

## 第3章 CORBA服务回顾

OMA，即对象管理体系结构，定义了一系列的服务，用来提供基本 ORB功能之上的更高系统级的服务。这些服务处理核心系统级功能，如持久性和事务、安全性、消息接发以及目录服务，它们形成了构建高级工具和业务功能的框架。服务规范主要处理各种服务，并定义 IDL接口，以及非形式化地描述这些接口的语义。只有很少数的服务规范来处理服务质量的问题。这就意味着大多数规范并没有描述实现必须提供什么质量的服务，或是应用程序如何通过标准化的接口来控制服务的质量。

与前面的章节一样，我们把注意力集中于 OMG规范的抽象世界与服务实现的现实世界之间的差距上，以及使用这些实现的应用程序。我们从非常高的层次上来看一下这些不同服务的可用性和重要性，讨论一些它们的优点和缺点；看一下服务质量和可伸缩性问题；以及 OMG当前的某些进展。这个讨论确认了 CORBA企业级解决方案所需的核心服务。它还设置了本书余下部分的范围，即讨论用户如何使用存在的服务来构建企业级 CORBA系统，以及如何能克服现实世界中的限制。

### 3.1 核心服务

首先看一下由 OMG定义的不同目录服务。最重要的目录服务是 CORBA命名服务，它提供结构化名字与对象引用之间相对的静态映射。服务器用命名服务公布名字 /引用映射，客户机可根据名字查找引用。这是很基础的服务，而且在现实世界中经常使用。服务的结构十分简单。但是，有很多问题需要讨论，例如选择公布的对象、管理大量的输入，以及结构化名字空间。在第 6章“对象定位”中将考察这些问题。另一个 OMG规定的重要目录服务是 CORBA对象交易服务，它通过把对象服务和一系列属性相关来提供一种灵活的方式，以输出和定位对象服务。这些属性支持由一个对象提供的描述服务的高级方法。一个主要的区别是客户机只需要知道要查找对象的种类，不必知道特定对象的准确名字。对于命名服务，有很多问题，例如服务提供空间的结构、实现的质量和强调的持久性机制。

请求处理的基本 CORBA模型是内在同步和紧耦合的。很多应用程序需要另外的异步和松耦合的消息接发机制，因为这是很多业务状况要求的模型。CORBA事件服务通过定义一基于同步、紧耦合的 CORBA请求模型的结构来处理这个问题，但它也通过引入中间事件信道这个概念来提供异步和松耦合的消息接发。事件服务规范实际上只定义了普通的角色，如事件提供者、用户、信道，以及它们之间的接口。这个规范并没有定义事件信道必须实现的服务质量，例如，所保证的发送与其所尽的努力相对应。事件服务的一个扩展是 CORBA通知服务，它通过定义一系列接口来解决这个问题，例如在每个信道和每个事件的基础上，这些接口可使用户能定义服务所需的质量。另外，通知服务还为事件过滤引入了高级机制，这对大规模应用程序是特别重要的。第 7章“消息接发”中还特别从可伸缩性的角度来讨论和比较这些服务。

企业系统一个重要的方面是安全性，这是设计者经常不想处理又不得不处理的问题。今

天的企业依赖于它们的 IT 系统，而保护它们的业务是极其重要的。分布式系统中的安全性不仅包括加密，还有像身份验证、授权和不可否认这些概念。CORBA 规范定义了安全性的一些层次，最近又有像安全套接字层协议标准化的进展，可以把这些都集成到 CORBA 标准中。在第 8 章“安全”中会讨论安全性的不同方面以及它们在 CORBA 系统中是如何处理的。

## 3.2 数据库集成和事务处理

持久性、数据库集成和事务处理是复杂的问题，由一些 OMG 标准来考虑。

最初的持久对象服务（persistent object service, POS）是有缺陷的。它基于两层存储模型，而且极其复杂。另外，它没有集成处理与持久性相关主题的其他服务，例如事务和并发性。因此，OMG 决定完整校订持久性规范，即现在的持久状态服务 2.0（persistent state service, PSS）。新的 PSS 2.0 和对象事务服务（object transaction service, OTS）、对象值（object-by-value, ObV）以及可移植对象适配器（POA）等服务有了更紧密的集成，而且这在实践中证明是有用的。

和持久性紧密相关的是 CORBA 对象事务服务（OTS）。OTS 通过提供像事务管理、事务传播和基于提交的两阶段事务解决方法等基本特性来使 CORBA 环境中的分布式事务处理成为可能。传统上，事务处理系统在映像中是相对重要的，它要基于复杂的 TP 监控技术。特别地，两阶段提交事务解决方案的高成本加到了其作为重要技术的印映像中。在第 12 章“分布式事务处理”中将讨论 OTS 如何在没有 TP 监控的背景中使得分布式事务处理更易于被人们访问。在第 13 章“用户会话”中将讨论 OTS 如何有用，即使在单数据库环境中也同样如此，而且在没有真的两阶段提交处理的负担下，如何帮助完成在三层环境中保持数据一致的艰难任务。

处理关系传统上是数据库管理系统的范围。RDBMS 在第一层处理关系，即直接在数据库服务器中处理它们，例如执行复杂的连接以把行加入关系。ODBMS 使用户能以极其有效的方式在高度复杂的对象图中导航。通常导航逻辑在第二层，即在某个数据库客户机中实现。大多数 ODBMS 还提供机制以把查询直接送到数据库服务器，这样查询就可在第一层进行最有效的评估。CORBA 关系服务使关系可被第三层访问：CORBA 对象间的关系可以表示为 CORBA 对象。这在理论上是一个很好的主意，因为它使用户能用标准的 CORBA 机制来表达和导航关系。实际上，第三层中的导航关系由于简单的性能原因是受到严重的限制的：从一个对象到另一个对象的每一步都包括多个远程激发。不同于 ODBMS 客户机，CORBA 客户机没有标准的方法在本地捕捉对象和关系的信息。因此设计一个在第三层中使用关系服务的实现来访问关系的系统时必须小心。关系服务的使用一般被限制在性能不是最重要问题，而且需要导航的关系数量相对较少的场合。在第 10 章“数据库集成”中会对此进行更详细的讨论。

关系服务的一个有趣方面是和 CORBA 生命周期服务的紧密集成，为复合对象提供生命周期管理。生命周期服务致力在 IDL 层次上对象生命周期管理的问题，并使用工厂和工厂查找器等高层次概念。注意，这是和第 2 章“CORBA 回顾”中讨论过的伺服对象和 CORBA 对象的基于对象适配器的生命周期管理根本不同的方法。把生命周期服务和关系服务结合起来，以解决复合对象的生命周期管理问题看上去是强制性的，但读者也要仔细看一下前面所讨论过的性能问题。

CORBA 查询服务的定义并不只由执行与数据尽可能相近的查询的需求来驱动，还要由对象集合的一般需求来驱动，而这些对象集合允许选择、插入、修改和删除成员。CORBA 查询

服务试图通过集成象 SQL92和OQL93等标准来避免重构基础。但是，缺少商业上可用的实现使得作一个关于规范质量的声明变得很困难。

另一个关于持久性和事务的重要主题是并发性控制。同样，并发性控制是另一传统的 DBMS范围。数据库级别上的并发性控制是一复杂的课题，像 SQL92这样的标准中，在并发问题方面是很模糊的。同样，CORBA并发服务在IDL层次上处理并发性，定义接口来执行像加锁和解锁这样的任务。在大多数情形中，我们发现应用程序在数据库级处理并发，而不是使用更高级层次的抽象，如 CORBA并发服务——主要是因为大多数并发服务的实现没有提供和商业数据库管理系统的紧密集成。在一些场合需要更高级的并发机制，例如，为平行地管理人们工作而具有特定策略的某种工作流管理系统。本书在几个不同的地方讨论并发控制问题：如第12章“分布式事务处理”和第13章“用户会话”中。

### 3.3 可伸缩性问题

当CORBA系统在规模、复杂性和功能上增强时，对于调配这些系统的企业来说，它们就变得更加重要。大规模的CORBA系统必须致力于解决负载平衡和容错性这些难题。不幸的是，目前没有可用的CORBA标准来直接解决负载平衡和容错性。在第15章“负载平衡”和第16章“容错性”中将论述这些主题。

最后，CORBA系统开发者必须处理检测、调配和系统管理。检测是OMG完全没有涉及的领域。类似地，调配仍然是各个ORB供应商用自己专有的解决方案来解决的问题。系统管理是OMG正在开展的领域，但在编写本书时还没有具体的标准可用。在第17章“系统管理与维护”中会覆盖这三个主题。

### 3.4 小结

很多CORBA服务对于建造大规模CORBA系统是很关键的。在本书的第二部分“核心服务”中将讨论这些关键服务，并着重于对象定位、消息接发和安全性。本书的第三部分论述了持久性、数据库集成和事务处理这些问题。这部分和核心服务部分分开，是因为现时没有集成化的解决方案，初始的持久对象服务正在被新出现的持久状态服务规范所代替。还有，我们发现这个领域中使用CORBA服务并不总是有意义的，例如，在本地数据库并发控制机制比并发服务更有作用的场合中。本书的第四部分阐述了重要的可伸缩性问题，如资源管理、容错性和负载平衡，这些在目前都没有被CORBA规范覆盖。

还有其他很多的CORBA服务，如特许服务、时间服务和特性服务，以及涉及了其他课题的规范，如复合文档管理。以后读者会知道这些服务将会怎样实现，以及怎样用来建造现实世界的CORBA应用程序。