

基于.NET平台ORM技术阳光采购系统的设计

任 钢

(中国科学技术大学 计算机科学技术系, 安徽 合肥 230027)

摘 要:在基于.NET平台的管理信息系统中,面向对象的业务处理和关系数据存储存在着一定的矛盾。在分析.NET平台的数据库处理的技术基础上,设计了一个对象/关系数据库映射模式,并结合这个模式创建了一个通用的数据操作框架CFC,实现了对对象/关系数据库映射。CFC框架使软件开发人员能够以面向对象的方法进行对象数据存取,而不必了解后台关系数据库的实现。该框架实际应用到了广东电网公司阳光采购系统。

关键词:招标;.NET框架;对象关系映射

中图分类号:TP311.1

文献标识码:A

文章编号:1673-629X(2008)05-0203-05

Application and Design of SunBid Information System Based on ORM in .NET Framework

REN Gang

(Dept. of Computer Sci. and Techn., Univ. of Sci. and Techn. of China. Hefei 230027, China)

Abstract: The strong conflict exist between the persistent object and relationship database in the management information system based on .NET platform. Bring forward ORM model which map between object and RDS(relation database system). Based on .NET framework, ORM(object-relational mapping model) is designed and integrated to data operating framework named CFC depending on analyzing RDS access model in .NET framework. The programmer call objects operations and object-oriented ways without concerning the implementation of bottom relationship database. The CFC framework is applied to SunBid Information System of Guangdong Power Network Corp.

Key words: invite public bidding;.NET framework;ORM

0 引 言

评标是项目招标投标工作的核心和关键环节,评标工作的公平性、公正性以及科学性对确保招标方和投标各方的正当权益、使招标投标工作在经济建设中发挥实效具有决定性意义。文中基于.NET平台的ORM技术的广东电网公司阳光采购系统进行深入的研究,并提出设计方案和实现过程。

1 关键技术

1.1 .NET 框架

.NET Framework^[1]是微软推出的新一代的计算平台,其作用是建立一个可以快速构建 Web 应用、建立网络应用程序的计算平台。.NET 框架由公共语言运行时(CLR)、共享对象类库和 ASP.NET 组成。

1.2 ASP.NET

ASP.NET^[2]用于编写动态 Web 页。ASP.NET 建立在 .NET Framework 类的基础之上,并提供了由控件和基础部分组成的“Web 程序模板”,大大简化了 Web 程序和 XML Web 服务的开发。ASP.NET 程序可用多种语言开发,本系统采用 C#^[3]语言编写。

1.3 ADO.NET

ADO.NET^[4]是微软提供了一种高性能访问数据源的策略。ADO.NET 技术是一个基于标准的、面向创建的、分布式数据共享应用程序的编程模型。ADO.NET 主要包含四个主要专业组件(Connection, Command, DataReader 和 DataAdapter)和一个综合组件 DataSet。

1.4 对象关系/映射模式

对象关系/映射模式(ORM 模式),即 Object-Relational Mapping,是把对象模型表示的对象映射到基于 SQL 的关系模型结构中去,同时支持对象到关系数据库表的持久化。

收稿日期:2007-08-30

作者简介:任 钢(1971-),男,四川泸州人,硕士研究生,研究方向为软件架构、软件工程。

性的变化,把对象属性转化为 Delete, Insert, Select, Update 的 SQL 语句。在转化的过程中,包括对象的检索条件和传递的参数。

SqlBuilder 类是一个面向 .Net Framework 环境的对象/关系数据库映射类。对象/关系数据库映射用来把对象模型表示的对象映射到基于 SQL 的关系语言中去。DbContextHelper 提供数据提交、数据查询和获取数据的方法,可以大幅度减少开发时人工使用 SQL 和 Connection, DbDataAdapter, DbCommand, DbParameter 等专有处理对象的时间。

实现过程是 DataHelper 作为 DbContextHelper 的管理类,类似于工厂模式。由 DataHelper 创建出 DbContextHelper 实例化对象。主要的处理方法都是由 DbContextHelper 来实现,包括 ExecuteDataSet, ExecuteNonQuery, ExecuteReader, ExecuteScalar, ExecuteUpdate。如图 2 所示。

在实例化 DbContextHelper 的同时把数据源类型参数传递进去,返回当前的数据源类型接口,这个接口通过 OracleDataAccessProvider, SqlServerData Access Provider 和 OleDbDataAccessProvider 类中的一种去实现。这样,根据 C# 的数据访问模式,在相应的 Provider 生成出相应的 Connection, DbDataAdapter, DbCommand, DbParameter 等。比如, OleDbDataAccess Provider 就生成 OleDbConnection, OleDbData Adapter, OleDbCommand, OleDbParameter。

2.2 系统总体结构

采用 B/S 模式下的多层结构,包括以下 4 个层次:

表示层、业务逻辑层、持久层和数据层。图 3 表现了业务逻辑层和数据层及表示层的关系。

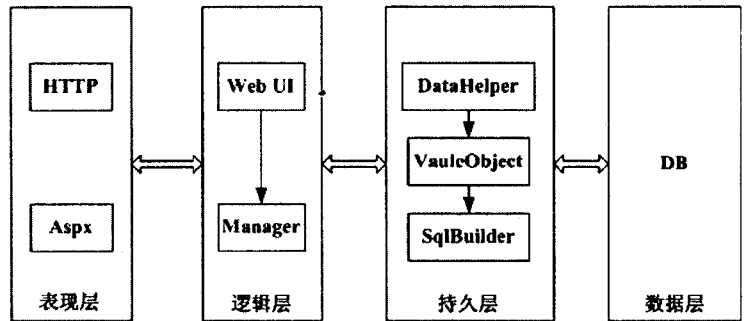


图 3 阳光采购系统多层结构应用模式

2.2.1 表示层

在表示层中,.NET 是由 ASP.NET Web Forms 来实现的。Web Forms 是基于服务器端的动态页面技术,服务器端根据客户端的请求和提交的信息,动态地生成与客户端浏览器类型无关的通用 HTML 页面。

2.2.2 业务逻辑层

业务逻辑层(事务逻辑层)包括业务管理层和数据访问层。业务服务子层主要处理一些业务逻辑和业务规则。先用表现层获取信息,然后依据获取的数据,进行业务处理,转化为持久对象,提交到专用数据操作层。

2.2.3 持久层

把关系数据映射为 VO(Value Object)对象。直接对 VO 对象进行操作。由 VO 属性的 SqlBuilder 把 VO 转化为关系数据库的 SQL,通过通用数据操作类 DataHelper 完成对数据库的操作。DataHelper 主要解决如何从数据库中提取和保存数据,并支持并发和事务管

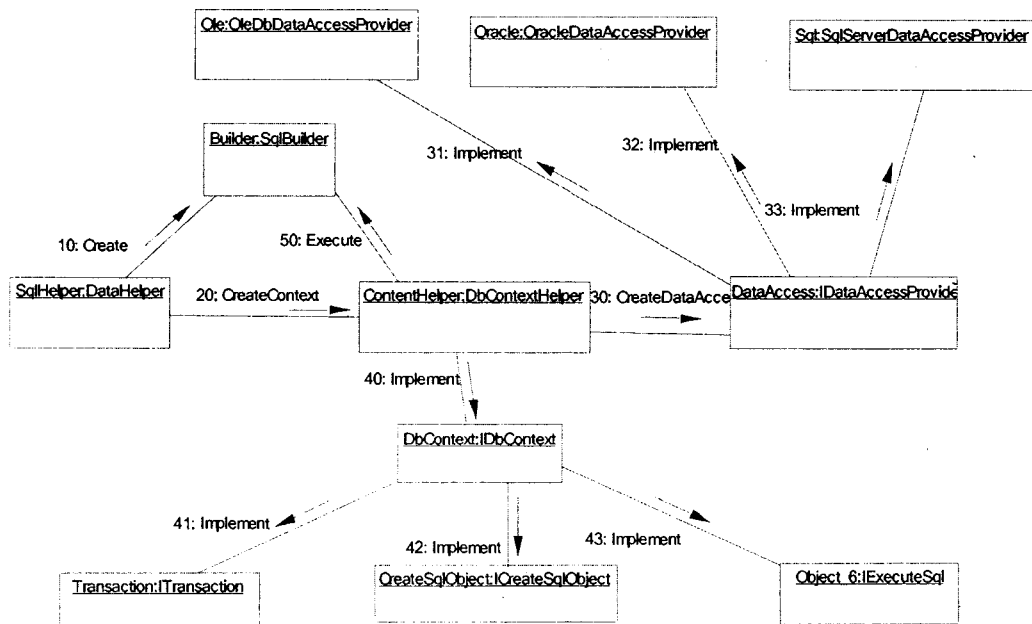


图 2 ORM 数据访问框架实现过程

理。

2.2.4 数据层

数据层就是 EIS 内容, 主要是数据库管理系统。VO 作为一个数据传输对象在各个层次间传递信息和数据。这样, 在开发过程中开发人员不用知道这些数据具体是如何进行操作的, 极大地降低了软件体系中各个功能部件的耦合性, 提高了软件的可重用性、可伸缩性、灵活性和可扩展性。

2.3 系统的功能模块

阳光采购系统的功能模块包括邀标管理、评标管理、辅助邀评标管理、基本资料管理、综合查询分析、系统管理, 见图 4。

2.4 系统实现

2.4.1 数据修改和提交模式

通过 O/R 映射的实现过程, 见图 5。

代码实现:

```
public class BidProductRightSer : System.Web.UI.Page {
    private int projectID;
    private int bidaimID;
    private void butSave_Click(object sender, System.EventArgs e) {
        //获取标的产品规格
        BidAim bidAim = BidAimManager.CreateInstance ( Convert.
        ToInt32(Request.QueryString["BidAimID"]));
```

图 5 阳光采购系统数据修改和提交实现模式

```
BidProductCollection bidProducts = BidProductManager. GetList
(new BidAim(bidaimID));
foreach(BidProduct bidProduct in bidProducts) {
    BidProduct bidpro = new BidProduct(bidProduct. ID);
    bidpro. RightValue = Convert. ToInt32(Request["bidProduct_" +
    bidProduct. ID]);
    BidProductManager. Update(bidpro);
}

public class BidProductManager {
    public static void Update(BidProduct bidProduct) {
        try {
            //执行 Command 对象命令, 把 BidProduct 对象数据更新到数据库
```

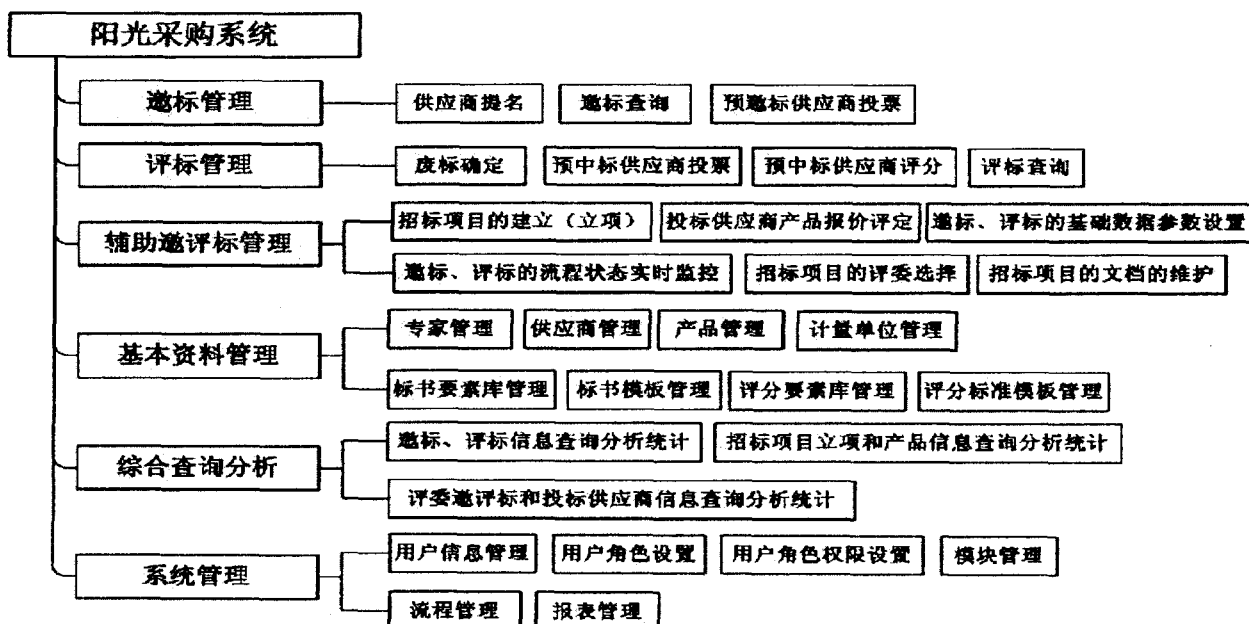
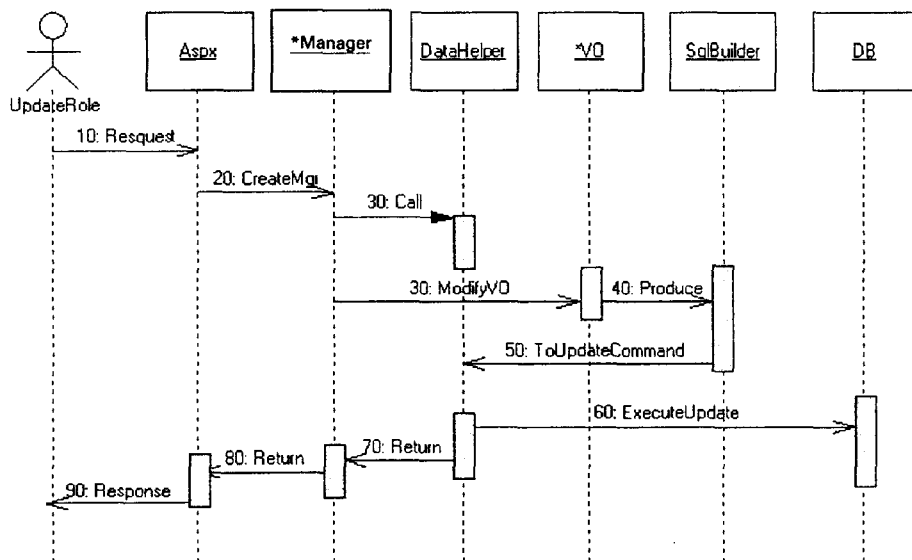


图 4 阳光采购系统功能模块

```

DataHelper. ExecuteUpdate (bidProduct. SqlBuilder. ToUpdateCom-
mand());
return;
} catch (Exception e) {throw new ApplicationException(e. ToString
());}
}
}

```

2.4.2 数据查询模式

通过 O/R 映射的实现过程,见图 6。

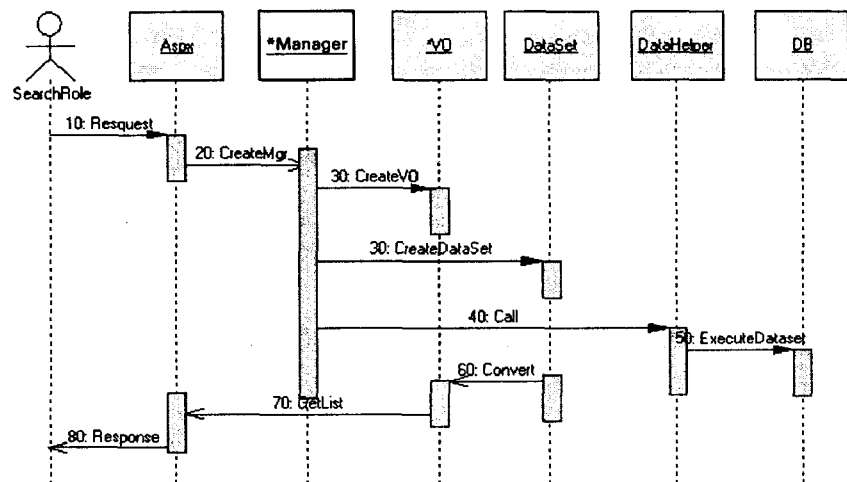


图 6 阳光采购系统数据查询实现模式

代码实现:

```

public class BidAimList: System. Web. UI. Page {
private int projectID;
private void BindGrid() {
//数据绑定到 CustomDataGrid 控件的数据源
CustomDataGridBidAim. DataSource = BidAimManager. GetList
(new Project(projectID), false);
CustomDataGridBidAim. DataBind();
if (((string)Session["currentpage"] != "0")) {
if (System. Convert. ToInt32(Session["currentpage"]. ToString())
<= CustomDataGridBidAim. PageCount) {
CustomDataGridBidAim. CurrentPageIndex = System. Convert.
ToInt32(Session["currentpage"]. ToString());
CustomDataGridBidAim. DataBind();
Session["currentpage"] = "0";
}
}
}
public class BidAimManager {
public static BidAimCollection GetList(Project project, bool isOnly-
Active) { StringBuilder strBuilder = new StringBuilder();
strBuilder. Append("SELECT * FROM BidAim WHERE ProjectID
=");
strBuilder. Append(project. ID. ToString());
BidAimCollection bidAims = new BidAimCollection (); //声明
BidAim 对象集合
try {

```

//从数据库返回数据到 DataSet 中

```

DataSet ds= DataHelper. ExecuteDataset(strBuilder. ToString());
//把 DataSet 中的数据依行转换成相应的对象
for(int i=0;i< ds. Tables[0]. Rows. Count;i++ ) {
BidAim bidAim= new BidAim();
DataRow dr= ds. Tables[0]. Rows[i];
//给 BidAim 对象赋属性时禁止追加相关值到 SqlBuilder 对象中
bidAim. IfAppendToSqlBuilder= false;
//给 BidAim 对象各属性赋值

```

```

bidAim. ID = Convert. ToInt32 ( dr
["BidAimID"]);
bidAim. Code= dr["Code"]. ToString();
...
bidAims. Add(bidAim);
return bidAims;
}
catch (Exception e) {throw new Applica-
tionException(e. ToString());}
}

```

3 结束语

评标是工程项目招投标工作的核心和关键环节。文中提出了基于 .NET Framework 平台的 ORM 技术思想和思路,对系统的结构和功能进行了设计和开发。实践证明:使用 ORM 开发应用系统,不仅可以提高程序的可维护性,也使开发工作变得更加灵活、方便,更适用于大型系统的团队开发,大大提高了编程的效率。

同时,该系统在实际运行中可提高物资采购透明度,保证采购的物资质量,降低采购价格,减少用户采购成本,使用户的集中采购取得极其明显的效益。经过多年的系统运行,阳光采购系统取得了明显的经济效益和社会效益^[6]。

参考文献:

- [1] Bellinaso M. ASP. NET Web 站点高级编程[M]. 北京:清华大学出版社,2002.
- [2] Lair R, Lefebvre J. ASP. NET 开发人员手册[M]. 张俊,何长松译. 北京:电子工业出版社,2002:8-10.
- [3] 微软公司. C# 语言参考手册[M]. 裴剑峰译. 北京:清华大学出版社,2001:3-4.
- [4] Riordan R. ADO. NET 程序设计[M]. 李高健,孙瑛霖译. 北京:清华大学出版社,2002:23-25.
- [5] 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(4):56-57.
- [6] 乔海涛,朱洪波,张晓霞,等. 广东省广电集团有限公司阳光采购系统概述[J]. 广东电力,2003(4):77-79.