

# 基于.NET混合模式的铁路水电收费系统优化设计

韩磊, 史百战, 邱建东

(兰州交通大学机电技术研究所, 甘肃兰州 730070)

**摘要:**在现行的铁路水电收费系统效率低下且不能满足收费处理中新需求的前提下,开发了本系统。它基于.NET开发平台,利用ASP.NET,ADO.NET,C#等技术的新特点结合相适应的算法,后台数据库采用SQL SERVER与之无缝连接,实现了C/S与B/S混合模式的水电合一收费系统。现场应用表明,利用以上技术后新系统处理速度加快,操作便捷,很好地取得了预期的效果。

**关键词:**C/S;B/S;.NET;反射技术;模板列

中图分类号:TP311.13

文献标识码:A

文章编号:1673-629X(2007)08-0226-04

## Design of Double-Patterns Water-Electricity Charging System in Railroad Based on .NET

HAN Lei, SHI Bai-zhan, QIU Jian-dong

(Mechatronic Technology Research Institute, Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract:**The current water-electricity charging management system can not satisfy the requirements of dealing with water-electricity information among railroad department. The system based on .NET development platform and used new characteristics of ASP.NET and ADO.NET, C# and other technology, combining with the relevant algorithm. The database background used SQL Server to interconnect with the whole system seamlessly. The system implement the application of hybrid model(C/S and B/S) in the water-electricity charging management system. The paper describes the various functional modules and the key technology that used. Field application showed that use the technology above can improve the processing speed a lot. The whole system have convenient operation and achieve the desired good results.

**Key words:**C/S;B/S;.NET;reflection;templatecolumn

### 1 开发的可行性和必要性

本系统是在库尔勒供电段旧收费系统不能有效支持目前工作的情况下提出的。原有处理系统功能针对性不强,可扩展性差,使得一些收费过程中出现的矛盾无法靠系统自身升级扩展来解决,并且随着收费工作的不断发展,也产生了很多新的工作需求。具体问题包括:随着用户增加但系统运行十分缓慢,影响了正常收费;数据管理死板无法根据实际工作需求灵活收费,影响效率;计划增加预收款处理,及用户通过互联网查询水电交、欠费情况等;及时通过留言板响应用户的水电使用要求等。随着编程工具及计算机新技术的不断出现,为现有应用系统的优化、扩展提供了更好的实现平台,同时铁路信息化改造也要求行业的规范化,

基于此开发了混合模式的铁路水电收费系统。系统在库尔勒供电段的应用表明其极大提高了收费效率,完善了用户的实际需求。

### 2 系统总体设计

#### 2.1 系统结构设计

C/S模式与B/S模式是目前两种运用最为广泛的系统体系模式,C/S即客户机服务器模式,而B/S为浏览器服务器模式。C/S模式与B/S模式有着各自结构上的优缺点:C/S人机交互灵活方便,并可以实现数据的快速传送和安全存储,但在远程数据管理方面存在不足;B/S模式的系统易于集成和扩展,可以提供远程的数据服务和管理。水电收费系统采用以C/S体系结构为主、B/S体系结构为辅的混合技术框架,C/S构架完成收费人员各种复杂的管理操作,实现强大的数据维护、统计分析、报表打印等功能。对于客户来说,他们的目的是查询或对水电供应情况的反映和监督,

收稿日期:2006-11-13

作者简介:韩磊(1974-),男,陕西人,硕士研究生,研究方向为计算机应用;史百战,兰州交通大学副校长,教授,研究方向为信息仿真。

利用 B/S 构架技术实现网上客户对预交款余额及欠费的查询。系统结构图如图 1 所示。

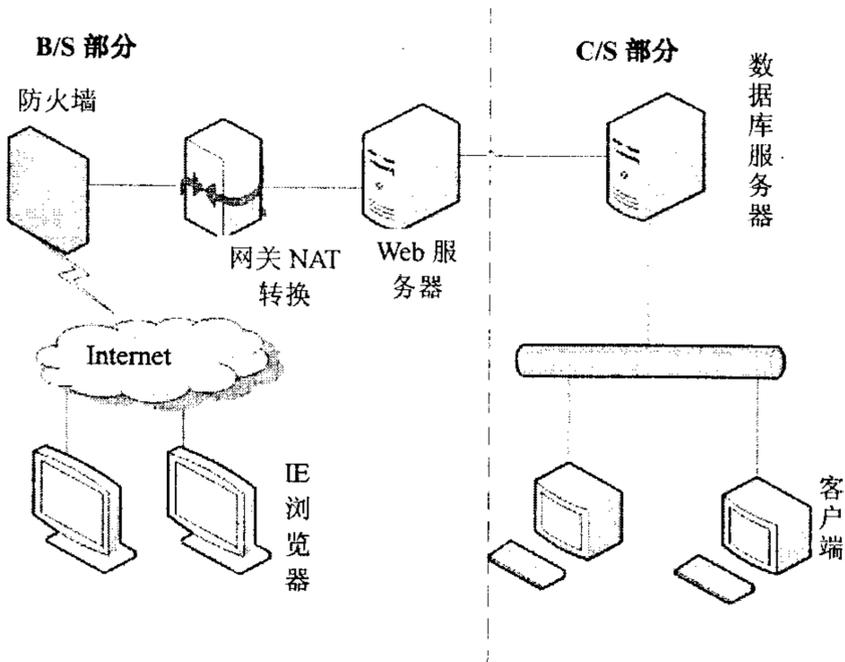


图 1 系统结构图

### 2.2 开发环境选择

本系统采用 .NET 平台,.NET 可以支持高性能桌面应用程序,以及企业规模的 WEB 应用程序的开发。采用了其中的 C#,ASP.NET,ADO.NET 来完成代码编写。后台数据库选用微软的 SQL SERVER2000 实现了代码与数据的无缝连接。

在 C/S 模式的开发工具中,客户端采用 VS.NET 首推的 C# 进行界面的开发,C# 是 VS.NET 中引入的全新编程语言。作为 C 的派生语言,它具有简单、先进和完全面向对象的特点。.NET 通过项目模板、设计器、属性页码、WIFORM 窗口,及大量第三方控件为它做了强大支持。

在 B/S 模块开发中,采用 ASP.NET 技术,它可用来创建动态的 WEB 页,所有 ASP.NET 代码(包括服务器脚本)都经过编译,具有可提供强类型、性能优化和早期绑定及其他优点。代码一旦经过编译,运行库会进一步将 ASP.NET 编译为本机代码,从而提供增强的性能,执行时比 ASP 的直译方式快很多,另外,ASP.NET 也提供快取(Cache)的能力,有效地缩短服务器的应答时间。ADO.NET 是 Microsoft 提供的断开式数据存取解决方案,适应于基于 WEB 的分布式软件应用环境。与传统的 ADO 技术相比,主要体现在基于前者的应用程序只在检索或更新数据时才连接数据库,完成事务处理后,即关闭数据库;而后者必须保持数据库的持续连接。前者得到的数据以 DataSet 数据集的形式存储,以 XML 的格式传输。

### 2.3 系统设计目标

(1)可扩展性:有了新的需求,新的性能可以容易

地添加到系统中,并且不影响现有性能。

(2)可修改性:系统某一部分的代码需要修改时不会破坏系统的现有结构,也不会影响到其他部分。

### 2.4 系统的安全设计

采用 C/S 与 B/S 混合模式中的安全问题主要表现在 B/S 模式上,由于 B/S 部分与公网有接口,系统为了达到系统资源的安全、数据安全和通信安全的目的,在系统设计时,B/S 部分使用了三级安全机制以防止信息的泄漏和非法用户对数据的修改和破坏。

(1)设置了防火墙作为水电收费管理系统的第一级防线,隔离了外界对服务器的直接访问,有效地防止“黑客”的入侵。

(2)在防火墙和 WEB 服务器间设置网关,通过 NAT 转换屏蔽访问其他端口的服务,只对访问 80 端口的 WWW 服务开放,保证了服务器数据的安全。

(3)在用户登录时,通过对用户密码的验证,保证不同用户的访问权限和服务权限。

(4)在 C/S 部分首先对用户进行了权限分配,权限分配的单位是菜单的命令项,每个不同权限的用户登录时动态生成各自不同的系统菜单,另外考虑了基于数据库的安全性,以确保数据库中表的完整性和正确性,将数据表划分级别以保证用户按权限访问不同的级别,对 App.config 中字符连接密码采用哈希加密以防止通过 SQL SERVER 直接注册登录数据库。

### 2.5 系统功能模块介绍

功能模块见图 2(只给出电的部分)。

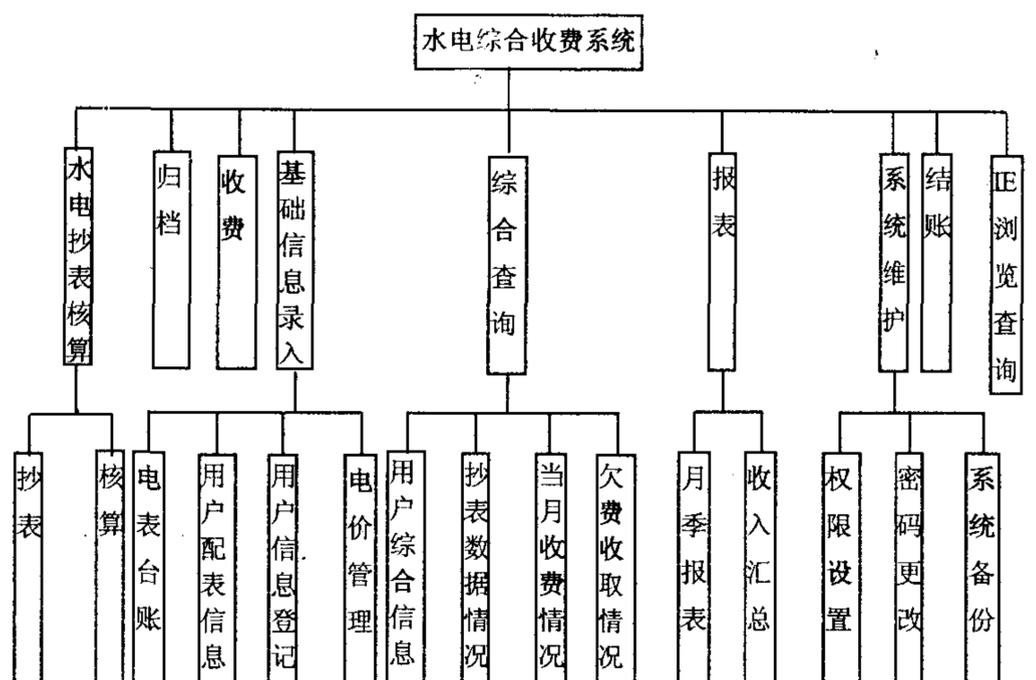


图 2 系统模块结构图

(1)基础信息录入。

用户信息登记:对用户信息进行录入、修改、删除;用户配表信息:该功能是对每个用户的编号、姓名根据电表台账资料进行关联查询、修改、插入和删除操作;电表台账:电表基本信息管理;电价管理:对所有的用电单价进行管理,可以修改电单价,增加新的单价,删

除作废的单价。

#### (2) 水电抄表。

抄码输入:抄码录入人员在此完成抄表数据录入工作;抄码修改:为了维护数据的正确性和稳定性,本月抄码可以修改,底码修改需要有此权限,可以通过权限设置授权后才能修改。

#### (3) 核算发行。

为了保证数据的一致性,发行后的数据不能再抄表,开始进入收费程序,系统又考虑了如果发行后的某一节点下大批用户需要重新抄表的情况,由此增加了回退功能,确保能在回退后再次抄表。

#### (4) 归档。

把当月数据归入历史数据库,初始化下月抄表数据表,并把新增用户纳入到抄表及收费表单。对以前用户的历史交费记录进行查看,分析、监视用户用电量及交费情况,根据预交款和交款情况形成水电欠费记录,预收款冲账:对余额大于水电费的用户,系统自动从用户余额中扣除当月水电费。

#### (5) 用户缴费。

用户缴费:按表卡号或按用户名或按地址多途径查询用户,有模糊查询功能,收取水电费,打印发票。收费时用户可以预交款,且当收费员无零钱可找时,系统会自动将多交纳的金额转入本月余额,避免了找零的麻烦;当用户来交费时,而该用户有欠费时系统会出现一个窗口,提示该用户有欠费。

#### (6) 报表。

收入汇总;月季报表:该功能按季统计水电费,打印报表。

#### (7) 系统维护。

设置密码:每个收款员可以在这里修改自己的密码;权限设置:系统管理员可以在此设置每个收款员的操作权限;数据备份:该功能可以方便地进行完全备份和差异备份;数据恢复:该功能把备份出来的数据恢复到后台数据库。

#### (8) 用户 IE 查询。

用户水电信息清单查询,包括预交款余额、欠费情况、历史清单;BBS 公告板,用于供电信息的通知及用户留言。

#### (9) 系统结账。

包括登录结账,时段结账,按发票号段结账,方便收款员在结账时与其实收对账。

### 3 系统的技术特点

(1)开发了大量的公共类,设计了部分接口。

包括返回数据连接,返回数据表,保存系统登录信

息的类、常用函数类等,尽量做到类间的低耦合,易于类的复用、修改、扩展。

(2)利用 .NET 反射技术动态生成系统菜单<sup>[1]</sup>。

反射是 .NET 中的重要机制,通过反射,可以在运行时获得 .NET 中每一个类型的成员。动态生成便于系统功能模块的扩充,如有新模块只需将模块名加入表中即可读出,为用户权限分配提供了方便的条件。

首先根据菜单树状结构规划了其节点间的存储结构,以二维表的形式存于数据库表中,然后当用户登录时根据权限表逐级读入树状关系的菜单,动态生成时利用反射得到各命令项所对应的 WINFROM 类。以下代码为利用反射得到各命令项所对应的 WINFROM 类:

```
private void CommandSelected(object sender, System.EventArgs e)
{
    //强制转换 sender 为 MenuCommand 类型提取其中包含的命令项信息
    Crownwood.DotNetMagic.Menus.MenuCommand Clicked =
    (Crownwood.DotNetMagic.Menus.MenuCommand) sender;
    string x = Clicked.Text;
    SqlConnection l_conn = SDSFS.DB.Oper.DBConn("gd-mis");
    string l_str = "select label from public.-bm.-menu where mc = '" + x + "'";
    SqlDataReader l_dr = SDSFS.DB.Oper.DBReader(l_conn, l_str);
    l_dr.Read();
    //命令项对应的窗体名从表中取出
    string l_formName = l_dr["label"].ToString();
    l_dr.Close();
    //利用反射根据文本 l_formName 得到各命令项所对应得 WINFROM 类名
    System.Windows.Forms.Form
    NewWinForm = (System.Windows.Forms.Form) Activator.
    CreateInstance(System.Type.GetType("SDSFS." + l_formName + ""));
    NewWinForm.MdiParent = this;
}
```

(3)优化树的遍历算法加快检索速度<sup>[2,3]</sup>。

当抄表时工作人员通过树来检索用电区和分区及用电客户时,子节点(用电分区)较多,利用传统指针作为储存结构的遍历算法会因访问数据库服务器频繁而导致系统运行缓慢,系统中利用树的层次特点采用前缀相同后缀逐级递增的编码表示节点关系作为其存储结构,一次性将数据表利用 ADO.NET 读入到内存的数据集 DATASET 中,再进行遍历的办法,检索速度加快,基本上没有延迟。部分代码如下:

```

static void nodeGetChild(DataTable L_Get, string fatherbh)
{
foreach(DataRow dr in myDataTable.Rows)
{
string path = dr[nodeName].ToString();
string text = dr[nodebh].ToString();
//找到子节点
if ( (path.Length == fatherbh.Length + 2) && path.
StartsWith(fatherbh))
{
DataRow MyDR = L_Get.NewRow();
MyDR["sbbm"] = text;
L_Get.Rows.Add(MyDR);
nodeGetChild(L_Get, path);
}
}
}

```

#### (4) 合理使用存储过程。

水电费抄表后要进行核算,会对多个表进行批量处理,这时考虑使用传参到数据库服务器用存储过程来提高效率,但是实际使用时当核算节点太多时逐一传送节点编号又加大了系统开销,反而使处理变慢,本系统把需要多次传参的操作单独拿出来在 SQL 语句中通过条件表示为集合 where in (参数列表)一次处理,其他传参少但一次处理多个表的使用存储过程,二者结合使用后速度提高了 2/3。

#### (5) 使用多线程。

系统归档时调用存储过程处理工作周期长,独占资源无法知道当前处理进度,为了使得程序更加人性化,此时采用了多线程来改变调用存储过程的进程独占 CPU 资源的情况,一个进行调用存储过程,另外一个显示进度,二者同步进行实现了对当前进度的实时监控。

#### (6) 数据的备份和恢复。

数据的备份和恢复重要且复杂,本系统为用户提供了良好易操作的界面,采用了一星期一次完全备份、每天一次差异备份的办法,既使数据有了安全保障,又保证了备份数据存储量不至于增长速度太快。

(7) B/S 部分采用模板列页面多级调用时进行多参数传递<sup>[4,5]</sup>。

在 B/S 模式下实现了类似于 C/S 的页面逐级调用返回。

下面是利用模板列进行 URL 多个变量参数的传递方法的代码:

```

<ItemTemplate>
<asp:HyperLink runat="server" Text="查看"
NavigateUrl='<%# "usepowerdetail.aspx? person=" +
DataBinder.Eval(Container, "DataItem.person") + "&date="
+
DataBinder.Eval(Container, "DataItem.monthid") + "&url=
userpowerlist.aspx"% >'
ID="Hyperlink_detail" NAME="Hyperlink_detail">
</asp:HyperLink>
</ItemTemplate>

```

## 4 结束语

通过算法的优化、方法与新技术的巧妙组合弥补了原系统功能上的不足,提高了处理速度,精简了部分无用的功能,使本软件的行业针对性更强,又通过 B/S 部分如 IE 浏览查询和 BBS 留言,增加了与用户间的沟通,提高了服务响应速度,使得收费系统更加人性化。本系统现在库尔勒供电段试用,它的应用将极大提高铁路供电段的工作效率。

#### 参考文献:

- [1] 甄 镛. .NET 与设计模式[M]. 北京:电子工业出版社, 2005.
- [2] 罗宾逊,内格尔. C# 高级编程[M]. 李敏波 译. 北京:清华大学出版社, 2005.
- [3] 王晓东. 计算机算法设计与分析[M]. 北京:电子工业出版社, 2001.
- [4] 章立民. 用实例学 ASP.NET[M]. 北京:电子工业出版社, 2005.
- [5] Sussman D, Homer A. ASP.NET 高级编程[M]. 北京:清华大学出版社, 2002.

(上接第 225 页)

- Part II: Problem Statement[J]. Signal Processing, 1991, 24 (1): 11-20.
- [2] Cichocki, Unbehauen R. Robust Neural Networks with On Line Learning For Blind Identification and Blind Separation of Source[J]. IEEE Trans on Circuit and Systems, 1996, 43 (11): 894-906.
- [3] 吴小培, 李晓辉, 张道信. 多变量数据分析及应用研究[J]. 电子测量与仪器学报, 2004, 18(1): 51-56.

- [4] Lee T W. Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources[J]. Neural Computation, 1999, 11(2): 409-433.
- [5] 杨福生, 洪 波. 独立分量分析的原理与应用[M]. 北京:清华大学出版社, 2006: 97-101.
- [6] Hyvarinen A. A fast fixed-point algorithm for independent analysis[J]. Neural Computation, 1997, 9: 1483-1492.