



1 多层分布式和 ASP、DCOM

原先的客户/服务器体系结构是为局域网/广域、企业级网络而设计的,它能为某些模式相对固定的应用和相对稳定的数据需求提供服务。随着网络技术和面向对象技术的发展,全球的服务器和数千万的用户正在互相连接起来,这就要求当前的计算机体系发生变革以适应这一变化,从而满足一个几乎完全无序的超大规模的网络环境。分布式对象技术及基于 Internet 的多层分布式应用开发就是在这样的环境中产生的。

从逻辑上来说,分布式应用程序至少由如下三层构成:表示逻辑层、业务逻辑层、数据逻辑层。

表示逻辑层:它是应用中直接面向用户的部分,主要完成应用的前端界面处理,即人机界面处理。

业务逻辑层:它实现应用的业务规划处理,决定程序的流程。

数据逻辑层:它是应用中对数据进行管理的部分,主要完成应用对数据的存取、更新、管理等工作以及访问数据的安全性、完整性、一致性。

随着 Web 技术的发展,出现了基于 WWW 的 C/S 模型,也叫 B/S 模型,本质是三层 C/S 结构,它综合了浏览器、信息服务和 Web 等多项技术,面向全球通信与 Internet,具有无可比拟的先进性。

对于分布式应用来说,三层逻辑上已经完全分开,物理上可能分布在不同机器上,使用不同的操作系统,使用不同的网络协议,但各层之间要实现相互通信,为此微软提出了分布式组件对象模型(DCOM),DCOM定义了分布式环境中 COM 对象之间通信和交互操作的一套标准 APIS 和接口,它进一步发展了 COM 技术,使得组件间的通信能跨越网络,通过 Internet,位于不同机器上的组件能自由通信。在 DCOM 的支持下,各层之间通过组件实现相互通信从而构成一个完整的分布式应用,并且与机器的物理位置、操作系统、协议都无关。

利用 ASP 和 DCOM 技术构建多层分布式应用



谢军 (南昌大学南区计算中心 330029)

摘要: 在论述 ASP 和 DCOM 技术基础上,阐述 ASP 和 DCOM 在多层分布式上的具体应用。

关键词: ASP DCOM MTS 多层分布式

ASP (动态服务器页面) 技术的出现使得我们可以方便地进行 WEB 页面与数据库和 DCOM 组件的集成,利用它,可以调用 WEB 服务器上注册的 ActiveX 组件来执行任务,它是基于 Microsoft NT4.0 和 IIS3.0/4.0 Personal Web Server on Win9x 的开放式脚本开发环境,是新一代动态 WEB 数据库应用开发方案,利用它可以实现复杂的 WEB 应用,开发动态高效、交互的 WEB 服务器端应用程序。由于它在服务器端执行,很好地保证了浏览器/服务器结构中数据的一致性。

是基于 NT4.0 之 IIS4.0 及采用 ASP 技术,实现 WEB 页面与数据库及 DCOM 组件的集成,让各层之间通过组件实现相互通信从而构成一个完整的分布式应用系统。

系统结构框图如图 1 所示。

在多层分布式应用中,业务逻辑层是十分重要的一层。利用 Visual C++ 可以方便地创建这一层。微软提供的两个开发 Wizard 使得创建 DCOM 对象的工作变得更加容易,应用 Wizard 产生一个基本的框架,可以在此基础上根据需要添加自己的 IDL 代码,对象 Wizard 则帮助在最后编译前装配 DCOM 对象,包括以下几个步骤:

(1) Visual C++ 的 ATL COM 应用向导生成一个 IDL 基本框架文件。

(2) 根据需要在 IDL 文件中增加微软的界面



2 “汽车代理商 Internet 网络销售系统”的多层分布式应用

“汽车代理商 Internet 网络销售系统”就

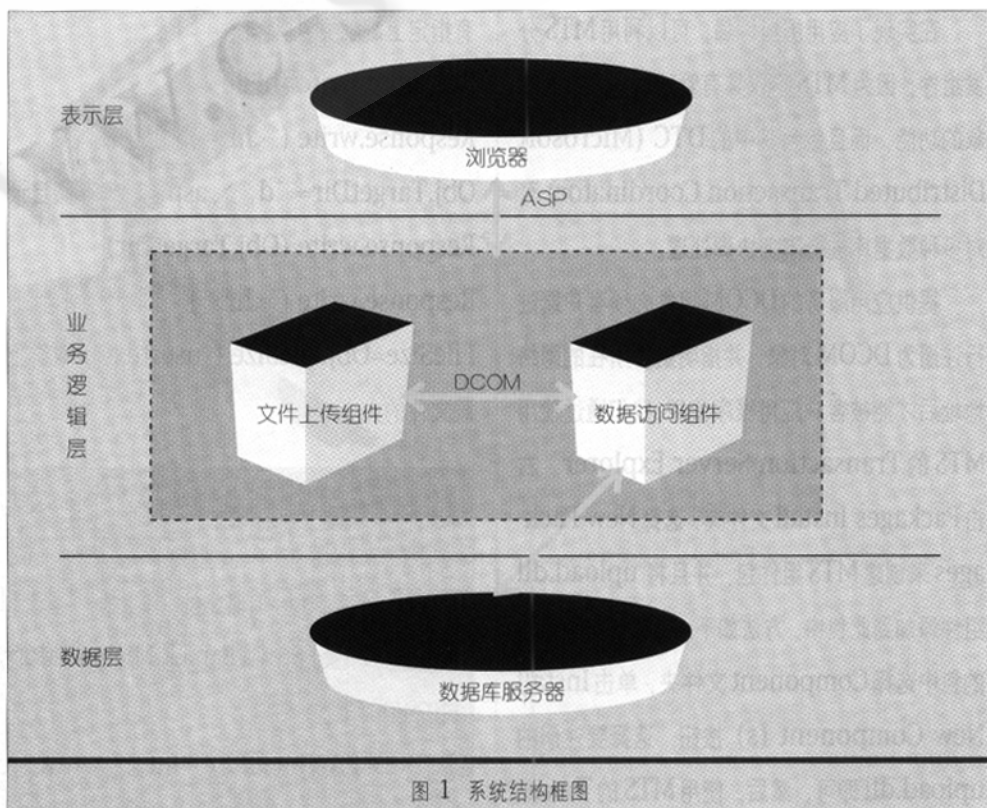


图 1 系统结构框图

定义语言 MIDL。

(3) 使用 MIDL 编译程序编译 IDL 文件, 生成必要的 proxy 和 sstub 代码。

(4) 使用 ATL 对象向导添加你要的 COM 对象。

(5) 编译并建立最后的 COM 对象。

(6) 编写应用, 访问该 COM 对象。

(7) 将该对象移到远程机器上并在那里登记。

文件上传组件 Upload 是利用 RFC1867 文件上传协议实现, 通过浏览器将文件上传到 WEB 服务器。其中接口是 DCOM 组件的重要属性, 它提供了若干个属性和方法给客户端程序调节器用。客户程序通过接口和组件通信。组件接口定义代码略。

DCOM 虽然提供跨局域网在不同的计算机上建立 COM 对象实例的方法, 但不能为多层应用系统提供完整的基础, 而需要由 MTS 进行打包管理。组件包对 MTS 组件包管理, 包括资源的集中存储、激活设置及对事务处理的支持。所以中间层的实现其实是在 MTS 中安装 COM 服务器构成。MTS 处理服务器上的对象实例、进程和线程管理、同步与共享资源, 以及安全等。

在实现了应用层组件后, 可以利用 MTS 分发组件, 因为 MTS 不仅具有管理这些分布式对象的功能, 而且包含其中的 DTC (Microsoft Distributed Transaction Coordinator) 可对多种数据库进行事务协调处理。

提供应用服务的 DCOM 组件必须客户端进行注册为 DCOM 对象, 并指明组件所在的网络地址后才能被客户应用程序使用。我们通过使用 MTS 的 Transaction Server Explorer, 右击 Packages Install 文件夹, 选择 New Packages 来创建 MTS 组件包, 并且将 upload.dll 组件添加到此包中。方法如下: 在准备安装组件的包中选择 Component 文件夹, 单击 Install New Component (s) 按钮, 选择要注册的 upload.dll 即可。然后, 使用 MTS 的 Trans-

action Server Explorer 的导出 (Export) 功能实现包的部署。

在组件已经完成后, 就可以编写 upload.asp, 并在其中调用注册好的 DCOM 组件, 并自由调用对象方法。程序代码如下:

```
<%@ Transaction=Required %>
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR"
Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<% Set Obj=Server.CreateObject
("Upload.File")
Response.write ("<hr>")
Response.write ("<hr>")
FileName=Obj.Form ("usertext") 取得
文件名
Response.write (FileName)
Response.write ("<hr>")
bFile=Obj.IsFile ("usertext")判断用户是否
有指向上载的文件
Response.write (bFile)
Response.write ("<hr>")
Obj.TargetDir="d:\asp" 设置存储目录
Response.write (Obj.TargetDir)
Response.write ("<hr>")
FileSize=Obj.FileSize ("usertext") 取得上
载文件的大小
```

```
Response.write (FileSize)
```

```
Response.write ("<hr>")
```

```
Obj.SaveFile ("userfiletext") 存储上载
的文件
```

```
Response.write ("<hr>")
```

```
Response.write ("<hr>")
```

```
ObjectContext.SetComplete
```

```
ObjectContext.SetAbort
```

```
%>
```

```
</BODY>
```

```
</HTML>
```

由于 MTS 负责组件的分布事务控制, 因此每个组件只须关注自己的业务逻辑, 仅有的额外操作是在处理完成后调用上下文对象的 SetComplete/SetAbort, 向 MTS 通报处理结果, 而由 MTS 根据两阶段提交协议决定提交或回滚事务。显然, 如果事务处理机制需要由组件负责实现的话, 完成上述流程的工作量将是十分巨大的。



3 结束语

总之, DCOM 技术和 ASP 技术为开发基于 Internet 的多层分布式应用提供了极大的便利, 提高了组件的访问透明性, 有效地进行数据库操作, 增强了重用性和扩展性, 降低了系统资源消耗, 更易于维护与升级系统, 为企业促管理, 提高企业竞争力提供了重要的技术途径。可以预见, 分布式对象与 ASP 结合将成为构造 WEB 应用系统的最常用方式, 基于 Internet 的多层分布式应用也将会越来越广。 ■

利用 ASP 和 DCOM 技术构建多层分布式应用



参考文献 1 骆文俊, 黄上腾, “基于 MTS 构建分布式医疗信息系统” 计算机工程 2000.5,

2 [美] Steve Banick Michael Morrison 著, “Visual InterDev6 开发使用手册”, 机械工业出版社。

3 “DCOM Architecture” <http://www.microsoft.com/com/compaper.asp>.