

一种基于 C/S 的分布式

PowerBuilder

数据传送方法

湘潭大学计算机科学系 万天明 孟志青

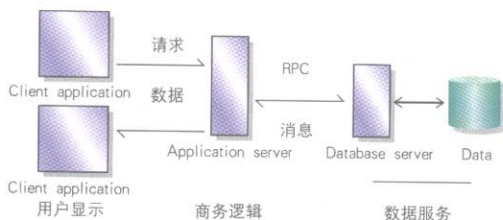
本文针对分布式 PowerBuilder 环境中数据传送问题提出了一种基于 C/S 的数据传送技术,并结合实例说明实现这种技术的方法。

引言

在分布式 PowerBuilder 中传送数据时大多数开发人员会发现不允许声明标准对象类型的远程用户对象存在的问题,开发人员仅能由定制的用户对象创建远程对象,这使得创建一个 DataStore 用户对象并把它声明为定制的用户对象的一个实例变量。不幸的是 PowerBuilder 在试图保存代理对象时会检测这种破坏性活动,警告用户,并创建没有实例变量的代理影像,这样就不能直接把对象作为一个属性访问或由一种方法返回对象,导致传送数据时无法返回信息。因此,有必要提出一种在远程对象之间转换数据的方法,它使用数组和允许开发人员直接把 DataWindow 行分配到结构中的新的 DataWindow 语法来解决这一问题。

数据传送结构

为了解决上述问题,我们在开发实用系统中提出了一种数据传送结构,以 Client/Server 的结构为基础,采用了一种 N 级结构,它把应用程序组件分为三个独立的逻辑组件:用户显示、商务逻辑和数据服务,如下图所示。



用户显示在客户机上负责与用户交互并根据用户请求调用相应的逻辑规则;数据服务由专用的服务器提供数

据库管理和操作;商务逻辑则驻留在一个或若干个应用服务器上,执行具体的事务逻辑并通过 SQL 等方式向数据库服务器提出数据或其他资源的请求。使用这种分级结构的优点如下:

1. 面向对象化

模块化和封装是任何面向对象开发的最终目标,逻辑分界能使这一目标较容易地实现。

2. 多种数据源

用户级向逻辑级发出请求,然后逻辑级决定使用哪个数据源请求,通过使用一个由用户级使用的调用接口,逻辑级可以对任可用的数据源中进行请求。

3. 安全性高

访问特权可以指定或内置于三个级中的每一个,以使提供三个级别的安全性。

4. 维护和管理容易

用户端只处理用户接口,不涉及数据的存取,而商业逻辑的更新只对中间修改,不用更新整个系统。

软件支撑环境可由数据库管理系统软件构成,如 VFP、ORACLE、SYBASE SQL 等,网络环境为 WIND OWS NT4.0,通信协议采用 WINSOCK(SERVER, CLIENT)。

数据传送方法的实现

下面我们给出上述数据传送结构的实现过程,我们以数据库的顾客表作为一个实例使用。

1. 对象的创建

(1)首先创建一个结构对象 s_customer_data。该结构对象的元素和数据类型与顾客表相同。

```
create s_customer_data
{
    int id
    string fname
```

```

string lname
string address
string city
date birthday
integer age
char sex
string memo
};

```

(2)再创建两个对象:DataWindow 对象和 DataStore 对象。这个 DataWindow 对象称为 d_customer_data, 它选择顾客表中的每一列。这个 DataStore 对象称为 u_ds_customer_data, 它是与新的 DataWindow 对象相关联的。

(3)定义定制的不可视类用户对象。最后一个对象是远程对象 ro_customer_data。同时定义该对象类的一个私有实例变量 ds_customer_data, 它声明的类型为 u_ds_customer_data, 另外还必须提供一个函数, 该函数从 DataStore 对象提取数据并返回到一个正在请求的客户机应用程序。代码如下:

```

//s_customer_data GetCustomerData(long a_lRow)
s_customer_data sCustomerData
sCustomerData=ds_customer_data.Object.Data
[a_lRow]
Return sCustomerData

```

最后的工作是把 ro_customer_data 设置成 AutoInstantiate 并保存代理对象 p_customer_data 及用户对象, 特别注意应把代理对象和结构对象保留在服务器和客户机应用程序都用到的一个常用库中。

2. 应用程序服务器的启动

建立一个处理远程请求的应用程序服务器所需的 PowerScript 如下:

```

//Declared at a global or instance level would
be the following:
//Transport g_DPBBTransport
//Create the sever transport object
g_DPBBTransport=Create Transport
//Set the communications driver.
//This can be NamedPipes, Winsock, OpenClientSever,
or Local.
g_DPBBTransport.Driver="Winsock"
//Set the application server name

```

```

g_DPBBTransport.Application= "Test1"
//Listen out for clients on the selected
communication protocol.
g_DPBBTransport.Listen()
If g_DPBBTransport.ErrCode<>0 Then
    MessageBox("DPB Server Test1", "Failed to
start!")
End if

```

3. 把客户应用程序连接到应用程序服务器上

我们实现此步骤的 PowerScript 如下:

```

//Declared at a global or instance level would
be the following:
//The connection object to connect to the app
server for accessing
//the remote object
Connection g_DPBBConnection
//The proxy user object p_customer_data is used
to declare the client
side pointer
p_customer_data g_roService1
g_DPBBConnection=Create Connection
g_roService1= Create p_customer_data
g_DPBBConnection .Application= "Test1"
//A server IP address
// g_DPBBConnection.Location= "196.96.166.000"
//or a local test connection
//g_DPBBConnection.Location= "Test1"
//or some other named location.
g_DPBBConnection.Location= "Test1"
//Change to the network driver you are using.
g_DPBBConnection.driver= "Winsock"
If g_DPBBConnection.ConnectionToServer()<>0 Then
    MessageBox("Client", "Error Code=" +&
String(g_DPBBConnection.ErrCode)+&
"~r~nError Text="+g_DPBBConnection.ErrText)
    HALT CLOSE
End If
g_roService1.SetConnect (g_DPBBConnection)

```

4. 数据传送

现在可以在客户机应用程序代码中使用远程对象和代理对象的一个实例提取数据,

代码如下:

```
//Assume that there is a global variable of type
p_customer_data
//called ip_customer_data
Long lRow, lRowCount
s_customer_data sCustomerData []
//Also,assume that we coded an additional method
that returns the current
//number of rows in the datastore object.
lRowCount=ip_customer_data. Customer DataRow
Count ( )
//Loop through the remote DataStore pulling into
the array.
//NOTE: we back fill the array to get the best
performance.
For lRow=lRowCount To 1 Step -1
    SCustomerData [lRow] =ip-customer-data.
    GetCustomer Data(lRow)
Next
```

我们现在就可以利用数据窗口或者其他控件把结构数组变量中的值显示出来, 这里不再多述。

5. 中断连接

通过服务器控制台 DataWindow 可中断客户端到服务器的连接, 方法是从选择所要求的客户机, 将其 Userid 值传递到连接对象的 RemoteStopConnection 函数, 代码如下:

```
long ll_row
string ls_userid
ll_row=dw_console. getselectedrow(0)
if ll_row=0 then
    messagebox("disconnect error", "please select a
client! ")
    return
end if
ls_userid=dw_console.getitemstring(ll_row,"userid")
if messagebox("Disconnect confirmation",&
    "Are you sure you want to disconnect" +ls -
userid+ "?", &
    Question!, YesNo!, 2)=1 then
    in - connection.RemoteStopConnection(ls -
userid)
end if
```

结束语

通过上述的方法, 在分布式 PowerBuilder 数据传送当中, 重点是类用户对象的定义, 此对象经常在一个应用程序和多个应用程序中重复使用。开发人员可以使用 DataStore 或 Data Window 对象或者甚至直接使用数据, 而不必在客户机上生成结构。除了支持基于 Client/Server 的分布式应用, PowerBuilder 还提供优质服务 Web.pb 使得应用可以在 Internet 上分布, 通过调用远程对象的服务可以为客户端的浏览器提供动态内容, 因此, 这一方法具有广泛的应用价值。■

参考文献

- 1 Simon Gallagher & Simon Herbert 著, 曹康等译, PowerBuilder 5.0 程序设计大全, 机械工业出版社, 1997.9
- 2 尹晓勇, 于洁, 用 PowerBuilder 建立基于 C/S 的分布式应用, 计算机系统应用, 1999.8
- 3 孟志青, 刘玉葆, 基于数据仓库的商业销售分析与决策系统的一种建造方案, 计算机应用, 1998.8
- 4 王坚强, 赵柴厚, 基于 C/S 模式分布式计算的实现, 管理信息系统, 1999.8

