层开发者通过调用接口来实现视图; 数据模型和控制器的功能在通过代码构造器实现; 数据模型对应数据层中数据单元, 通常在数据库系统中对应数据库的一个表. 这种设计提供的一个多层系统, 实现的系统在开发和维护、重构方面有明显的优点. 将用户显示(视图)从动作(控制器)中分离出来, 提高了代码的重用性. 将数据(模型)从对其操作的动作(控制器)分离出来, 可以设计一个与后台存储数据无关的系统. 就 MVC 结构的本质而言, 它是一种解决耦合系统问题的方法.

系统模型如图 1 所示.

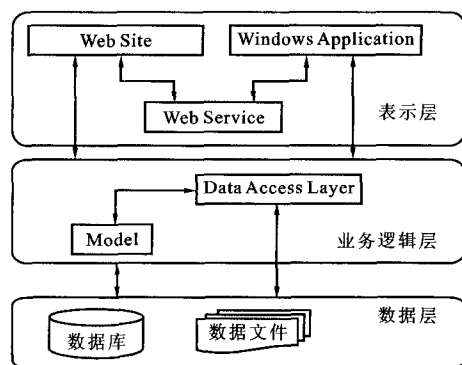


图 1 系统模型

对于数据层, 数据来源于数据库, 也可以来源于数据文件, 如 XML 结构化数据文档, 本模型只研究 MSSQL 数据库数据源. 数据层采用 ADO、NET 或 Linq 技术与业务逻辑层交互^[6]. 业务逻辑层的 Model 和 Data Access Layer 模块由代码构造器自动完成, 对于表示层的 Web Site 或 Windows Application, 可以直接调用业务逻辑层的接口, 也可以通过 Web Service 来调用业务逻辑层的接口.

3 数据访问模型的分析与设计

数据访问模型的总体设计思想是屏蔽表示层开发者对底层数据库的依赖性, 能够快速从模型中获取指定的数据, 并能将组织好的数据存储到数据库、操纵数据源对指定数据的删除与更新; 数据模型屏蔽不同数据库之间的差异性, 能够完成数据库到数据模型的映射.

基于以上思想, 数据模型在设计时分为三个部分. (1) 数据访问提供者. 为了满足当数据源数据库发生变化时对上层的屏蔽, 数据模型设计了一个数据访问代理, 该代理共提供了两类功能. 首先数据访问模型提供了一个数据连接对象, 针对不同的数据库, 进而作不同的修改, 提供不同数据库的连接, 并提供连接字符串属性, 完成数据源地址变化时的数据移植. 其次数据访问者对数据源数据提供基本的数据访问对象模型, 如对数据库的读取、删除、插入、更新等操作. (2) 数据表映射对象模型. 数据访问模型包含了数据库中每一个数据表的数据表映射对象模型, 以满足数据表示层与数据模型的数据快速交换和标准交换. 每一映射对象模型提供数据表每一字段的读写属性. (3) 数据表映射对象操纵访问模型. 该模型为表示层提供了数据表访问接口, 表示层访问者能够快速地将组织好的数据送入该模型进行处理, 包括数据的读取、更新、插入、删除、校验等. 整个模型的设计要求对表示层提供的接口友好、齐全与安全, 对数据库层交换数据要求灵活、高效.

4 数据访问模型的验证与实现

下面通过设计与实现基于数据访问模型的数据访问模型构造器, 来验证数据模型的可行性和高效性. 构造器的设计基于 MSSQL 数据库, 为 VS.NET 应用程序和 WEB 应用程序开发提供中间访问层代码.

代码构造器通过读取 MSSQL 数据库等相关设计信息, 从而动态生成业务逻辑层的 Model 和 Data Access Layer, 这样做的目的是快速地通过用户需求, 完成数据层的建立, 达到 RAD 的开发需求. 为了标准化和团队开发的需要, 数据库的建立必须满足团队所定义的标准, 要求所有建表和初始化字典必须写在存储过程中, 并且写好注释, 对于每一个表的创建, 都必须包含 if + drop table + create table 三条语句, 数据字典数据的插入写在建表之后, 临时测试数据写在另一个存储过程中, 以方便表示层开

发调试时使用,这样当需要对数据库进行初始化时,只需要运行建表的存储过程,即可得到一个完整、干净的数据库环境,运行含有临时测试数据存储过程即可得到一个含有测试数据的数据环境.为了完成数据库中表字段的反射,对存储过程和注释系统有特殊的要求,每一列都有注释,格式如:“——中文名称;”以及每一个表创建完毕最后必须以“);”结尾,这样的存储过程具有可读性和可维护性,方便团队开发.当团队开发期间,数据库的创建和数据字典的生成都在存储过程中完成,临时测试数据单独存放.当一个小组成员进行单元测试时,若生成了大量的垃圾数据,其可通过重新运行存储过程,达到恢复原始状态的要求.当程序移交或移植时,也可通过运行此存储过程,清空测试数据和重建数据表.一旦程序正常运行,此存储过程必须控制运行权限,甚至删除.

系统按照以下步骤完成数据模型的映射:

(1) 登录代码构造器,可以有两种登录方式,以连接字符串或 IP、用户名、密码、数据库名方式,并指明生成的数据模型对象代码存放的位置.

(2) 登录后选择所对应的表示层程序类型. 对于 WebSite , 系统不提供命名空间, 以便整个 Web 系统的访问, 对于 Windows Application, 要求提供命名空间.

(3) 生成具有通用操作类 DbOperator. 该类完成连接字符串和连接对象的提供, 完成对数据库所有表的读写更删. 为其他对象的 Data Access Layer 提供基础操作.

(4) 读取自定义数据表的名称和 ID,通过读取系统表 sysobjects 获取.

(5) 遍历所有自定义表,以表名为参数,依次读取每个表表结构.通过系统表 INFORMATION_SCHEMA.COLUMNS 与 INFORMATION_SCHEMA.KEY_COLUMN_USAGE 左关联获得.

(6) 根据表结构,依次建立每个表的 Model 和 Data Access Layer. 通过 `StreamWriter` 类将获得信息写入文件.

(7) 建立表头反射类 TableHeadHandle. 通过系统表 syscomments 和 sysobjects 左关联, 读取存储过程内容. 如果存储过程过长, 系统表将以多条记录存储, 为此必须按照 sys. syscomments. colid 升序排序, 将 sys. syscomments. text 字段内容连接起来, 再进行查询.

5 应用实例测试

数据访问模型构造器构在诸多的开发过程中,都显示了其高效性、可维护性、便捷性.下面是笔者运用文中提出的开发方法在宁波物流网实际开发中的应用,图2所示为宁波物流网首页测试截图.

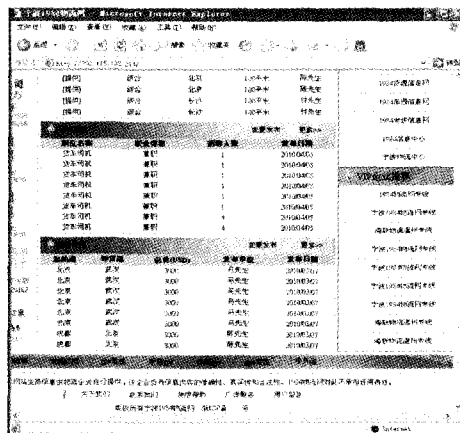


图 2 宁波物流网截图

在宁波物流网实际开发中,我们对中间层整个业务代码开发合计 4 076 行,其中由数据访问模型构造器产生的代码共计 3 999 行,程序员以 partial 类添加的代码行共计 77 行,整个开发过程中修改数据库 7 次,修改后由数据访问模型构造器重新产生代码,由于加入了 partial 类,需要调整数据访问模型构造器产生的代码,合计调整 21 行.图 3 中选中部分是由数据访问模型构造器产生的代码,未选中部分是由程序员手工添加的代码.整个使用过程表示层开发人员对数据的操纵自由度和易用性表示满意.

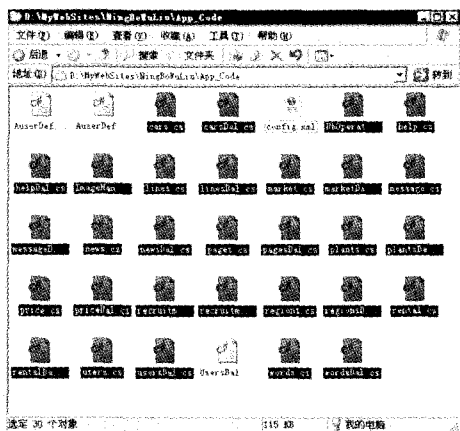


图 3 数据访问模型构造器产生的代码

- techniques based on color features and image bitmap[J]. Information Processing and Management, 2007, 43(2): 461—472.
- [2] 高燕, 胡学龙, 李玲, 等. 基于小波多尺度分析的综合特征图像检索[J]. 微电子学与计算机, 2010, 27(4): 21—28.
- [3] 宋琳琳, 王相海. 一种基于 HVS 加权颜色特征的图像检索算法[J]. 微电子学与计算机, 2008, 25(6): 60—62.
- [4] Swain M J, Ballard D H. Color indexing[J]. International Journal of Computer Vision, 1991, 7(1): 11—32.
- [5] Stricker M, Dimai A. Spectral covariance and fuzzy regions for image indexing[J]. Machine Vision and Applications, 1997, 10(2): 66—73.
- [6] Chan Y K, Liu Y T. An image retrieval system based on the image feature of color differences on edges in spiral scan order[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2003, 17(8): 1417—1429.
- [7] Chan K W, Chan K L. Optimisation of multi-level block truncation coding[J]. Signal Processing: Image Communication, 2001(16): 445—459.
- [8] Mathias E, Conci A. Comparing the influence of color spaces and metrics in content-based image retrieval [C]// Proceedings of the International Symposium on Computer Graphics, Image Processing and Vision. Rio de Janeiro, Brazil: IEEE, 1998: 371—378.
- [9] Kekre H B, Sudeep D Thepade. Boosting block truncation coding with kekre's LUV color space for image retrieval[J]. International Journal of Electrical, Computer, and Systems Engineering, 2008, 2(3): 172—180.
- [10] Guoping Qiu. Color image indexing using BTC[J]. IEEE Transactions on Image Processing, 2003, 12(1): 93—101.

作者简介:

邹彬男, (1986—), 硕士研究生. 研究方向为图像处理与识别.

(上接第 3 页)

6 结束语

文中提出了一种在分层开发模式下引入数据访问模型的设计方法, 设计了数据访问模型构造器, 该构造器能够自动生成系统开发层的 Model 和 Data Access Layer 层, 解决了以往手工编写该层代码且容易出错的问题, 开发测试过程中减少的代码工作量尤为明显, 为系统的 RAD 开发提供了基础. 相对于其他的开发方法, 该方法加快了程序开发的进度, 节约了程序开发过程中的人力、物力和时间, 且能够快速响应用户的需求. 目前基于数据访问模型和数据访问层构造器的设计方法已由笔者设计实现并应用到宁波物流网设计开发测试中, 取得了良好的效果, 具有较强的实用性.

参考文献:

- [1] iansommerville. 软件工程[M]. 北京: 机械工业出版社, 2007: 43—44.

- [2] iansommerville. 软件工程[M]. 北京: 机械工业出版社, 2007: 44—45.
- [3] 邵维忠, 杨芙清. 面向对象的系统分析[M]. 北京: 清华大学出版社, 2006.
- [4] Cohan Sean, Glazer Hillel. An agile development team's quest for CMMI maturity level 5[C]// Proceedings — 2009 Agile Conference. Chicago, USA: IEEE, 2009 (8): 201—206.
- [5] 王海燕, 周思方. 基于 Web 服务的校园网格异构数据源统一访问接口[J]. 微电子学与计算机, 2009(8): 179—181.
- [6] 林·巴斯, 保罗·克莱门茨, 瑞克·凯兹曼. 软件构架实践[M]. 北京: 清华大学出版社, 2006.
- [7] 李勇, 陈明, 王斌. 基于 Linux 的物探软件快速开发架构模型[J]. 微电子学与计算机, 2009(9): 110—113.

作者简介:

王合闯男, (1975—), 博士研究生. 研究方向为虚拟现实与仿真、软件工程.