

Enterprise CORBA



软件开发技术丛书

CORBA

企业解决方案

(日) Dirk Slama

(美) Jason Garbis 著

(澳) Perry Russell

李师贤 郑红 吴涛 等译



机械工业出版社
China Machine Press

Prentice Hall PTR

00107648

TP316.4
03

软件开发技术丛书



CORBA 企业解决方案

(日) Dirk Slama

(美) Jason Garbis 著

(澳) Perry Russell

李师贤 郑红 吴涛 等译



机械工业出版社
China Machine Press

CORBA是分布式对象计算领域的主流技术之一。本书三位原作者在CORBA技术的应用上具有丰富的经验,并对CORBA中的接口定义、安全性、实体命名、事务、通信选择、两层或三层体系结构、数据库集成、容错性、系统开发与管理等方面有着深刻的了解。本书全面介绍基于CORBA的企业信息系统开发的各个方面,为系统设计人员构建真正大规模企业级CORBA系统提供专业的指导。本书适合于高级软件开发人员、大学计算机专业研究生和CORBA研究人员阅读参考。

Dirk Slama, Jason Garbis, Perry Russell: Enterprise CORBA.

Authorized translation from the English language edition published by Prentice Hall PTR.

Copyright © 1999 Dirk Slama, Jason Garbis and Perry Russell. Published by Prentice Hall PTR.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2000 by China Machine Press.

本书中文简体字版由美国Prentice Hall PTR授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号:图字:01-1999-2350

图书在版编目(CIP)数据:

CORBA企业解决方案/(日)斯拉摩,(美)哥毕斯,(澳)罗塞尔著;李师贤等译.-北京:机械工业出版社,2001.1

(软件开发技术丛书)

书名原文:Enterprise CORBA

ISBN 7-111-07565-X

3542/11

I.C… II.①斯… ②哥… ③罗… ④李… III.分布式处理系统-应用软件
IV.TP316.4

中国版本图书馆CIP数据核字(2000)第50961号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:赵红燕

北京市密云县印刷厂印刷·新华书店北京发行所发行

2001年1月第1版第1次印刷

787mm×1092mm 1/16·18印张

印数:0 001-5 000册

定价:35.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译 者 序

CORBA以其平台独立性和语言独立性，在异构、分布式环境下占据优势，它可使应用程序开发者更多地关心其本身的应用领域开发，而较少地考虑低级的分布式系统程序设计。CORBA将对象接口的定义和对象的实现分开，允许对象用不同的程序语言编程实现，以实现大规模分布式系统。

本书主要面向对CORBA已有基本了解，并希望进一步深入研究CORBA系统开发实际需求的读者，以及通过使用CORBA技术进行大规模企业级应用系统开发的程序员和设计者。

本书尽可能以CORBA规范为中心，力图实用，提供的指导和解决方案是针对大规模分布式系统开发的实际需求，并着重于使用现有的可用技术。书中论述了与建立企业级CORBA系统相关的各个方面，同时也提供了相应的一些实例。

参加本书翻译的有李师贤、郑红、吴涛、王迪、刘鸥、马壮展、李智，全书由李师贤教授审校。由于译者自身的知识局限，译文中不足之处在所难免，欢迎读者批评指正。

2000年8月

序 言

我们假定用户需要在一复杂的软硬件环境中编写应用程序，或是需要把一整套组件/应用程序集成到一灵活的企业级系统中。我们还假定用户强烈要求自己的应用程序能运行起来，或者自己的系统能够集成起来，但对于重新创建各个部件却不感兴趣。这样，用户就需要得到尽可能多的帮助，帮助的形式既可以是能购买到的可实现软件，也可以是能采用的技术。

也许这种复杂性是由于用户的应用程序/企业必须运行在分布式硬件系统上而产生的，因此必须解决网络带来的边界问题。又或许用户不得不处理由于不同的组件用不同的语言编写所引起的问题；又或许这些组件需要运行在不同的操作系统上。又或许用户（作为世界上唯一的建筑师或经理）正处理一个完全同种类的环境，但应用程序/组件必须越过不同部门或公司的边界进行访问。

如果真的出现以上情况，用户就很可能要用到对象请求代理来帮助越过这些边界——而实际上所用的代理可能就遵循着CORBA标准。使用CORBA的原因在于系统中的组件能越过这些边界进行相互通信，而用不着编写一个软件层来提供越过边界的桥梁。CORBA提出的不止是基本的通信需要，还有其他很多的内容。就像在点到点通信中，CORBA也能提供变化，如同多对多通信和存储转发一样。对象可做成安全的；对象可以存储在数据库中；数据库可以越界访问；在一单独的事务中，多个数据库可自动更新；系统可以在不同的层次上进行监控和管理；对象可以通过名字或是用户希望的属性来寻找。

那么，本书在哪些方面可以起到作用呢？分布式系统研究大概始于1980年或更早些时候，在接近二十年的时间内，我们对它的理解不断加深。这种理解实际是集中在如接口定义、安全性、实体命名、事务、通信选择、两层或三层体系结构、数据库集成、容错性、系统管理等领域。用户如果对自己系统中重要的方面有着深刻的理解，那么就能更快更好地实现该系统。本书涉及了以上的所有领域，另外，对于CORBA环境，还给出了采取什么方法比较好的实践性建议。

Dirk、Jason和Perry对这些领域有着深刻的了解。他们研究过调查结果，还担任过多个与这些问题相关项目的顾问。他们的贡献在论述了与现代系统相关的各个方面，并把这些方面依次和CORBA框架联系起来。CORBA在系统中很容易用来克服多种限制，本书描述了用户想要应用层上用到的技术，以便最好地利用CORBA，并使自己的系统尽可能简单地实现。

CORBA的发展方向何在？早期（到90年代中期），CORBA的目标是提供跨越边界的框架。第二个阶段（到1998年），内容有所丰富（安全性、事务、多通信、数据库集成等），并且在CORBA中还集成了很多其他技术。因此，CORBA在使用了COM的环境中同样工作良好；它采纳了Java作为首要语言，并加入了其他设施以允许它在企业内使用；而且CORBA能安全地越过防火墙使用。那么往后几年的目标是什么呢？

首先，和其他技术的集成还会继续。EJB在这里很重要——或更具体地说，是在Java环境中

持续加入功能设施，以允许编写多特征的客户机和服务器。CORBA和标准企业应用程序的集成也会作为有用的功能设施提供，而不是为每个应用程序编写代码。

其次，CORBA会变得越来越容易使用，现在出现的程序员生产工具就是明证。在这里一个关键领域是客户机和服务器或至少是它们中的一部分基于脚本的生成。C++和Java代码可以以这种方式产生，这就减少了人们必须编写的和应用相关的代码数量。使用工具的其他优点在于让系统管理者有更多的监控和控制设施可用。

第三，组件这个概念变得越来越重要。组件是一个对象，里面只含有业务级的代码，而没有和安全性、事务、命名等方面相关的代码。因此，一个组件可以给定不同的安全属性而不用改变对象的代码，在某种意义上，它可以“融”入不同的安全环境而不用改变它的代码，即使这些安全环境是以不同的方式实现的。类似地，组件是否是事务性的，不是由组件本身决定，而是由它所融入的环境（或容器）来决定。

业务级代码和基础级代码的分离是提高生产率的一个关键方面。这样做的一个好处是软件在所需执行的环境中的变化可以留待以后证明。同样的组件可以融入一纯CORBA容器、或是一EJB容器、或是一COM容器。这样，如果组件融入一COM容器中，它可以自己用COM来进行通信，也可以在使用了自动桥梁的其他环境中进行通信。如果它融入到一CORBA容器中，CORBA就是它自身的通信环境，就需要一自动的桥梁来从COM中访问它。当这些发生时，业务级的代码不用改变。

这些改善都在进行当中，并会减少跨越边界的软件的编写任务。自动化并不会减少对有经验的软件工程师和设计师的需要，他们对分布式计算的重要技术的理解，以及在不同的情况下如何找出最适合的技术仍然是很重要的。我非常感谢本书的三位作者，他们花了大量时间，不辞劳苦地把这些技术和CORBA关联起来，并把多年的研究和开发结果公诸于众，书中没有多余的空话。此外，他们还指出了这些技术的相关发展方向。

Sean Baker

IONA Technologies的创始人之一

前 言

欢迎阅读《CORBA企业解决方案》。本书的主要目的是为用户提供建立真正大规模的企业级CORBA系统的指导。显然，分布式系统工程是一复杂的课题，不可能提供一个单独的、通用的模式来立即解决所有的问题。通常，并不存在“最好”的解决方案，解决方案的选择取决于很多因素——技术的和非技术的。本书基于我们三位作者在近几年中所获得的经验，这些经验来自于对世界各地的许多CORBA用户进行的培训和提供的咨询，所涉及的企业分别有制造业、金融、电信和交通。我们希望本书能够覆盖和实际项目相关的大多数主题，而且为用户提供的指导和解决方案不是抽象或学术性的，而是集中于信息系统开发的实际需求。

完成类似于本书这样的项目，即要覆盖从现实世界角度看来如此大型而又复杂的问题，会有两个普遍性的问题。第一，对于单独一个人（或即使是三个人）来说，不可能了解该领域的每一个细节，并把学术见解和面向效果的现实世界经验结合起来。第二，企业发展如此迅速，以致于很难写出一本长时间不过时的书。

最后，本书面向的是具有使用CORBA必需知识的程序员和设计者。即，读者要有自己建立CORBA系统的经验，并对事情有自己的看法。

为了向读者提供与我们及其他读者交流的机会，我们建立了一个“CORBA企业解决方案”的网站：www.middlewarehouse.com。这个站点以布告栏的方式构建，设计为对企业级CORBA进行更深入讨论的平台。希望这个主页还能解决上面提到的两个问题：提供对各单独问题的更详细的技术探讨和最新的信息。这个网站不断更新，将变成一个与中间件和企业系统开发相关的知识源。请和作者相会于：www.middlewarehouse.com。

本书是如何组织的

本书分为五个部分，各有不同的侧重点：

- 第一部分 基础。
- 第二部分 核心服务。
- 第三部分 数据库集成和事务处理。
- 第四部分 可伸缩性问题。
- 第五部分 工程化CORBA系统。

下面分别给出对这五部分的简要概述。

第一部分：基础

本书假定读者早已对CORBA有了扎实的掌握，不打算再对CORBA体系结构作详细介绍。在第一部分“基础”中，我们完成了两件事。首先，我们提供了核心ORB规范和高层CORBA服务的关键性回顾，因为这是为建立企业级CORBA系统提供技术基础的框架（第2章和第3章）。

其次，我们提供了理解本书的一些基础性信息。特别是，我们引入了StockWatch系统，这是在本书中作为实例系统而使用的（第4章）。最后，我们讨论了IDL设计的性能实质，因为这对于本书余下章节的大部分讨论有着重要的影响（第5章）。

第二部分：核心服务

某些CORBA服务在建立企业级CORBA系统中比其他服务更为重要。在第二部分“核心服务”中，我们把重点放在这些最重要服务中的某些服务上。第6章讨论如何在CORBA系统中定位对象和服务及使用CORBA命名服务和CORBA交易服务等。第7章对CORBA和消息接发作一全面的讨论。这包括了基于ORB的单目和多目发送，还有如事件服务和通知服务等高级服务。第8章讨论如何在CORBA系统中设计和实现安全策略。

第三部分：数据库集成和事务处理

输出存储在数据库中的信息也许是CORBA最常见的用法^①。第三部分“数据库集成和事务处理”讨论了这一主题。首先，我们提供了对对象持久性的一般概述（第9章）。然后，基于独立于CORBA的对象持久性的介绍，再着重于讨论CORBA对象的持久性（第10章）。如果没有对CORBA和事务处理的讨论，CORBA和数据库集成的讨论就不能完成。在对CORBA环境中的事务进行了一般的讨论后（第11章），我们着重于分布式事务处理（第12章）。这包括了对CORBA对象事务服务的讨论。我们以对用户会话的讨论（第13章）来结束第三部分。这一章涉及几个高深的课题，包括长事务和短事务以及乐观加锁和悲观加锁的对比。

第四部分：可伸缩性问题

CORBA规范提供了很多特性，以用于允许实现非常大规模的分布式对象系统。第14章，解释了如何利用其中某些特征来扩大CORBA服务器的伸缩性。特别是，本章覆盖了内存管理策略、连接管理和多线程。第四部分“可伸缩性问题”还覆盖了和建立大规模CORBA系统相关的一些重要问题，而这些CORBA系统目前还没有被现有的CORBA服务规范提到过。特别是，它包括了负载平衡（第15章）和容错性（第16章），最后，系统管理和维护（第17章）对大规模系统也是很重要的。

第五部分：工程化CORBA系统

本书的最后部分描述了基于系统开发的CORBA系统工程化内容。我们回顾了本书前面部分的技术性讨论，并接下来讨论工程化过程的重要性（第18章）。一个重要的结论是传统的面向对象分析和设计并不能一对一地映射到分布式对象计算。我们解释了传统的OO开发过程如何进行调整以反映分布式对象计算的细节。另一个和系统工程化相关的主题是自动化，在第19章中提及，这一章还包括了对与CORBA相关的CASE工具、代码生成和过程向导的讨论。

① CORBA服务中和这个复杂问题相关的显然就是核心服务。因为这是一重要而又复杂的课题，我们决定把对数据库集成和事务处理的讨论分开进行，都放在核心服务之后。

CORBA一致性

我们尽力使这本书以CORBA为中心，而不是着重于特定的ORB实现。在很多场合中，我们认为实现了这一目标。本书力图实用，着重于现今可用的技术。在某些情况下，我们提供了一些与供应商有关、和CORBA不一致的技术或特征的讨论，这通常是因为我们觉得这种讨论对纯CORBA一致性有一定价值（而且很多情况下，CORBA规范会逐渐覆盖这方面）。我们三位作者都为IONA Technologies专业服务部门工作。因此，很自然地，读者会找到一些与IONA特定解决方案相关的讨论。但是，我们希望已经把这些情况控制到最少，并尽可能保持CORBA的一致性。

目 录

译者序
序言
前言

第一部分 基 础

第1章 绪论	1
1.1 概述	2
1.2 简要的术语表	2
1.3 图形符号	3
第2章 CORBA回顾	6
2.1 CORBA的演变	6
2.2 BOA代和POA代	7
2.2.1 BOA代	7
2.2.2 POA代	7
2.3 激发生命周期	7
2.3.1 CORBA请求	7
2.3.2 激发生命周期: 评估标准	9
2.3.3 激发生命周期: 评估BOA代	10
2.3.4 激发生命周期: 评估POA代	13
2.4 对象生命周期	17
2.4.1 CORBA对象	17
2.4.2 对象生命周期事件	18
2.4.3 早期绑定和后期绑定	19
2.4.4 CORBA对象实现的分类	20
2.4.5 CORBA对象实现和内存管理	20
2.4.6 对象生命周期: 评估准则	21
2.4.7 对象生命周期: 评估BOA代	21
2.4.8 对象生命周期: 评估POA代	23
2.5 小结	25
第3章 CORBA服务回顾	26
3.1 核心服务	26
3.2 数据库集成和事务处理	27

3.3 可伸缩性问题	28
3.4 小结	28
第4章 简单例子概述	29
4.1 简介	29
4.2 StockWatch组件	30
4.2.1 StockWatch接口	31
4.2.2 数据库模式	32
4.2.3 扩展StockWatch的系统体系结构	33
4.3 证券管理器组件	35
第5章 性能要求	36
5.1 IDL设计的性能实质	36
5.1.1 访问模式	36
5.1.2 影响性能的重要因素	40
5.1.3 设计例子	42
5.1.4 对象的合适性	45
5.2 传递大量数据	46
5.2.1 迭代器	46
5.2.2 按值传递对象	47
5.2.3 专有的流解决方案	47
5.2.4 音频/视频流	48
5.3 小结	48

第二部分 核心服务

第6章 对象定位	49
6.1 定位对象的模型	49
6.2 CORBA对象定位服务	50
6.2.1 CORBA命名服务	51
6.2.2 CORBA交易对象服务	55
6.3 定位对象的其他方法	59
6.3.1 使用对象引用字符串	59
6.3.2 使用工厂对象	60
6.3.3 特定于ORB的方法	60

6.3.4 自举	61
6.4 选择对象定位机制	61
6.5 选择公布对象	62
第7章 消息接发	63
7.1 CORBA和消息接发	63
7.2 ORB对消息接发的支持	63
7.2.1 异步单向激发	64
7.2.2 延迟同步激发	66
7.2.3 消息接发的实现	66
7.3 面向消息中间件的需求	67
7.4 当前面向消息中间件产品	67
7.5 相关的CORBA服务	67
7.5.1 CORBA事件服务	67
7.5.2 CORBA通知服务	70
7.5.3 即将实现CORBA服务	71
7.6 多目消息接发	71
7.6.1 网络术语	71
7.6.2 网际协议IP	71
7.6.3 单目和多目	72
7.6.4 多目传送实例研究: OrbixTalk产品	73
7.7 小结	76
第8章 安全	77
8.1 安全的概念	77
8.1.1 认证	77
8.1.2 授权	78
8.1.3 加密	78
8.1.4 数据完整性	78
8.1.5 不可否认	78
8.1.6 证件和委托	79
8.2 企业系统安全需求和策略	79
8.2.1 分析	79
8.2.2 设计	80
8.2.3 实现	81
8.3 CORBA安全	81
8.3.1 CORBA安全服务规范	81
8.3.2 安全互操作/SecIOP规范	83

8.3.3	ORB-SSL集成规范	84
8.3.4	CORBA/防火墙规范	84
8.4	实际的解决方案	85
8.4.1	企业CORBA安全: 系统的 协调运作	85
8.4.2	内部Intranet系统	86
8.4.3	外部Internet系统	88

第三部分 数据库集成和事务处理

第9章 对象的持久性	91
9.1 简介	91
9.1.1 RDBMS与ODBMS	92
9.1.2 阻抗失配问题	92
9.1.3 对象持久性服务的特征	94
9.2 访问关系数据库	94
9.2.1 本地的SQL API	94
9.2.2 开放连接	95
9.2.3 OO SQL封装类	95
9.3 对象/关系映射	95
9.3.1 基本映射技术	96
9.3.2 水平分割	97
9.3.3 垂直分割	98
9.3.4 统一	99
9.3.5 自顶而下和自底而上	99
9.3.6 对象/关系代码生成	100
9.3.7 对象/关系映射的局限	101
9.4 对象数据库	101
9.5 数据一致性	102
9.5.1 事务	103
9.5.2 并发控制	103
9.5.3 数据同步	104
9.6 小结	105
第10章 数据库集成	106
10.1 系统体系结构	106
10.1.1 三层结构	106
10.1.2 N层结构	107
10.2 相关的OMG工作	107

10.3 集成的各方面	108
10.4 CORBA业务对象	109
10.4.1 阻抗失配	110
10.4.2 对象/关系代码生成	111
10.4.3 对象关系	111
10.5 ODBMS	112
10.6 CORBA关系服务	112
10.7 CORBA查询服务	113
10.8 特定于应用程序的解决方案	113
10.9 权衡	114
10.9.1 做实际工作的层次	114
10.9.2 事务和并发	115
10.9.3 对象标识	115
10.9.4 伺服对象管理	116
10.10 无状态伺服对象	116
10.11 有状态伺服对象	116
10.12 数据库适配器	117
10.13 海量对象CORBA系统	118
第11章 CORBA环境下的事务	119
11.1 两层和三层结构	119
11.2 客户机控制事务和服务器控制事务	119
11.3 服务器控制事务	119
11.3.1 每操作事务模型	119
11.3.2 分段事务模型	121
11.4 客户机控制事务	121
11.5 小结	122
第12章 分布式事务处理	123
12.1 事务处理	123
12.1.1 事务处理系统	123
12.1.2 事务处理监控器	124
12.1.3 对象事务监控器	124
12.1.4 ACID事务	125
12.1.5 事务管理	126
12.1.6 X/Open DTP	127
12.2 CORBA对象事务服务	128
12.2.1 总览	129
12.2.2 用OTS编程	129

12.2.3 事务资源	132
12.2.4 恢复	132
12.2.5 OTS体系结构	132
12.2.6 并发	134
12.2.7 OTS提供了多少ACID	135
12.2.8 多少ACID是合适的	136
12.2.9 事务分布的分类	137
12.3 高级的事务模型	139
12.3.1 持久队列	139
12.3.2 工作流和多事务	140
12.3.3 Saga事务	141
12.3.4 嵌套事务	142
12.4 小结	144
第13章 用户会话	146
13.1 短期事务与长期事务	146
13.2 并发控制	147
13.2.1 锁定	147
13.2.2 时间戳排序	147
13.2.3 乐观并发控制	148
13.2.4 乐观并发控制与商业DBMS	148
13.3 三层环境下的会话	149
13.3.1 客户机控制事务: OTS	150
13.3.2 服务器控制事务: 每操作	150
13.3.3 服务器控制事务: 分阶段	151
13.4 案例研究: 保险合同管理器	152
13.4.1 系统需求	153
13.4.2 设计	154
13.4.3 体系结构	160
13.4.4 实现	161
13.4.5 管理版本的层次	162
13.5 小结	163

第四部分 可伸缩性问题

- 第14章 服务器资源管理165
 - 14.1 内存管理165
 - 14.1.1 对象分类166
 - 14.1.2 对象生命周期事件166

14.1.3 伺服对象池模式	167	16.1.5 容错与客户机	220
14.1.4 案例研究: 股票价格迭代器	171	16.2 当前CORBA对容错的支持	220
14.1.5 分布式内存管理	172	16.3 容错性与复制	221
14.1.6 小结	173	16.3.1 应用程序状态: 服务器和 数据库的复制	221
14.2 连接管理	173	16.3.2 应用程序服务器策略	222
14.2.1 连接和CORBA规范	174	16.3.3 讨论	229
14.2.2 在证券管理器例子中的连接	175	16.4 案例研究——一个高可用性的 CORBA系统	229
14.2.3 可伸缩性的方法	177	16.4.1 引言	229
14.2.4 小结	181	16.4.2 数据库复制	231
14.3 线程管理	182	16.4.3 高可用性	231
14.3.1 多线程概述	182	16.5 未来CORBA对容错性的支持	235
14.3.2 多线程的CORBA客户机	183	16.6 小结	237
14.3.3 多线程的CORBA服务器	184	第17章 系统管理与维护	238
14.4 小结	189	17.1 系统管理	238
第15章 负载平衡	190	17.1.1 网络管理	238
15.1 负载平衡的必要性	190	17.1.2 活动系统和系统基础设施	240
15.2 应用程序分区	191	17.2 管理活动系统	240
15.2.1 水平应用程序分区	192	17.2.1 应用程序监控	240
15.2.2 垂直应用程序分区	192	17.2.2 性能监控	242
15.2.3 分区策略	193	17.2.3 存储管理	242
15.3 复制	194	17.3 系统基础设施维护	243
15.4 复制机制	196	17.3.1 调配CORBA系统	243
15.4.1 定位机制	196	17.3.2 管理变动	244
15.4.2 迁徙机制	202	17.3.3 CORBA环境中的版本化	245
15.4.3 状态管理	204	17.4 小结	246
15.5 复制框架	207		
15.5.1 组件复制	207	第五部分 工程化CORBA系统	
15.5.2 集中器模式	208	第18章 工程化过程的重要性	247
15.5.3 BOA激活模式	212	18.1 CORBA和软件工程过程	247
15.6 衡量负载平衡的花费	212	18.1.1 对象管理组	248
15.7 小结	214	18.1.2 传统软件工程过程	248
第16章 容错性	215	18.1.3 使软件工程过程适合于反映分布 式处理	248
16.1 什么是容错性	215	18.2 CORBA组件模型	250
16.1.1 需要一种实用的容错方法	216	18.2.1 统一建模语言	250
16.1.2 故障类型	217		
16.1.3 故障检测	217		
16.1.4 故障通知	219		

18.2.2 使UML适合于反映分布处理	251	第19章 自动化工程过程	266
18.3 建模CORBA组件	254	19.1 CASE	266
18.3.1 组件特性	255	19.2 代码生成	266
18.3.2 例子	256	19.2.1 基于IDL的代码生成	267
18.3.3 建模组件的一般原则	257	19.2.2 基于模型的代码生成	269
18.3.4 再论阻抗失配	258	19.2.3 对不同方法的讨论	269
18.4 将建模结果映射到实现	261	19.3 过程向导	271
18.4.1 UML建模元素	261	19.4 小结	271
18.4.2 实现元素	263	第20章 结论	272
18.4.3 模型与实现	264	20.1 困难仍然存在	272
18.5 小结	264	20.2 希望已出现在地平线上	272

第一部分 基础

在企业内建立和调配软件系统是一复杂的任务。CORBA (Common Object Request Broker Architecture, 公共对象请求代理体系结构) 为完成这一任务提供了功能强大的框架。使用CORBA, 可以更容易地开发异构分布式系统。但是, 要开发出好的系统仍然是相当困难的。CORBA规范对于开发实际系统中错综复杂的情况所给出的指导甚少。本书的目的在于帮助CORBA系统开发者学习有关开发大规模CORBA系统所需的设计要点、方法和值得考虑的问题。

为了恰当地讨论这些复杂的课题, 需要一个关于术语和符号的常用词汇表, 以便对对象适配器的一些重要方面有共同的理解, 以及对CORBA系统的性能特征达成共识。这就是本书第一部分的目的。

第1章定义了一些公共的术语, 并引入了在本书中使用的一整套图形符号。

第2章从应用程序员的角度来考察CORBA规范。本书并不是要讨论规范中的细节, 而是集中于两代主要的ORB——基于基本对象适配器 (BOA) 的ORB和基于可移植对象适配器 (POA) 的ORB。在此引入了两个重要的课题, CORBA激发的生命周期和CORBA对象的生命周期, 并从BOA ORB和POA ORB的角度讨论它们。

第3章简要介绍了作为CORBA规范一部分而定义的某些服务, 这当中的许多服务在本书后面会涉及。

范例系统将在第4章描述。它提供了本书很多讨论中需要的上下文知识。这个范例很简单, 易于理解, 但又足够综合, 能够说明CORBA系统中的很多方面。

最后, 第5章讨论了CORBA系统的性能方面, 并特别着重于IDL (接口定义语言) 设计。

第1章 绪 论

近年来, CORBA已从一学术研究课题转变为主流技术。各组织正在建立和调配实际的CORBA系统, 并使用CORBA技术来解决行业中的基本问题, 涉及的行业范围从金融到电信, 从保险到制造业, 从医疗到石油化工。

CORBA结合了计算机工业中两个重要的趋势: 面向对象软件开发和客户机/服务器计算。但CORBA并不只是面向对象远程过程调用机制。OMG的对象管理体系结构是一个定义了不同抽象层次的框架。核心ORB提供了网络编程复杂性的抽象。CORBA服务以面向对象方式提供了典型的系统级功能。CORBA工具则提供了解决特定领域问题的标准化方法。

虽然以对象请求代理进行编程已成为主流技术, 但应用高级CORBA服务来建立大规模企业级CORBA系统并不是小事。CORBA和CORBA服务提供了某种层次的抽象, 但对于像事务处理、

安全性和消息接发等大多数疑难问题，复杂性仍然是固有的。

本书着重于使用当今可用的技术来建立企业级CORBA系统。它反映了我们这些作者作为CORBA咨询人，在多个不同软件项目中使用CORBA所积累的经验。我们会不时提到CORBA界的一些未来发展，但重点仍是我们在现实世界中已使用和可利用的技术。

这意味着，一方面我们着重于可用作商品的某些核心服务和在实际中已经用到的技术的应用，另一方面，也说明了目前不是所有的CORBA服务都是商业上可用，不是所有在大规模企业系统中的问题都被已存在的CORBA服务规范完全覆盖。在没有标准服务可用的情况下，我们会尝试描述一些我们使用的基于已有CORBA技术的模式和技巧。

1.1 概述

本书分为五部分。每一部分表达了CORBA系统不同的一面。第一部分基础约定了在本书中使用的通用词汇。我们定义了经常使用的术语，并会引入在本书图表中所用到的图形符号。我们回顾了CORBA的体系结构，尝试识别出独立于特定CORBA版本的CORBA公共元素。这是很重要的，因为以下的很多章节都依赖于这些元素，而且我们不想老是要区分不同的CORBA版本。我们还回顾了CORBA服务，讨论不同服务的当前状态和适用性。这形成了本书余下部分的基础，剩下的部分将更详细地讨论某些核心服务的应用。另外，第一部分介绍了StockWatch例子，它将在本书中被引用。最后，第一部分讨论了一些CORBA系统设计的基本性能实质。

第二部分覆盖了CORBA对多数企业系统必要方面的支持。我们会涉及CORBA对对象定位、消息接发、安全性等基本需求的支持。

第三部分讨论的是与CORBA和持久性相关的问题。首先，我们一般性地提及对象持久性。这种讨论的结果是形成对CORBA和数据库集成进行讨论的基础。在对CORBA事务作一简要介绍后，我们会涉及分布式事务处理和CORBA对象事务服务。最后，我们讨论关于处理包括用户在内的长期事务的方法。

第四部分介绍可伸缩性问题。当CORBA系统规模变大时，本部分所涉及的课题就变得更重要。服务器资源管理讨论了处理连接、内存和线程的方法。负载平衡和容错性这两个在某种程度上相关的主题也被涉及，如同组件测试、调配和管理一样。

第五部分介绍工程化CORBA系统，作出结论，研究了CORBA和分布式对象计算对传统的面向对象软件工程过程的影响。我们还把形式化扩展到UML（统一建模语言）符号，以便有助于更精确地对系统的CORBA细节进行建模。最后，我们考察哪一种现行的CASE系统、建模工具和代码生成工具的扩展能帮助把基于CORBA的软件开发过程自动化。

1.2 简要的术语表

这里不打算给出所有与CORBA和分布式对象计算可能相关的术语的解释。下面仅列出一些关键术语，并描述它们在本书的用法：

- **CORBA对象** 既是有标识的抽象实体，也是用IDL定义的接口，也是实现。在2.4.1节“CORBA对象”中会更详细讨论为什么CORBA对象是抽象的。
- **伺服对象(servant)** 是实现了CORBA对象功能的具体编程语言实体。

- **客户机，服务器** 术语客户机和服务器作为角色使用：客户机激发驻留在服务器中的CORBA对象。一个服务器也可以扮演客户机的角色，激发另一个服务器中的对象。
- **CORBA组件** CORBA组件由一系列CORBA对象实现，通常比一单独的对象能提供更复杂的服务。为利用组件提供的服务，组件的用户要与该组件实现的不同对象交互。第18章含有对组件更详细的讨论。
- **组件入口点** 组件通常会提供一入口点，即一个专用对象，作为用户接触组件的第一点。从这个对象开始，组件的用户可以获得该组件提供的其他对象的引用。入口点对象经常注册到命名服务或交易服务，以令该组件的服务可以公用。

1.3 图形符号

想像一下这样的情况：某个人在一块白板上画了两个方框，在右边的方框内加入一个圆圈，并画一条从左边的方框到右边方框里圆圈的直线。大多数讨论过CORBA系统设计或实现的人都会立即知道，这通常是讨论和CORBA相关课题的开始：一个客户机激发一个服务器中的远程对象。这个特别的符号对于很多简单的情形能解决得很好，因为它允许用户用图表来描述一特定的例子。但同时也是问题：它只是一个例子——即它不是很普遍。它就像是在对象层次上建模，而不是在类的层次上建模：对象图对讨论具体的例子是有用的，但我们用类图以一般的方式来表达设计。为保持本书中的模型尽可能一般化，书中使用基于统一建模语言（UML）的类图。

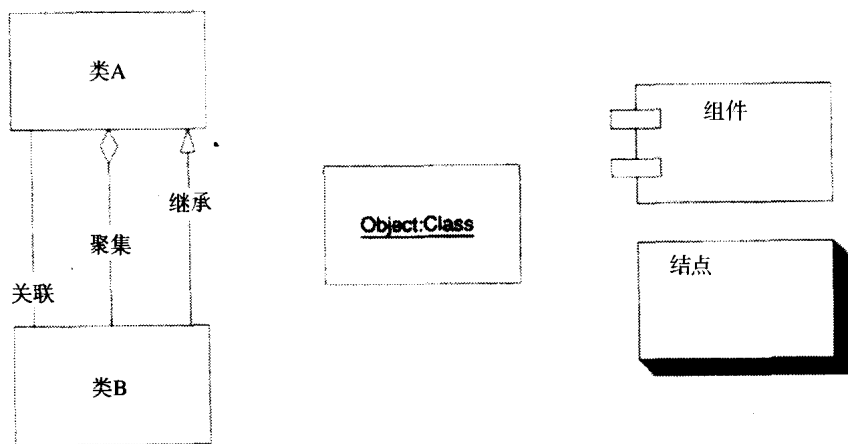


图1-1是一些标准UML元素的概貌。一个类以一个方框表示，并伴有粗体显示的类名字。对象以一个方框表示，并伴有下列线显示的对象名称和类名称，两个名称之间用冒号隔开。包括关联、聚集和继承等重要关系类型，在图中如类A和类B间的符号所示。组件是一个可执行的软件模块，有自己的标识和定义好的接口。结点是一个计算的单元，即一个硬件。

为反映CORBA的特殊内容，书中对标准UML进行了一些扩展。第18章中有对这些扩展的详