

基于 B/S 的开放式工资管理系统的设计及实现

陈晓红 赵燕锡 刘红霖 (长沙中南大学商学院 410083)

摘要: 针对企业管理模式集中化、开放化的趋势以及工资管理系统中工资计算和费用分配业务规则灵活多变的特点, 从应用模式和应用程序两个方面分析和设计了一种基于B/S的开放式工资管理系统, 并给出了关键实现技术。

关键词: B/S 开放式 Windows DNA 工资管理系统 数据库

1 引言

随着互联网的飞速发展, 企业商业模式开始转向电子商务, 其管理模式也从分散式管理转向集中式管理, 从封闭式管理转向开放式管理。在新的管理模式下, 一方面由于企业所面临的环境的变化在加快, 其业务规则更趋于动态、灵活、多变, 另一方面企业又希望通过互联网进行网络协同工作, 提高工作效率和资源分配效率。我们在为一家商业企业开发工资管理系统时, 该企业对系统的开放性要求非常高。

该企业是一家主要从事有色产品贸易的企业, 规模较大, 发展非常快, 且各个办事处与总部相距甚远。因此, 该企业既要求工资项目、工资计算公式和工资费用分配方式能灵活设置, 又要求应用能支持远程操作, 其需求可概括如下:

- (1) 可以灵活设置工资项目以及工资项目之间的计算公式;
- (2) 可以灵活设定工资费用的分配方式;
- (3) 按照计算公式核算工资, 并生成工资单;
- (4) 按照设定的方式进行费用分配, 并自动生成凭证;
- (5) 自定义报表并输出;
- (6) 可以远程查询工资。

2 系统设计

2.1 设计思路

就应用的开放性来讲, 可以将其分为应用模式的开放性和应用程序的开放性。前者是指应用具有灵活的体系结构, 伸缩性好, 便于根据用户的需求

对系统进行扩展; 后者是指业务规则是开放性的, 即业务规则不是固定不变的, 可以由用户通过应用程序修改^[1]。

目前应用模式已经发生了很大的变革, 从原有的单层应用模式(主机终端模式)和传统的C/S(Client/Server)两层模式迅速向多层扩展, 多层应用的具体实现可以分为两种:

(1) 将传统的C/S应用程序发展成多层应用程序, 以胜任与企业相适应的各种任务。

(2) 利用Internet/Intranet技术, 构建B/S(Browser/Server)模式的Web应用程序。我们考虑到B/S的“瘦”客户端、良好的开放性以及用户的远程查询的需求, 故采用了B/S模式。

开放式应用程序的实现方法多种多样, 我们的设计思路是将业务规则结构化分解, 然后存储到数据库表中, 尽量避免将易变的业务规则封装到应用程序中, 这样通过修改表中的数据就可以实现规则的变化^[1]。

2.2 应用体系结构

为了保证应用模式的开放性, 我们采用了

Windows DNA (Windows Distributed interNet Architecture) 平台。DNA平台分为三个层次, 即外部表现层、事务逻辑层和数据层, 外部表现层主要是人机交互的界面, 其解决方案包括基于页面和基于EXE程序两种, 前者是指客户机不是装载整个应用程序, 而是应用程序的页面, 后者指需要客户机的一些功能, 这些应用程序是可执行的, 可以安装在客户机上^[3]。事务逻辑层是应用程序的脉搏, 在这一层主要按照事务处理规则和工作流程处理数据, 故该层需要编制大量的程序。通过这些程序去进行大批量处理, 事务支持, 大型配置, 信息传送, 该层主要包括三项服务: COM+, MSMQ和IIS, 通过这些服务, 应用程序可以以COM的形式进行扩展。最后是数据层, 这是资源管理层, 与事务逻辑层相比, 没有数据的处理, 而是定义了大量数据的管理任务, 这一层常用的对象模型是OLE DB对象模型, OLE DB是高速、低级而不受限的。在具体应用程序的开发过程中, 常用其简化模型ADO (Active Data Objects) 来进行数据处理。

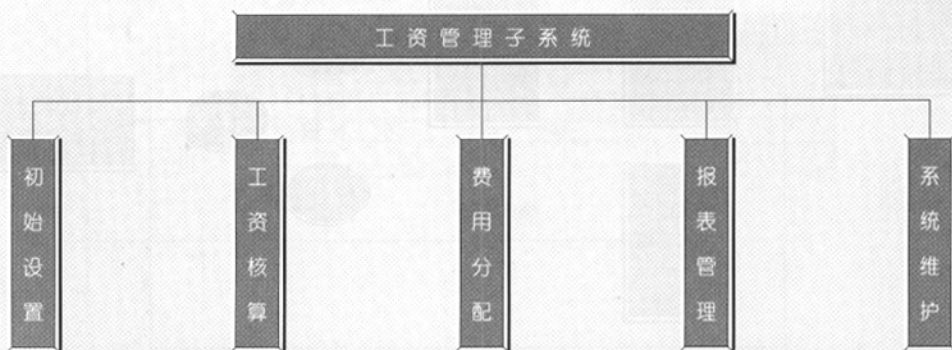


图1 工资管理系统功能结构图

DNA 多层次的分布式应用程序模型提供了一系列服务和工具, 这些服务和工具把用户终端、Web 服务器和数据库服务器等联合在一起, 从而为我们提供了以页面的形式构建用户界面, 以COM 的形式进行扩展的一种开放式的应用体系结构。

2.3 功能模块设计

按模块划分的原则, 以功能划分的模块聚合度高, 藕合度低^[2]。因此, 首先我们抽象出工资管理系统中的基本操作(如工资项目的增加、修改和删除), 并将相类似或相关联的操作按照其功能特性进行归并形成层次的功能结构, 在这个功能结构中, 底层或较低层的操作其业务规则相对来说是固定的, 我们将其封装在程序中, 当用户利用这些基本的操作构建其应用时, 我们将用户定义的规则存储到数据库的表中, 系统运行时可以访问相应的表获取规则进行业务计算, 而用户需求发生变化时, 只需要修改数据库中的规则。如图1是工资管理系统的功能结构图:

(1) 系统设置, 包括工资项目设置、税率设置和计算公式设置。

(2) 工资变动, 提供当月工资的录入、修改(包括成批修改), 所有工资数据的查询以及重算工资的功能, 重算工资主要是指用户在当月录入工资数据之后又对初始设置进行了修改时引用新的业务规则重新计算工资。

(3) 费用分配, 主要包括职员类别设置、分配

科目设置、费用分配和重新分配四部分, 职员类别设置是对职员的分配类别(如管理人员、业务人员等)进行设置; 分配科目设置是对每一类别职员的某一工资项目(一般是应发工资)费用分配的科目进行设置; 费用分配是指按照相应的类别和科目设置进行分配; 重新分配是指对本月的费用分配进行取消, 从而使得系统恢复成费用分配之前的状态(在未进行费用分配的时候可以进行工资数据的录入工作)。

(4) 报表管理, 包括报表设置和报表输出两大部分, 系统默认提供工资单、费用分配单两种报表, 其中费用分配单不可以修改和删除, 工资单可以修改但不可以删除, 同时用户可以通过报表设置自定义报表, 报表输出主要用来查询和打印相应报表。

(5) 系统管理, 提供口令修改、切换用户等功能。

2.4 数据库设计

按照关系型数据库设计原则, 为每一个实体建立一个表, 表的字段由该实体中与系统有关的属性确定, 系统中主要的表包括工资项目表(计算公式为工资项目的一种属性)、税率表、职工工资表、职工类别表、分配科目表、职工类别对应表、费用分配表和报表设置表, 另外还有一些用户系统管理的表如用户表、角色表和权限表等。

为了保证系统的实现, 我们在主要的表中自定义了一些规则, 以工资项目表为例, 我们将工资项

目分为固定项目、变动项目和统计项目三种类型, 固定项目每月的数据基本不变, 变动项目数据每月都变, 统计项目每月的数据需要通过设置的公式进行计算, 同时我们还设一个标志字段用以标识对工资项目公式的操作和计算的先后:

0 表示计算公式不能修改和删除, 且在计算工资的时候要计算在最后的项(如实发工资);

1 表示计算公式不能修改和删除, 且在计算工资的时候要计算在次后的项(如税款);

2 表示计算公式可以修改但不可以删除(如应交税所得额);

3 表示计算公式可修改和删除, 且计算工资的时候要计算在前的项, 主要指用户定义的工资项目, 前三类工资项目为系统预设的项目。

3 系统实现

3.1 开放式应用模式的实现

基于B/S模式的数据库开发技术包括很多种, 传统的有CGI(Common Gateway Interface)、ISAPI、NSAPI等方式, 但CGI执行效率低, 而ISAPI、NSAPI技术又太复杂, 目前常用的是ASP方式, ASP是Microsoft Active Server Pages的缩写, 它是一个服务器端脚本环境(Web Server Scripting Environment), 可以用来创建、运行交互式高性能的Web Server应用程序^[4]。在DNA平台上, ASP程序一般位于事务逻辑层, 它会通过Web服务器(在这里为IIS)生成表现层上的界面代码, 在这种情况下, 处理技术分布在表现层和事务逻辑层之间, 如图2是DNA平台上ASP的处理模式:

在具体的执行中, IIS负责缓存来自特定客户的HTTP请求, 该请求由HTTP转化为可以访问COM环境的脚本ASP请求, 通过COM对象模型, 从而实现与后台DB、各类组件进行通信, 同时, COM模型提供事务服务, 以确保同一事务中处理的一致性。

3.2 开放式应用程序的实现

(1) 公式设置, 工资核算的开放性表现在用

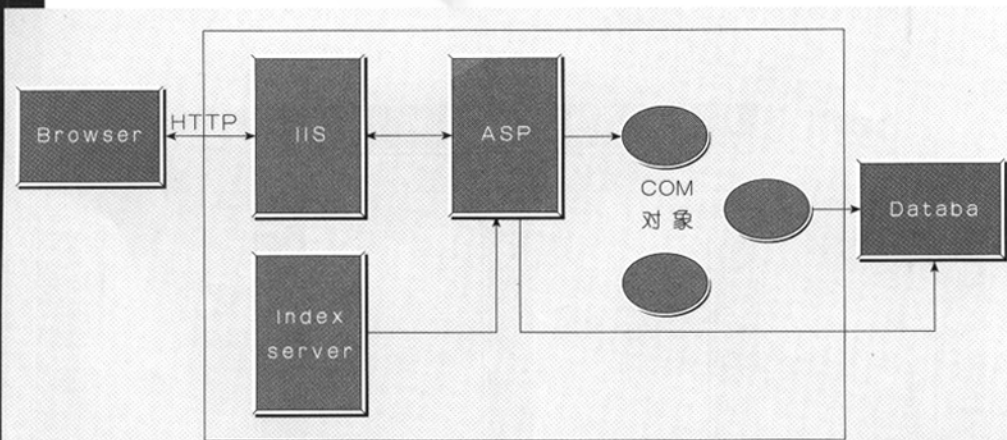


图2 基于DNA的ASP处理模式

用户可以自定义工资项目、用户可自定义统计项目与其他工资项目之间的计算公式及用户可以在当月多次核算工资。对于工资项目，我们提供输入界面让用户进行设置，用户可以定义项目名字、项目类型（变动项、固定项和统计项）、是否生效等属性，系统在用户提交后将相关信息保存到表中，而对于统计项目的计算公式，我们将其从应用程序中抽象出来，并放置到数据库中。计算公式是某统计项目与其他项目（包括其他统计项目）之间的一种函数关系式，在实现中我们可以将其看作一个字符串。用户通过工资设置面板（如图3）进行公式的设置，系统根据用户的选择，将其操作转换为一个由工资项目和算术符号组成的字符串，然后保存到表中。如个人所得税的计算公式字符串为 $(XM1-800) \times SL-sskcs$ ，在这里XM1指应发工资，SL指税率，sskcs指速算扣除数。

(2) 工资计算。设置好项目、税率和计算公式之后，就可以根据这些设置进行计算。首先，我们通过循环利用表格和表单构造输入界面，纵向显示职工姓名，横向显示各工资项目，并使每一个表单的输入框的名字为职工编号、*和工资项目组合，如编号为1001的职工的应发工资(xm1)所对应的文本输入框的名字为“1001*xm1”。工资输入和修改界面如图4。

在处理页面，对于固定项目和变动项目我们利用request对象通过for循环获得职工各工资项目的数值，对于统计项目我们定义了以下五个函数，然后通过这五个函数来计算其值：

function read(gsmes,zyid,rq) 分解计算公式，参数为公式字符串、职员编号和日期，返回值为数字或者运算符

function inysf(w) 判断是否是运算符，参数为为一字符

function yxj(aa,bb) 判断运算符优先级，参数为两个运算符，优先规则基本上按四则运算规则，但只有同边括号和#号两两比较时才返回等于。

+、-、×和÷两两比较时返回大于。

function operate(shu1,yszb,shu2) 执行计算操作，参数为数量1、运算标志和数量2，返回计算结果。

function jstjxgs(gsmes,zyid,rq) 计算统计项，参数为公式字符串、职员编号和日期。该函数调用以上四个函数，最后返回统计项的计算结果。其关键代码如下：

```
.....
调用read函数，取出公式的单元,返回值为数字或者运算符号
w=read(gsmes,zyid,rq)
while not((w= '#')and(ysf(ysfop)= '#')) ysftop
为运算符堆栈的栈顶，#表示公式的结束
if inysf(w)=0 then 如果是数字
shuzi(shuzitop)=w 将数字压栈
```

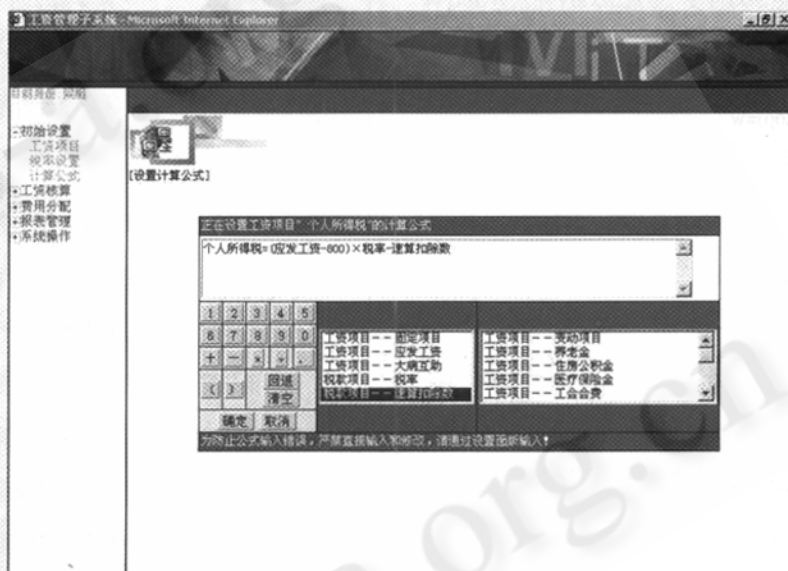


图3 工资录入界面

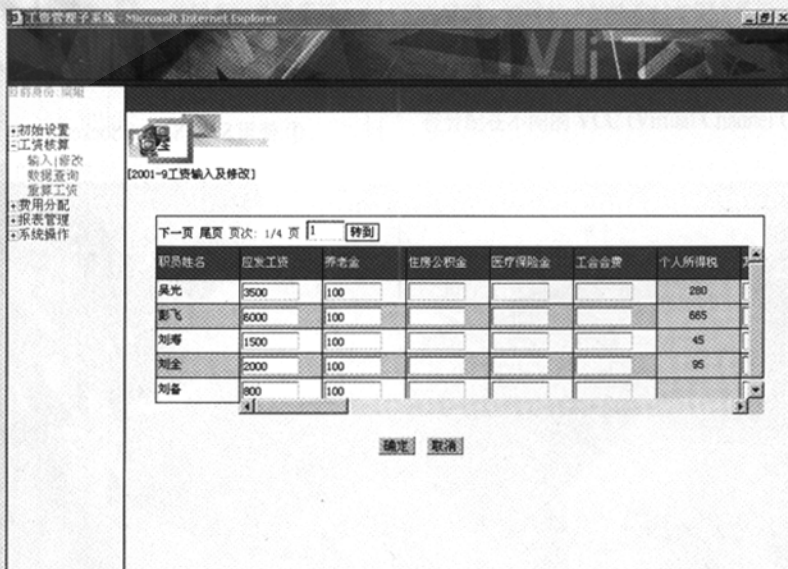


图4 工资录入修改界面

Design and Implementation of Open Salary Management Systems Based on B/S

```

shuzitop=shuzitop+1 改变栈顶指针
w=read(gsmes,zyi d,rq)
else 如果是运算符,则与运算符栈中原有运算符
比较并进行相应操作
Select Case yxj(ysf(ysftop),w)
Case '<' 如果新的运算符比原运算符优先级高,则
压栈
ysftop=ysftop+1
ysf(ysftop)=w
w=read(gsmes,zyid,rq)
Case '=' 如果优先级相同,则将原有符号出栈
ysftop=ysftop-1
w=read(gsmes,zyid,rq)
Case '>' 如果新的运算符比原运算符优先级低,则
将数字栈和符号栈相应的数字和符号出栈,并将计
算结果压栈
ysbz=ysf(ysftop)
ysftop=ysftop-1
shu1=shuzi(shuzitop)
shuzitop=shuzitop-1
shu2=shuzi(shuzitop)
shuzitop=shuzitop-1
newshuzi=operate(shu1,ysbz,shu2)
.....

```

```

将运算出来的结果赋值给 shuzi(shuzitop)
shuzitop=shuzitop+1
shuzi(shuzitop)=newshuzi
End Select
end if
wend
.....
jstjxgs=shuzi(shuzitop) 循环结束后,输出统计项
结果
.....

```

公式设置和工资计算就构成了工资核算的实现过程,费用分配和工资报表的实现过程跟其类似,费用分配先对员工类别,分配科目(不同类别员工的工资费用分配的科目不一样)进行设置,然后利用设置的规则进行计算,并将计算结果保存到费用分配表中,报表管理的开放性主要表现在可以让用户设置报表,系统让用户通过报表设置界面将报表输出规则保存到报表设置表中,在报表输出的时候,系统从报表设置表中检索出相应的规则,并按规则检索数据,从而输出报表。

(3) 系统安全。在广域网上运行的工资管理系统,安全性特别重要,在安全方面我们主要采取如下措施加以控制:

① 使用 SSL (Secure Sockets Layer) 维护应用程序

的安全,以提供了一种安全的虚拟透明方式来建立与用户的加密通信连接,从而保证用户与应用程序间交换的重要信息不被截取。

② 应用软件提供用户身份认证,功能权限控制,操作日志监控等权限控制和安全防护手段,即将用户的权限细分到每一个最终菜单,并将用户在系统中的操作记录到应用操作日志中。

③ 使用大型数据库完善的安全机制,即将操作系统和数据库的权限相结合,对敏感数据通过多种交叉的权限控制其存取,给连接计算机的用户赋予各种不同的权限,如查看、修改和删除等。

4 结束语

基于B/S模式的工资管理系统通过其灵活的体系结构,易于扩充,便于企业集中管理资源,优化配置和提高工作效率;同时通过业务规则的灵活设置,将原来只能由开发人员完成的工作以简单灵活的方式交给用户,具有相当的开放性和灵活性,延长了系统的寿命周期。此外,以上设计思路在报表管理、单据管理、计划管理等基于B/S的系统中具有较好的借鉴作用。■

- 1 赵永强、朱霞,开放式计划管理系统的设计及实现,计算机系统应用,2001(10)。
- 2 陈晓红、吴良刚,管理信息系统理论与实践,中南工业大学出版社,1997-10。
- 3 Scot Johnson 著,Active Server Pages 详解,电子工业出版社,1999-8。
- 4 Richard Anderson Chris Blexrud 等著,ASP3 高级编程,机械工业出版社,2000-10。
- 5 Robert J.Oberg 著,深入学习:COM+ 高级编程,电子工业出版社,2001-1。
- 6 用友 iERP 战略, <http://www.ufsoft.com>。

