

ASP.NET三层架构的概念及关系分析

哈尔滨铁道职业技术学院 乔 冰

【摘要】 分层次体系结构是按层次组织软件的一种软件体系结构，其中每一层建立在低一层的软件层上。通常的ASP.NET三层架构就是将整个业务应用划分为：表示层（UI）、业务逻辑层（BLL）和数据访问层（DAL）。很好地实现了高耦合低内聚的思想。三层架构使得用户操作部分、逻辑操作部分和数据访问部分分开，每一层由相应的人员来完成，各层之间互不影响，使得应用系统的开发、升级、维护得到了大大简化。在种种优越性下，三层架构已经成为进行商业系统开发的首要选择。

【关键词】 三层架构；ASP.NET；表示层；业务逻辑层；数据访问层

选择合适的系统体系架构是一项影响整个应用系统设计的关键工作。使用三层架构可以使得系统的结构更加清楚，分工更加明确，有利于系统的分工合作和后期维护。B/S模式的三层结构是一种简单、成熟，并得到普遍应用的应用程序架构，它将应用程序结构划为分三个相对独立层，包括用户表示层、业务逻辑层和数据访问层。每一层只实现该层内相对独立的功能，而当任何一层发生变化时，只要保持层间接口关系不变，则其他各层都不会受到影响。三层结构是一种严格的分层协作，即数据访问层只能被业务逻辑层访问，业务逻辑层只能被表示层访问。用户通过表示层将请求传送给业务逻辑层，业务逻辑层完成相关业务规则和逻辑，并通过数据访问层访问数据库获得数据，然后按照相反的顺序依次返回，将数据显示在用户界面层。

在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构，微软推荐的分层式结构一般分为三层，从下至上分别为：数据访问层、业务逻辑层（又或成为领

域层）、表示层。

所谓三层体系结构，是在客户端与数据库之间加入了一个“中间层”，也叫组件层。这里所说的三层体系，不是指物理上的三层，不是简单地放置三台机器就是三层体系结构，也不仅仅有B/S应用才是三层体系结构，三层是指逻辑上的三层，即使这三个层放置到一台机器上。三层体系的应用程序将业务规则、数据访问、合法性校验等工作放到了中间层进行处理。通常情况下，客户端不直接与数据库进行交互，而是通过COM/DCOM通讯与中间层建立连接，再经由中间层与数据库进行交互。

一、系统各层次原理

三个层次中，系统主要功能和业务逻辑都在业务逻辑层进行处理。

1. 用户表示层（USL）

用户表示层也叫用户界面层，封装入机界面的所有表单和组件，是应用系统与系统用户之间的直接接口。主要用于显示由业务逻辑层动态传送过来的数据信息，结合使用相应的HTML标记和样式表定义来实现；用

户录入数据信息并通过简单的校验后，经由数据接口传送给业务逻辑层下载论文。

2. 业务逻辑层（BLL）

业务逻辑层主要完成对应用系统相关业务规则和逻辑的封装，在为用户表示层访问提供功能调用的同时，它又通过调用数据访问层所提供的功能来访问数据库。业务逻辑层主要根据系统设计的需要，通过构建系统的关键对象类，实现系统的大部分逻辑控制功能。

3. 数据访问层（DAL）

数据访问层只能被业务逻辑层访问，并且系统只通过它访问数据库来获取数据。当数据访问层根据业务逻辑层的要求与数据库进行交互时，主要完成从数据库中提取数据库记录的查询和插入、修改与删除数据库记录的更新数据功能。众所周知，数据库访问是动态信息管理应用系统中最频繁、最消耗资源的操作，所以必须优化对数据库的访问策略，以提高系统的性能和可靠性。B/S模式的三层结构是一种严格的分层定义，它首先将应用系统复杂的开发工作划分为相对

LPC2132是基于一个支持实时仿真和嵌入式跟踪的32位ARM7TDMI-S CPU，并带有64kB嵌入的高速Flash存储器，128位宽度的存储器接口和独特的加速结构使32位代码能够在最大时钟速率下运行。其内部集成了8路10位分辨率的A/D转换器、1个10位的D/A转换器、2个32位的定时器/外部事件计数器，支持在系统编程（ISP）和在应用编程（IAP），支持低功耗模式。

(2) 数据采集电路的设计

对蓄电池进行检测的关键在于对蓄电池参数采样的精确程度，因而采样电路设计得是否适当对整个系统至关重要。由驱动板输入的8路模拟信号经运算放大器后，经模拟开关选通后，再经电压跟随器，由LPC2132内置8路10位高精度的自校准模数转换器进行A/D转换。而温度测量是单独分配的一个采样通道，负责对环境温度的监控。各路蓄电池参数检测采用相同的采集通道分别采样，而不是设计每组一个采集通道，利用模拟开关或继电器进行切换采样。这样不但保证了数据采集精度一致，而且避免了因频繁切换测量通道造成的仪器工作不可靠。

(3) LCD显示及键盘模块

LCD选用TM240128A大屏幕液晶显示器，能显示15行20列中文信息，包括：巡检状态显示、单节电压查询、放电状态显示、单组

放电查询、故障显示及查询等。键盘模块采用程序查询方式，通过I/O口进行扫描查询，实现参数设置、状态查看、手动检测等功能。还能对运行参数如电池个数、电压标准值、电流报警值、容量标准值、放电时间等进行在线修改。

(4) 通讯电路

为方便存储和调用每次检测数据，本监测仪还设计了串行接口单元用来进行与上位机之间的通信，上位机可通过监控软件来监视设备的状态。LPC2132拥有两个符合16C550工业标准的异步收发器UART0和UART1，允许在串行链路上进行全双工的通信。使用了MAX232芯片进行RS232电平的转换，MAX232芯片内置一个电压转换器，只需输入5V电压就可得到RS232接口所需的电压。

4. 软件设计流程

软件设计所需要完成的功能主要包括系统初始化（包括系统自检、相关外围设备的初始化）、主要参数设置（包括蓄电池的性能参数、通讯参数以及测量参数）、键盘和LCD液晶显示管理、A/D转换、中断服务、参数修正、串行通讯等程序，整个参数检测过程的流程如图3所示。

5. 结束语

不同环境下的实践证明，采用32位ARM-

7TDMI-S LPC2132为核心的蓄电池在线监测仪智能化程度高，测量准确，能及时发现蓄电池组存在的早期故障电池，准确判断电池当前的运行状况，能有效地对蓄电池进行实时在线检测，及时报告故障信息，友好的界面使得用户能方便地察看系统运行的实时参数，通过设置权限，可以在线修改系统参数和进行控制，取得了较好的应用效果。

参考文献

- [1] 桂长清. 阀控铅酸蓄电池组的均匀性[J]. 通讯电源技术, 2001(3).
- [2] 周立功. ARM嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2005.
- [3] 李芳培. 铅酸蓄电池健康诊断中的内阻测量法研究[D]. 南京航空航天大学, 2009.
- [4] 杨培君. 基于LPC2368的低压配电智能监控系统设计[D]. 西安电子科技大学, 2011.
- [5] 郭亮. 蓄电池在线监控系统的研究[D]. 山西大学, 2011.

本文缘由于2011年湖南铁道职业技术学院院级课题“基于ARM芯片的多路蓄电池在线监测系统的设计与实现”（项目编号：K201126）。

作者简介：刘红兵（1975—），男，湖南株洲人，大学本科，学士，讲师，主要从事应用电子技术及电子信息技术的教学。

简单的小分块，然后在每一层中只实现系统相应层的功能设计，层间的交互由相邻层对应的功能模块进行调用，信息传递只由接口进行传送。业务实体组件体现的是现实生活中的业务数据，而数据访问逻辑组件则是在数据库中检索数据或把业务实体数据保存到数据库。

二、系统各层次职责

1. 用户表示层 (USL)

UI (User Interface) 层的职责是数据的展现和采集，数据采集的结果通常以 Entity object 提交给 BL 层处理；

Service Interface 侧层用于将业务或数据资源发布为服务 (如 WebServices)。

2. 业务逻辑层 (BLL)

BL (Business Logic) 层的职责是按预定的业务逻辑处理 UI 层提交的请求。

(1) Business Function 子层负责基本业务功能的实现；

(2) Business Flow 子层负责将 Business Function 子层提供的多个基本业务功能组织成一个完整的业务流。(Transaction 只能在 Business Flow 子层开启)。

3. 数据访问层 (DAL)

Resource Access 层的职责是提供全面的资源访问功能支持，并向上层屏蔽资源的来源。

(1) BEM (Business Entity Manager) 子层采用 DataAccess 子层和服务 Access 子层来提供业务需要的基础数据/资源访问能力。

(2) DataAccess 子层负责从数据库中存取资源，并向 BEM 子层屏蔽所有的 SQL 语句以及数据库类型差异；DB Adaptor 子层负责屏蔽数据库类型的差异；ORM 子层负责提供对象—关系映射的功能；Relation 子层提供 ORM 无法完成的基于关系 (Relation) 的数据访问功能。

(3) Service Access 子层用于以 SOA 的方式从外部系统获取资源。Service Entrance 用于简化对 Service 的访问，它相当于 Service 的代理，客户直接使用 Service Entrance 就可以访问系统发布的服务。Service Entrance 为特定的平台 (如 Java、.Net) 提供强类型的接口，内部可能隐藏了复杂的参数类型转换。

(4) Config Access 子层用于从配置文件中获取配置 object 或将配置 object 保存倒配置文件。

三、系统各层次规则

1. 系统各层次及层内部子层次之间都不得跨层调用；

2. Entity object 在各个层之间传递数据；

3. 需要在 UI 层绑定到列表的数据采用基于关系的 DataSet 传递，除此之外，应该使用 Entity object 传递数据；

4. 对于每一个数据库表 (Table) 都有一个 DB Entity class 与之对应，针对每一个 Entity class 都会有一个 BEM Class 与之对应；

5. 有些跨数据库或跨表的操作 (如复杂的联合查询) 也需要由相应的 BEM Class 来提供支持；

6. 对于相对简单的系统，可以考虑将 Business Function 子层和 Business Flow 子层合并为一个；

7. UI 层和 BL 层禁止出现任何 SQL 语句。

四、系统各层次优缺点

1. 三层结构的优点

(1) 提高程序的可维护性：三层架构层次分明，开发人员可以只关注整个结构中的其中某一层；

(2) 可以并行开发，提高了开发的进度；

(3) 提高系统的安全性：可以降低层与层之间的依赖；

(4) 提高系统的扩展性：三层结构最大的好处是在扩展应用系统时，改动的部分不会影响到大局，比如要给这个系统增加 Oracle 数据库的功能，只需要改动数据层就可以了，其他层保持不变。

2. 三层结构的缺点

(1) 降低了系统的性能。如果不采用分层式结构，很多业务可以直接造访数据库，以此获取相应的数据，如今却必须通过中间层来完成。

(2) 有时会导致级联的修改。这种修改尤其体现在自上而下的方向。如果在表示层中需要增加一个功能，为保证其设计符合分层式结构，可能需要在相应的业务逻辑层和数据访问层中都增加相应的代码。

(3) 增加了开发成本。“三层结构”开发模式，不适用于对执行速度要求过于苛刻的系统，如在线订票、在线炒股等。它比较适用于商业规则容易变化的系统。

五、ASP.NET 三层架构的实现

1. 数据访问层

当开始着手实现程序时，首先要做的便是设计数据库。而表示层要做的便是应用面向对象中类对象相应知识，设计相应的类和方法来实现对数据库的操作。ASP.NET 数据访问层的代码都放到特定文件夹 App_Code 中，这样在网站初始化时，此特定文件夹中的代码会自动进行编译，便于检查出错误和提高程序应用时的效率。一般对应数据库中的表都要建立实体类，封装其变量及属性，其中变量对应数据库中字段。数据访问层建立于系统中模块相关的类，然后将系统中对应的功能在类中以方法的方式完成。用这些方法为业务逻辑层提供服务。为了更好地完成对数据库的操作，可以将一些对数据库通用的操作封装成类，然后在数据访问层中进行调用，其中最具有代表的是微软提供的 SQLHelper 类。

2. 业务逻辑层

业务逻辑层为介于表示层和数据访问层之间处理逻辑关系的一层。在业务逻辑层里建立相应的数据访问层相应类实例，然后调用数据访问层的方法。由于在数据访问层类方法中已经近似完成了对应的业务逻辑层

所要完成的功能，但业务逻辑层仍是不可缺少的。所有的业务上的控制，如判断数据的有效性，都逻辑层完成。有时候逻辑层只是简单的传递调用的数据访问层的方法的返回值。但是保留此层对以后进行维护有很大的帮助。如：public static int majorNameToID (string name) {return M_MajorDAL.majorNameToID (name); } 以上代码便是仅仅返回了数据访问层类的静态方法的返回值，返回了专业名对应的专业 ID。

3. 表示层

表示层只处理接收数据、显示数据和判断输入数据的有效性等问题。ASP.NET 中表示层由数量不等的网页文件组成，这些网页各自独立，相互之间存在不同程度的关联。表示层为面向用户的部分，其好坏直接影响到用户的评价。表示层整体框架做出来后，其页面修饰便由网站美工来完成。

以上介绍了 ASP.NET 技术的三层结构模式。通过介绍分层规划，体现了利用 ASP.NET 技术编写三层结构的清晰逻辑，并有效地实现了系统中各功能层的相对独立，使系统具有更强大的灵活性、可扩展性和可维护性；而且系统分层管理使分级分布部署成为现实，可极大地提高系统的性能和安全性。

参考文献

- [1] 谭政, 邢剑宽, 郑翔. 软件体系结构 [M]. 清华大学出版社.
- [2] Paul Sarknas. ASP.NET 电子商务高级编程 [M]. 清华大学出版社.
- [3] 孙琳俊, 陈松. 新手学 ASP.NET 3.5 网络开发 [M]. 电子工业出版社.

作者简介: 乔冰 (1982—), 男, 山西榆次人, 硕士, 现供职于哈尔滨铁道职业技术学院。