



Common Object Request Broker Architecture

面向二十一世纪的软件总线

CORBA

技术及其应用

汪 芸 博士 编著 顾冠群 审

东南大学出版社

Common Object Request Broker Architecture

面向二十一世纪的软件总线

CORBA 技术及其应用

汪 芸 博士 编著 顾冠群 审

东南大学出版社

TP3

内容提要

CORBA 规范(通用对象请求代理体系结构)是由 OMG 组织制订的规范,是新一代分布计算技术的工业规范主流之一。本书主要论述了 CORBA 技术的组成、特色、运行原理、实现技术和使用方法,通过实例加深读者对 CORBA 技术的理解和认识。本书中的所有实例均可在东南大学自行开发的 ORBUS 系统中运行。本书可作为高等院校本科生高年级或者研究生的分布计算技术课程的教材,也可作为需要了解分布计算新技术人员、CORBA 应用开发人员以及有志于从事 CORBA 技术研究和应用的人员的参考书。

图书在版编目(CIP)数据

CORBA 技术及其应用/汪芸编著. —南京:东南大学出版社,1999.4
(女博士书系)
ISBN 7-81050-454-1

I. C… II. 汪… III. 分布式处理系统-应用软件, CORBA IV. TP338

中国版本图书馆 CIP 数据核字(1999)第 11630 号

责任编辑 于 金
特邀编辑 刘友鹏

东南大学出版社出版发行

(南京四牌楼 2 号 邮编 210096 Tel: 025-7711295)

出版人: 洪焕兴

江苏省新华书店经销 常熟第八印刷厂印刷

开本: 787mm×1092mm 1/16 印张: 17 字数: 403 千

1999 年 6 月第 1 版 1999 年 6 月第 1 次印刷

印数: 1~2000 册 定价: 45.00 元

序 1

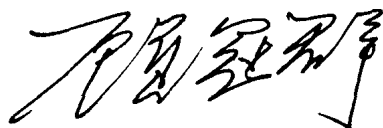
分布计算系统又称为中间件。从层次上看，它处于系统的中间层；从功能上看，向下它可以传递和处理上层用户的各种请求，向上它可以屏蔽下层的实现细节，提供各种增值服务。因此，分布计算系统起到了承上启下的作用，为用户构造分布应用提供了强有力的支持工具。

分布计算技术的不断发展与相关技术的发展是密不可分的。随着计算机网络技术和应用的发展，分布计算技术逐步成为热点。在 80 年代中期，国际标准化组织 ISO 推出了 RPC (Remote Process Call, 远程过程调用) 标准，极大地推动了分布处理技术的规范化和标准化。到了 1992 年，开放软件基金会 OSF 颁布了 DCE (Distributed Computing Environment, 分布计算环境) 标准。在这个标准中，它不仅吸收了 RPC 作为其远程调用的方式，而且对分布处理进行了抽象定义，是分布计算技术发展过程中的一个里程碑。非常可惜的是，遵从 DCE 的实现和产品不多，造成了这个标准始终未能得到广泛的应用。

对象技术的发展和成熟，给分布计算技术带来了新的发展方向。国际非盈利组织 OMG 于 1991 年制订了 CORBA (Common Object Request Broker Architecture, 通用对象请求代理体系结构) 规范，并不断增加新内容和新技术，版本也不断升级。目前，国际范围内已经有大量的企业、政府和银行等运用了 CORBA 技术，取得了显著的效益。在国内，CORBA 研究和应用也在深入。CORBA 技术的成功在于，它除了能够解决由于多个系统层次上的异构带来的“孤岛”问题，还在理论和实现技术上扩展了客户/服务器模式，使得系统具有良好的可伸缩性，便于系统的开发和升级，保护了已有投资。随着在 CORBA 技术中引入组件技术，CORBA 技术的优势将进一步得到增强。

本书是国内第一本系统地论述 CORBA 技术的中文著作。自从 1994 年由我担任课题组长、由全国十家高校和科研单位参加的国家“863/CIMS”软课题“集成框架可行性研究”开始，作者汪芸和她所在的研究小组就开始了 CORBA 技术的研究，在国家 863/CIMS 主题支持下，还实现了一个 CORBA 系统，积累了大量的理论研究和系统实现经验，为本书的写作打下了坚实的基础。本书不仅系统地阐述了 CORBA 技术的组成及其关系，而且通过实例加深读者对 CORBA 技术的理解。本书还讨论了采用 CORBA 技术如何构造应用，从理论上和实践上都给读者以指导。因此，我认为本书是一本不可多得的好书，将有力地推动 CORBA 技术在国内的研究和应用。

特作序，期望我国的分布计算技术水平能够赶上并且超过国际先进水平。



顾冠群

1999 年 5 月

序 2

The year that this book is published also marks the tenth anniversary of the Object Management Group. It has been a fruitful decade for the organization. In the 1980s distributed application programming was still a black art, requiring esoteric socket programming skills and the creation of custom application protocols. Designing and debugging the code for a new application was error-prone and time-consuming. Unsurprisingly, the result was that computer networks were always used for the same few, tried-and-tested applications, like sharing printers and sending electronic mail.

OMG was established in 1989 by computer engineers who saw the need for a new set of tools to unlock the full potential of computer networks. OMG's first published specification was for a piece of software so novel that the organization had to invent a new name for its function – “Object Request Broker”, or ORB for short. They envisioned an ORB allowing the parts of an application to be spread across separate computers, but programmed almost as easily as if they were on the same machine. OMG's own ORB design was called CORBA, for “Common Object Request Broker Architecture”. Since it was released in 1991 it has become an unqualified success, implemented by dozens of independent products and deployed in hundreds of mission-critical applications as varied as the CNN Interactive Web site, cable TV systems in Hong Kong and the IRIDIUM global mobile telephone system.

Meanwhile, as the basic CORBA technology has spread, OMG used the same open technology specification process to create specifications for essential support services for CORBA, such as transactions support, name lookup and security. More recently, the organization has begun creating domain-specific services for such diverse applications as air traffic control and medical records maintenance – and once again, all building on the same underlying CORBA technology.

As OMG enters its tenth year of creating open specifications for distributed application integration, it is my great pleasure to welcome the arrival of another milestone – this book, which to the best of my knowledge is the first Chinese text on CORBA. It arrives at an opportune moment, with increasing application development based on this important technology. I look forward to ten more years of growth in the use of CORBA, not least because of the contribution of this important text.



Andrew Watson
Vice President and Technical Director
Object Management Group

前言

记得在 1994 年春夏之交,国家“863”CIMS 专家组已经确定要实施一项关于集成框架可行性研究的软课题。当时,已经有国外企业成功应用 CORBA 技术而获得巨大经济效益,并增强企业竞争能力的报道。新的技术和机遇在向我们招手。这个软课题的实施标志着我国正式开始对集成框架技术进行跟踪和研究。集成框架技术的基础部分就是遵从 CORBA 规范的软件总线系统。

完成了以顾冠群院士为课题组长的软课题后,我们参加了由航天部二院、东南大学、上海交通大学和清华大学参加的国家“863”“九五”重大攻关项目“并行工程集成框架关键技术”,承担其中的一个子课题“并行工程企业层框架关键技术”。到 1997 年底,我们已经实现了遵从 CORBA 2.0 规范的 ORBUS 系统,在之后的一年中,把这个系统提供给其他三家单位,作为他们的开发平台。到 1998 年底,整个项目顺利地通过了国家科技部组织的验收和由航天部组织的鉴定。专家们一致认为,项目的总体水平达到了国际 90 年代中期水平。

为了推动 CORBA 技术在我国的应用,我们发布了免费的 ORBUS1.0 可执行版本。但是,用户们常常会来询问有关 CORBA 技术的知识,为没有一本好资料而烦恼,这对 CORBA 技术的推广和应用是极为不利的。于是,我们萌生了写一本关于 CORBA 技术的书籍的想法,专门系统地介绍 CORBA 技术知识和开发应用过程,这个想法得到了东南大学出版社的支持。经过大家的努力,才有今天读者能看到的这本书。

写到这里,我必须向大家介绍我所在的 CORBA 研究小组。这个小组从承担软课题开始成立,到现在已经有五年的时间了。它由顾冠群院士直接领导,以分布计算技术为主要研究内容,主要的力量是在校的博士生和硕士生,包括博士研究生谢俊清、兑继英、沈卓炜和硕士研究生杨昕、尹良钟、蔡敏、吕承宁、张恂、杨志宏、孙昌平和易建超。这是一个非常团结和活跃的集体,相互帮助,协同工作,共同探讨技术问题,畅谈对技术发展的看法。正是这个小组,在半年时间内,完成了 ORBUS 系统的开发。该系统的性能测试表明,它已经达到了国外主流产品的性能指标。他们还参加了本书部分章节的写作工作和繁琐的书稿录入、编辑和修改工作。我想如果没有他们的辛勤工作,这本书是不可能这样顺利出版的,应该说这本书是 CORBA 研究小组工作和智慧的结晶。在这里,我向他们表示深深的谢意。

我特别要感谢顾冠群院士,他不仅领导我们研究工作,而且在百忙之中为本书审稿,提出了极其宝贵的意见,并为本书作序,对我们提出了殷切的希望。同时,还要感谢国际 OMG 组织技术委员会副主席 Andrew Watson 先生,为了推动 CORBA 技术在中国的发展,特地为本书作序,表示了 OMG 组织对我们所做工作的支持和肯定。我们引以为自豪。

希望本书能够对有志于 CORBA 技术研究和应用的人们有所帮助。书中如有疏漏和不妥之处,敬请读者不吝赐教。

汪 芸

1999 年 5 月于南京东南大学

作者简介



汪芸，女，1967年5月出生于江苏省苏州市。1989年毕业于南京大学计算机科学系软件专业，获理学学士学位，后分别于1994年6月和1997年3月获得东南大学计算机科学与工程系工学硕士和工学博士学位。毕业留校后，任教于东南大学计算机科学与工程系，从事计算机网络协议、分布处理和网络服务质量等方面的研究和教学工作。先后参加国家“863”高技术项目和国家教育部科研项目7项，发表论文30余篇，其中有多篇被收录到SCI和EI检索。获1998年国家教育部科学技术进步一等奖。

联系地址：东南大学计算机科学与工程系。电子邮件：yunwang@seu.edu.cn。

目 录

序 1	顾冠群
序 2	Andrew Watson
前言	汪芸
第 1 章 CORBA 简介	1
1.1 CORBA 的发展历程	1
1.2 CORBA 的主要内容	2
1.3 CORBA 的技术特色	5
1.4 CORBA 产品一览	7
1.5 ORBUS 系统简介	9
第 2 章 一个简单的银行应用	13
2.1 基本的实现步骤	13
2.2 银行应用的 IDL 描述	13
2.3 编译 IDL 接口	14
2.4 产生的程序	15
2.5 实现接口	19
2.6 提供一个服务器	23
2.7 服务器注册	24
2.8 编写客户方主程序	24
2.9 运行描述	29
第 3 章 ORB 系统组成及其运行原理	31
3.1 ORB 系统组成结构	31
3.1.1 对象引用 (Object Reference)	32
3.1.2 客户	32
3.1.3 对象实现	33
3.1.4 ORB 核心和 ORB 接口	33
3.1.5 对象适配器和基本对象适配器 (BOA)	39
3.1.6 OMG IDL 语言和 OMG IDL 语言到编程语言的映射	42
3.1.7 IDL 桩和动态调用接口 (DII)	43
3.1.8 IDL 构架和动态构架接口 (DSI)	48
3.1.9 接口库和实现库	50
3.2 ORB 系统工作原理	54
3.2.1 ORB 系统运行过程概述	54

3.2.2 运行实例	55
3.3 ORB 间互操作机制	62
3.3.1 互操作的概念	62
3.3.2 互操作桥接机制	63
3.3.3 可互操作的对象引用 IOR	66
3.3.4 带互操作的请求执行机制	66
3.3.5 ORB 间互操作协议	67
第 4 章 IDL 语言和 IDL 编译器	70
4.1 IDL 语言	70
4.2 IDL/C++ 映射	73
4.2.1 IDL/C++ 语法结构映射	73
4.2.2 服务器方映射	95
4.3 IDL/C++ 编译器	97
4.3.1 编译器系统结构	97
4.3.2 输入处理层中的编译器前端	98
4.3.3 目标代码输出层中的编译器后端	99
4.4 IDL/Java 映射与编译器	100
4.4.1 命名	100
4.4.2 模块映射	101
4.4.3 支持类 (Holder Classes)	101
4.4.4 助手类 (Helper Classes)	102
4.4.5 基本类型映射	103
4.4.6 常量映射	105
4.4.7 枚举类型映射	106
4.4.8 结构类型映射	107
4.4.9 联合类型映射	108
4.4.10 序列类型映射	109
4.4.11 数组类型映射	110
4.4.12 接口类型映射	110
4.4.13 异常类型映射	113
4.4.14 any 类型映射	114
4.4.15 嵌套类型映射	116
4.4.16 typedef 类型映射	116
第 5 章 COM/CORBA 互操作	118
5.1 COM 和 CORBA 的比较	118
5.1.1 体系结构	119

5.1.2 编程方式	121
5.1.3 性能	128
5.2 COM/CORBA 互操作原理	129
5.2.1 COM/CORBA 互操作模型	129
5.2.2 接口映射	130
5.2.3 接口映射示例	131
5.2.4 接口组成映射	141
5.2.5 对象绑定和生命周期	144
5.2.6 异常处理映射	146
5.2.7 互操作接口	147
5.2.8 DCOM/CORBA 互操作	149
5.3 COM/CORBA 互操作产品介绍	151
5.3.1 Orbix ActiveX	151
5.3.2 Orbix COMet	152
5.3.3 Orbus COMBridge	152
5.4 COM/CORBA 互操作示例	153
5.4.1 COM 客户访问 CORBA 对象	153
5.4.2 CORBA 客户访问 COM 对象	159
第 6 章 CORBA 对象服务	160
6.1 CORBA 对象服务的概念	160
6.2 主要的 CORBA 公共对象服务	161
6.2.1 名录服务	161
6.2.2 事件服务	170
6.2.3 安全服务	177
6.3 其他的 CORBA 公共对象服务	188
第 7 章 CORBA, WWW 和 Java	191
7.1 CORBA 和 WWW	191
7.1.1 现阶段的 WWW 体系结构	191
7.1.2 分布对象技术与 WWW	193
7.2 CORBA 与 Java	196
7.2.1 Java 简介	196
7.2.2 CORBA 与 Java 的比较	198
7.2.3 Java ORB	199
7.3 构建 Web 上的 CORBA 应用	203
7.3.1 开发步骤	203
7.3.2 简单 WWW 银行系统的开发	204

7.3.3 开发中应注意的问题	212
第8章 CORBA 技术的应用.....	215
8.1 几种基于 CORBA 技术的体系结构.....	215
8.1.1 Oracle 公司的 NCA 体系结构	215
8.1.2 Sybase 公司的 ACA 体系结构.....	217
8.2 CORBA 技术在制造型企业中的应用.....	218
8.2.1 OMG 制造业技术委员会的工作	219
8.2.2 PTC 公司的 Windchill 体系结构	220
8.2.3 SEMANTECH 组织的 CIM 框架	222
8.3 CORBA 技术在电信领域中的应用.....	223
8.3.1 OMG 电信特别兴趣小组的工作	223
8.3.2 CORBA 技术在智能网络中的应用.....	223
8.3.3 CORBA 技术在电信网络管理中的应用.....	225
8.4 CORBA 技术在其他领域中的应用.....	228
8.5 CORBA 应用的构建	229
第9章 CORBA 技术的发展.....	233
9.1 实时 CORBA 技术	233
9.1.1 原始线程优先级	233
9.1.2 CORBA 优先级	234
9.1.3 CORBA 优先级映射	235
9.1.4 互斥接口	236
9.1.5 服务器方的配置	236
9.1.6 客户方配置	239
9.2 CORBA 环境中的异步消息技术.....	240
9.2.1 异步方法调用模型	240
9.2.2 与时间无关的方法调用模型	247
9.2.3 消息服务质量	247
附录: 术语索引	251
参考文献	257

第1章 CORBA简介

随着Internet的广泛运用,将应用扩展到局域网、广域网甚至Internet上已成为用户的普遍需求,这样,分布计算成了新的热点。在分布计算环境中,异构性是一个十分明显的特点。一个典型的分布环境包括有大型主机、UNIX工作站和PC机,各种机器所采用的操作系统和网络通信协议也是千差万别。在这样的异构环境下实现信息和软件资源的共享将十分困难,而一个健壮分布计算框架将为分布应用软件的开发带来极大的好处。自80年代末期以来,对应用软件的可扩展性和可重用性的要求也使得面向对象的分析、设计和编程技术得到了广泛的运用。将面向对象的思想应用到分布环境中,分布对象计算(Distributed Object Computing,简称DOC)的概念就应运而生了。目前,OMG组织制订的CORBA(Common Object Request Broker Architecture,通用对象请求代理体系结构)标准与Microsoft的COM/DCOM(Component Object Model / Distributed COM,组件对象模型/分布组件对象模型)标准、Sun公司的Java RMI(Java Remote Method Invocation,Java远程方法调用)标准一起形成了分布对象计算领域的主流。本书将着重探讨CORBA技术的组成、特色、运行原理、实现技术和使用方法,对CORBA技术和其相关技术进行分析比较,并研究CORBA技术的新发展。

1.1 CORBA的发展历程

OMG(Object Management Group)组织是一个国际性的非盈利组织。到1998年初已经拥有会员830多个,其中包括著名的信息系统开发商和软件开发公司如IBM、HP、DEC等,还包括许多研究机构和大学等成员。目前,OMG在中国有两个成员,东南大学是我国大陆唯一的成员,台湾Acer公司也是其正式成员。

OMG组织成立于1989年,其发起单位有BNR欧洲有限公司、DEC公司、Expersoft公司、HP公司、IBM公司、ICL公司、IONA技术有限公司、NCR公司、Novell USG公司、Object Design公司、SUN Microsystem公司和SunSoft公司。该组织的工作职责是,为应用开发提供一个公共框架,制订工业指南和对象管理规范,加快对象技术的发展。其主要技术目标是,使得基于对象的软件在分布异构环境下具有良好的可重用性、可移植性和互操作性,从而能够在由多种主流硬件平台上运行多种操作系统构成的异构分布环境中,方便地建立异构分布应用系统。

为了实现上述目标,OMG组织成立后不久就制订了OMA(Object Management Architecture,对象管理体系结构)参考模型。该模型描述了OMG的规范所遵循的概念化的基础结构,其核心部分是ORB(Object Request Broker,对象请求代理)。ORB提供了一种机制,通过这种机制,对象可以透明地发出请求和接收响应。基于ORB机制

就可以充分利用分布的、可以互操作的对象构造可以互操作的应用。这种优势是明显的，用户可以在不了解实现交互细节的情况下，建立共享资源的应用。

针对ORB，OMG组织制订了CORBA规范，并于1991年颁布了CORBA规范1.1版本，而后经历了1.2版本，又于1995年颁布了CORBA规范2.0版本，该版本对前一版本内容作了一定修改。OMG组织在1997年颁布了CORBA规范2.1版本。该版本增加了COM/CORBA互操作规范。1998年2月CORBA 2.2规范问世，该版本增添了POA（Portable Object Adapter，可移植的对象适配器）和IDL到Java语言的映射规范。与此同时，OMA体系结构中增加了领域接口这一部分，表明OMG组织已经从最初的制订CORBA内部运行方式转向开始注重CORBA技术与应用相结合的问题。CORBA公共服务规范也作了相应的修订。

目前即将发布CORBA 3.0规范，该规范采用支持Enterprise JavaBeans（EJB）规范的对象封装方案，使EJB成为CORBA的组件方案。CORBA 3.0对象能动态加载各种组件以及传输、事务和邮件交换协议。这就意味着，如果与CORBA 3.0规范相容的服务器遇到EJB，它就能在运行时加载Java虚拟机并运行这些对象。CORBA 3.0规范还包括一些新特性：将应用程序工具和环境用于CORBA组件的设计；支持CORBA对象在异步实时信息队列中进行传输；采用服务质量（QoS）技术规范；增加新的技术规范，以便实现与传统环境的结合等。

CORBA技术本身的发展与CORBA技术应用的发展是不可分割的，CORBA技术的广泛深入应用推动了CORBA技术的不断成熟，而CORBA技术的进一步完善又促进CORBA技术应用的实施和推广。

本书中，如果不做特别说明，CORBA规范是指CORBA 2.0规范。

1.2 CORBA的主要内容

OMA参考模型从总体上抽象地描述了OMG组织推出的面向对象技术所包含的内容，以及模型中各组成部分之间的关系。OMA参考模型如图1-1所示。

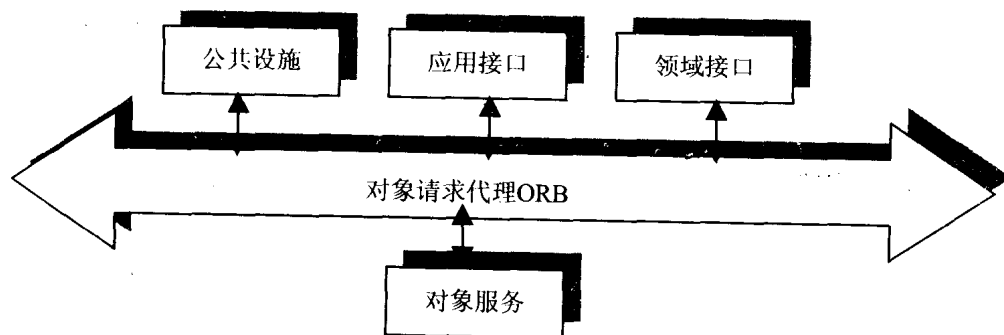


图1-1 OMA参考模型

OMA体系结构由五部分组成：

(1) 对象请求代理ORB (Object Request Broker)

负责对象在分布环境中透明地收发请求和响应,它是构建分布对象应用、在异构或同构环境下实现应用间互操作的基础。

(2) 对象服务 (Object Services)

是为使用 and 实现对象而提供的基本服务集合。在构建任何分布应用时,经常会用到这些服务,而且这些服务独立于应用领域。例如生命周期服务定义了对对象的创建、删除、拷贝和移动的方法;但是它并不指明在应用中如何实现这些对象。

针对对象服务,OMG组织制订了CORBA服务(CORBAServices: Common Object Services Specification)规范,简称为COSS规范。COSS规范由一组接口(Interface)和服务行为描述构成,其接口一般使用OMG IDL语言描述。目前的COSS规范主要包括以下内容:名录服务(Naming Service)、事件服务(Event Service)、永久对象服务(Persistent Object Service)、生命周期服务(Life Cycle Service)、并行控制服务(Concurrency Control Service)、外部化服务(Externalization Service)、关系服务(Relationship Service)、事务服务(Transaction Service)、查询服务(Query Service)、许可证服务(Licensing Service)、特征服务(Property Service)、时间服务(Time Service)、交易服务(Trading Service)和安全服务(Security Service)。由此可见,COSS规范几乎包含了分布系统和面向对象系统的各个方面,它的每个组成部分也都非常复杂和全面。

(3) 公共设施 (Common Facilities)

是向终端用户应用提供的一组共享服务接口,但这组服务不像对象服务那样基本。例如系统管理、组合文档和电子邮件服务就可以归入公共设施中。

(4) 应用接口 (Application Interfaces)

是由销售商提供的、可控制其接口的产品,它相应于传统的应用表示。因此OMG组织不对它们做标准化工作。应用接口处于参考模型的最高层。

(5) 领域接口 (Domain Interfaces)

是为应用领域服务而提供的接口。例如,现在OMG组织为PDM (Product Data Management, 产品数据管理)应用制订了规范,在该规范中定义了8个使能器,描述了12个功能模块及其接口,明确了PDM系统内部以及PDM系统与其应用系统之间交互的方式。

CORBA规范是针对OMA参考模型中的对象请求代理ORB而制订的。CORBA规范定义了IDL语言 (Interface Definition Language, 接口定义语言)及映射、单个ORB和ORB间互操作机制。随着CORBA规范版本的升级,其内容也不断丰富。单个ORB的体系结构如图1-2所示。

在该体系结构中,描述了以下主要的内容:

(1) IDL语言通过说明对象的接口来定义对象。一个接口包括一组命名的操作和相应于这些操作的参数。注意,虽然IDL语言提供了用于描述被ORB操纵的对象的概念框架,但是ORB在运行时并不需要IDL源代码。只要桩例程或者运行状态下接口库中的等价信息是可用的,ORB就能通过特定的方式正确地完成其功能。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编译器完成。OMG IDL语言的语法类似于C++。CORBA 2.0规范中定义了OMG IDL语言到具体编程语言的映射，包括OMG IDL语言到C、C++和SmallTalk语言的映射。OMG IDL语言到Ada和COBOL语言的映射定义在CORBA 2.1规范中，到Java的映射定义在CORBA 2.2规范中。

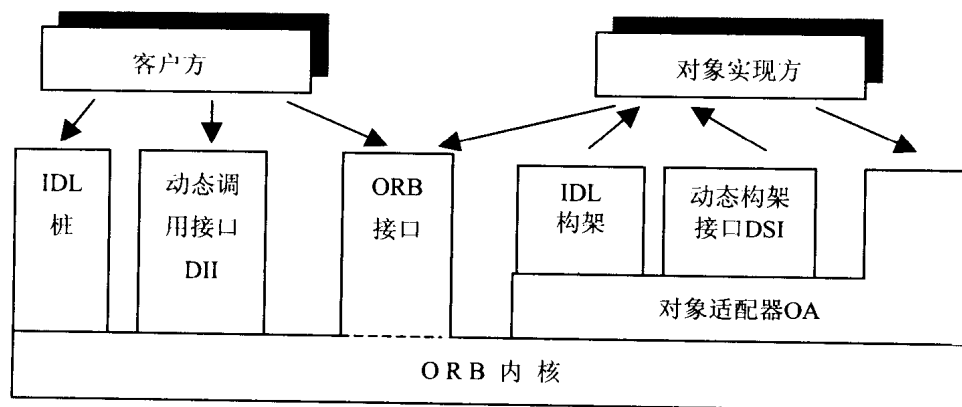


图1-2 单个ORB体系结构

(2) ORB核心提供了客户—对象实现间实现透明通信的方法，它可以屏蔽对象实现位置、实现方式、状态和通信机制等细节以及不同实现间可能存在的差异。

(3) 对象适配器位于ORB核心和对象实现之间，它负责服务对象的注册、对象引用的创建和解释、对象实现的服务进程的激活和去活、对象实现的激活和去活以及客户请求的分发。

(4) IDL桩为客户提供了静态调用方式，IDL构架为客户提供了静态实现方式。IDL编译器编译描述服务对象接口的IDL文件，生成对应于具体编程语言的IDL桩和IDL构架程序。IDL桩负责把用户的请求进行编码，发送到对象实现端，并对接收到的处理结果进行解释，把结果或异常信息返回给用户；IDL构架对用户请求进行解码，定位所请求的对象的方法，执行该方法，并把执行结果或异常信息编码后发送回客户。

(5) 动态调用接口DII (Dynamic Invocation Interface) 和动态构架接口DSI (Dynamic Skeleton Interface) 提供了动态调用方式和动态实现方式。某些情况下客户预先不知道服务对象的接口信息，需要通过查询或者采用其他的手段获得服务对象的接口描述信息，然后使用DII动态调用ORB核心接口的方法来构造客户请求并发送到对象实现。在对象实现方可以使用DSI动态分发用户请求的机制，以便动态地处理客户方的请求。

客户和对象实现所采用的方式并不一定要一一对应，也就是说，客户方支持的静态和动态两种调用方式，对象实现方支持的静态和动态两种实现方式，经过组合后得到的四种方式都可能出现。例如，客户方可能使用静态调用方式，而对象实现方使用动态构架接口，反之亦然。

(6) 在动态方式下, 需要查询相应的服务对象的接口描述信息 (在静态方式下, 这些信息由IDL文件来描述), 这些信息由接口库提供。接口库通常以IDL描述文件为其输入, 将接口描述信息进行处理后存放在文件、数据库或者其他形式的存储机制中, 并提供一组标准的调用接口供客户查询使用。服务对象实现的描述信息也由接口库提供。

(7) CORBA2.0规范还为不同厂商ORB系统之间实现互操作制订了规范。它定义了GIOP (General Inter-ORB Protocol, 通用ORB间通信协议), 对数据编码的格式、消息格式和对传输协议的要求都作出了详细的规定。由于GIOP是一个抽象的协议, 它必须映射到具体的传输协议和传输机制中去, CORBA2.0中支持GIOP到TCP/IP协议 (Internet Inter-ORB Protocol, 简称IIOP) 的映射。同时, 在CORBA规范中还定义了用于特定应用环境的协议ESIOP (Environment Specific Inter-ORB Protocol, 简称ESIOP)。ESIOP协议也需要映射到指定的环境, 已定义的映射是DCE-CIOP (DCE Common Inter-ORB Protocol)。

1.3 CORBA的技术特色

CORBA规范的产生有其特定的技术背景, 主要有以下几个方面:

(1) 面向对象技术的兴起

面向对象技术不同于传统的设计和实现技术, 它把数据以及相关的方法封装在一起, 构成对象。通过封装的方法更加精细地描述数据以及方法的作用域, 使用继承特性来扩展这些作用域。再加上对象引用和对象实现的分离, 使得由于对象的数据和方法改变而造成的影响, 在不涉及引用方法的情况下, 局限于该对象的本身, 从而大大方便了软件对象的修改和扩展。对象的继承性使得可以利用已有对象的功能来构造新的对象, 有效地提高软件的开发效率。因此, 面向对象技术的发展, 在提高软件可重用性和软件开发质量方面起到了积极的推动作用。

(2) 客户/服务器模式的普遍应用

这种计算模式可使得客户尽量少了解实现细节, 从而便于客户应用的开发。该模式具有不对称性, 即服务器不会主动询问客户的请求, 而是被动地执行指定服务。客户/服务器模式适用于由不同计算能力的计算机和设备组成的网络环境。

(3) 集成已有系统及屏蔽通信和实现细节的需求

一般地说, 用户已经积累了一批极具价值的系统, 通常这些系统改造和重写的代价很高, 所以在构造新系统时, 用户希望能够利用这些资源。因此, 如何有效地将已有系统集成或移植到新系统中, 并且在将来尽可能使它们不再成为“孤岛”系统, 是用户应用开发中提出的迫切需求。

(4) 现有分布处理机制和方法存在着不足之处

现有典型的分布处理系统和标准有RPC (远程过程调用) 和DCE (分布计算环境)

等。它们虽然是按照客户/服务器模式构造的系统，但是客户程序和服务器程序之间的调用关系是静态的，基于它们所构造的系统也是静态的，并且可以使用的编程语言必须与RPC和DCE实现语言相一致，从而限制了用户的可选语言范围。

CORBA技术在充分发挥新技术的优势和克服已有方法缺陷的基础上，引入了自身的新特色，使之成为新一代的分布计算技术中有影响力的工业规范。

CORBA规范的技术特点是：

(1) 引入了代理(Broker)的概念。一个代理至少可以有三个方面的作用：完成对客户方提出的抽象服务请求的映射；自动发现和寻找服务器；自动设定路由，实现到服务器方的执行。这样，用户在编制客户方程序时就可以避免了解过多的细节，而只要完整地定义和说明客户需要完成的任务和目标。

用代理来处理客户/服务器模式环境下应用或者应用系统之间交互的信息，称为请求(Requests)和响应(Response)。代理根据客户方的请求，选择一个或若干个服务器进行处理，使客户方无需考虑服务器方的处理细节，只需最后把服务器处理的结果作为对客户方请求的响应通知客户。客户方和服务器方相互独立无关，使得软件易于修改、移植和维护。

(2) 所实现的客户方程序与服务器方程序的完全分离。这与以面向过程调用机制为基础的客户/服务器模式存在着很大的不同。客户将不再同服务器发生直接的联系，而仅仅需要同代理进行交互。客户与服务器之间就可以有更加灵活的关系存在。在调用方式保持不变的前提下，服务器可以自由地修改或者升级，客户方程序也可以自由地按照要求进行更改，而无需通知对方。

(3) 将分布计算同面向对象的概念相互结合。这样，可以提高软件重用率、控制冗余度等，由此带来诸多好处。

(4) 提供了软件总线的机制。所谓软件总线是指CORBA系统定义了一组接口规范，任何应用程序、软件系统或工具只要具有与该接口规范相符合的接口定义，就能方便地集成到CORBA系统中，而这个接口规范独立于任何实现语言和环境。也就是说，CORBA规范支持服务对象的位置透明性、实现透明性、执行状态透明性和通信机制透明性，只要按照OMG IDL接口描述语言对服务对象所提供的服务进行描述，客户应用和服务对象之间就可以透明地交互运行。应用软件或构件能够在“软件总线”(Software Bus)之上实现“即插即用”(Plug & Play)。

(5) 分层的设计原则和实现方式。CORBA规范仅定义了ORB中需要用到的基本对象，封装了相应的属性和方法。而面向应用的对象定义则可以在OMA的应用对象、领域对象或应用开发环境中逐步分层定义和实现，使CORBA系统核心始终是一个精炼的实体，而各种复杂功能和应用可以由核心扩展和延伸。

CORBA规范充分利用了现今各种技术发展的最新成果，将面向对象的概念糅合到分布计算中，定义了一组与实现无关的接口方式，引入代理机制分离客户和服务器，使得CORBA规范成为开放的、基于客户/服务器模式的、面向对象的分布计算的工业标准。