

# 异构环境下数字图书馆的互操作研究

周 西 京

(五邑大学 江门 529020)

**摘 要** 提出了当前数字图书馆及情报信息系统在异构网络环境下互操作存在的问题,分析了 CORBA 技术的特点及其在基于 Internet 的数字图书馆方面的应用,指出 CORBA 技术是目前解决异构环境下网络互操作存在问题的较好的方案。

**关键词** CORBA 数字图书馆 异构网络环境 互操作

数字图书馆作为社会信息基础设施,由于不同组织进行建设的目的、方式、运行手段不同,从而在技术实现上采用的计算机平台、网络协议、体系结构也各不相同。为了对数字图书馆的资源利用最大化,向整个社会提供信息服务网络应用层的数字图书馆应用软件,就必须和不同底层信息基础交互,以获取用户所需要的信息,也就是需要解决已有系统间的互操作问题。这在技术上会遇到各种各样的难题,原因是已有系统的建设并非以同一种标准进行。已经有很多社会团体在不同程度上进行了数字图书馆的建设或者电子资源的积累,这些都是未来数字图书馆的资源基础。由于这种建设的开发目的、技术基础以及运行手段都各不相同,就造成了不同系统提供信息服务的操作平台、网络协议等各不相同,但作为向整个社会提供信息基础设施服务的数字图书馆建设,不可能把这些信息基础设施舍去重建,而必须充分利用这些现有的信息资源,由于数字图书馆具有分布式、异构性的特点,建设数字图书馆面临的最大问题之一就是必须在已有的信息基础设施之上设计实现更高层次的信息服务协议,使已有的各种信息服务在该层次上达到统一,从而解决异构系统间的互操作问题。

## 1 CORBA 技术

CORBA 是 OMG(Object Management Group)提出的通用对象请求代理体系结构(Common Object Request Broker Architecture)的技术规范,它是对当今快速增长的软硬件的协同工作能力的要求而提出的方案。CORBA 允许应用程序和其它的应用程序通讯,而不论它们在什么地方或者由谁来设计。CORBA 定义了接口定义语言(IDL)和应用编程接口(API),从而通过实现对象请求代理 ORB(Object Request Broker)来激活客户机/服务器的交互。它定义了如何跨越不同的 ORB 提供者而进行通信。

CORBA 的核心是 ORB,ORB 其实就是对象总线,定义了异构环境下对象透明地发送请求和接受响应的机制,所起的作用就像一个中间人一样,使各个对象能够互传请求。通过 ORB,一个客户可以很简单地使用服务器对象的方法而不论服务器是在同一机器上还是通过一个网络访问。ORB 截获调用后负责找到一个对象实现这个请求,传递参数和方法,最后返回结果。客户不用知道对象在哪里,是什么语言实现的,它的操作系统以及其它和对象接口无关的东西。CORBA 对象的互通信要以对象请求代理

为中介,这种互通信可以在多种流行通信协议(如 TCP/IP 或 IPX/SPX)之上实现。在实现 TCP/IP 上,来自于不同开发商的 ORB 用 IIOP(Internet Inter-ORB Protocol)标准协议进行通信。

ORB 能够作为中间件,其支持的机制是接口定义语言(IDL)。IDL (Interface Define Language) 说明对象的边界和接口,与对象和 ORB 无关,但将分布式服务的提供者和客户连接在一起。IDL 语言实质上提供了一种手段,通过它特定的对象实现就能告诉客户哪些操作是可用的,并且应该如何激活这些操作。IDL 语言是一种描述性语言,客户要能真正使用 IDL 源文件中所定义的对象,必须把这些对象映射到指定的程序设计语言实现的对象。这部分工作由 IDL 编译器来完成。因此,在所有的 ORB 中都有 IDL 编译器。所以 CORBA 具有平台无关性、语言无关性。

## 2 CORBA 的体系结构

常用的分布式对象技术有三种参考模型,它们分别是微软公司提出的 COM/DCOM 技术、SUN 公司提出的 JAVA/RMI 以及 OMG 的 CORBA 技术。

CORBA(Common Object Request Broker Architecture)是 OMG (Object Management Group)提出的一系列有关对象技术的规范之一,它是 OMG 发布的 OMA(Object Management Architecture)参考模型的核心——\* \* \* \* \* ORB 的功能描述与约定。

OMG 于 1989 年成立,发展至今其成员包括了除 Microsoft 的几乎所有计算机厂商,因此它提出的概念性结构 OMA 参考模型有着极广泛的支持基础。在 OMA 参考模型中定义了 CORBA 的四个主要部分:应用对象(Application Objects)、对象服务(Object Services)、公共设施(Common Facilities)和对象请求代理(Object Request Broker,简称 ORB)。其中 ORB 是 CORBA 规范的核心,它定义了 CORBA 的对象总线;应用对象是指所有以 CORBA 为运行环境的应用;对象服务定义了为分布对象所提供的系统级的基本功能;公共设施定义了能够直接被应用对象所使用的功能。

CORBA 规范给出了 ORB 的基本结构及其各部分的功能描述,它包括:接口定义语言(IDL)、静态调用接口(IDL Stub)、ORB 界面、动态调用接口(DII)、静态框架接口(Static Skeleton)、动态框架接口(DSI)、对象适配器(Object Adapter)、界面库(Interface Repository)、对象实现库(Implementation Repository)和 ORB 间互操作协议 IIOP。这些部分的关系如图 1,图中的箭头表示调用或

执行关系。

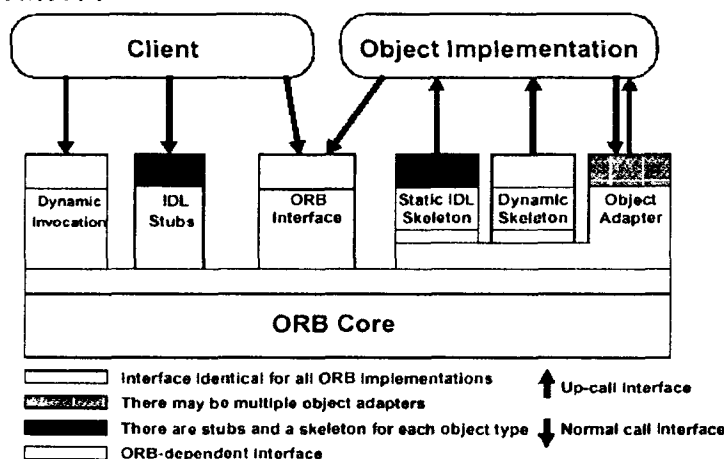


图1 单个 ORB 体系结构

图1表示了客户、服务的执行对象与 ORB 接口间的关系。ORB 能识别语言是 OMG 的接口定义语言 IDL。它可完整地确定部件的接口,即客户所需要使用接口的全部信息。IDL 定义的接口经 IDL 编译器编译后能产生客户的桩(Stub)及执行对象的框架(Skeleton)这类能与 ORB 通信的接口。

a. 接口桩 Stub。是在编译时确定的静态接口,它位于客户对象的本地,接受客户的请求,对客户来说相当于远程的执行对象,接口桩向 ORB 提交请求。b. 动态调用接口(DII)。用于编译时不能确定的请求,与接口桩作用相同。c. 框架(skeleton)。分动态框架和静态框架,它针对对象来说代表了远程客户的作用,可在本地调用执行对象服务,并与 ORB 通信。d. 对象适配器。提供执行对象可以访问 ORB 服务的基本接口,其作用是产生及解释对象引用、安全交互、登记和执行等。ORB 可以由各种方式来实现,只要满足 CORBA 规定的规范即可,具体 ORB 实现之间亦可存在接口关系。

对象的界面通过 IDL 定义,这样界面的定义可与对象的实现独立开来,最终使得对象的实现和对象的调用可以用不同的编程语言来实现。IDL 编译器将对象的 IDL 文件编译成客户方的存根(Stubs)和服务方端的框架(Skeleton)。客户方根据 IDL Stubs 使用静态方式调用对象服务,或者根据界面库中的 IDL 描述信息采用动态调用方式搜索可用的服务,找到这些服务的接口并构造使用这些服务的请求。对象实现在执行客户请求时,通过对象适配器获取 ORB 的服务。对象适配器是对象访问 ORB 服务的主要通道,它为实例化的对象服务提供运行环境,接受客户请求并传送给服务对象。此外对象适配器还负责为服务对象分配对象 ID,以及将对象类和实例化对象注册到对象实现库中。对象实现库包含了允许 ORB 查找和调用对象实现的相关信息,它是 ORB 进行对象匹配的场所。ORB 接口则是为客户方和对象实现提供几个局部性的基本服务。

CORBA 2.0 增加了 Internet Inter-ORB 协议(IIOP)来解决 ORB 互操作的问题,IIOP 是在 TCP/IP 协议之上的通用主干协议,它是 CORBA 消息交流的协议,每个 CORBA 兼容的 ORB 必须实现 IIOP 或提供一个与 IIOP 相连的半桥。

CORBA 通过 ORB 和 IDL 在客户方和服务方之间实现互操作,它将客户和服务方都抽象成对象,只是规范这些对象的接口定义,而并不关心它们功能的具体实现,客户方与服务方之间的

通信由 ORB 这个代理来完成,对象不必关心通信的细节,CORBA 技术的应用为面向对象的程序设计带来了方便。

### 3 基于 CORBA 的数字图书馆分布式系统集成方案

通常情况下,数字图书馆的用户(服务消费方)需要获取另一个数字图书馆(服务提供方)的信息,这些数字图书馆在网络层是相通的,处于局域网或广域网环境中,数字图书馆之间可以是异构的。本文给出了基于 CORBA 的分布式应用系统集成思路。

a. 各个信息消费方向 CORBA 的注册对象注册,通知 CORBA 服务器关于信息消费方的 IP 地址、端口号、对象关键字,使 CORBA 服务器获得对象实现在运输层地址信息;b. 各信息提供方通过 CORBA 服务方的对象引用达成和 CORBA 服务器的底层通信,将信息传递给 CORBA 服务器;c. 将收集到的信息通过对其参数的区分进行分类处理,分别进行保存(保存到某一特定目录或数据库中)或进行格式化转换,以便向信息消费方发布;d. 将已分类的信息送至 CORBA 的事件服务对象中的事件通道;e. 通过事件通道,采用 PUSH 或 PULL 的方式将文件传递给信息消费方。

采用 CORBA 技术方案具有如下优点:a. 屏蔽了底层的软硬件系统环境,有效、可靠;b. 采用有三层体系结构与服务消费方的注册机制,使系统具有很强的可扩展性,信息提供方和消费方增加时只需要作很少的改动;c. 一个 CORBA 服务器可以管理很多服务提供方和消费方,并且不会发生冲突;d. 由于对象引用中包括了服务器中的 IP 地址、端口号、对象关键字,整个体系可以构建在广域网上,具有很强的开放性;e. 采用 CORBA 技术开发的应用系统作为可重用的部件,可以根据需要通过 CORBA 接口任意连接,从而降低开发成本和系统维护费用。

### 4 结束语

根据数字图书馆的网络协议、应用软件平台的异构性、在地理位置上的分布性及数字图书馆之间的互操作性,分析了 CORBA 技术特点及在数字图书馆集成方面的优势,CORBA 完全可以解决数字图书馆面临的上述问题。另外应指出 CORBA + Java 的组合更具有发展潜力。Java 提供了一个概念性清晰、结构紧凑的分布式计算模型和构件互操作的方法。而 CORBA 作为一种集成技术,提供了对各种功能模型进行构件化处理并将它们集成在一起,可以扩展 Java 在网络、语言、组件。操作系统中的各种应用,Java 在简化了大型 CORBA 系统中的代码分配,其中内置的多线程和垃圾收集使编写可靠的网络对象更为方便。

### 参考文献

- Barry M. Leiner The NCSTRL Approach to Open Architecture for the Federated digital library. D-Lib Magazine, 1998; (12)
- Robert Kahn, Robert Wilensky A Framework For Distributed Digital Object Services. D-Lib Magazine, 1995; (5)
- OMG CORBA v2.4.2 <http://www.omg.org/format/01-02-33.2,2001>
- Piroz, M.. Exploit Distributed Java Computing With RMI. <http://www.ncworldmag.com,1998>
- David, R. Java RMI & CORBA A Comparison of Two Technologies. <http://www.davitreilly.com,2004-11-27>

(责编: 勤梅)