

基于 COM 的分布式多层 应用软件开发

许斌 张浩文 (昆明理工大学 99 级计研 650011)

摘要 本文详细介绍了 COM 的一些基本概念和构件间的通信机制, 阐述了结合构件技术与分布式计算的多层体系结构的开发乃是软件开发的发展趋势。最后, 结合云南联通寻呼公司的选号实时开机系统的开发, 介绍其完整的开发模式。

关键词 COM 构件 三层结构 ADO ASP B/S 体系 中间件

1 软件开发的现状

计算机以软件体系结构而论经历了以文件为基础的结构, 再到客户/服务器结构。目前许多系统都使用二层 C/S 这种结构来设计, 随着二层 C/S 计算模式不断成熟, 也就日益暴露出其局限性。首先, 由于只有两层结构, 许多应用逻辑都必须在客户端实现, 且仅能对特定的客户, 一旦业务发生变更, 其维护成本将变得相当高昂, 有数据显示, 维护的成本占到了系统生命周期的 60%, 从软件工程的高度看, 其扩展性和可重用性都比较差。其次, 随着 INTERNET 的爆炸式发展, 分布式计算成为必然趋势, 而 C/S 体系中, 客户与服务器的关系绑定过死, 无法适应 INTERNET 中异构异地的现实, 于是多层系统的设计思想出现了。

多层系统的基础为三层计算模式, 即表示层、业务逻辑层和数据服务器。现较流行的为三层 C/S 和 B/S 结构。中间件既作为客户端与服务器的通信代理, 又作为业务逻辑的具体实现层, 使得整个系统的可扩展性和可

重用性都极大增强, 且使得客户端与服务器实现松耦合, 以适应 INTERNET 的分布式计算模式。

为了与多层系统体系这一结构相适应, 构件, 这一先进的编程方法得到迅速发展。构件并不是新生事物, 它是面向对象编程中类的更高层抽象——组件 (COMPONENT), 它具有类的继承、重载、多态等封装特性, 使其间的交互仅仅依赖于接口, 但其本质区别在于可用不同语言实现。构件意义上的重用是二进制上的重用!

多层系统与构件这一计算体系与编程技术的有效结合, 为 INTERNET 上的分布式计算提供了有力的支持。

2 COM 及通信过程

COM, 即组件对象模型, 是 MICROSOFT 提出的一种以组件为发布单元的对象模型, 它提供了组件之间进行交互的规范, 因为组件对象之间的规范不依赖于任何特定的语言, 所以 COM 也可以是不同语言协作开发的一种标准。

早期的 COM 可看作是单机环境

中的对象代理请求, 而目前的 DCOM 则是将 COM 间的调用扩展到分布式环境中。COM 是通过一系列底层的 API 函数的 COM 库实现的。

COM 中, 对象和它的使用者之间的交互采用请求应答模式, 其过程为: CLIENT 向 COM 发出创建组件对象请求; COM 库通过注册数据库查询 SERVER 位置; COM 库定位并激活 SERVER; SERVER 创建对象实例并传回接口指针给 CLIENT, CLIENT 调用对象的接口与 SERVER 对象透明通信。

3 MICROSOFT 的多层分布式构件开发解决方案——DNA

Windows Distributed interNet Application (简称为 Windows DNA 或 DNA), 是在 Windows 平台上开发 INTERNET 应用所推出的方案。

3.1 B/S 体系

B/S 体系称为浏览器、服务器体系。从某种意义上看是三层 C/S 的特定演变, 即客户端以 IE 浏览器形式实

现,业务逻辑层以 IIS 服务器 (INTERNET INFORMATION SERVER) 产生 WEB 页面。与三层 C/S 相比,其系统伸展性和客户端可维护性都有较大提高。

3.2 ADO 与 ASP

为了统一与各数据的接口, MICROSOFT 推出了 OLEDB, 但 OLEDB 过于低级难以直接使用, 于是 MICROSOFT 随后推出了一个较为高级且简单的 COM 对象—— ADO, 让程序员存取数据, 由于 ADO 是建立在 OLEDB 之上的, 因此拥有广泛的与各种数据库相连的接口。

为了适应 B/S 这一计算模式, MICROSOFT 推出 ASP 技术, 所谓 ASP 就是在 HTML 中内嵌脚本语言, 然后以脚本语言使用 ASP 对象来取得 HTTP 要求的信息, 使用 ADO 存取数据库, 当处理完毕之后再使用 ASP 对象把结果回传给客户端的浏览器。MICROSOFT 定义了六大内嵌对象以供程序员使用。

利用 ASP 技术访问数据库的过程如图 1:

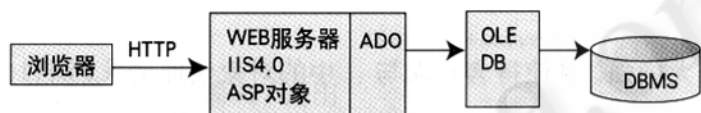


图 1 利用 ASP 访问数据库

3.4 ASP 技术的发展——定制 ASP 对象与 MTS

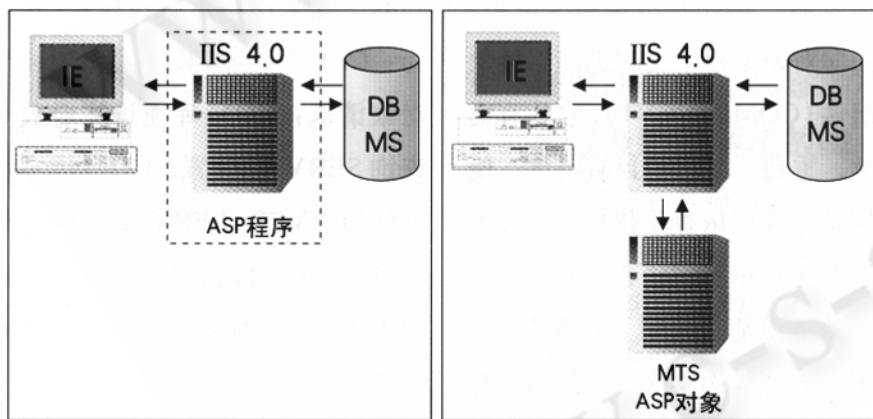


图 2

MICROSOFT 提出 ASP 技术基本解决了 B/S 下的动态网页交互问题, 但是由于仅提供了六个内嵌 ASP 对象, 使得大量业务逻辑依靠脚本语言来实现, 脚本语言并不能很好的解决面向对象的设计且需 IIS 来解释执行, 因此简单的使用 ASP 难以实现复杂的业务需求且系统效率低下, 业务逻辑的错误容易造成整个 IIS 服务器的死锁。

自定义的 ASP 对象正是基于 COM 这一先进编程思想, 它可根据需求封装特定的业务逻辑。并以 EXE 或 DLL 文件形式产生, 执行效率大幅提高。

随着自定义 ASP 这种 COM 对象的增加, 由各种组件所构成的中间件需要 MTS 负责所有事务处理、并发性处理, 和资源管理、安全性管理等系统级特性处理。业务逻辑从 WEB 服务器中独立出来, 变化如图 2 所示:

自定义 ASP 是 ASP 技术的革命性发展, 它改变脚本语言为 COM, 为 DNA 提供有力支持。

4 应用实例

寻呼事业经过近十年的发展市场竞争日益激烈, 寻呼的先期用户中, 有相当大的一部分转入使用移动电话, 停止使用原来的寻呼机。为了合理配置资源, 联通寻呼公司决定将停用三年以上的寻呼号源回收, 将其中一部分优质号源 (如以 6、8、9 结尾) 与裸机的地址码相配, 给顾客以免选号的优惠。经销商将用户选中的寻呼号与裸机的地址码通过特服号码 1292222222 告知联通寻呼公司的开机办, 通过实现选号开机系统将寻呼号与地址码有效动态绑定, 实现开机功能。

需求分析显示, 联通寻呼共有 126、128、191、198 四个异构的数据库, 分别记录有该台号属下的寻呼号、地址码、频点、速率等数据。接受传呼请求时通过寻呼号获得地址码、频点、速率等通信要素的映射, 最终通过 TNPP 接口与通信设备相连实现传呼。这四个数据库分置于异地的电信分局, 省公司下设计费中心集成有全部台号的用户数据, 用于联机处理用户收费等。实时选号开机系统必须依不同台号完成用户资料的录入相应台号所属的数据库, 并完成计费中心数据的相应录入。

整个系统实现的难点在于异构数据库的集成, 显然可用一些成熟的工程技术如依靠串口通信实现数据库间的一致更新。但考虑到电信业务的迅猛发展, 日后面临四台合一的需求, 决定采用中间件技术来屏蔽异构数据源, 采用构件技术来提高系统的可扩展性。

基于上述考虑, 我们的系统分为三层: 数据库层、应用逻辑层, 用户界面层, 用户界面采用 IE, 通过内置 ASP 来同应用逻辑层构件交互, 应用

逻辑层运行于 MTS 环境下, 享受 MTS 提供的各种 POOLING 技术和负载均衡技术, 来提高运行效率和构筑容错系统。界面及应用逻辑层中的自定义 ASP 对象由 C++BUILDER 5 实现。

流图的方法仍然在工程中取得了很好的效果。数据流图采用自顶向下, 分层细化的思想来刻画需求, 底层数据流中的文件可映射为数据库中的表, 不再细分的加工封装为基础构件, 而上一层数据流图中的上级加工封装为更高层的功能构件, 功能的实现依靠 DCOM 接口调用下层基础构件。以次类推, 顶层数据流图中的加工封装为协调构件, 负责产生、调用下一级构件的功能。

应用逻辑层拥有发空机, 领、放、销寻呼号, 开、停机, 统计, 4 个协调构件, 限于篇幅仅对开、停机构件的实现与其他构件通信关系作简要说明, 见图 3。

需要说明的是, 我们可以将大多数可细分的业务逻辑直接封装在一个构件中, 这暂时提高了编程的效率, 但考虑到日后的软件复用与扩展需要在工程中找到适宜的折衷。

以 C++BUILDER 为开发工具时, 4 个协调构件定义为定制 ASP 对象, 主要负责处理 HTTP 的要求, 基础构件适当地以函数形式封装于功能构件中, MTS 对象实现功能构件并处理结果主页内容, 但 MTS 对象不知道如何从数据库中取得它需要的数据, 因此它调用了数据对象以便取得它需要的数据。■

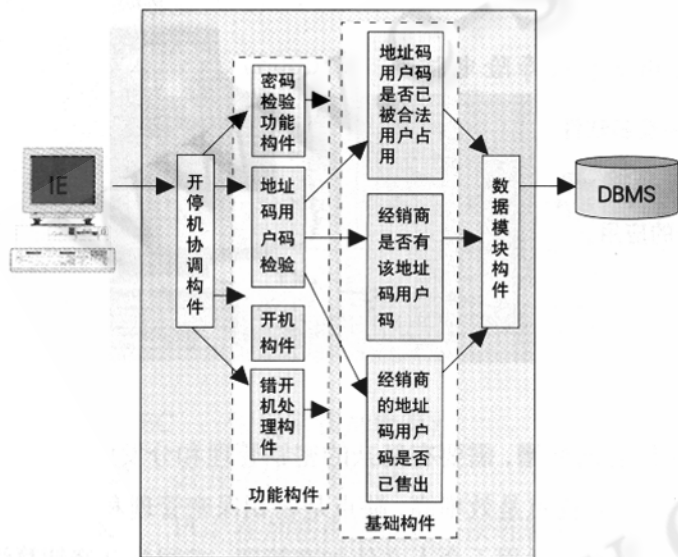


图 3 运行在 MTS 环境中的业务逻辑层

应用逻辑层的设计乃是本系统成败的关键。使用 UML 建模语言似乎是潮流所趋, 但笔者使用成熟的数据