

# CORBA 与 DCOM 桥接设计与实现

祁明龙, 阚文第, 杨 俊

(武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)

**摘 要:** DCOM 与 CORBA 是目前广泛使用的分布式技术, 在开发大型的企业级分布式应用系统时, 优先考虑使用的分布式技术就是 DCOM 或 CORBA 这两种中间件技术。有时一个系统需要同时使用 DCOM 与 CORBA。由于这两者在具体的技术实现上有较大差别, 使得它们在跨平台的互联时, 不能直接实现 DCOM 与 CORBA 的互访。因此, 如何实现系统中 DCOM 与 CORBA 直接通信成为一个关键问题。提出了一种实现 DCOM 和 CORBA 通信的桥接框架, 是一种“动态桥”, 所需的桥接口可以随时添加到桥里, 它解决了 DCOM 和 CORBA 之间不能通信的问题。

**关键词:** 通用对象请求代理架构; 分布式组件模型; 桩; 骨架; 接口定义语言; 桥

**中图分类号:** TP311.5

**文献标识码:** A

**文章编号:** 1673-629X(2008)05-0105-03

## Design and Implementation of Bridging Between CORBA and DCOM

QI Ming-long, KAN Wen-di, YANG Jun

(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

**Abstract:** DCOM and CORBA are the currently widely used distributed technology. In developing large distributed enterprise application, developers would firstly think about using DCOM or CORBA. Sometimes, a system has to use DCOM and CORBA simultaneously, because of business need. But there is some difference in detail technical realization, which makes it impossible to achieve direct communication between DCOM and CORBA. How to achieve the communication between DCOM and CORBA in system becomes a key problem. Introduces a bridging architecture of DCOM and CORBA, which is a dynamic bridge; any needed bridge API can be added to it anytime. It resolves the problem between DCOM and CORBA.

**Key words:** CORBA; DCOM; stub; skeleton; IDL; bridge

## 0 引 言

面向对象中间件联合了对象与分布式技术在平台、编程语言透明性等方面, 支持分布式对象应用集成。CORBA、DCOM、RMI 是三个顶级的面向对象中间件系统, 它们分别使用自己的独立框架来建立分布式应用。这些系统在最高层上是类似的, 但在底层上却不相同。以 CORBA 和 DCOM 为例, 这两个系统都把对象看作离散的功能单元, 都对客户端隐藏具体实现细节, 但是在对象描述、对象使用以及对象模型组件的组织方式方面却各不相同<sup>[1]</sup>。因此, CORBA 与 DCOM 之间的相互操作是不能直接实现的。故而, 需要找到一种解决不同系统对象之间互操作的方案, 来实现一个系统中的对象跟另一系统中对象的交互。文中主要研究有关 CORBA 对象与 DCOM 对象之间的交互协作。

## 1 CORBA 与 DCOM 简介

### 1.1 CORBA

CORBA 是 OMG 制定的一种标准的面向对象应用程序体系规范。CORBA3.0.3 核心规范在 2004 年发布。规范允许软件开发人员编写可以与其它对象交互的对象, 而无需知道其它对象如何以及在哪里实现。CORBA 的一个主要特点是它支持构建和集成异构分布式环境下的面向对象的软件构件。CORBA 协议的核心部件是对象请求代理(ORB), 它是实现 CORBA 客户端与被请求对象之间通信的核心<sup>[2]</sup>。客户端通过 IDL 桩或 DII(Dynamic Invocation Interface)与 ORB 核心通信。由于 IDL 存根表示了客户端实现语言与 ORB 核心实现语言的对应, 故客户端可以用 ORB 核心支持的任一种语言进行开发<sup>[2]</sup>。

### 1.2 DCOM

分布式组件模型(DCOM)应追溯到 1990 年, 当时为了支持 Windows 系统的“剪切-粘贴”操作, 引入了动态数据交换(DDE)和对象链接和嵌入(OLE)。随后, 在 1993 年引入 OLE2, 即 OLE Automation。这使得

收稿日期: 2007-08-28

作者简介: 祁明龙(1962-), 男, 博士, 副教授, 研究方向为分布式计算与网络数据库。

一个文档类型可以嵌入到另外一个文档当中。微软把这种底层的通信模式命名为组件模型(COM),并把它作为开发基于组件的单机应用程序的通用基本组织。今天的 DCOM 是 COM 的一个扩展,主要是为了开发多机应用系统,是微软公司开发的一种基于面向对象的开发和部署软件组件的编程模式。DCOM 把对象接口从应用中分离出来(类似于 CORBA),并定义二进制调用类;它也定义了接口定义语言 MIDL(Microsoft IDL)。接口定义语言编译器根据接口定义生成以 C、C++ 或者 Java 为目标语言的客户端代理和服务端桩代码<sup>[3]</sup>。现在,微软把 COM/DCOM/OLE Automation 技术统一叫做 ActiveX。DCOM 的扩展,使得 COM 处理器可以运行在不同的机器之上,包括远端对象通信、透明定位和接口到分布式安全服务。

## 2 CORBA 与 DCOM 桥接

### 2.1 互操作模型

一般地说,为了实现两个不同的对象系统之间的协作功能,必须做到的就是,不同系统中的对象之间能够相互“看到”对方,就好像这些对象存在于一个共同的对象系统里一样。具体一点说,就是客户端能够获取并调用服务器上的对象,反之亦然。例如,一个工作在 CORBA 模式下的客户端应该能够“看到”DCOM 服务器上的对象,就好像这个 DCOM 对象是一个 CORBA 对象;反之,一个工作在 DCOM 模式下的客户端应该能够“看到”CORBA 服务器上的对象,就好像这个 CORBA 对象是一个 DCOM 对象<sup>[2]</sup>。

CORBA 与 DCOM 之间的协作方式可以分为单向协作与双向协作。单方向的协作就是 CORBA 对象只能访问 DCOM 对象,DCOM 对象不能访问 CORBA 对象,或者相反;而双向的协作,则是 CORBA 对象与 DCOM 对象可以相互访问<sup>[2]</sup>。图 1 为互操作模型。

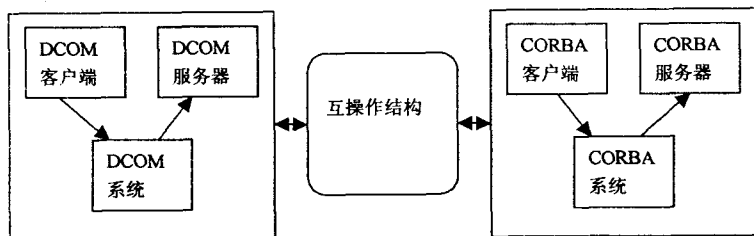


图 1 CORBA/DCOM 互操作结构

这里给出一种 CORBA 与 DCOM 桥接框架,它支持 CORBA 对象与 DCOM 对象之间的双向通信。为了实现这种桥接框架,使得 CORBA 对象与 DCOM 对象之间能够双向通信,有几个难题需要解决:第一,有效访问。如何使得客户端能够调用服务器上的方法,如何使得客户端的请求必须能够被服务器能够识别;第

二,消息的传送。客户端的请求如何才能够发送到服务器系统。CORBA 与 DCOM 都有自己的通信协议,但是这两个协议之间是互不兼容的。因此,这些问题就是这种互操作体系结构要解决的重点。

### 2.2 桥

为了实现 CORBA 与 DCOM 系统的双向桥接,这里引入“桥”的概念,“桥”是软件组件,它们位于客户端与服务器的通路之间,既可以位于客户端,又可位于服务器之上,也可以置于中间机器之上。图 2 是桥在系统中的角色<sup>[4]</sup>。

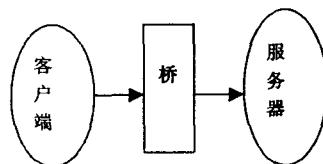


图 2 桥的角色

桥的设计被定位在应用级别,它屏蔽了 CORBA 与 DCOM 之间的不兼容性。两个系统的内部细节以及底层协议对桥是透明的,桥只是通过在不同的对象访问协议之间做出相应的转换,实现客户端访问由不同对象系统实现的服务器对象。远程调用由标准的 I-IOP(Internet Inter-ORB Protocol)来执行。在这一点上,桥与通过在客户端与服务器调用编组函数的 COM 代理对象和 CORBA 代理对象类似。

### 2.3 CORBA 与 DCOM 桥模型

对于双向通信的桥模式,需要两种形式的桥,即 DCOM-CORBA 桥与 CORBA-DCOM 桥。这两种形式的桥的功能是完全一样的,它们共同完成从客户端到服务器的交互协作。这里只以 DCOM/CORBA 桥为例进行阐述<sup>[1]</sup>(如图 3 所示)。

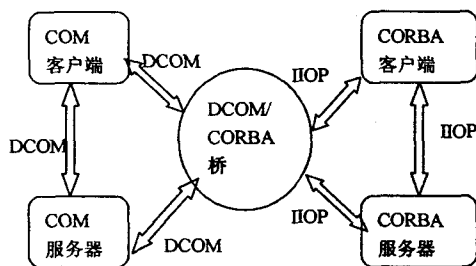


图 3 DCOM/CORBA Bridge

### 2.4 桥组件

桥的主要构成元素如图 4 所示。

#### 2.4.1 IDL 转换器

IDL 转换器的功能是把一个系统的 IDL 代码转换为另一个系统的 IDL 形式的代码。经过转换,一个系统中的对象就可以被其它的系统获取。在转换过程中会同时产生两个系统的客户端与服务端端的 IDL 的

对象定义代码。因此,IDL 转换用来生成其它系统可以获得的对象定义。IDL 转换为每个系统生成的不关联的 IDL 文件,由不同系统本地 IDL 编译器分别编译,产生本地系统的对象代码、代理代码、桩代码和编码代码。

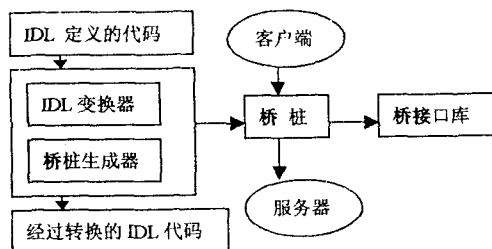


图4 桥的主要构成元素

#### 2.4.2 桩生成器

桩生成器的功能是在 IDL 转换的时候为每一个对象创建桥桩代码。桥的 IDL 生成器把代码添加到桥模版文件中,生成桥桩代码。这一过程包括转换方法参数、调用目标服务器方法和转换异常代码。桥桩代码还负责桥被加载和卸载时初始化和终止程序。

#### 2.4.3 桥接口库

桥接口库内置了通用的用于参数转换和异常处理的方法,是一个静态库。在程序的链接阶段就可以直接连接到桥代码中去。桥库简化了桥桩的生成,有助于整体结构的扩展。库的接口独立于客户端系统与服务器系统,因此,新的接口可以随时添加到库中。这些独立的接口使得桥可以被不同的编译器所编译;在其它的操作系统下可以被完全重新实现,而不用对桥桩生成过程做过多的修改。

### 2.5 桥的参数转换

对于每一个被调用的方法,桥都执行以下参数转换工作(如图5所示)<sup>[5]</sup>。

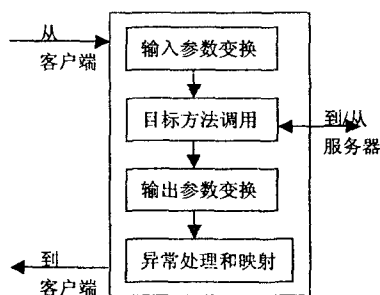


图5 参数转换流程图

首先,转换输入参数。桥把来自客户端对象系统的输入参数转换为等价的服务器对象系统能够识别的数据类型,这一转换工作包括作为参数传递的对象引用。然后,调用目标对象方法。客户端请求的方法在服务器上被调用。桥使用经过转换后的参数以及服务器需要的附加参数在服务器上调用被请求的方法。其

次,输出参数转换。如果有从服务器回传的结果参数,桥就会把这些参数转换为等价的客户端系统可以识别的数据类型。最后,异常的处理和映射。所使用方法在服务器上执行过程中,往往会有异常发生,当异常发生的时候,服务器就会捕获并处理异常,桥的异常处理模块就会对这些异常进行处理,然后映射成客户端能够识别的消息。

### 2.6 桥的实现

在这一部分,用一个例子来演示如何建立 DCOM 客户端/CORBA 服务器桥。CORBA 服务器实现一个银行自动提款机接口<sup>[2]</sup>:

```

Module BANK{
  Interface ATM{
    Enum type {CHECKS,CASH};
    Struct DepositRecord{
      String account;
      Float amount;
      Type kind;
    };
    Void deposit(in DepositRecord depositVal);
  }
}
  
```

目的就是使用 DCOM 系统的客户端来调用这个 CORBA 服务器提供的 ATM 接口方法。流程如下:

(1)ATM 接口转换。这里用 ORB 的 IDL 到 C++ 的翻译器对这个 ATM 接口进行转换,生成 CORBA 桩(stub)头文件、CORBA 桩代码、CORBA 骨架(skeleton)头文件和 CORBA 骨架代码。由桥的 IDL 转换器和桥桩生成器生成 DCOM MIDL 定义文件、桥头文件、桥客户桩以及桥服务器桩。

(2)编译 DIDL 定义文件。用微软的 MIDL 编译器编译由(1)生成 DCOM MIDL 定义文件,产生头文件(包含 ATM 接口中的方法)、代理和桩文件、接口的 GUIDS 定义文件以及类型库文件。

(3)搭建服务器、客户端和桥。利用以上过程产生的桩文件、头文件等文件分别构造服务器、客户端和桥类文件,然后分别编译与链接。

至此,所有的工作已经完成,然后就可以运行所搭建的程序,对程序进行测试。

### 3 结束语

在 CORBA 与 DCOM 桥接方面的研究中实现了 CORBA 与 DCOM 两个不同系统之间的互操作,为开发分布式应用提供了便利。当然,如果要是大型的项目开发中,为了加快开发进度,降低开发难度,往往

(下转第 111 页)

目较少,采用边界近邻最小二乘支持向量机由于需要寻找边界近邻样本,在训练时间上高于最小二乘支持向量机,但随着样本数量的增加,边界近邻最小二乘支持向量机训练时间将小于最小二乘支持向量机。所以从总体上来看,边界近邻最小二乘支持向量机在提高识别率的同时也减少了样本的训练时间。

#### 4.2 实验二

本实验的数据均来自 UCI 数据库<sup>[7]</sup>,实验数据如表 2 所示。在实验过程中,Iris 和 Letter 均采用原始数据;对于 Wine 和 Segment,由于属性值之间差异太大,故利用式(10)作了预处理。其中  $x_i$  是属性值, $\bar{x}$  是均值, $S$  是标准差。

$$x'_i = (x_i - \bar{x})/S \quad (10)$$

实验数据和实验结果分别如表 3 和表 4 所示,从实验结果可以看出运用边界最小二乘支持向量机来处理多类问题从总体上提高了样本的识别精度和识别速度。但对于大样本问题,由于样本的数量比较多,边界近邻最小二乘支持向量机产生的支持向量也比较多,这样增加了样本的学习时间。实验中可以采用聚类的方法减少类中心样本,以减少边界样本的搜寻时间来减少样本的学习时间。

表 3 实验数据列表

数据名称	训练集	测试集	类别数	属性值
Iris	75	75	3	4
Wine	118	59	3	13
Segment	1540	770	7	19
Letter	16000	4000	26	16

表 4 实验结果

数据名称	训练集		测试集	
	分类正确率(%)	学习时间(s)	分类正确率(%)	花费时间(s)
Iris	97.33	1.860	96.00	0.016
Wine	99.85	1.375	98.33	0.031
Segment	99.74	92.35	95.58	0.042
Letter	98.93	2653.4	94.30	56.38

(上接第 107 页)

都是利用商业化的桥接产品。这里对 CORBA 与 DCOM 桥接方面的研究,是为了探讨更好的桥接模式,开发更好的桥接产品,推动分布式应用的发展。

#### 参考文献:

- [1] Kraus E. Interworking Methodologies for DCOM and CORBA[D]. Tennessee: Faculty of Computer and Information Science, East Tennessee State University, 2003.

## 5 结束语

针对多分类方法中的 1-a-r(one-against-rest)方法存在由于训练样本之间的不平衡将对精度产生影响,存在不可识别区域等不足,提出了利用边界近邻最小二乘支持向量机来解决多分类问题。有效地解决了 1-a-r 方法中存在的由于训练样本之间的不平衡将对精度产生影响及存在拒分区域的问题,实验结果表明该方法的有效性。然而,文中的方法还存在一定的局限性。如在训练过程中的参数如何确定,如何减小噪声对训练精度的影响等,这些问题都有待于进一步的探索研究。

#### 参考文献:

- [1] Vapnik V. The Nature of Statistical Learning Theory[M]. New York: Springer, 1995.
- [2] Suykens J A K, Vandewalle J. Least Squares Support Vector Machine Classifiers[J]. Neural Processing Letter, 1999, 9(3): 293-300.
- [3] De Kruif B J, De Vries T J A. Pruning error minimization in least squares support vector machines[J]. IEEE Trans. Neural Networks, 2003, 14(3): 696-702.
- [4] Hoegaerts L, Suykens J A K, Vandewalle J, et al. A comparison of pruning algorithms for sparse least squares support vector machines[C]//in: Proceeding of International Conference on Neural Information Processing 2004. Calcutta, India: [s. n.], 2004: 1247-1253.
- [5] Botton L, Cortes C, Denker J, et al. Comparison of Classifier Methods: A Case Study in Hand-writing Digit Recognition [C]//International Conference on Pattern Recognition. [s. l.]: IEEE Computer Society Press, 1994: 77-87.
- [6] Krebel U. Pairwise Classification and Support Vector Machines [C]//Scholkopf B, Burges C J C, Smola A J. Advances in Kernel Methods; Support Vector Learning. MA: MIT Press, 1999: 255-268.
- [7] UCL Machine Learning Group. Elena database[EB/OL]. 2003. <http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm#stuff>.

- [2] OMG. Common Object Request Broker Architecture: Core Specification[M]. [s. l.]: OMG, 2004.
- [3] Norman R J. CORBA and DCOM: Side by Side[EB/OL]. 1998-05. [www.DistributedComputing.com](http://www.DistributedComputing.com).
- [4] 钟灿, 钟本善, 周熙襄. COM 和 CORBA 的桥接应用[J]. 电子科技大学学报, 2003, 32(2): 188-191.
- [5] Tolba M F, Fathy S K, Ismail H M. Design and Implementation of Interworking Architecture[J]. IJICIS, 2003, 3(1): 10-21.