大规模软件构架技术

王映辉　冯德民　编著

科学出版社
www.sciencep.com

内 容 简 介

大规模软件构架技术是近几年发展起来的一个重要分支学科。本书比较全面地描述了大规模软件构架的关键技术，揭示了大规模软件构架的内涵。本书共7章。第1章简要总结了面向对象技术；第2章给出了分布式处理技术的内涵、开放式分布处理ODP的参考模型和体系结构；第3，4章描述了中间件技术和该技术支持下的几种分布构件模型技术；第5章阐述了软件Agent和MAS技术；第6章总结了各种构件模型的集成方法和技术；第7章给出了基于大规模软件构架技术的应用实例，即数字城市的软件构架模型

本书可作为构建分布式环境系统、企业电子商务平台系统，特别是空间信息平台软件的科研人员、高等院校教师和研究生的参考书。

图书在版编目(CIP)数据

大规模软件构架技术/王映辉等编著. —北京：科学出版社，2003

ISBN 7-03-011517-1

Ⅰ.大… Ⅱ.王… Ⅲ.软件开发 Ⅳ.TP311.52

中国版本图书馆CIP数据核字（2003）第038324号

责任编辑：赵卫江/责任校对：郝 岚

责任印制：吕春珉/封面设计：一克米工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

2003年6月第 一 版 开本：B5（720×1000）

2003年6月第一次印刷 印张：13

印数：1—4 000 字数：240 000

定价：25.00元

（如有印装质量问题，我社负责调换〈路通〉）

作者简介

王映辉，分别于1989年、1999年和2002年毕业于陕西师范大学（本科）、西南石油学院（硕士）和西北大学（博士），同时分别获得计算机软件学士、矿产普查与勘探硕士和计算机软件与理论博士学位。现为陕西师范大学计算机科学学院副教授，从事计算机软件与理论教学与科研工作。曾有10多年的石油工程软件开发经验，并荣获厅局级科技进步奖多项。近几年作为主要完成人员参加过863高科技攻关项目、国家自然科学基金项目、陕西省科学基金项目和西安市科技攻关项目等多项研究工作。近三年内在核心及以上级别的期刊上发表论文近20篇。目前的主要研究方向是大规模软件技术、信息可视化技术与WebGIS。

冯德民，教授。长期从事计算机软件与理论教学与科研工作，先后主持或作为主要参加人员完成了多项国家级和省部级科研项目。在核心及以上级别的期刊上发表论文20多篇。现为陕西师范大学计算机科学学院院长，主要研究方向为软件工程和算法理论。

前　言

计算机软件的发展从面向过程到面向对象，直到面向软件 Agent，无论从开发方法上，还是从开发工具上都发生了翻天覆地的变化，从而使人们的开发模式从简单的单机模式发展到了复杂的分布式大规模软件集成模式。

1. 计算模式的发展

在计算技术领域，随着微处理器技术和计算机网络技术的不断发展，计算模式经过了以下几次变迁。

(1) 集中式计算模式

从 1945 年现代计算机时代开始到 1985 年前后，计算机设备既庞大又昂贵，即使是小型机，每台也价值数万美元。因此，大多数机构也只有有限的几台计算机。为了节省成本，在一个系统中往往以一台主机（mainframe）为主，连接着若干个终端设备。所有的数据存储和计算都在主机上进行，终端设备只负责为用户发出计算请求和显示计算结果。我们称这种方式为集中式计算模式。

(2) 桌面计算模式

随着集成电路技术的不断进步，20 世纪 80 年代中期开始出现了微型计算机。从 8 位、16 位、32 位，到今天 64 位的 CPU 机型，发展非常迅速。许多 PC 机和工作站具备了以前大型计算机的能力，可以存储大量的数据且能进行相对复杂的计算，而价格却非常便宜，可以被一个机构大量采用。计算机也由此脱下了高贵的外衣，走入了寻常百姓之家。因此计算模式的主流从主机转移到了用户桌面。我们称这个阶段的模式为桌面计算模式。

(3) 分布式计算模式

进入 20 世纪 90 年代，计算机技术最显著的进步之一就是高速计算机网络技术的飞速发展。局域网 LAN 使得同一建筑内的数十甚至上百台计算机连接起来，使大量的信息能够以 $10^8 \sim 10^9$bps 或更高的速度在计算机间传送。广域网 WAN，尤其是 Internet 的迅速普及，使得全球范围内的数百万台计算机连接起来得以进行信息交换，改变了人们传统的获取和处理信息的方式。随着计算资源的网络化，拥有个人计算机或工作站的广大用户，迫切需要共享分布于网络上的丰富信息资源，用以廉价获得超出局部计算机能力的高品质服务，并逐步实现具有计算机支持的协同工作，因此在多个资源上进行分布式处理就变得越来越迫切。从简单的数据共享到多个服务的先进系统，大量的计算转移到了网络环境下

的各种资源和个人桌面，这就是分布式计算模式。分布式计算技术成为影响当今计算机技术发展的关键技术。

2. 分布式计算模式的特征

分布式计算（distributed computing）技术是在近 20 年来影响计算技术发展的最为活跃的因素之一，它的发展经历了两种不同的技术路线：理想路线和现实路线。

（1）理想路线（传统意义上的理解）

对于分布式计算或者称分布式系统、分布式应用，不同的学者给出了不同的定义。A.S.Tanenbaum 认为一个分布式系统可以看作是一些独立的计算机集合，但是对这个系统的用户来说，系统就像一台计算机一样。这个定义有两个方面的含义：第一，从硬件角度来讲，每台计算机都是自主的；第二，从软件角度来讲，用户将整个系统看作是一台计算机。Carl L.Hall 则将分布式计算定义为通过多个独立的计算机处理来完成一个特定的任务，每个处理可以在相同或不同的计算机平台上以并行或串行的方式进行，通过通信协议相互协作完成任务，从而实现了把计算负担分散到多个能通信的计算机上。对于分布式应用系统，P.H.Enslow 作了如下描述：包含许多物理资源和逻辑资源；通过网络通信实现信息交换；有一个高层的操作系统能够对整个系统进行管理；系统对用户透明；系统中各部分资源既相互独立又相互配合。

从上面的定义不难看出，系统中计算机的互连和各部分在网络中的分布仅仅是分布式计算的必要条件，分布式系统的统一的逻辑特性才是其充分条件，主要特征有以下几点：

①透明性。就是让用户将一些机器集合的协同工作看作是在一台机器上完成的。其主要包括：位置透明，即用户不需知道软、硬件资源（如 CPU、文件和数据库）的位置，资源的名字不应含有资源的位置信息；迁移透明，即资源无需更名就可自由地在系统中流动，外界不需要知道系统为使资源均衡而改变对象的位置；复制透明，即系统可以随意地为文件和其他资源进行附加拷贝而无需用户知道；并发透明，即多个用户可以自动共享资源；并行透明，即系统活动可以在用户没有感觉的情况下并行发生；存取透明，即隐藏数据表示和调用机制的异同；失败透明，即将出错和恢复事件隐藏在对象内部，以达到容错的目的；持久性透明，即对象里隐藏着所用资源的变化，如处理器资源、存储资源的冻结与解冻；重定位透明，即改变一个接口的位置不影响与之编联的其他接口；提交透明，即一组对象发生作用的次序不影响结果的一致性等。

②灵活性。可以根据不同的情况，用最有效的方式将工作负荷分配到可用的机器上，最大限度地合理利用资源。

③可靠性。即系统可以屏蔽错误。通过把工作负载分散到众多的机器上，单个芯片的故障最多只会使一台机器停机，而其他机器不会受任何影响。理想条件下，某一时刻如果有5%的计算机出现故障，系统将仍能继续工作，只不过损失5%的性能。对于某些关键性应用，如证券交易或核反应堆控制系统，采用分布式系统可以保证其高可靠性。

④可伸缩性。系统可在需求增长的时候，通过增加资源，对系统能力进行灵活地扩充。

由此可见，理想路线试图在互连的计算机硬件上部署全新的分布式操作系统，全面管理系统中各自独立的计算机，呈现给用户单一的系统视图。在20世纪80年代，学术界普遍追求这一目标，尽管产生了许多技术成果和实验系统，但却没有被用户和市场接受。这些技术总称为分布式计算机系统技术。对于分布式计算机系统的定义一直不太明确的一个主要原因是，不知采用什么样的模型来构造、管理和协调使用基于网络（包括局域网、城域网和广域网）中的各个计算机系统。但是，对于分布式计算机系统，当时追求的一个共同的特点是其含有多个处理单元，每个处理单元都既能从事自己的活动，又能协同处理一个大任务。

（2）现实路线（Internet/Intranet下的理解）

随着人们对Internet/Intranet的进一步使用，人们的生活方式和获取信息的方式发生了根本性的变化。在分布式软件的构架和研究方面，那种一味地追求逻辑上完全统一的研究方式已经失败。经验告诉我们，统一协议下的松散式协同工作模式更容易满足人们的需求。目前的分布式研究，不再追求严格的、逻辑上完全统一的系统对分布在各个角落的计算机进行管理，而是用像CORBA或DCOM或CCM等这样的分布式构件模型和中间件技术，结合软件Agent进行基于网络的全方位分布式框架和环境的构架。

由此可见，现实路线是在网络计算机平台上部署开放分布式计算环境，提供开发工具和公共服务，支持分布式应用，实现资源共享和协同工作。在20世纪90年代，工业界普遍追求这一技术路线，产生了一系列行之有效的技术和广为用户接受的产品。当前，人们所说的分布式计算技术通常是指在网络计算机平台上开发、部署、管理和维护以资源共享和协同工作为主要应用目标的分布式应用技术。本书中的讨论也是指这种技术，并称为开放分布式技术。

无论如何，开放分布式技术像分布式技术一样也在不断地发展之中，要给出一个确切的定义是不可能的，但概括起来是指在独立的计算机集合系统中，通过网络通信来开发、部署、管理和维护以资源共享和协同工作为主要应用目标的分布式开发环境。相对于集中式计算模式，开放分布式计算模式在性能价格比、计算能力、可靠性、伸缩性和解决问题固有的分布性上占有明显的优势；而对于桌面计算，分布式计算在数据共享、设备共享、通信、灵活性等方面则显示出了无

可比拟的优势。当然，事物总是一分为二的，开放分布式计算技术也面临着一些急需解决的问题，如没有一个权威统一的标准、缺乏开发分布式系统的支撑软件、网络负载饱和引起的问题、数据安全等问题。

3．开放分布式计算模型

进入90年代，开放分布式计算的实现主要依赖于经典的客户机/服务器计算模型。它将分布式应用分解为客户和服务器两大部分，客户机首先发出请求，服务器在接到客户的请求后提供服务。这种方式不同于主机的计算模型在于：充分利用了客户端计算机的计算能力，每一个客户机都是一个独立的计算单元，有自己的处理器和存储器，负责处理应用系统的显示逻辑，如图形用户界面、信息预处理等，而把复杂的计算，如业务逻辑、数据处理，交给了服务器。通过平衡客户机和服务器的负载以实现分布式资源和信息的共享。目前Internet上最流行的WWW应用就是一种开放分布式的客户机/服务器结构。

WWW应用是一种多层结构，分为客户端、应用服务器、数据库服务器。客户端可以通过浏览器（browser）以标准的通信协议访问分布在网络上的各种多媒体信息，如文本、图像、声音、视频等。浏览器的技术简明易用，一旦用户学会了使用浏览器，也就打开了运用系统上各种信息资源的大门；应用服务器端通过Web服务器提供各种应用服务，包括各种业务处理逻辑；客户机不必关心服务器的具体位置，它可以把分布在网络上的许多服务器当成是一台巨大的“虚拟主机”，各个应用服务可以是并行的也可以是串行的，通过中间件用以消除通信协议、数据库查询语言、应用逻辑与操作系统之间潜在的不兼容问题；数据库服务器就是DBMS（Database Management System），负责管理对数据库数据的读写，迅速执行大量数据的更新和检索。

上述计算模型是在共享分布资源的应用背景下形成的，只是实现完全的分布式计算的一个中间步骤。随着对象模型、构件技术、软件框架技术和Web技术的不断进步，在新的应用需求冲击下，分布式计算开始向分散对等的协同计算方向发展，开放分布式对象技术正成为分布式计算的主流。

对象模型、构件技术、框架技术、软件Agent技术和Web技术的融合彻底改变了系统的构造方法。在开放分布式系统中，对象被用来表示分布的、可移动的、可通信的实体；构件化的软件开发方法使对象被加在网络上、集成在框架和中间件上，达到跨平台的互操作和较高的可伸缩性；Web技术使应用对象可以在Internet这个开放的计算平台上移动。这是一个全新的计算模式，核心是可互操作的对象，即软件对象间可透明地进行相互通信，彼此地位是对等的，可使用对方的服务，而不管这些对象是处于同一编址空间，还是不同的编址空间，或是根本不同的机器上。

在开放分布式对象市场中有3种主要的竞争技术：包含760多个成员的对象管理组织（OMG）的公共对象请求代理体系结构（CORBA）；SUN公司的Java远程方法调用（RMI）和微软公司的分布式构件对象模型（DCOM）。

当前，CORBA和Java以其平台独立性，影响已大大超过了微软的DCOM。事实上Java和CORBA在体系结构上是互补的。CORBA不仅仅只是一种对象请求的中介工具，它还是一个非常完美的分布式对象平台，它使Java应用软件得以在不同的网络中、在不同的操作系统上顺利运行，可以跨越不同语言、不同构件的边界而畅通无阻；Java也不仅仅只是一种新语言，它还是一种可移动对象系统，简化了大型CORBA系统的代码分布处理，它的字节码使对象行为流动化，为CORBA实现流动运行打开了大门。人们发现，Java正是编写客户机和服务器CORBA对象的最佳语言，它内部的多线程机制，无用存储空间收集和出错管理的功能，使其能很容易地编写出健壮的网络化对象代码。作为Internet上影响最大的Web应用，CORBA和Java的加入，不仅大大改善了Web应用的质量，如客户端可直接动态调用服务器上的程序、多对象服务群平衡处理输入的客户请求等，而且对于开放分布式对象的普及起了积极的作用。

开放分布式计算模型广泛应用于目前的软件开发中，从小规模的企业软件的组建，到中大型跨区域、跨国际企业的软件构架，无处不见它的痕迹。基于这种计算模型的技术称为大规模软件构架技术。

在大规模软件构架之中，人们充分利用面向对象的技术，为了达到在分布环境下的软件高度重用，制定了不少标准，其中最有影响的是微软的DCOM、OMG的CORBA和SUN的EJB。然而，在这种异构环境下，不仅包含了不同的硬件，也包含了复杂多样的系统软件和应用软件。经验告诉我们，若要采用集中式管理是绝对行不通的。这种环境的一个最大特点是在松散的基础上讲究一致性和协同性。因而，处于该环境中的用户需要遵循一种协议，进行逻辑上的统一。同时，大量信息的共享加之网络带宽的限制，要求软件具有智能性，同时还应具有移动性，既靠移动的软件处理分布在网络环境下的数据和信息，争取保持数据处于原地不动，从而降低大量数据在网上传递带来的网络堵塞。软件Agent，特别是移动Agent支持下的MAS（Multiple Agent System）为我们提供了一种很好的解决途径。

如果将每一个软件Agent看成一个对象，利用开放分布式构件模型将它们进行整合，将是未来大规模软件开发的主流模式。

本书正是沿着这条思路，从介绍面向对象技术开始，对中间件、分布式构件模型、软件Agent和大规模软件集成等技术做了较详尽的阐述。最后，给出了一个大规模软件构架的应用范型——数字城市。

本书注重软件的构架思想和基本概念，也强调应用技术与方法，它不仅适应

于电子商务、企业信息系统、WebGIS、数字城市乃至数字地球等大规模软件的开发者、应用人员与管理人员阅读参考，也可作为高等院校的高年级学生或研究生的教材或参考书。

由于时间紧，本书难免有许多不足或错误之处，恳请读者批评指正！

本书在编写过程中引用了大量的参考文献，在每章的最后专门列出。在此向被引用文献的所有作者表示最衷心和最诚挚的感谢！

作 者

2003 年 4 月

目　录

第 1 章　面向对象技术

面向对象（OO）是一种认知方法学。它既提供了从一般到特殊的演绎手段(如继承等)，又提供了从特殊到一般的归纳形式（如类等)。面向对象基于信息隐蔽和抽象数据类型概念，把系统中所有资源，如数据、模块以及系统看成“对象”，每个对象封装数据和方法，而方法实施对数据的处理。其实，面向对象技术中有关模块化、数据抽象和信息隐蔽等概念也继承了 20 世纪 70 年代软件工程的成果，面向对象技术的贡献是在对象这一更高的层次上进行了抽象。面向对象是一种程序设计方法学。当今计算机正朝着分布式处理、网络化和软件生产工程化的方向发展，而面向对象方法学作为实施这个目标的关键技术之一，显然和计算机领域的总体发展相一致。目前，面向对象技术除了在应用领域的拓广和技术实现的完善等方面正在进行大量的工作外，还试图在一个比“对象”更具有“动态性”和更具有“人性”含义的层次上进行抽象，有人称之为面向智能体(agent oriented)技术。

面向对象的技术把人们自然思考问题的方式作为它的基本原则，为了实现这个原则，在问题域中，必须建立直接表达事物以及事物间相互联系的概念，同时还必须建立适应人们一般思维方式的描述范式。在各种各样面向对象的方法中，对象（object）和消息传递（message passing）是分别表示事物及其相互联系的概念；类（class）和继承（inheritance）是适应人们一般思维方式的描述范式；而方法（method）是允许作用于该类对象上的各种操作。这种对对象、类、消息和方法的程序设计范式的基本点在于对象的封装性（encapsulation）和继承性。通过封装能将对象的定义和对象的实现分开，通过继承能体现类和类之间的关系，以及由此带来的动态绑定（dynamic binding）和实体的多态性（polymorphism），从而构成了面向对象的基本特征。

1.1　对　　象

客观世界的问题都是由客观世界的实体及其间的关系构成，客观世界中问题的集合构成了问题空间或问题域，而实体是问题域的对象。显然，“对象”有大小之分、有“粒度”的概念。在面向对象的程序设计思想中，自然界中的任何事物都被看作对象，世界上的各个事物都是由各种“对象”构成的，这些对象可以属于同一个类，也可以属于不同的类。复杂的对象可由某些简单的对象构成，甚

至这个世界也可以由一些最原始的对象经过层层组合而成。

1.1.1 对象的概念

在面向对象的思想中，对于任何一个对象，都可以采用属性（property）、方法（method）和事件消息（event message）三个方面来描述，这被称为 PME 模型。其中，属性是对象所具有的性质和特征，这些特征可能是看得见摸得着的，也可能是内在的；方法是对象所具有的动作和行为，即使一些无生命的对象也可找出它的方法来，如桌子的倒下、杯子的破裂等；事件是指对象能识别并做出反应的外部刺激，是通过事件消息进行传递的；事件消息是事物关联的桥梁和纽带。

由此可见，对象是一个具有局部属性状态和操作（方法）集合的实体。

本质上，利用计算机解题是借助某种语言的规范，对被计算的实体实施某种动作，以此动作的结果去映射解。我们把计算实体称之为求解域的对象。从动态的观点来看，对象的操作就是对象的行为或方法。问题域中对象的行为是极其丰富的，而解空间对象的行为是极其死板的。因此，只有借助于极其复杂的算法才能操纵解空间对象而得到解。传统的程序设计语言限制了程序员定义解空间对象。而面向对象语言提供了“对象”概念，这样，程序员就可以自己定义解空间的对象。从存储角度看，对象有一片私有存储，其中有数据，也有方法。其他对象的方法不能直接操作该对象的私有数据，只有对象自己的方法才能操纵它。从对象的实现机制看，对象是一个自动机，其中私有数据表示了对象的状态，该状态只能由自己的方法改变它。每当需要改变对象的状态时，只能由其他对象给该对象发送消息，对象响应消息后，按照消息模式找出匹配的方法并执行之。

在面向对象的设计中，对象是应用域中的建模实体，是系统的基本运行单位。换句话说，对象是具有特殊属性和行为方法的实体。对象占有存储空间且具有传统程序设计语言的数据，如数组、字符串和记录等。给对象分配存储单元就确定了给定时刻对象的状态；与每一个对象相关的方法定义了该对象上的操作。所有对象在外观上都表现出相同的特性，即固有的处理能力和通过传递消息的统一的联系方式。

1.1.2 对象的特性

在面向对象的系统中，对象是构成和支撑整个软件系统的最重要、最基础的细胞和基石。每定义一个对象，就增加了一个具有丰富内涵的新的抽象数据类型。对象主要有如下 3 个特性。

（1）模块独立性

从逻辑上看，一个对象是一个独立存在的模块。对外界对象来说，只需要了

解它具有哪些功能，而无需了解它是如何实现的，也无需了解隐蔽在模块内部的信息。也就是说，模块内部状态不会受外界的干扰，也不会涉及其他模块。也就是说，模块之间的依赖性极小，各模块可独立为系统所选用，也可以被程序员重用，而不用担心会影响或破坏其他模块。

（2）易维护性

由于对象的功能实现细节被隐蔽，所以对对象功能的修改、完善等都局限于对象的内部，并不会涉及外部对象，这就使得对象和整个系统变得非常容易维护。

（3）动态连接性

客观世界中各式各样的对象并不是孤立存在的，它们之间存在各种各样的联系。正是这些对象之间的相互作用和相互联系，才使世界变的丰富多彩。在面向对象系统中，通过事件激活机制产生的消息，将各个对象动态地联系在一起，从而形成一个有机的整体。这就是对象的动态连接性。

1.2　类和实例

在面向对象的程序设计中，对象是程序的基本单元。具有相同特性的对象可以归并到一类中去。在面向对象程序设计中，只需定义一个对象类就可以得到若干个实例（instance）对象了。

一个类描述了该类型的所有对象的性质。具体来说，类由方法和属性数据组成，包括外部特性和内部实现两个方面。对象的外部特征是对象类通过描述消息模式及其相应的处理能力来定义的，而对象的内部实现是通过描述内部状态的表现形式及固有处理能力的实现来定义的。类的形式如图 1-1 所示。

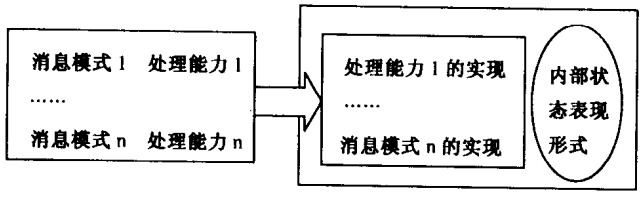

图 1-1　类的形式

从类的形式上来看，类实际上是抽象数据类型 ADTs（Abstract Data Types）的具体实现。数据类型是指数据的集合和作用在其上的操作集合。抽象数据类型的特点是使用和实现的分离，实行封装和信息隐蔽。从使用者的角度看，只要了解该抽象数据类型的规格说明，就可以利用其公共服务来使用这些类型，而不关