

COM/MTS 对象组件技术在远程数据传输中的应用

Application of Remote Data Transfers Based on COM Technology

殷立峰 (山东大学计算机科学与技术学院)
(山东政法学院司法信息系)

摘要:COM(Component Object Mode)对象组件技术是 Microsoft 公司提出的一种分布式应用开发技术,本文介绍了在 Power builder 软件开发平台上,基于这种技术原理和 MTS 微软事务服务器技术,在医院 HIS 和医保 MIS 信息系统之间远程数据传输的研究和实现。

关键词:对象组件技术 远程对象代理 传输对象 数据集

1 引言

COM 是英文 Component Object Model 的缩写,中文的意思是组件对象模型,它定义了一套可以进行程序代码共享与复用的软件开发技术标准,提供了对远程对象的调用功能,使得客户端和服务端对象和对象的方法可以相互进行远程调用。此外 COM 结合 MTS 微软事务服务器提供了对事务管理的支持,通过 ODBC 接口可以实现对数据库事务型的访问。基于 Power Builder 对 COM/MTS 和数据存储对象的技术支持,我们根据上述原理,采用 Power Builder 编程设计了医疗保险管理信息系统和医院管理信息系统的数据接口,成功实现了参保人员住院诊疗明细数据的实时、定时远程传送和在线医疗费用报销结算。

2 基于 COM 原理技术的接口系统软硬件架构和工作原理

根据 COM 技术原理,我们设计的系统软硬件接口体系结构如图 1 所示。

从硬件结构来看,我们把医疗保险信息中心作为服务器端,把医院端设计成为客户端。客户端和服务端采用无线传输方式构建数据通信网络。其中医保信息系统数据库安装在服务器端中心服务器上,在数据库服务器前端安装一台计算机作为服务器端程序服务器,负责针对客户端远程数据的收发和对数据库的访问;在医院客户端,安装一台计算机作为前置计算机,内置医院、医保接口数据库用以保存参保病人住院

信息、诊疗信息、报销结算参数信息、医疗保险三大目录基础数据、参保人员就诊明细费用和医保对照信息、以及其他参保人员在线结算用其他信息,同时安装具有参数下载、就医人员医保台帐数据下载、结算数据上传等功能的客户端程序,并安装独立执行实时上传参保人员住院信息以及诊疗原始处方数据的程序模块。

得益于 COM 技术的支持,软件体系架构从传统的 C/S 客户机、服务器模式,变成了实现方式为三层的客户端、应用服务器、数据库服务器应用体系结构,其体系结构示意图如图 2。

医保数据库服务器安装具有事务支持功能的大型数据库。主要保存医疗保险参保人员账户数据、医疗费用、基金和系统相关数据信息。

应用服务器端由远程对象(remote Objects)和传输对象(transport Objects)组成,传输对象具有服务器的定义信息、通信驱动器信息、数据在网络中的传递方式等属性信息,其功能是监听医院客户端的连接请求和事务处理请求,我们通过服务器端程序脚本中构建实例对象去实现。远程对象是基于 MTS 数据库事务访问技术原理创建的,用于数据库访问的一些非可视 datastore 类实例对象,主要用于完成将数据库访问结果构建成数据集功能。

客户端由用户界面、远程对象调用代理、连接对象组成。用户界面负责与用户的交互,主要完成医院诊疗、检查、药品等项目和医保相应项目的匹配对照,医疗费用报销预结算、结算等业务工作;连接对象通过

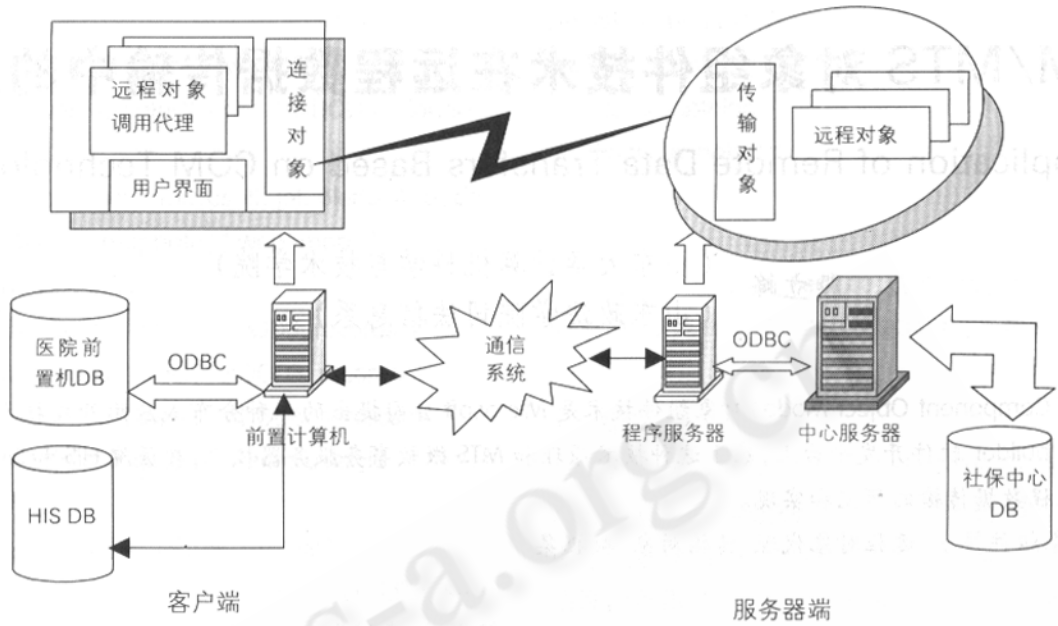


图 1 医院 HIS 和医保 MIS 信息系统软硬件接口

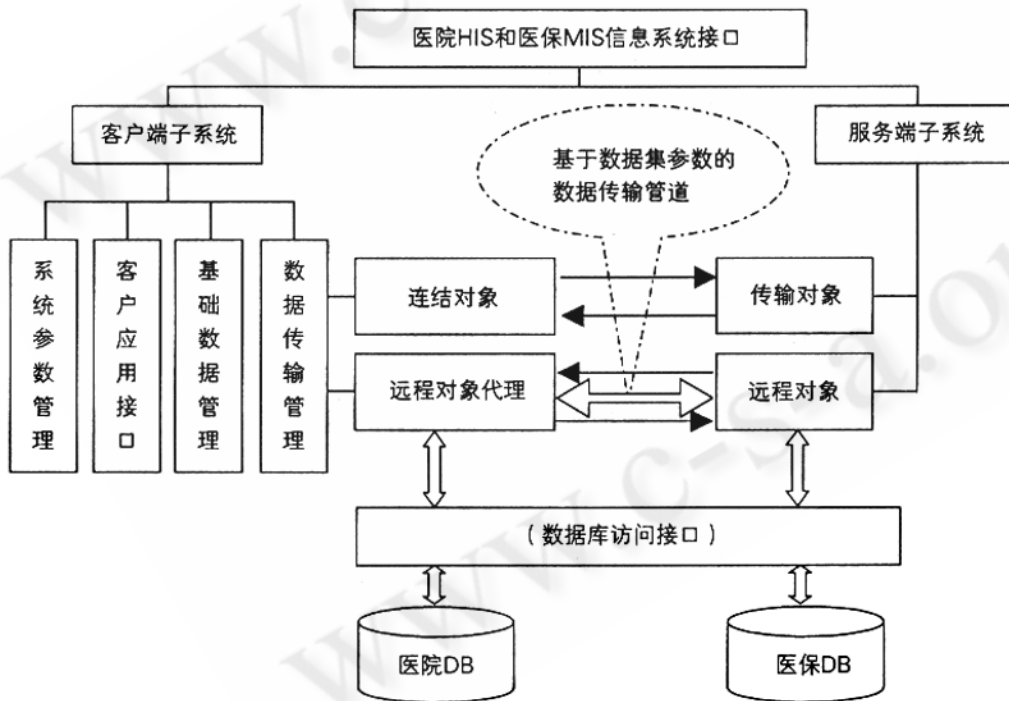


图 2 医院 HIS 和医保 MIS 信息系统软件体系结构图

以使得客户端应用与服务端远程对象的实例进行交互,从而形成客户端连接服务器端远程对象的纽带,客户端通过这个纽带实现对远程对象的调用,远程调用对象的参数是服务端构建的数据集,通过这种数据集参数传递机制在客户端和服务端建立了一个字符流通道,从而实现数据的传输应用。

3 数据传输的具体实现方法和流程

3.1 采用 ODBC 建立与数据库的连接

采用 ODBC 开放式

数据库接口技术,分别在客户端和服务端建立应用程序和医疗保险数据库的连接。

3.2 医院参保人员就诊原始数据信息准备

设计 HIS(医院管理信息系统)和医保 MIS(医疗保险管理信息系统)数据交换接口,构造数据传输接口函

在客户端程序脚本中实例化,使用其有关的属性信息如服务器应用,服务器主机名称,通信驱动器等将客户端应用和服务器端应用连接起来,建立通信连接;远程对象调用代理实际上就是远程对象的类定义在客户端的实例化,它在客户端应用和服务器端应用中必须确保有相同的名称,从而使得程序执行时,本地类定义可

数,利用动态连接库技术封装这些函数,通过在 HIS 中对这些函数的调用,动态实时的实现 HIS 数据向前置机接口数据库的传递,为下一步远程数据传输和网上在线医疗费用结算做好数据准备。

3.3 服务器端应用的实现

使用 Power Builder 的在服务器端创建服务端 SERVER 程序项目 COM_SERVER,项目中创建对象用于服务器端系统工作参数、个人医保台帐数据等的下载,以及客户端结算后个人台帐数据、原始诊疗处方数据、医保报销界算数据等的上传;创建 SUPER_SERVER_CLASS 用户对象,该用户对象里面主要根据 MTS 技术原理,封装了一些具有数据库访问,并将访问返回的结果从数据存储 DS_STORE 对象传递到 DATA_SET 数据集或者反之将 DATA_SET 数据集中的数据传递到 DS_STORE 对象中的方法。其中的数据集实际上是采用一些方法,将数据库的无论何种数据取出一律转变成字符串值并按规律装配成一个数据管道模型,这些方法是该种原理系统数据传输的核心部分。其中主要方法的程序如下:

```
方法一
// 功能: 将数据存储对象 ds 中的数据构建数据集 name
PUBLIC long put ( string name, datastore ds ) //name
保存数据的管道, ds 数据存储对象
long vcount //数据存储对象的列数(字段个数), long
vi, vj //长整型变量
string wvalue1, vitem_name[ ], vitem_type[ ] //字符串和
字符串型数组变量
dec wvalue3
datetime wvalue4 //时间日期型变量
//判断值参 NAME 是否为空
if isnull( name ) or trim( name ) = "" then
    this. puterr( -103, '调用 put 时参数' + name + '为
空')
    return -20
end if
//统计数据存储对象列数(字段个数),并取出字段名
称和类型
vcount = integer ( ds. describe ( " DataWindow. Col-
umn. Count" ) )
```

```
for vj = 1 to vcount
    vitem_name [ vj ] = lower ( ds. describe ( " #" +
string( vj ) + ". Name" ) )
    vitem_type [ vj ] = left ( ds. describe ( " #" + string
( vj ) + ". ColType" ) , 4)
next
//将数据存储对象 DS 中的数据取出,在 name 中构建
数据集
name = trim( name )
pri_newset( name ) //初始化数据集
for vi = 1 to ds. rowcount( )
    for vj = 1 to vcount
        choose case vitem_type[ vj]
            //将数据对象 ds 每一列的数据取出,转化成字符串
数据,赋值给 wvalue1
            end choose
        pri_putitem( vitem_name[ vj ] , wvalue1)
    next
    pri_saverow( )
next
pri_endset( name )
return 0
```

方法二

```
PUBLIC none get ( string name, datastore ds )
//name 保存数据的管道, ds 数据存储对象
// 功能: 将数据 name 的数据取出,还原成原本类
型,并且放这些数据到存储对象 DS 中,其程序实现和
put( string name, datastore ds ) 基本类似,是它的逆过
程。
```

在服务器端,将上述 SERVER 程序的对象方法封装成一个 COM 类型的组件,发布到 Win2000 服务器的组件服务中,并授权访问用户和密码,设置其不间断运行,以随时监听来自客户端的服务调用请求。

3.4 客户端应用的实现

使用 Power Builder 的在客户端创建应用客户端程序,该程序除包含系统与用户的接口外,主要还包括一些如服务器端的数据窗口类的存储对象实例和用户 COM_CLIENT 以及传输对象 CLIENT_TRANS。在这些对象中的方法就主要完成远程对象调用,实现数据上传功能。下面的程序代码就是上传原始处方数据的函

数,其中 `set_script` 就是保存数据并传输数据的载体,它比较形象的描绘出了数据传输的技术原理。

```
PROTECTED BOOLEAN UPLOAD_SCRIPT( )
```

//上传原始处方数据函数,是 `CLIENT_TRANS` 对象的方法之一

```
//变量定义
```

```
//准备要传输的处方
```

```
dw_upload_script.settransobject( sqlca )
```

```
dw_upload_script.retrieve( )
```

```
//计算传输的次数
```

```
vstep = dw_upload_script.rowcount( )/i_block_
num
```

```
for vk =0 to vstep
```

```
client.reset( )
```

```
//每次传输 i_block_num 行处方到 server
```

```
//client.put 方法将数据存储对象的数据压入
数据集
```

```
//client.run 方法调用远程对象 server_trans 并
完成处方数据的上传
```

```
//将这次传输过的处方置上传时间
```

```
next
```

```
return true
```

可以看出,在客户端通过调用远程服务端的传输对象,通过数据集参数作为载体实现了数据的远程传送。

4 结论

根据 COM 技术原理,设计了一种信息系统之间数据交换的软硬件体系架构,并通过具体编程实现了远程数据传输。实际应用表明,这种方案能够保证大量数据的正确无误的远程传输,是分布式程序设计技术在数据传输中的一种探索,为远程数据传输开辟了一条新的道路。

参考文献

- 1 杨秀章, Power Builder 8.0 开发指南,北京清华大学出版社,2003.