

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系统并非是无足轻重的。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越发重要。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“什么是真正的 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白板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。

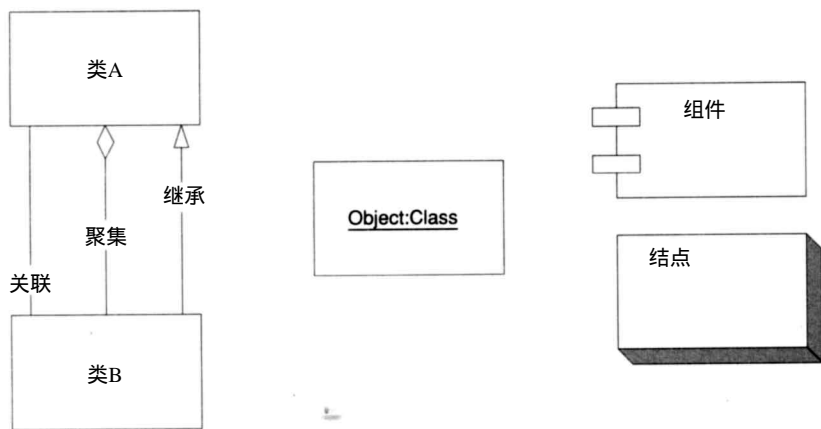


图1-1 重要的UML元素

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，并伴有下划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。组件是一可执行软件模块，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细讨论。我们使用UML定式来定义这些扩展。要定义的第一个定式是 <<CORBA Interface>>，用T连接符号表示，如图1-2所示。一个类可伴随有一个 <<CORBA Interface>>，表示该类的实例可通过CORBA远程激发的正常方式来访问。

我们定义的第二个定式是 <<CORBA Component>>，就像定义在上面的术语表一样。<<CORBA Component>>在本书中用来组合CORBA系统元素。<<CORBA Component>>的元

素只能通过 CORBA IDL 接口访问。<<CORBA Component>>的符号是一 UML 组件，在边界上伴随有传统固定风格的 ORB 符号，如图 1-3 所示。

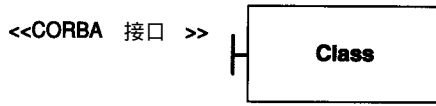


图1-2 <<CORBA Interface>>定式

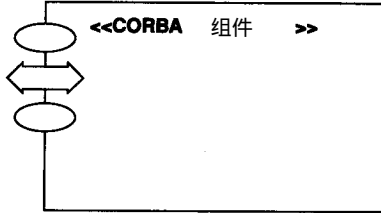


图1-3 <<CORBA Component>>定式

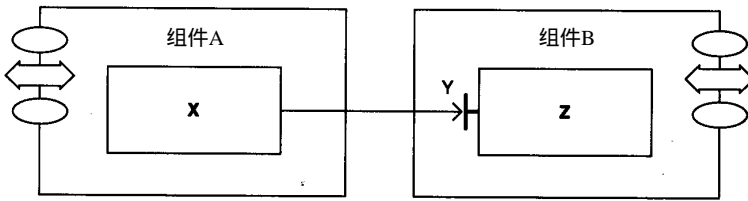


图1-4 UML扩展用法例子

图 1-4 显示了使用这些 UML 扩展的一个很简单的例子。组件 B 实现了类 Z，而类 Z 暴露 CORBA 接口 Y。组件 A 实现了类 X，类 X 通过 CORBA 接口 Y 访问组件 B 中类 Z 的实例。类 X 本身不暴露接口，因此它对于组件 A 外的元素都不可访问。

最后，我们必须承认“方框和圆圈”符号对一些白板类的讨论是有意义的，所以书中会经常使用图形符号。本书所使用的 CORBA 系统图形描述的基本元素如图 1-5 所示。主机以一个小的计算机符号表示。操作系统进程以正方形框表示，可能还会指向它们所运行的主机。CORBA 对象以带有阴影的圆圈表示，并有 T 型连接以指出 CORBA 接口。有时在图中显示代理对象也是可以的。在这些情况下，代理以带阴影的小圆圈显示。CORBA 消息以小信封符号表示。

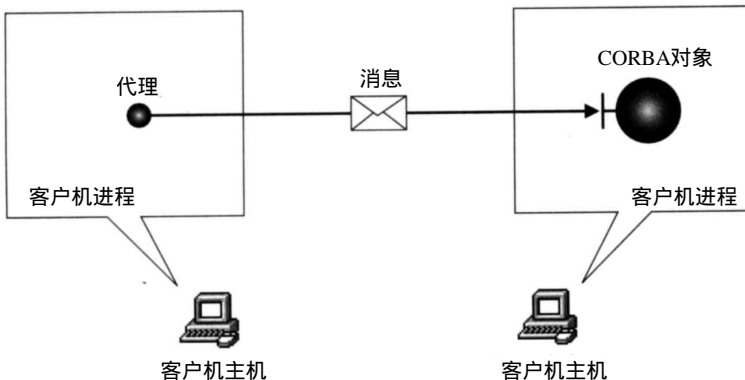


图1-5 符号图解