文章编号: 1007- 2985(2007) 04- 0043- 03

# PDM 系统中编码管理和图纸批阅的设计方案*

贺　静
(长沙大学机电工程系, 湖南 长沙　410003)

摘　要: 通过基于 3 层分布式应用、用 PowerBuilder 工具结合 Adaptive Server Anywhere 数据库管理系统联合开发, 使系统分为 3 个不同的级. 将用户界面和企业逻辑分离开来, 克服了 2 层模式所带来的缺乏安全控制、缺乏安全性和客户端负载重等问题, 实现了 PDM 系统中编码管理和图纸批阅的设计.

关键词: 编码管理; 图纸批阅; 设计
中图分类号: TP311. 131　　文献标识码: A

目前, 许多制造企业仅仅停留在记录零部件和装配图的初级水平上, 这使得产品设计人员和工程人员很难得到所需要相关信息. 为了缩短产品开发时间, 一个很重要的前提就是要将产品有关的信息与过程进行有效地管理和控制, 否则, 就会造成数据混乱, 而且可能引起极大的损失[1-4]. 当前, 由于面向对象的数据库还不能很好地为企业提高支持, 关系数据库又存在许多不足如: 面向记录、不支持设计过程、缺乏协调工作机制等; 这些弱点导致在处理非结构化数据时显得力不从心; 远远不能满足现代企业对信息管理的需要. PDM 技术正是在这 3 个不同层次要求的驱动下产生的一门新兴的数据处理技术, 它把数据库的数据管理能力、网络的通信能力和其自身数据的控制能力结合在一起, 通过有效管理和控制所有与产品有关的信息, 以满足企业对信息管理较高层次的需求[5-7]. 本文通过基于 3 层分布式应用, 以 windows 系统为平台, 用 PowerBuilder 工具结合 Adaptive Server Anywhere 数据库管理系统联合开发, 利用分布式计算最大限度的发挥客户/服务器体系结构的优势, 将它在传统的客户机/服务器两级结构中再增加一个称作应用服务器的中间级, 用以执行复杂的商业逻辑计算, 使系统分为 3 个或多个不同的"级": 用户接口级、企业逻辑级和数据访问级. 通过 3 层结构, 用户界面与企业逻辑分开, 通过这种方法将用户界面和企业逻辑分离开来, 克服了两层模式所带来的缺乏安全控制、缺乏安全性和客户端负载重等问题, 实现了编码管理和图纸批阅的设计.

## 1　分布式应用的技术实现

PDM 采用分布式 PowerBuilder. 在该体系结构下把应用程序分为 2 个部分: 客户应用和服务器应用. 服务器应用和客户应用共同完成用户所需的任务. 一般来说, 客户应用与用户进行交互, 而服务器应用则为客户应用提供后台服务. 下图是服务器应用和客户应用的结构.

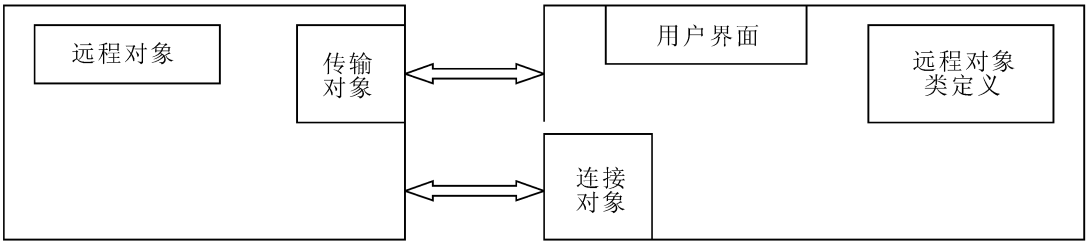


图 1　服务器应用和客户应用的主要组成

### 1. 1 开发服务器应用( Bmgj_server. pbl)

服务器应用有 2 个重要的组成部分: 传输对象(Transport Object) 和远程对象(Remote Object) . (1) 传输对象负责建立客户

* 收稿日期: 2007- 04- 20
作者简介: 贺　静( 1975- ) , 女, 湖南株洲人, 长沙大学机电工程系讲师, 主要从事 CAD/CMA 研究.

连接、处理客户的服务请求. 它在服务器应用的脚本中被初始化, 其属性为 PowerBuilder 提供了处理客户所需的信息. (2) 远程对象是定制(非可视化) 用户对象, 它包含在一个位于远程服务器上的应用程序中, 客户可以像访问本地对象那样调用远程对象的函数.

1.1.1 创建用户界面 服务器应用的界面仅用于启动和结束监听程序以及监控客户的连接.

1.1.2 管理客户连接 当服务器启动时, 同时开启监听客户连接, 具体代码如下:

```
// 声明全局变量: transport mytransport
long ll_rt
this.SetTranspool(12, 16, 10)// 设置事务缓冲池
mytransport = create transport// 实例化传输对象
mytransport.driver = "Winsock"// 指定通讯驱动程序
mytransport.application = "10015/tcp"// 指定服务器应用
mytransport.location = "hejing"
ll_rt = mytransport.Listen()// 监听
IF ll__rt <> 0 THEN
MessageBox("提示","启动服务器失败" + string(ll_rt)
+ mytransport.errtext)
ELSE
    MessageBox("提示","启动服务器成功")
END IF
```

当客户向服务器应用请求建立连接时, 服务器允许建立连接, 在事件 ConnectionBegin 中建立如下代码:

```
SQLCA.DBMS = "odbc"
SQLCA.DBParm = "ConnectString='DSN=bmxt;UID=
dba;PWD=sql'"
Connect using sqlca;
IF sqlca.SQLCode <> 0 THEN
    MessageBox("提示","数据库连接失败" + SQLCA.
SQLErrText)
    return noconnectprivilege!
ELSE
    return connect with adminprivilege!
END IF
```

1.1.3 建立远程对象 远程对象是服务器应用的基本构建模块. 每个远程对象都是定制的类(非可视化的) 用户对象. 服务器应用必须包含每个远程对象的完全实现, 客户只需包含代理对象. 本课题在服务器端建立了 2 个远程对象用于实现企业的商务逻辑:

a) cnvo_data

b) cnvo_data1

1.1.4 访问数据库 服务器应用通过数据存储与数据库进行交互, 客户应用通过数据窗口控件显示从服务器检索出的数据. 为了使客户与服务之间的数据同步, 使用了 4 个函数: GetFullState(), SetFullState(), GetChanges(), SetChanges().

1.2 开发客户应用(Bmgj_client.pbl)

客户应用包含 3 个重要的组成部分: 用户界面, 连接对象(Connection Object), 远程对象的类定义. 用户界面包括所有与用户交互的窗口和菜单, 同时也包含每个动作的处理脚本; 连接对象负责与服务器应用进行连接, 并向服务器提交服务请求; 远程对象的类定义用来访问远程对象实例.

1.2.1 开发用户界面

1.2.2 与服务器建立连接 客户应用必须与服务器应用连接才可以完成相应工作, 这一工作通过连接对象来完成, 具体代码如下:

```
// 全局变量 connection myconnect
long ll_rt
myconnect = create connection// 实例化连接对象
myconnect.driver="WinSock"// 指定通信驱动程序
myconnect.application = "10015/tcp"// 指定服务器应用
名
myconnect.location = "hejing"// 指定服务器的位置
ll_rt = myconnect.ConnectToServer()// 与服务器建立连
接
IF ll_rt <> 0 THEN
MessageBox("提示","连接失败!" + string(ll_rt) +
myconnect.errtext)
ELSE
MessageBox("提示","连接Server成功!")
END IF
open(w_main)// 打开客户应用界面
```

1.2.3 生成代理对象(Bmgj_proxy.pbl) 服务器应用中的每个远程对象在客户应用中都有一个对应的类定义, 客户应用中的类定义既可以是远程对象的完全实现, 也可以是提供远程对象接口表现的一个代理对象. 本课题采用代理对象的形式. 创建工程 Bmgj_proxy.pbl, 该工程包含两个代理对象: cnvo_data 和 cnvo_data1, 分别与远程对象 nvo_data、nvo_data1 对应.

1.2.4 访问远程对象 与服务器建立连接后, 客户应用即可以通过代理对象调用远程对象函数, 访问远程对象的实例变量. 具体代码如下:

```
// 实例化远程对象 cnvo_data
// 全局变量 cnvo_data nvo_data
long ll_rc
ll_rc = myconnect.createinstance(nvo_data)
```

```
if ll_rc<>0 then
messagebox("createinstance 失败",ll_rc)
end if
//实例化远程对象 cnvo_data1
//全局变量 cnvo_data nvo_data1
ll_rc = myconnect.createinstance(nvo_data1)
if ll_rc<>0 then
    messagebox("createinstance 失败",ll_rc)
end if
```

## 2　编码管理的技术实现

根据 CAD 文件编码管理标准建立属于企业的编码规则库, 企业在编码之前, 通过先定义符合企业产品特点的编码规则, 系统自动将编码规则保存到数据库. 编码时, 企业通过用户界面选择产品所属的特征代号, 在该特征代号下进行申请图号的操作, 编码模块会根据预定义的编码规则自动生成文件的编码, 并将之存入产品信息库以备查询.

## 3　图档批阅的技术实现

### 3.1 建立图文档数据库

用户根据图文档的属性(装配图、部件图、零件图) 将图文档分门别类存入数据库, 方便查询.

### 3.2 图文档批阅

企业从数据库提取图文档, 利用系统提供的图文档批阅工具对其进行操作, 常用的批阅工具有直线、云状线、圆、弧、自由画笔及文字标注等. 批注内容用红色画笔绘在特定的图层中, 并单独存盘, 不影响原图形的独立性, 此部分利用 PowerBuilder 的 OLE 工具实现.

## 4　结语

本文通过在 windows 系统的平台上, 利用 PowerBuilder 工具结合 Adaptive Server Anywhere 数据库管理系统联合开发来进行了编码管理和图纸批阅方案的设计, 通过方案设计, 使得分布式计算在 PDM 中得到实现, 并能使 PDM 最大限度地发挥客户/服务器体系结构的优势.

参考文献:


[1] 刘士军, 孟祥旭, 杨承磊. PDM 中的编码管理设计研究 [J]. 计算机工程与应用, 2002, (9): 79- 82.

[2] 刘增进. PowerBuilder7.0 数据窗口技术详界 [M]. 北京: 电子工业出版社, 2000.

[3] 郑若忠, 宁　洪. 数据库原理 [M]. 北京: 清华大学出版社, 1999.

[4] 王建涛, 方明伦, 俞　涛. 基于 PDM 的产品数据集成管理 [J]. 计算机辅助设计和制造, 1998, (3): 31- 33, 52.

[5] 叶晓俊, 蒲明辉. 产品图纸文档管理 [J]. 计算机辅助设计和制造, 1998, (3): 34- 36.

[6] 汪　利, 王石刚, 邹慧君. 中小企业 PDM 设计及实现 [J]. 中国机械工程, 1998, 9(9): 69- 71.

[7] 莫欣农, 高奇微. 产品数据管理系统的实施 [J]. 计算机辅助设计和制造, 1998, (7): 18- 25.


# Coding Management and the Design Research of Drawing Read Overmarginalis In PDM


HE Jing

(Department of Mechanical & Electrical Engineering, Changsha University, Changsha, 410003, China)



**Abstrct**: System is designed to have three different grades on three-layer distributed application by combining PowerBuilder and Adaptive Server Anywhere. By the Separation between User interface and business logic conquers, some problems such as the lack of safety control, security and the overloading on client side have been conquered, and it realizes coding management and the design research of drawing read over marginalia in PDM.



**Key words**: Coding Management; drawing read over marginalia; design

(责任编辑　易必武)