《电子技术应用》 www.ChinaAET.com

基于.NET 的 Web 框架在用电信息采集系统中的应用

韩 勇，高会生，顾 博，刘 柳
(华北电力大学 电气与电子工程学院，河北 保定 071003)

摘 要：针对传统 ASP、PHP、JSP 等技术开发的 Web 应用系统层次不够分明、业务分工不够明确等不足，结合 MVC 设计模式、Ext JS、Spring .NET 和对象持久化 NHibernate 等技术，提出了一种基于.NET平台的 Web 应用程序框架，并结合实例详细分析了该应用框架在用电信息采集系统开发中的应用。基于该框架开发的用电信息采集 Web 应用系统面向接口编程，细化了传统三层式结构的软件开发，实现了页面显示逻辑、业务应用逻辑和数据访问逻辑的高效分离，具有松耦合性以及很强的可扩展性。

关键词：信息采集；MVC；Ext JS；Spring.NET；对象持久化
中图分类号：TP311.52 文献标识码：A 文章编号：1674-7720(2012)20-0005-04

Application of Web framework based on .NET in power consumption information collection system

Han Yong, Gao Huisheng, Gu Bo, Liu Liu
(School of Electrical and Electronic Engineering, North China Electric Power University, Baoding 071003, China)

Abstract: According to the shortage of web application system developed using ASP, PHP, JSP techniques, such as arrangement of ideas is not clear, business division is not enough limpid, and based on the technology of MVC pattern, Ext JS, Spring.NET and the object persistence with NHibernate, a framework of web applications based on .NET platform is presented. And an application example in power consumption information collection system based on this framework is analyzed in detail. Power consumption information collection system based on framework uses Interface-oriented programming, divides the tier of traditional three tiers' software development, implements the effective separation among page express logic, business application logic and data access logic, and has loose coupling and strong expansibility.

Key words: information collection; MVC; Ext JS; Spring.NET; object persistence

传统的用电信息采集系统主站软件开发大多采用 C/S 模式，其数据采集和业务应用等操作均在一个业务内网中完成，其数据仅能提供给业务内网所覆盖的少数内部相关人员使用，数据使用效率不高。基于 B/S 模式开发的用电信息采集系统将应用程序处理部分集中于服务器端完成，用户界面统一采用浏览器，无需安装客户端程序，任意授权上网客户均可获得最快捷的个人网上交互式服务。与此同时，维护人员也不再为程序的维护工作奔波于每个客户机之间，而把主要精力放在功能服务器的程序更新工作上。

然而在 B/S 模式开始盛行时，基于 Web 开发的用电信息采集系统大多采用 ASP、PHP、JSP 等技术，将业务逻辑和页面显示混合在一起，极其不利于分工与协作；而且在业务逻辑中采用内嵌 SQL 语句的方式完成数据访问，一旦数据库或者类定义中一方发生变化，就会导致系统的大幅修改，不利于系统的维护。针对以上不足，本文结合 MVC 设计模式、Ext JS、Spring.NET 和对象关系映射 NHibernate 等多项技术，提出了一种基于.NET 平台的 Web 应用框架，并分析了该框架在用电信息采集主站系统中的实际应用，为解决 Web 应用的不足提供一种解决方案。

1 Web 应用框架研究

1.1 MVC 设计模式

MVC 架构是许多交互和界面系统的构成基础，其核心是实现系统不同层次间的松散耦合。它把一个应用任务的输入、处理、输出流程按照模型、视图、控制器的方式进行分离，同时各个模块之间相互独立，提高了灵活性和可重用性。

《电子技术应用》 www.ChinaAET.com

《电子技术应用》 www.ChinaAET.com

在.NET 平台下,ASP.NET 提供了一个很好的实现 MVC 设计模式的类似环境。开发者通过在 ASPX 页面中结合 Ext JS 框架开发用户接口来实现视图;控制器的功能在逻辑功能后台代码(.aspx.cs/.ashx.cs)中实现;模型通常对应于系统的业务应用部分。

1.2 Ext JS 框架

Ext JS 是一个用 JavaScript 编写、与后台技术无关的前端 Ajax 框架,可以用来开发富有华丽外观的富客户端应用,能使 B/S 应用更加具有活力。Ext JS 融入了面向对象的概念,让开发者可以像理解其他面向对象语言一样,灵活地运用 JavaScript 语言,在支持面向对象的同时还提供了丰富的跨浏览器 UI 组件,灵活采用 XML/JSON 数据源进行开发,使得服务端表示层的负荷真正减轻,从而实现客户端的 MVC 应用。

1.3 Spring.NET 框架

Spring.NET 是一个开源的应用程序框架,它能够提供宽广范围的功能,例如依赖注入、面向方面编程(AOP)、ORM 类库整合等。依赖注入功能由框架提供的一种轻量级的控制反转 IoC 容器来完成。该容器改变了传统的在程序中强制声明对象的创建方法,实现了一种配置式的对象管理方式,降低了类之间的耦合度。AOP 为业务对象提供面向方面编程(AOP)的支持,完善了 IoC 容器的功能,为创建企业应用和使用声明式服务奠定了坚实的基础。ORM 类库整合为时下流行的 ORM 类库(如 NHibernate 等)提供了一个整合层,其中包含声明式事务管理等诸多功能。

1.4 NHibernate 框架

NHibernate 是一个基于.NET 的针对关系型数据库的对象关系映射 ORM(Object/Relation Mapping)框架。对象关系映射 ORM 的最主要目的是为了解决关系型数据库与面向对象编程技术中面向对象的类与数据库的表不是一一对应的“阻抗不匹配”问题。NHibernate 从数据库底层来持久化.NET 对象到关系型数据库,它封装了底层的数据库 SQL 操作,上层应用程序不需要知道数据库管理系统的类型、数据表的结构以及访问方法。当对象的模型不变,而只改变数据库管理系统或者改变数据表的结构时,则只须修改相应的 NHibernate 映射文件和配置文件,不需要对程序代码进行大的修改,因而使得应用程序具有较好的可移植性。

2 Web 应用框架设计

传统的 Web 应用系统中,通常将系统划分为 Web 表示层、业务应用层和数据层 3 个部分。采用 MVC 设计模式的 Web 应用系统,表示层被细分为视图层和控制器层,模型则通常对应系统的业务应用部分。在业务应用层将表示层、业务逻辑与数据访问进行分离,细分为业务逻辑层接口、业务逻辑层、数据访问接口层和数据访问层、实体层。Web 表示层只依赖于业务逻辑层接口和实体层,即在表示层由控制器调用业务逻辑层接口定义的方法,并处理返回的实体层数据;业务逻辑层实现了业务逻辑层接口,同时依赖于数据访问接口层和实体层,这一层实际是调用数据访问接口层中的方法组合为业务,并处理数据访问接口层返回的实体层数据;数据访问接口层定义了访问数据的底层方法;数据访问层实现接口中的所有方法;实体层负责整个系统中数据的封装及传递,定义的对象实体只有属性没有方法。系统应用框架如图 1 所示。

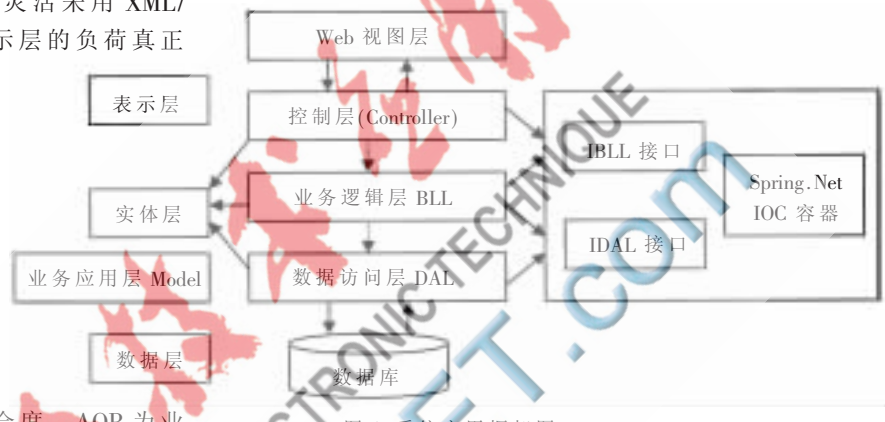

图 1 系统应用框架图

本框架设计中,视图层由 ASPX/HTML 页面结合 Ext JS 框架完成界面显示和 Ajax 请求;控制器功能在.ashx.cs 文件中实现,其主要负责用户请求和后台业务层的中转、接收并分析用户请求,调用业务逻辑层类完成请求,再分发给用户。数据库访问层采用 NHibernate 框架将关系数据库的数据映射成对象,实现以面向对象的方式操作数据库。系统中的各层之间借助于 Spring.Net 框架的 IoC 容器以松耦合的方式组合在一起,即表示层的控制器访问业务逻辑层时,调用的是业务逻辑层接口 IBLL,具体的业务对象实现则由 Spring.NET 框架的 IoC 容器动态注入。同理,业务逻辑层的实现调用的是数据访问层的接口 IDAL,同样由 IoC 容器注入具体的数据访问层实现。

综上所述,本框架具备如下优点:

(1)将 Web 开发的三层架构进行细分,结构清晰,功能完备,使得系统的开发更加有条理、更加便捷。

(2)对于大型的企业应用,前端页面显示复杂,共享代码较多,使用 MVC 模式分离显示与业务逻辑就使得共享代码便于管理和修改,降低了依赖性。

(3)采用富客户端 Ext JS 框架。将显示逻辑从服务器端转移到客户端,服务器端仅负责业务逻辑的处理和运算,并把处理的结果以纯数据的形式发送给客户端,由客户端负责具体的显示和交互,解决了以往 Web 应用系统性能低下、效率低、开发出的界面千篇一律等问题。

《电子技术应用》 www.ChinaAET.com

《电子技术应用》 www.ChinaAET.com

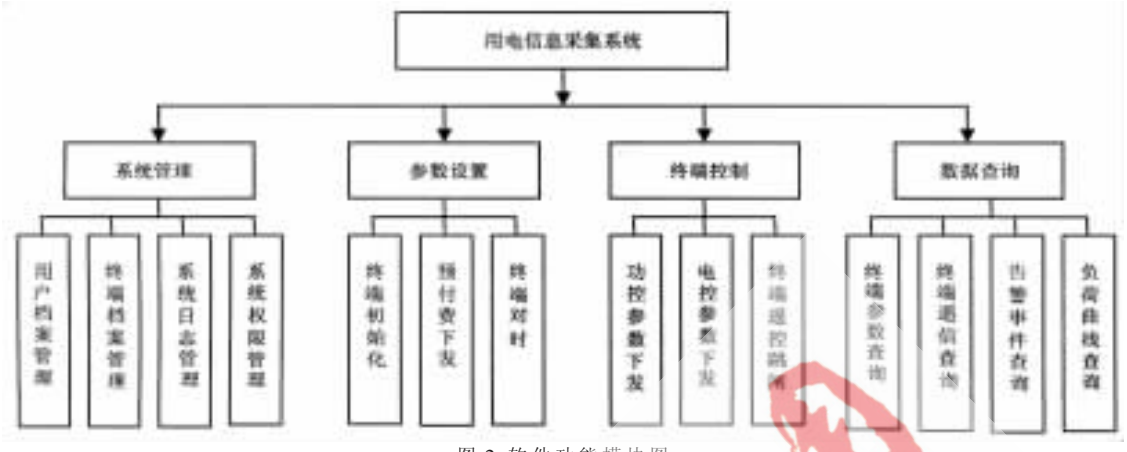

图 2 软件功能模块图

(4)框架面向接口编程,充分利用 Spring.NET 依赖注入特性,改变了传统的在程序中强制声明对象的创建方法,实现了一种配置式的对象管理方式,降低了层次之间的耦合度。

(5)将数据访问从业务逻辑中分离出来,采用 NHibernate 对象关系映射框架从数据库底层来持久化.NET 对象到关系型数据库,无需采用内嵌 SQL 语句的方式完成数据访问,提高了系统的可移植性。

3 采集系统软件设计

3.1 系统功能模块设计

用电信息采集系统是对电力用户的用电信息进行采集、处理和实时监控的系统,软件功能可归纳为系统管理、参数设置、终端控制和数据查询 4 大模块。软件模块如图 2 所示。

3.2 数据库设计

用电信息采集系统包含的数据实体根据不同功能可分为管理类实体、数据采集类实体和电网模型类实体 3 种类型,每个实体包含一个或者多个数据表。管理类实体主要包括与系统管理相关的实体,如系统功能、管理部门、岗位、操作员、操作员分组、用户、用户组、角色等;数据采集类实体主要指电能信息数据实体以及和电能信息数据采集相关的实体,如电表信息、终端信息、终端配置参数、终端控制状态、前置机、电能量、模拟量、任务配置、异常日志等;电网模型类实体主要指与电网设备和电网拓扑相关的各种实体,如区域、变电站、电压等级、变压器、线路、开关等。

4 基于 Web 应用框架的采集系统实现

在 Visual Studio 2008 中搭建起整个项目的解决方案,命名为 FK。通过建立不同的工程来划分层,每一个工程使用不同的命名空间。FK.Web 实现 Web 表示层;FK.IBLL 实现业务逻辑层接口;FK.BLL 实现业务逻辑层;FK.Model 实现实体层;FK.IDAL 实现数据访问层接口;FK.NHDAL 通过 NHibernate 实现数据访问层;FK.Controler 实现业务逻辑的调用。

下面以系统管理模块中添加终端设备操作的实现为例说明基于.NET 的 Web 应用框架在用电信息采集主站系统中的具体应用,其中涉及的业务逻辑包括将终端信息存储入库以及向前置机下发终端配置信息两部分,UML 类图如图 3 所示。

(1)实体层实现

实体层在 FK.Model. DeviceTable 类中实现,其中定义的对象实体只有属性没有方法。主要包括终端设备相

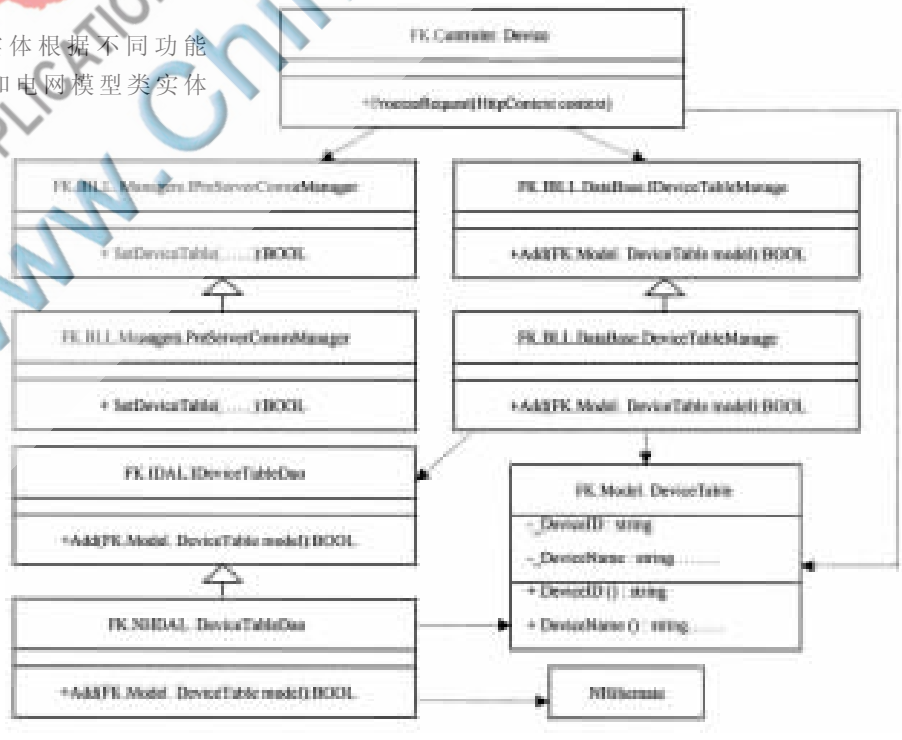

图 3 终端操作 UML 类图

《电子技术应用》 www.ChinaAET.com

《电子技术应用》 www.ChinaAET.com

关属性信息。部分代码如下：

```
namespace FK.Model { public class DeviceTable{
public virtual string DeviceID {get; set;}        //终端 ID
public virtual string DeviceName {get; set;}
        //终端名称 <! --省略其它属性信息--> }}
```

(2)数据访问层实现

数据访问层使用 Spring.NET 提供的 HibernateDaoSupport 作为基类，通过该基类的 HibernateTemplate 对象来完成数据访问操作。HibernateTemplate 封装了对象持久化的 CRUD 等基本操作（底层的数据库访问由 NHibernate 来具体实现）。这里由 DeviceTableDao 类来完成终端设备信息的存储入库。

数据库访问DeviceTableDao 类继承于接口IDeviceTableDao 和 HibernateDaoSupport 类。HibernateDaoSupport 类的 HibernateTemplate 属性通过 Spring.NET IoC 容器从外部注入，同时还通过 IoC 容器向 HibernateTemplate 中注入 SessionFactory，然后在 Spring.NET 中声明一个 SessionFactory 的对象。配置文件 Web.config 部分注入代码如下：

```
<! --将 id 为 HibernateTemplate 的对象注入到数据访问类-->
<object id = "DeviceTableDao" type = "FK.DAL.DeviceTableDao,FK.DAL">
<property name = "HibernateTemplate" ref = "HibernateTemplate"/></object>
```

(3)业务逻辑层实现

业务逻辑层通过 PreServerCommManage 类和 DeviceTableManage 类分别实现向前置机下发终端配置信息以及将终端信息存储入库等业务逻辑。在业务逻辑层的具体实现中，用到了数据访问接口 IDeviceTableDao 属性，通过 Spring.NET IoC 注入具体的数据访问实现，配置文件 Web.config 注入代码如下：

```
<! —声明 DeviceTableManage 业务逻辑类，将类 DeviceTableDao 注入到业务逻辑类-->
<object id = "DeviceTableManage" type = "FK.BLL.DataBase.DeviceTableManage, FK.BLL.DataBase">
<property name = " DeviceTableDao" ref = "DeviceTableDao"/></object>
```

(4)视图层实现

视图层主要由 HTML 结合 Ext JS 框架实现，其负责接收用户输入的终端信息，通过 Ext.Ajax 实现与控制层的通信，并显示控制层的业务逻辑调用执行结果，关键代码如下：

```
var amr = Ext.getCmp('createDevice');//获取界面上输入的终端设备数据，保存在 Dev.DeviceData 变量中
Dev.DeviceData ={DeviceName: amr.items.itemAt(0).items.itemAt(0).getValue(),<! --省略其他元素获取代码--> }
Ext.Ajax.request( params: { DeviceData: FK.Global.Encode(FK.Global.ToJson(Dev.DeviceData)) })//实现视图层与控制层的通信
```

(5)控制层实现

控制层通过视图层获取终端设备的输入数据，并调用业务逻辑层方法完成具体的业务逻辑，这里调用的是业务逻辑接口，具体的业务逻辑层通过 Spring.NET IoC 容器注入予以实现。实现文件 Device.ashx.cs 终端添加入库操作关键代码如下：

```
Spring.Context.IApplicationContext ctx = ContextRegistry.GetContext();        //建立 ApplicationContext 容器实例
FK.IBLL.DataBase.IDeviceTableManager
        //建立 DeviceTableManage 业务对象实例
sameDev= ctx.GetObject ("FK.BLL.DataBase.DeviceTableManage") as FK.IBLL.DataBase.IDeviceTableManager;
sameDev.Add(DeviceData);    //实现终端添加入库操作
```

综上所述，以系统设备管理模块中添加终端设备操作为例，从系统各层次详细分析了基于.NET 平台的 Web 应用框架在用电信息采集系统中的应用。

本文采用 Ext JS +ASP.NET MVC +Spring.NET + NHibernate 等技术构建多层 Web 应用程序框架，并分析了该框架在用电信息采集系统主站中的具体应用。本框架面向接口编程，具有松耦合性以及很强的可扩展性，比较适合用电信息采集系统多应用平台的开发。Spring.NET IoC 作为整个框架的容器，充分利用其依赖注入的特性，实现了组件间的松耦合；NHibernate 的本质是一个提供数据库服务的中间件，它的使用使整个框架更面向对象，同时支持多数据库；ASP.NET MVC 一改以往 Web Form 的方式，使界面和后台代码完全分开；Ext Js 则创建出更美观、用户体验更好的界面，能够很方便地完成 ASP.NET Web Form 很难完成的功能和界面。

参考文献

[1] 陈玮，沈雷.基于 MVC 模式的 Web 应用框架[J]. 微计算机信息，2009(15)：216-218.

[2] 李园，陈世平. MVC 设计模式在 ASP.NET 平台中的应用[J].计算机工程与设计，2009，30(13)：3180-3184.

[3] 任伟，林晓东.基于 Spring 框架和 Ext JS 的药品库房管理系统实现[J].计算机工程与设计，2009，30(18)：4312-4316.

(收稿日期：2012-05-22)

作者简介：

韩勇，男，1987 年生，硕士研究生，主要研究方向：电力系统通信。

高会生，男，1963 年生，教授，主要研究方向：通信网的管理和安全风险评估、信息处理。

顾博，男，1986 年生，硕士研究生，主要研究方向：电力系统通信。

《电子技术应用》 www.ChinaAET.com