基于.NET的三层结构在大豆专家系统中的应用

陈争光[1], 徐海燕[2]
(1.黑龙江八一农垦大学信息技术学院, 黑龙江大庆 163319; 2.大庆技师学院, 黑龙江大庆 163319)

摘要: 三层结构的软件开发技术是一种通用的模块化技术, 该技术旨在提高软件的可用性、灵活性和可伸缩性, 这种技术已经越来越受到人们的重视。介绍了三层结构的基本概念、特点及数据访问层和业务逻辑层的组成结构。并以大豆专家系统一个模块的开发为例说明了.NET环境下三层结构的开发技术的优点。

关键词: 大豆; 三层结构; .NET
中图分类号: TP311 文献标识码: A 文章编号: 1002-2767(2009)04-0135-03

Three-tier Architecture Based on .Net and Its Application in Soybean Expert System

CHEN Zheng-guang[1], XU Hai-yan[2]
(1. Information Science and Technology College of Heilongjiang August First Land Reclamation University, Daqing, Heilongjiang 163319; 2. Daqing Technician's Institute, Daqing, Heilongjiang 163319)

Abstract: The three-tier architecture is a versatile and modular infrastructure intended to improve usability, flexibility, interoperability and scalability. And this technique of layering software development is paid more and more attention. The essential conception, characteristic and the component of data access layer and business logical layer of three-tier architecture were discussed in this paper. The advantage of the three-tier architecture was illustrated by one of modules development under .net environment in soybean expert system.

Key words: soybean; three-tier architecture; . NET

1 三层结构介绍

所谓三层体系结构, 就是在传统的C/S结构的客户端与数据库两层之间增加一个新层, 形成一个三层结构[1-2]。三层是指逻辑上的三层[3-4], 每一层完成一个特定的逻辑功能。在软件体系结构设计中, 分层式软件结构是最常见, 也是非常重要的。微软推荐的分层式结构从下至上依次为: 数据访问层、业务逻辑层和表示层[5] (见图1)。

数据访问层: 有时候也称为是持久层, 其功能是负责对数据库的访问。通俗地讲, 就是实现对数据库表的插、删、改、查等操作。如果要加入ORM的元素, 那么就会包括对象和数据表之间的映射, 以及对象实体的持久化。

业务逻辑层: 是整个系统的核心, 它与这个系统的业务(领域)有关。如果涉及到数据库的访问, 则调用数据访问层。

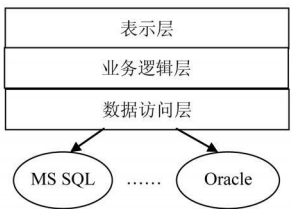

图1 三层的分层结构

表示层: 是系统的用户界面部分, 负责用户与整个系统的交互。在这一层中, 理想的状态是不应包括系统的业务逻辑。表示层中的逻辑代码, 仅与界面元素有关。

分层式结构的优势主要体现在[6]: 分散关注、松散耦合、逻辑复用、标准定义。

(1)分散关注: 开发人员可以只关注整个系统中的某一层;

(2)松散耦合: 可以很容易地用新的实现来替换原有层次的实现; 可以降低层与层之间的依赖性;

(3)逻辑复用: 有利于各层逻辑的复用。比如, 可以使用同一个业务逻辑来实现不同的表现层, 针对不

收稿日期: 2009-01-12
基金项目: 黑龙江省农垦总局资助项目(HNKXIV-09-01-02)
第一作者简介: 陈争光(1973-), 男, 湖北黄陂人, 讲师, 主要从事计算机技术在农业中的应用研究。E-mail: ruzee@sina.com。

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

同的客户端显示不同的界面。比如针对电脑和手机编写不同的界面，两者共用相同的业务逻辑。

(4)有利于标准化：一个好的分层式结构，可以使得开发人员的分工更加明确。一旦定义好各层次之间的接口，负责不同逻辑设计的开发人员就可以同时进行开发，软件开发周期可以大幅度缩短。同时，好的分层结构有利于软件的自动生成。

2　用 ASP .NET 部署三层结构的应用

Microsoft .NET Framework 是微软推出的下一代开发平台。从开发人员的角度说，.NET是一个公共语言平台的类库(FCL)，包含了近 100 个命名空间的近 5 000个类，还包括一个公共语言运行库(CLR)。

ASP .NET可以使用.NET平台快速方便地部署三层架构。ASP .NET革命性的变化是在网页中也使用基于事件的处理，可以指定处理的后台代码文件。.NET中可以方便地实现组件的装配，后台代码通过命名空间可以方便地使用自定义的组件。表示层放在 ASPX 页面中，数据库操作和逻辑层用组件来实现，这样就很容易实现三层架构。下面看看数据访问层和业务逻辑层的结构。

数据访问层(Data Access Layer，DAL)中(其模块结构如图 2 所示)，采用 DAL Interface 抽象出数据访问逻辑，并以 DAL Factory 作为数据访问层的对象工厂模块。对于 DAL Interface 而言，可以有支持不同数据库的具体实现，比如支持 MS-SQL 的 SQL Server DAL 和支持 Oracle 的 Oracle DAL 等实现，从而实现对不同数据库的操作。图 2 中 Model 模块则包含了数据实体对象。

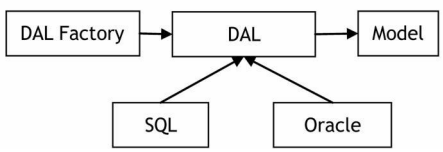

图 2　数据访问层的模块结构

在数据访问层中，完全采用了“面向接口编程”思想。抽象出来的 DAL Interface 模块，脱离了与具体数据库的依赖，从而使得整个数据访问层有利于数据库迁移。DAL Factory 模块专门管理 DAL 对象的创建，便于业务逻辑层访问。SQL Server DAL 和 Oracle DAL 模块均实现 DAL Interface 模块的接口，其中包含的逻辑就是对数据库的插、删、改、查、存储过程调用等操作。因为数据库类型的不同，对数据库的操作也有所不同，因此，SQL Server DAL 和 Oracle DAL 模块代码也会因此有所区别，当然，如果数据库类型确定，只需要实现其中的一个模块即可。

抽象出来的 DAL Interface 模块，除了解除业务逻辑层向下的依赖之外，对于其上的业务逻辑层，同样仅存在弱依赖关系[5]。图 3 所示为业务逻辑层(Business Logical Layer，BLL)的模块结构。

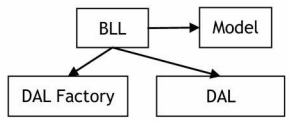

图 3　业务逻辑层的模块结构

BLL 是业务逻辑层的核心模块，它包含了整个系统的核心业务。在业务逻辑层中，不能直接访问数据库，而必须通过数据访问层。是通过接口模块 DAL Interface 间接来完成的。由于 BLL 与具体的数据访问逻辑无关，所以，业务逻辑层和数据访问层之间的关系就是松散耦合的。如果此时需要修改数据访问层的具体实现，只要不涉及到 DAL Interface 的接口定义的修改，那么业务逻辑层就不会受到任何影响。

3　三层结构在大豆专家系统中的应用

大豆专家系统为大豆生产提供一个辅助决策的技术支撑平台，它主要提供大豆栽培技术支持和生产决策支持数据(土壤肥力、病害、虫害和草害)，提供大豆资源数据库管理系统。系统根据用户提供的大豆叶片特征图像、大豆生长情况图像和大豆栽培过程中的问题，通过信息处理技术和专家系统的判断，为农户提供决策支持。该系统由多个部分组成，其中一部分通过 web 和用户之间进行交互，下面以系统 web 开发中的一个表为例说明三层结构在大豆专家系统中应用。大豆专家系统中有关大豆品种的数据表(Types)设计如图 4 所示。

说明	列名	可为空值	数据类型
ID列	id	否	unique identifier
品种名称	name	否	nvarchar(100)
品种来源	pzly	是	text
特征特性	tztx	是	text
产量表现	clbx	是	text
栽培要点	zpyd	是	text
适宜地区	sydq	是	text

图 4　Types 数据表设

针对表 Types 的数据实体类 Model 定义如下(代码中有省略)。

```
public class ModelTypes
{
    //字段
    private Guid id;
    private string name;
    private string pzly;
```


?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

```
……
//属性
public Guid ID
{
    get { return id;}
    set { id = value;}
}
public string Name
{
    get { return name;}
    set { name = value;}
}
public string Pzly
{
    get { return pzly;}
    set { pzly = value;}
}

//默认构造函数
public ModelTypes() {}
//重载构造函数
public ModelTypes(Guid id, string name, string pzly, string tztx, string clbx, string zpyd, string sydq)
{
    this.id = id;
    this.name = name;
    this.pzly = pzly;
……
}
}
```

数据访问层接口(interface)定义如下,其中的接口函数可以根据实际需要增减。

```
public interface ITypes
{
IList< ModelTypes> GetTypesList();//获取大豆品种列表
IList< ModelTypes> GetTypesList(string keywrods);//根据关键字搜索大豆品种信息
void Update(ModelTypes model);//更新数据
void Add(ModelTypes model);//添加数据
ModelTypes GetType(Guid guid);//得到某个大豆品种的信息
void DeleteType(Guid guid);//删除某个大豆品种
}
```

实现上述数据访问层接口,操作 SQL Server 类型的数据库的 SQL Server DAL 代码定义如下。

```
public class TypeDAL :ITypes
{
public IList< ModelTypes> GetTypesList()//实现接口的同名方法
{ 执行SQL 操作,完成指定的功能}
public IList< ModelTypes> GetTypesList(string keywrods)
{……}
……
public void DeleteType(Guid guid)
{……}
public void Update(ModelTypes model)
{ 执行SQL 操作,完成数据更新功能}
}
```

业务逻辑层 TypeBLL 类定义如下。

```
public class TypeBLL
{
private static readonly ITypes dal = new TypeDAL();//也可以通过工厂创建
public void Update(ModelTypes model)//实现数据更新
{
    dal.Update(model);//通过接口方法调用SQL Server DAL 实现数据库访问
}
……
}
```

表示层(Presentation Layer)用于处理人机交互。它主要的责任是处理用户请求,例如鼠标点击、输入、HTTP 请求、为用户显示信息等。表示层把用户的指令翻译、格式化后传送给业务层。表示层业务逻辑写在 ASP.NET 中的 aspx 或 aspx.cs 中。由于表示层只需调用业务逻辑层的代码,因此表示层的开发人员只需关心业务逻辑层的代码即可。又由于业务逻辑层将表示层和数据访问层分开,使得表示层和数据访问层之间没有编译依赖性,因此,可以使用两组人员分别独立开发、编译并调试,最后将两部分组装起来即可,大大提高了开发效率,缩短开发时间。此外,由于分层软件的模块化开发的特点,某层模块内的错误不会扩散到其他软件层次,使得软件维护过程变得相对简单。

4　结论

主要讨论基于.NET的三层结构及其在大豆专家系统中的应用。基于.NET 三层结构是目前 B/S 结构软件开发的方向。这种结构的软件具有开发效率高,不同模块结构之间的耦合度低,软件易于维护等特点。

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net