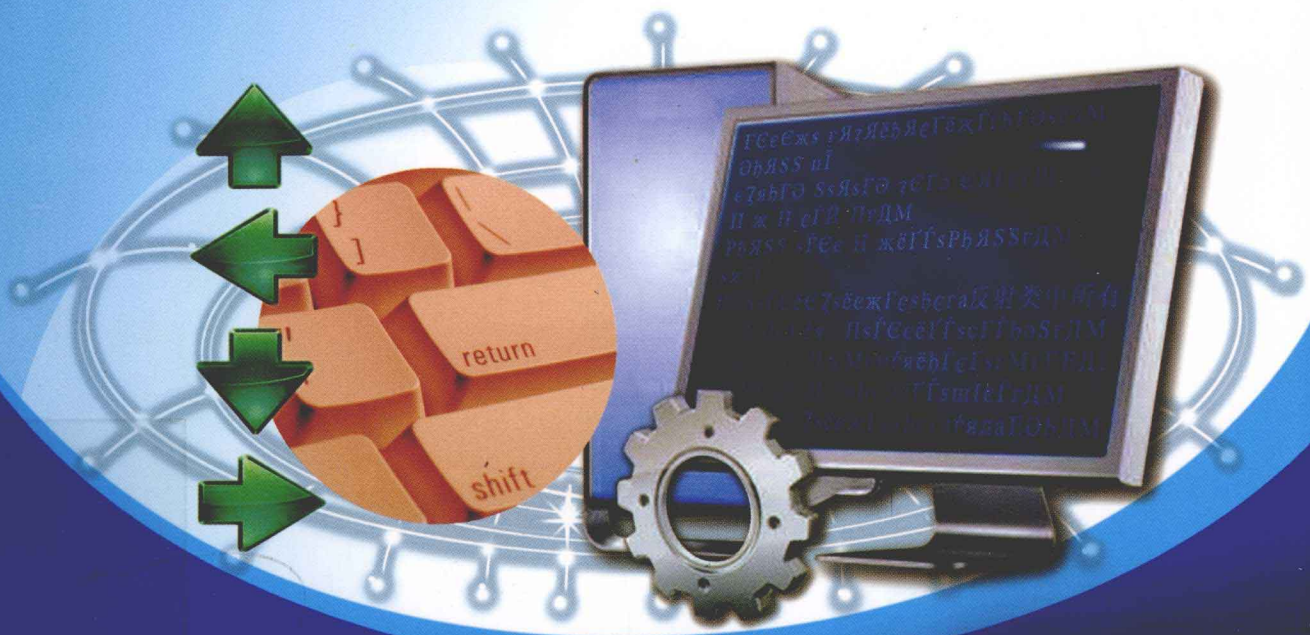
普通高等教育计算机类“十二五”规划教材

Java组件技术

主　编　赵　莉　孙喁喁　徐　飞
副主编　杨国梁　耿军雪

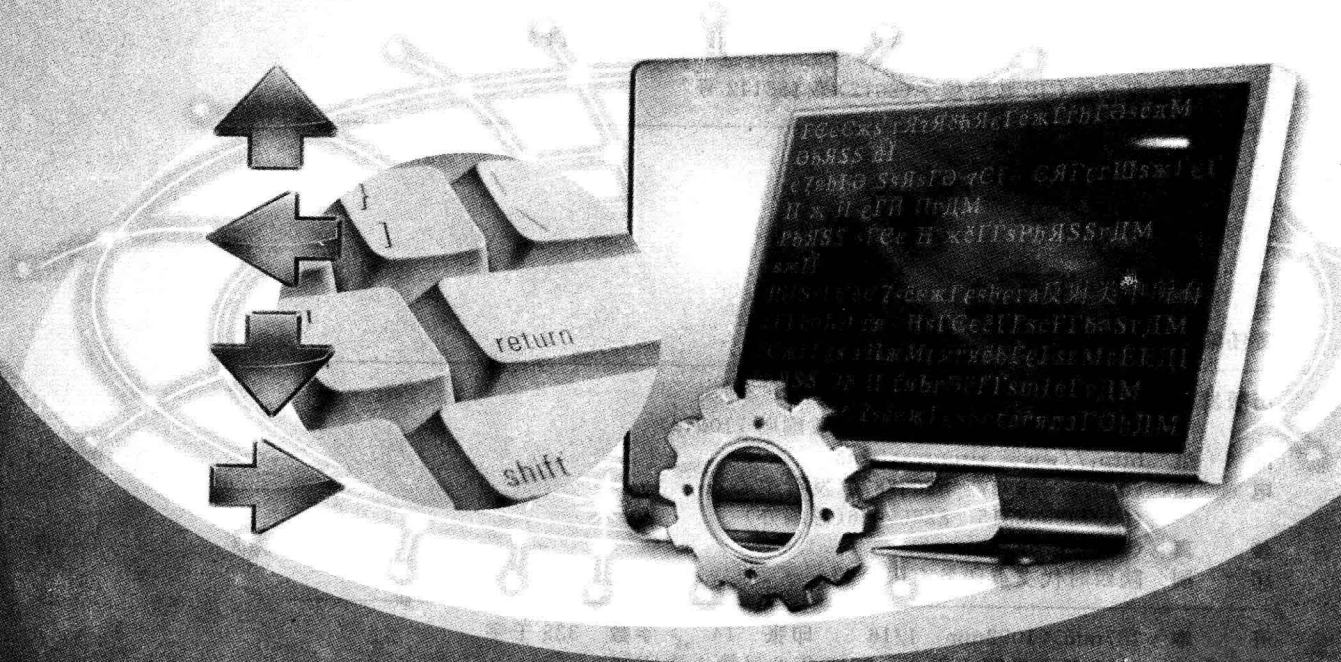


西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

普通高等教育计算机类“十二五”规划教材

Java组件技术

主　编　赵　莉　孙喁喁　徐　飞
副主编　杨国梁　耿军雪



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本书全面系统地介绍了基于Java的组件设计及开发方法，详细介绍了组件的相关概念及设计原则、Java反射及动态代理、企业级JavaBean、会话Bean、实体Bean、WebService框架及开发方法、Java消息服务、CORBA组件、事务、安全管理、配置组件、Socket通信组件、实体Bean开发实例等内容。各章均提供了丰富的实例，便于读者巩固知识，掌握组件设计的基本方法和技巧。本书力求概念叙述准确、严谨，描述简练，语言通俗易懂，使读者易于理解和掌握。

本书适合作为高等院校计算机及相关专业组件技术课程的教材和工程技术人员学习组件的参考书，也适合于编程开发人员培训、广大计算机技术爱好者自学使用。

图书在版编目(CIP)数据

Java组件技术/赵莉等主编. —西安：西安交通大学出版社，2012.9
ISBN 978-7-5605-4470-0

Ⅰ.①J… Ⅱ.①赵… Ⅲ.①JAVA语言-程序设计 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2012)第185142号

书　　名　Java组件技术
主　　编　赵　莉　孙喁喁　徐　飞
责任编辑　毛　帆

出版发行　西安交通大学出版社
　　(西安市兴庆南路10号　邮政编码710049)
网　　址　http://www.xjtupress.com
电　　话　(029)82668357　82667874(发行中心)
　　(029)82668315　82669096(总编办)
传　　真　(029)82668280
印　　刷　陕西时代支点印务有限公司

开　　本　787mm×1092mm　1/16　印张　14　字数　335千字
版次印次　2012年9月第1版　2012年9月第1次印刷
书　　号　ISBN 978-7-5605-4470-0/TP·573
定　　价　27.80元

读者购书、书店调货、如发现印装质量问题，请与本社发行中心联系、调换。
订购热线：(029)82665248　(029)82665249
投稿热线：(029)82664954
读者信箱：jdlgy@yahoo.cn

前　言

目前，计算机在各行各业中得到广泛的应用，随着需求的快速膨胀，计算机软件规模不断扩大，软件开发活动也变得日益复杂。这种情况在企业应用系统开发中表现得更为突出。企业应用系统中对并发、数据量、性能、可靠性要求很高，这些都对软件系统的设计和开发提出了严峻的挑战。如何应对复杂多变的客户需求，如何快速产出高质量的软件，经过多年的软件工程实践，组件技术被认为是快速发布高质量软件产品的关键技术。通过一系列基础组件的组装，可以像搭积木一样，迅速生成所需软件的主体结构和核心功能，应用程序开发者只需要编写特定的应用逻辑即可，这样大大加快了软件产品的开发速度。此外，组件都是经过充分测试的，因此在质量上也提供了坚实的保障，只有这样才能实现快速、高质量的软件开发。在组件技术中，如何捕捉和定义组件的需求，如何设计和开发组件，是组件技术的最核心部分。Java是备受开发者推崇的现代开发平台，提供了创建类和组件所需要的基础结构。Java提供封装、继承和多态性，以支持面向对象的编程等重要功能。Java还提供加载器和各种类型信息以支持组件。

本书内容丰富，教材内容体系、前后内容的衔接、知识点的安排由浅入深、循序渐进，力求做到重点突出、详析难点、解答疑点，使读者容易理解和掌握。全书围绕组件设计的思想，详细介绍了组件设计的原则和基于Java语言的组件设计方法，有利于学生扩充知识及后续课程的学习。本书在编写中力求做到概念叙述准确、严谨，语言简练通俗，适合读者自学。为便于学生更好地学习和理解组件的概念，在一些较难理解的部分尽量使用图解的方法，在例题中加入了部分代码注释，使读者低起点、高效率、高目标地掌握Java组件设计。书中示例丰富，覆盖面广，通过大量的例题来解释相关的概念和方法，将明确概念和着重实用相结合，有助于读者对基本概念和方法的学习，强调了编程能力训练的重要性，淡化了部分语法细节，实践性强。本书共分13章，围绕组件阐述如何有效使用Java的组件基础结构，各章的具体内容介绍如下。

第1章全面介绍了组件的发展现状及基本概念，并给出了组件设计的技术标准，以及在组件设计过程中组件设计者要考虑的一般原则。

第2章详细介绍了Java语言的反射语法，并给出了反射调用的实例。作为Java语言的高级语法，掌握反射语法为JavaEE多层分布式应用程序设计打下基础。

第3章主要对EJB的核心概念及EJB体系结构中的实体Bean、会话Bean进行介绍，为后续章节深入学习EJB设计与开发打下基础。

第4章主要介绍两种会话Bean的基本语法构成，和使用注释符进行Bean内属性、方法和结构描述的主要过程。掌握并区分两种会话Bean的特点和适用场合。

第5章主要介绍了实体Bean，实体Bean一直是EJB标准中的核心内容。本章对实体Bean的规范及持久化、实体Bean的操作、实体间的映射关系进行了详细描述，并详细说明了EJB 3的查询语句。

第6章主要介绍了WebService框架及其支持环境，以及WebService的开发方法，并通过

实例进行了详细的阐述。

第 7 章主要介绍了 JMS 的体系结构及编程模式，给出了通信域模型，并通过实例说明了点对点消息及主题消息。

第 8 章主要学习配置和使用简单的 CORBA 服务器对象的信息，了解使用 CORBA 服务器对象的主要特点及多样性，同时介绍了 CORBA 回调机制，允许服务器对象调用客户端的方法，以及在 BOA 中第一次注册临时对象时激活对象的技术。

第 9 章重点讲述了事务，首先介绍了事务的概念及事务的四个特征，然后介绍了事务模型，事务使用的注意事项，EJB 事务相关知识，声明性和编程型事务的划分等，最后介绍了分布式事务和两阶段提交协议。

第 10 章主要介绍了 EJB 安全管理，概述了 EJB 安全模型，详细介绍了 EJB 的安全机制、安全管理的整体考虑，最后着重阐述了 EJB 六种角色在安全管理中的职责。

第 11 章说明了如何从最初的设计一个配置组件，而最终经过一系列的重构，演化成一个通用的 XML 转换器，体现了对组件的定位和功能接口的高度抽取。

第 12 章对 Socket 组件客户端和服务器这两部分的设计和实现进行了详细的描述。

第 13 章给出了实体 Bean 的开发实例，通过例子说明如何开发实体 Bean。

本书适合作为高等院校计算机及相关专业组件技术课程的教材和工程技术人员学习组件的参考书，也适合于编程开发人员培训、广大计算机技术爱好者自学使用。

本书在编写过程中参考了大量文献、相关著作以及网络上的最新资料，在此表示衷心的感谢！尽管在本书的编写过程中投入了大量的时间和精力，但由于作者经验和水平有限，书中难免有疏漏和不当之处，敬请读者和专家批评指正。

编　者

2012 年 8 月

目 录

第 1 章　组件技术及发展

随着计算机和互联网的广泛普及和应用，计算机软件已成为信息时代社会的最重要的基础设施。广大用户对软件的数量与质量的要求也是与日俱增。那么显然用传统的软件开发方式，已经无法满足消费者的需求。在这样的环境下，软件构件技术是必然的选择，它不仅能缩短开发周期也节省了开发的成本，提高了软件的开发效率。那么什么是构件技术？

1.1　组件技术提出的背景

1968 年 NATO 软件工程会议，Mcllroy 在提交会议的论文《大量生产的软件构件》中，提出了“软件组装生产线”的思想。从那以后，采用构件技术实现软件复用，采用“搭积木”的方式生产软件，成为软件开发人员长期的梦想。软件复用是指重复使用“为了复用目的而设计的软件”的过程。就软件开发而言，软件复用包括：早期的函数复用，面向对象言语中的类的复用，以及互联网时代的完整软件体系的构件复用。

1999 年 2 月美国总统 IT 顾问委员会也在一份报告中列举了大量的事实论证 IT 技术对社会和国家以及人民生活的重要作用。建议美国政府加大对 IT 技术发展研究的投入。在建议重点支持的四大项目中，把软件列在首位。因为报告认为软件是信息时代社会的最重要的基础设施。然而现实上这个基础却相当脆弱和不可靠。软件越来越普及而且越来越复杂，但缺乏开发安全可靠的软件的适用技术。软件的生产能力远远满足不了飞速发展的实际需求。为此，报告建议重点支持四个方面的软件技术的发展和研究，第一个就是支持软件开发方法和构件技术的基础研究。什么是软件构件技术，为什么把它提得这么高，它究竟对软件的开发和应用有些什么作用，构件技术的突破对软件产业的发展会带来什么影响呢？

有效的软件复用是可以提高软件开发的效率和质量。建立在构件复用基础上的软件复用将会带来极大的价值，《Software Reuse》指出很多公司通过复用取得的成就使他们坚信，管理层可以期待获得如下优势：

①投放市场时间，减少为原来的 1/2 到 1/5；

②缺陷密度，降低为原来的 1/5 到 1/10；

③维护成本，降低为原来的 1/5 到 1/10；

④整体软件开发成本，降低大约 15%，长期项目可降低高达 75%。

基于这样的背景，软件构件技术在短短的数年间，迅速发展，到现在已经初具雏形，下面让我们来看看它的现状。

1.2　软件组件技术的现状

美国军方与政府资助的项目中，已建立了若干构件库系统，如 CARDS、ASSET、DSRS

等。由DARPA发起，由美国军方、SEI和MITRE支持的STARS项目在此基础上考虑了开放体系结构的构件库之间共享资源和无缝互操作的问题，并于1992年提交了ALOAF(Asset Library Open Architecture Framework，开放体系结构的构件库框架)Version1.2版本。这一报告体现了STARS对可复用构件库系统的认识，给出了一个构件库框架的参考模型，并就此实现了ALOAF规约作为该参考模型的实例，由此证明以公共元模型为基础，在构件库之间交换信息和创建易于移植的复用工具是可能的和必要的。

另外，中国在构件技术方面是处于领先水平的，我国已经建立有投入使用的构件库，并有大批项目在建设当中，如下所述。

①北京大学软件工程研究所。北京大学软件工程研究所是一个专注于软件工程及其相关领域的研究和实践的学术机构，其前身是建立于1983年的软件工程教研室，1999年正式挂牌成为北京大学的一个专业研究所。

近年来，研究所在所长杨芙清院士的领导下，对软件复用与软件构件技术进行了深入的研究。成果代号为“青鸟工程”，历经“七五”、“八五”、“九五”。青鸟工程在软件复用和构件技术领域成绩斐然。

②中国科学院软件研究所。中科院软件所软件工程技术研究中心，在首席研究员冯玉琳博士带领下，对构件技术深入研究，硕果累累。其中作为知识创新工程的成果的信息化基础软件核心平台是其代表。

③上海普元。普元是国内最早推进面向构件技术的厂商之一，也是目前国内唯一一家提供真正意义上的面向构件的互联网应用基础平台的专业化厂商。普元把崭新的互联网相关技术与先进的构件复用技术以及可视化开发技术完美地结合起来，创造了一套具有国际领先水平的面向构件的互联网的应用基础平台——EOS。

④互联网实验室。互联网实验室是我国著名的IT研究机构，长期从事构件技术及软件产业的研究，并于2004年1月16日发布了《面向构件的互联网应用基础平台研究报告》，是目前国内较为详尽的关于构件技术研究的专业报告。

1.3 组件的相关概念

1.3.1 组件

组件的英文名为“component”，也称为元件。实际上组件并不是一种新概念，它在许多成熟的工程领域有着十分广泛的应用。比如我们组装计算机，自己并不一定要了解CPU、主板、光驱等配件的工作原理，而只需要知道如何将这些配件组装在一起。

软件行业的组件系统比其它许多行业发展得都要慢。在计算机软件发展的早期，一个应用系统往往是一个单独的应用程序。随着人们对软硬件需要的不断增加，应用更加复杂，程序更加庞大，系统开发的难度也越来越大。

从软件模型的角度考虑，人们希望把庞大的应用程序分割成为多个模块，每个模块完成独立的功能，模块之间协同工作。这样的模块我们称为组件。这些组件可以进行单独开发、单独编译、单独测试；把所有的组件组合在一起得到完整的系统。许多人都认为，未来的应用程序都将利用组件实现。

组件化的软件结构为我们带来了极大的好处。但是为了能够通过组装现有的组件来创建应用程序系统，我们必须解决几个技术上的关键问题。

①采用一个标准方式来规范组件的定位和使用，这样将大大减少在人员培训上的开销，提高了组件的通用性。

②提供与对象进行交互操作的标准方式。组件和对象所处的具体位置不应用影响程序员的开发方式，也不妨碍它们之间的交互操作，即我们所说的“位置透明性”。

③要便于创建组件的版本。对软件的升级应用具有灵活性，组件的更新不会对现有的应用程序的运行造成不良影响。

④提供满足用户需要的安全性。

1.3.2　接口

了解了组件的基本含义后，我们还必须进一步理解接口(interface)的含义。接口描述了组件对外提供的服务。在组件和组件之间、组件和客户之间都通过接口进行交互。因此组件一旦发布，它只能通过预先定义的接口来提供合理的、一致的服务。这种接口定义之间的稳定性使客户应用开发者能够构造出坚固的应用。一个组件可以实现多个组件接口，而一个特定的组件接口也可以被多个组件来实现。

组件接口必须是能够自我描述的。这意味着组件接口应该不依赖于具体的实现，将实现和接口分离彻底消除了接口的使用者和接口的实现者之间的耦合关系，增强了信息的封装程序。同时这也要求组件接口必须使用一种与组件实现无关的语言。目前组件接口的描述标准是 IDL 语言。

由于接口是组件之间的协议，因此组件的接口一旦被发布，组件生产者就应该尽可能地保持接口不变，任何对接口语法或语义上的改变，都有可能造成现有组件与客户之间的联系遭到破坏。

每个组件都是自主的，有其独特的功能，只能通过接口与外界通信。当一个组件需要提供新的服务时，可以通过增加新的接口来实现，不会影响原接口已存在的客户，而新的客户可以重新选择新的接口来获得服务。

1.3.3　组件化程序设计

组件化程序设计方法继承并发展了面向对象的程序设计方法。它把对象技术应用于系统设计，对面向对象的程序设计的实现过程作了进一步的抽象。我们可以把组件化程序设计方法用作构造系统的体系结构层次的方法，并且可以使用面向对象的方法很方便地实现组件。

组件化程序设计强调真正的软件可重用性和高度的互操作性。它侧重于组件的产生和装配，这两方面一起构成了组件化程序设计的核心。组件的产生过程不仅仅是应用系统的需求，组件市场本身也推动了组件的发展，促进了软件厂商的交流与合作。组件的装配使得软件产品可以采用类似于搭积木的方法快速地建立起来，不仅可以缩短软件产品的开发周期，同时也提高了系统的稳定性和可靠性。

组件程序设计的方法有以下几个方面的特点：

①编程语言和开发环境的独立性；

②组件位置的透明性；

③组件的进程透明性；
④可扩充性；
⑤可重用性；
⑥具有强有力的基础设施；
⑦系统一级的公共服务。

1.4 组件技术标准

面向组件技术对一组类的组合进行封装，并代表完成一个或多个功能的特定服务，也为用户提供了多个接口。整个构件隐藏了具体的实现，只用接口提供服务。这样，在不同层次上，构件均可以将底层的多个逻辑组合成高层次上的粒度更大的新构件，甚至直接封装到一个系统，使模块的重用从代码级、对象级、架构级到系统级都可能实现，从而使软件像硬件一样，能任人装配定制而成的梦想得以实现。

目前主流的软件构件技术标准有：微软提出的 COM/COM+、SUN 公司提出的JavaBean/EJB、OMG 提出的 CORBA。它们为应用软件的开发提供了可移植性、异构性的实现环境和健壮平台，结束了面向对象中的开发语言混乱的局面，解决软件复用在通信、互操作等环境异构的瓶颈问题。

1.4.1 公共对象请求中介结构 CORBA

面向对象方法是软件构件技术的基础。为了真正实现软件构件化，还必须解决分布式计算和对象的互操作问题。因为按上述构件技术的目标，要求构件间能互操作，而且这些构件也允许分布式地放置在网上异构环境下的不同结点上。

为了协调和制定分布式异构环境下应用软件开发的统一标准，1989 年成立了一个国际组织，叫对象管理联盟(OMG)。加盟此组织的单位愈来愈多，现已有 750 多个单位，其中包括软件的开发供应商，软件用户和软件技术的研究院所等。经过多年的努力，已制定了一系列的标准规约，称为 CORBA(公共对象请求中介结构)。CORBA 的核心是对象请求中介(ORB)，是分布式对象借以相互操作的中介通道。另外还定义了最基本的对象服务构件和公共设施构件的规约。OMG 所定义的 CORBA 并不规定具体的实现。实现 CORBA 的软件由各个厂家自行开发。现已有多种可用的产品版本发布。

如上所述，CORBA 的核心 ORB 的作用是将客户对象(Client)的请求发送给目标对象(在 CORBA 中称为对象实现(Object Implementation)，并将相应的回应返回至发出请求的客户对象，如图 1.1 所示。ORB 的关键特征是客户与目标对象之间通信的透明性。在通信过程中，ORB 一般隐蔽了目标对象的以下内容。

①目标对象的位置：客户毋须了解具体目标对象所在的地址。目标对象可在同一机器的相同或不同进程中，也可在网络上另一机器的进程中。

②对象实现的方式：客户毋须了解具体目标对象是如何实现的，用何种语言写成的，也毋须了解该对象所在的操作系统和具体的硬件环境。

③对象执行的状态：当客户发送请求时，它毋须了解目标对象当前是否处于激活状态(即是否处于一个正在执行的进程中)。若有必要 ORB 可透明地激活该对象。

④对象通信机制：客户毋须了解 ORB 使用何种底层通信机制来发送请求和响应回答（如 TCP/IP，分享存储器及本地方法调用等）。

ORB 的通信透明性使得应用开发者可较少考虑低级分布式系统的程序设计问题，而更多地关心应用领域问题。

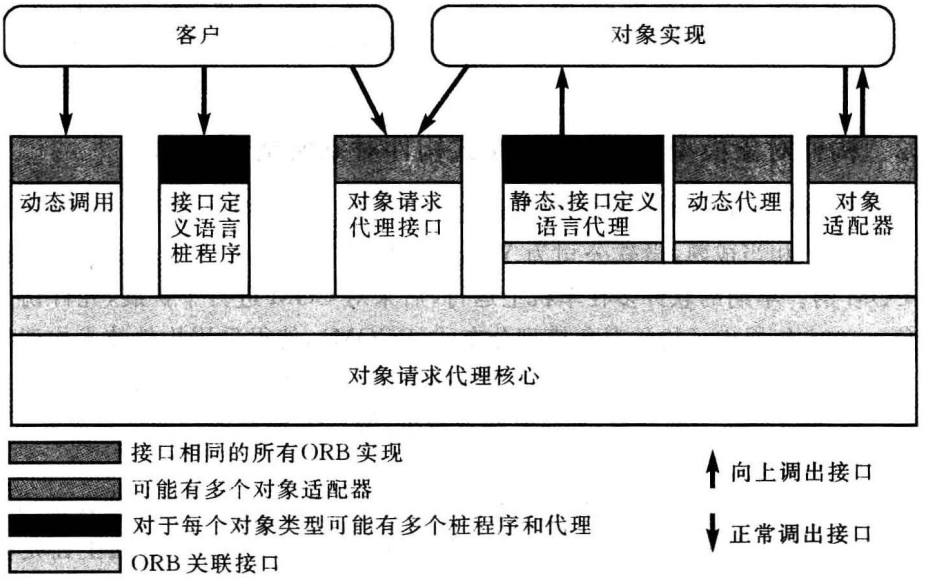


图 1.1　ORB 结构示意

OMG 接口定义语言 IDL 用于定义对象的接口。一个对象的接口指定该对象所支持的类型和操作，因而唯一定义了可用于该对象的请求形式。客户在构造请求时，必须了解对象的接口。如上所述，保持接口描述的“语言中性”对在异构环境中实现分布式应用是重要的。IDL 仅为一个说明式语言，而不是一个全面的程序设计语言。因此，IDL 本身并不提供诸如控制结构这样的特征，IDL 也不能直接用于实现分布式应用。相反，客户和对象的实现是采用具体的程序设计语言完成的。因此，ORB 所支持的特征必须能够在实现语言中访问。语言映射决定 IDL 的内容如何映射为具体程序设计语言的设施。IDL 编译器将具体接口定义翻译为目标语言代码。目前 OMG 已完成了从 IDL 到 C、C＋＋、Java、Smalltalk、Ada95、Cobol 等语言映射的标准化工作。

OMG IDL 编译器除了生成目标语言类型外，同时生成客户端的存根（Stub）和服务端的骨架（Skeleton）。存根是一个可有效创建和发送客户端请求的机制，而骨架是一个可将客户端请求传送至 CORBA 对象实现的机制。因为存根和骨架是直接从 CORBA 对象接口的 OMG IDL 描述中翻译而得，故存根和骨架通常是与特殊对象接口相关。通过存根和骨架发送和传递请求的方式通常称为静态调用。存根和骨架被直接嵌入客户应用和对象实现，因此，它们具有需调用的 CORBA 对象接口的所有静态信息。

除了使用存根和骨架的静态调用方式外，CORBA 提供了动态调用接口 DII 和动态骨架接口 DSI，前者支持动态客户请求调用，而后者支持将请求动态指派给对象。客户程序可通过使用 DII 对任何对象进行请求调用，而毋须持有对象的编译时信息。

CORBA 对象适配器(Adapter)的作用是配合对象实现和 ORB 本身的连接。Adapter 本身是一个对象,它使被调用对象的接口适配于调用对象所期望的接口。

CORBA 除了对核心 ORB 作了规定以外,还定义了对象服务和公共设施构件的规约。对象服务包括最基本和最常用的服务内容,如名字服务、事件服务等,而公共设施则包括范围更广的、建立在对象服务之上的服务,如用户界面、信息管理、系统管理和任务管理等。

CORBA 对应用系统未作具体规定,它可以建立在对象服务和公共设施之上,利用它们中的构件。

1.4.2 构件对象模型 COM 和分布式构件对象模型 DCOM

微软公司是也较早采用构件技术的公司之一。1993 年,微软公司提出了构件对象模型(COM)。此技术已相当成熟,微软公司为 Windows®和 Windows NT 开发的应用软件几乎都是基于 COM 的。早期的软件多在单机上运行,后来对 COM 进行了扩展,允许访问其它机器上的对象。1996 年提出了构件对象模型分布式(DCOM),使得采用构件技术构建网上的应用系统成为可能。除了 COM、DCOM 以外,微软还为开发分布式企业级应用软件提出了很多在 Windows NT®服务器上的服务,如微软作业服务(MTS)、微软因特网信息服务(IIS)、控件服务页面(ASP)、微软消息查询服务(MSMQ)等。

有人曾将 DCOM 和 CORBA 从程序设计结构、远程调用结构以及通信协议结构三个层次上进行了比较。虽然在基础原理和结构上有很多相近之处,但是在具体作法上还是有很大差异。也有人对 DCOM 和 CORBA 各自的优势和不足进行过评论。认为 DCOM 有较强的工具和系统的支持,另外由于有些功能已嵌入在操作系统中(特别是 Windows NT®,所以在降低花费上有优势。但是 DCOM 过多地依赖微软的操作系统平台,因而对异构网络环境,在兼容性方面会有不少问题。而正相反,CORBA 在支持多种平台和多种语言上具有优势,而且有比较广泛的独立开发商和用户及业界的支持。此外,CORBA 所采用的对象概念以及强调网络透明等在技术上也比较成熟。当然,CORBA 的不足之处是不如 DCOM 的支持工具那么多,另外在不同的开发商提供的 CORBA 实现之间的兼容性方面还有不少问题。但事物在不断发展,DCOM 和 CORBA 都会设法再改进自己的不足。

1.4.3 Java 和 Java2 环境平台企业版 J2EE

Java 语言由于巧妙地采用了虚拟机的机制,使得编译后产生的泛代码程序可以在各种平台上执行,从而做到了程序执行与平台无关。加之用 Java 编的 Applet 可以方便地用浏览器下载运行,Java 语言普及和发展得很快。Java 采用了构件技术,发展了 Java 构件(即 JavaBean)和企业级 Java 构件(即 EJB)。为了用构件技术组成实际的应用系统,最近又推出了 J2EE(Java2 环境平台企业版 Version 1.2 1999)和 Java 程序设计模型。

按照此模型组成的应用系统至少分为三层。

第一层是客户层,可以采用一般的浏览器或特制的客户软件。从服务器下载的 Applet 可以带有 JavaBean 一起在客户端执行。为了避免由于不同厂商提供的浏览器中虚拟机的差异,还专门提供了虚拟机软插件,做到程序的语义一致。为了保证安全,客户分防火墙内外,外客户只能从服务器进入,而内客户允许使用 RMI、IIOP 等直接访问 EJB。

第二层是中间层,即业务逻辑层。其中有两个包容器,一个是 Web 包容器,另一个是 EJB

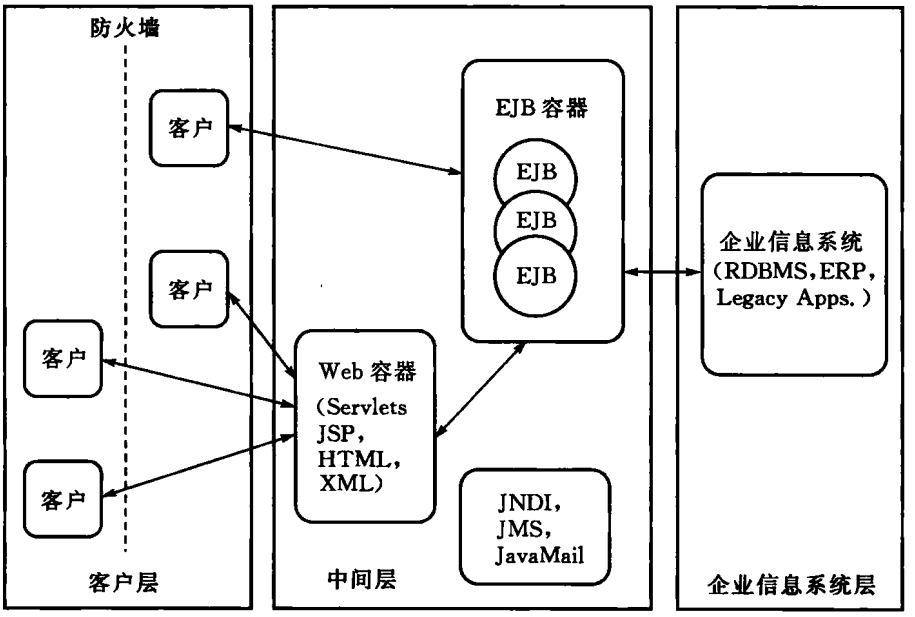


图 1.2　JavaEE 分层体系结构

包容器。Servlets Java 服务器页面(JSP)技术使人机界面的开发变得非常容易,而 Servlets 则方便为 Applet 等客户程序提供服务。简单的业务逻辑由开发人员编写业务 Bean,而复杂的业务逻辑则由 EJB 完成。

第三层是企业的信息系统。第二层的构件通过 JDBC(访问关系数据库),JNDI(Java 名子目录接口),JMS(Java 消息服务),Java Mail(发送和接收信件),Java IDL(与 CORBA 构件接口)访问第三层企业的信息系统。为了保护过去的投入,第三层可以与传统的应用软件、ERP 等建立联系。

1.5　组件技术与企业级业务软件

EBS 是 Enterprise Business Software 的缩写,即企业级业务软件。EBS 采用构件和集成新技术可以为企业软件用户带来如下一些好处。

构件技术可以使企业方便地、快速地、平滑地增加新的功能。新的构件同原有的构件可以集成在一起可靠地工作。系统解决方案可以特别灵活地、动态地重新配置,将一个构件替换为升级的新版本不必考虑对其它构件的适配。特别对那些需要灵活地,快速地对部分系统而不是整个系统升级换代的企业带来巨大的好处。采用构件技术允许对给定的任务采用不同的软件开发供货商提供的软件。企业在实现它的解决方案时具有选择产品的充分自由。企业可以容易地、灵活地将为企业特别设计的构件与整个系统集成使用,从而实现企业的特殊需求。基于构件的解决方案能够为进一步方便地扩展系统功能提供方便,因为定制的构件的接口也可以由用户特殊构件的使用。

企业业务软件开发商采用构件和集成新技术,还基于以下的考虑。

①软件产业面临的最大的问题是软件维护问题而不是软件开发问题。凡是以 EBS 为主

营业务的软件产业都要考虑长远一些，因为这类软件有一个比较长的生命周期，投资保护是一个重要方面。软件开发所依赖的技术和所实现的软件的功能作为一个连续的过程在不断发展，公司要在这整个过程中获取利润，这就导致了软件维护的问题。在IT基础设施不断更新的条件下一个软件公司如何能保持它的技术更新和功能增强的优势？在软件要适应用户企业特殊功能时，此问题特别突出。

②IT技术变更的速度远高于企业的业务变更的速度。软件技术更新周期愈来愈短对EBS软件影响太大。如客户/服务器技术、INTENET、B/S结构、Java、以及面向对象的通信机制CORBA等新技术层出不穷。从理论上讲这种变更还会继续下去。与此有明显对照的是业务过程和基本原则本身却改变相对甚少。采用构件技术后，技术的更新是以构件的形式实现的，而不必影响企业的业务过程。

③“职能分离”的设计原则。由于业务的改变远比技术的改变慢得多，因而在企业业务软件中将不同任务的软件按职能分离是一个非常适用的方法。如果系统中某部分进行了技术更新，只需将此部分的构件替换成新的构件无需改变另外的构件。但软件构件能够组合的前提是它们必须基于共同的设计。软件构件不应看作是孤立的单个的构件，还要考虑构件间交互的方式。软件构件必须用大家都能理解的语言来交谈，以保证能成功地协同工作。

④综合速度和灵活性两个关键因素。过去，企业基本上采用两种可能的方式获得和使用软件来支持企业的业务过程。一种是购买完整的标准的应用软件包，一种是由企业内部的IT部门来开发和维护应用软件以满足企业的需求。第一种方式有两个优点，一是购买软件实现相当快，二是不需内部开发部门的花费。但是，一个企业，只有自己开发软件或者将适合它的特殊需求的软件产品进行组合才能满足企业的IT需求。因而现代EBS领域的任务是将上述两种方式综合起来。既要软件包解决方案的速度，又要企业内部剪裁系统的开放性和灵活性。这就是建立EBS构件的基本思想。

1.6 组件技术的发展

从软件产业来看，无论是大的软件企业还是小的软件企业，目前很多都在做ERP。如果采用构件技术，小企业可以只做某些模块的构件，而大企业负责组装构件。这样，小企业就可以把构件卖给大企业，不仅大企业的成本降低了，小企业也能从中赚取利润。现在，国际上大的软件企业就是通过这种方式把一些软件工程的一部分外包给小企业，从而提高生产效率，提升规模化生产能力。

在这种新的软件开发方式下，软件公司将以开发构件为主要业务，提供规格化的软部件。系统集成商则汇总部件，组合成能完成不同功能的软构件，将自己的核心技术构件化。正是这两者之间分工的泾渭分明，将软件行业工业化逐渐推向成功。

可以想像，未来的软件产业将划分为三种业态。

第一个是构件业，类似传统产业的零部件，这些构件是可以买卖的。国家工程研究中心的构件库现在已经具备了这样的职能。

第二个是集成组装业，相当于汽车工厂，根据市场的需要先设计汽车的款型，然后到市场上采购通用零部件，特别需求还可以委托专门生产零部件的企业去设计生产，最后把这些零部件组装在一起。