

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

专业程序员快速成长之路丛书(2)

EJB

最新技术开发指南

北京希望电子出版社 总策划
李树仁 王锋锐 马凡 等 编写



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

专业程序员快速成长之路丛书 (2)

EJB

最新技术开发指南

北京希望电子出版社 总策划
李树仁 王锋锐 马凡 等 编写



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

前 言

首先，很荣幸由我来代表三位作者撰写本书的前言。

□ 什么是 EJB

EJB 的全称是 Enterprise Java Bean，是 Java 中的商业应用组件技术。EJB 并不是一个产品，它是一种创建在服务端可扩展的、事务处理的、多用户安全的企业级应用。EJB 是 Java 服务器端服务框架的规范，软件厂商根据它来实现 EJB 服务器。应用程序开发者可以专注于支持应用所需的商业逻辑，而不用担心周围框架的实现问题。

□ 本书读者对象

本书的适用对象是所有对 EJB 技术感兴趣的读者，特别是希望能创建企业应用的开发者。

□ 本书的特点

本书的目的是从使用的角度提供一个详细学习和使用 EJB 的指南，从 EJB 的基本概念学习到具体应用的开发，包括会话 Bean、实体 Bean、消息驱动 Bean、事务处理等概念及其使用方式。为了更好地讲解 EJB 相关技术，本书有大量的图示和例子，这有助于读者对 EJB 的学习。

□ 本书章节安排

本书的章节安排是这样的：

第 1 章主要介绍 EJB 中的基本概念，介绍 J2EE 体系结构的发展历史、EJB 的体系结构等。

第 2 章主要讲述 EJB 组件的基本结构。

第 3 章主要介绍 EJB 的配置和部署过程。首先介绍部署描述符和 JAR 文件，然后讲述部署的具体过程，并且结合一些具体的应用服务器以供读者理解。

第 4 章主要介绍 EJB 中应用服务器的基本概念，了解应用服务器的历史、EJB 容器的作用等。

第 5 章主要介绍 EJB 体系结构定义的在应用开发和部署过程中的 6 个不同角色。

第 6 章主要介绍会话 Bean 的开发。

第 7 章主要介绍实体 Bean 的开发。

第 8 章主要讲述会话 RMI 规范。

第 9 章主要介绍消息驱动 Bean 的基本知识及其应用方法。消息驱动 Bean 是 EJB 2.0 中新出现的一种，它为 EJB 应用中的消息服务提供了更好的实现。

第 10 章主要介绍一个基本企业应用所需了解的事务相关内容。

第 11 章主要讲述 EJB 中的安全管理。

第 12 章主要介绍 Web 的各种组件及其与 EJB 之间的关系。

第 13 章主要介绍 EJB 的应用决策、入门实例和一个综合实例。

另外还安排了 3 个附录，其内容分别是：EJB API、词汇表、优化 Entity Beans 的七条守则源码参考。

□ 致谢

在这里，我要感谢两位合作者王锋锐、马凡，是我们的共同努力促成了本书的圆满完成。请允许我对这本书的作者做一简要介绍。

李树仁 中国科学院计算机网络信息中心数据库应用研究室工作，计算机应用专业硕士。发表论文 10 余篇，主要从事软件开发和研究工作，有丰富的软件开发经验，做过很多的国家科研项目，对 Java 技术有很深的研究，出版了多本有关 Java 方面的书籍，现为中国科学院计算机网络信息中心副研究员。

王锋锐 中国航空结算中心亚科技中心工作，软件设计专业硕士。发表论文 3 篇，具有较高的学术造诣，主要从事软件开发工作，接触过较多的 IBM 大型机和相关的技术开发。

马 凡 中国科学院计算机网络信息中心数据库应用研究室工作，计算机及其应用软件专业毕业。有丰富的软件开发能力，做过多个部级项目和国家级科研项目，熟悉 C 和 C++，精通 Java 编程，在软件开发项目的管理及控制上有一定的实践经验和心得。

同时也要感谢北京希望电子出版社对我们的信任和大力支持，感谢清华大学的朋友张威以及武建军、刘欣蒙对本书的关心、支持与帮助。没有这些朋友们的帮助，就没有这本书的问世。

另外，参加本书编写工作的人员还有：杨力、孙功、黄卫荣、史严、姜心、吴波、何颖、刘海鹏、梁锦宗、曾春明、李超、曹鹏、朱柘、谢立、晏烁、陆志良、姜楠、贺文、郑鹏、张强等。在此，对他们为本书所付出的辛勤劳动一并表示感谢！

由于时间仓促，再加上作者水平有限，书中难免会有一些缺陷和不足之处，希望广大读者不吝赐教并提出宝贵意见。

李树仁

2002 年 7 月于北京

北京希望电子出版社网站 (www.bhp.com.cn) 欢迎您!



www.bhp.com.cn



欢迎你来“未来希望”的世界



2002年3月5日
16:16:55
上书店: “本站最新”

北京希望
电子出版社

首页 | 书盘目录 | E-Book | 技术支持 | 读者俱乐部 | 经销商园地 | 本社简介 | 网上书店 | 本站指南

最新图书 | 热点图书 | 隆重推出 | 热点光盘 | 本月排行 | 购物车

计算机普及

Macromedia公司与北京希望电子出版社合作,首次在中国出版它们的ATC专用教程。本教程全套5册,均出自于Macromedia公司软件开发专家之手,其全面性、严谨性和权威性是无可置疑的。这一点是本套教程与市场上林林总总的Macromedia软件教科书的重大区别。凡通过它们的ATC的培训、考核者,即可得到Macromedia公司的认证,为此,Macromedia公司在全中国建立了培训、考试、认证体系。

- Macromedia Flash 5 标准教程
- Macromedia Fireworks 4 标准教程
- Macromedia Dreamweaver 4.02 (中文版) 标准教程

[查看考点名录](#)

编程指导

- 因特网Internet高级编程教程
- 解读黑客内幕大曝光
- 全面引爆XSP网站开发——XSP/Cocoon/XML核心技术内幕
- Delphi 6深入编程技术
- Visual C++ 6.0数据库高级编程
- ASP.NET/XML深入编程技术
- Ultravox 4/JSP/XML高级实例教程
- Microsoft VB.NET开发人员指南

每周新书

- 排版大师Adobe PageMaker 7.0短期培训教程
- Pro/ENGINEER 2000i零件设计实训教程
- 中文 Adobe Photoshop 6.0 特效设计
- 跟我学用多媒体——精通Authorware 6.0
- 数字角色动画设计秘笈
- 未来希望
- 3ds max 4三维艺术与技巧
- 中文PowerPoint2002演示文稿制作实例教程
- 网站设计基础与实例教程
- 数码相机大观
- 童年的故事: 3ds max 4/Photoshop 6/Premiere 6千禧MTV角色动画大制作
- Pro/Engineer 2000i基础与实例教程
- 轻轻松松学用数码相机
- 网络程序设计教程

下载专区

本社图书资源配套源文件、源代码下载
Hope 3D室内装修软件
希望屏幕保护
本社2001年书目下载(免费阅读)
最新书目下载(免费阅读)
平面滤镜 3D 插件
查看Motion Capture技术资料及RIP动作文件下载

北京希望电子出版社网站是一个完备的电子商务交易系统,集信息发布、客户服务、网上销售为一体,为广大读者和希望图书产品经销商提供全方位的服务。

信息发布——每日更新 在北京希望电子出版社的网站里有数千种图书光盘的数据资料,读者和经销商可以了解到图书光盘的所有信息,可以看到书的目录、内容提要、精彩章节,相关资料,也可以了解书的印刷、开本、定价等情况。如果您不知道想找的某一本书在什么地方,您可以在“书盘检索”的窗口里输入书名、作者、书号等信息查询。同时,网站数据每日更新,并发布本社经营活动的相关信息和业界动态。

技术支持——全程服务 技术支持是我社倡导“服务第一”理念的体现,在这里我们向读者介绍IT行业的新技术,新软件,新动态;组织大家在BBS讨论热点技术性问题,并有专家回答技术咨询。只要您是希望的用户,就可得到希望的技术支援,解除用户的后顾之忧。在这个栏目里您看到大量精彩的计算机图形图像作品和Flash动画,此外,年轻的读者还可以在“读者俱乐部”发表自己的作品。

网上书店——方便快捷 在网站可以方便地选购图书光盘,付款方式灵活,如果您有招商银行的“一卡通”,可以在网上付款。所有客户都可以享受到免费邮寄的服务。此外网上购书还有各种优惠,总有意想不到的惊喜等着您!

只要您点击 www.bhp.com.cn 决不会空手而归!

北京希望电子出版社网站

“书目信息”服务办法

为了帮助您尽早了解我社的新书出版信息,我社充分发挥自身计算机软件企业的技术优势,在本社的网站(www.bhp.com.cn)上开设了书目信息的服务项目,它可以使您每天与出版社同步得到新书出版信息,也可以为您提供本社数年来的全部书盘目录。

在网站查找、下载本社书盘目录的方法如下:

1、在网站的首页查找新书。在浏览器地址栏处输入 www.bhp.com.cn 之后便进入我社网站的首页,首页的右侧有“每周新书”栏目,该栏目每天更新,增添我社的当日新书。在首页上方的栏目条里,有“最近新书”栏目,包括近两个月的新书介绍。

2、在“书盘目录”栏目里,有本社图书光盘的分类目录。通过分类目录,可以找到同类图书光盘的目录和系列书目录,最后到达的是书的详细介绍,包括书的版权页信息、本书内容提要、目录等等。

3、在“经销商园地”栏目里,设有“书目下载”专栏。内容包括:1995年——2000年的书盘目录下载;1999年书盘目录下载;2000年书盘目录下载;2001年内书盘目录按出版月份下载。书目下载是专门为希望电子图书经销商,书店提供的信息服务,与我社的图书目录单内容相同,是图书目录单的电子版,您可按此书目向我社定购图书。

书目下载文件为 zip 格式的压缩文档,解压缩后为 Word 格式。下载的书目操作步骤是:

- (1) 进入本社的网站(www.bhp.com.cn)“经销商园地”/“书目下载”栏目。
- (2) 点击所选取的连接,出现对话框,选取“将该文件保存到磁盘”,点击“确定”。
- (3) 在“另存为”对话框选择存放文件的路径,点击“保存”按钮后计算机便开始下载。
- (4) 下载完毕,启动自己的“zip 软件”,沿着前面文件的存放路径找到该文件,解开压缩,打开文件阅读。

如果您的计算机里还没有 zip 软件,可以在我们的网站下载一个。地址是:首页/技术支持/工具箱/常用工具/Winzip80

目 录

第 1 章 EJB 综述	1
1.1 EJB 基本概念	1
1.1.1 分布式对象	1
第 1 章 EJB 综述	1
1.1 EJB 基本概念	1
1.1.1 分布式对象	1
1.1.2 服务器端组件模型	3
1.1.3 应用服务器	6
1.2 J2EE 体系结构	6
1.3 EJB 的体系结构	9
1.3.1 Enterprise Beans 组件	10
1.3.2 EJB 角色	11
1.3.3 EJB 架构的设计目标	12
1.4 EJB 规范	13
1.5 EJB 编程模型	19
1.6 小结	23
第 2 章 EJB 组件基本结构	24
2.1 EJB 基本类型	24
2.2 Home 接口	25
2.3 Remote 接口	27
2.4 Bean 类	28
2.5 EJB 组件的约束	28
2.6 小结	31
第 3 章 EJB 的配置与部署	32
3.1 J2EE 的部署描述符	32
3.1.1 EJB 部署描述符	32
3.1.2 容器管理的 EJB 部署描述	35
3.1.3 部署描述符元素说明	37
3.2 EJB JAR 文件	41
3.2.1 JAR 文件简介	41
3.2.2 JAR 文件的目的	46
3.2.3 EJB 客户端 JAR 文件	47
3.3 EJB 的部署过程	47
3.4 在应用服务器上的部署过程	48
3.4.1 LDAP 无状态会话 Bean	48
3.4.2 ProfileTest——CMP 实体 Bean	51
3.5 开发 EJB 的步骤实例	53
3.6 小结	57
第 4 章 EJB 容器	58

4.1	应用服务器.....	58
4.1.1	应用服务器的现状.....	58
4.1.2	应用服务器的发展过程.....	60
4.1.3	应用服务器的发展方向.....	61
4.2	EJB 容器.....	64
4.3	选择一个应用服务器.....	70
4.3.1	技术原则.....	70
4.3.2	实例分析.....	71
4.4	小结.....	72
第 5 章	EJB 角色和开发过程.....	73
5.1	EJB 角色.....	73
5.1.1	Bean 开发者.....	74
5.1.2	应用组装者.....	75
5.1.3	部署者.....	76
5.1.4	EJB 容器提供者.....	76
5.1.5	EJB 服务器提供者.....	77
5.1.6	系统管理员.....	78
5.2	EJB 开发过程.....	78
5.3	小结.....	83
第 6 章	会话 Bean 应用开发.....	84
6.1	无状态的会话 Bean.....	85
6.1.1	无状态会话 Bean 简介.....	85
6.1.2	无状态会话 Bean 的生命周期.....	85
6.1.3	一个无状态会话 Bean 实例.....	87
6.1.4	开发一个无状态会话 Bean 实例的步骤.....	92
6.2	有状态的会话 Bean.....	97
6.2.1	有状态会话 Bean 简介.....	99
6.2.2	有状态会话 Bean 的活化和钝化.....	99
6.2.3	有状态会话 Bean 的生命周期.....	100
6.2.4	一个有状态会话 Bean 的实例.....	101
6.3	会话 Bean 的客户端.....	104
6.4	两种会话 Bean 的选择.....	114
6.5	开发实例.....	114
6.6	小结.....	133
第 7 章	实体 Bean 应用开发.....	134
7.1	理解实体 Bean.....	134
7.2	实体 Bean 的客户视图.....	135
7.2.1	Home 接口.....	135
7.2.2	Remote 接口.....	136
7.2.3	主键和对象标志.....	136
7.2.4	实体 Bean 句柄.....	137
7.3	容器管理的持久性.....	138

7.4	Bean 管理的持久性.....	143
7.5	JDBC 简介.....	149
7.6	EJB 查询语言.....	152
7.7	实体 Bean 的生命周期.....	155
7.8	实体 Bean 的配置与部署.....	156
7.9	如何选择实体 Bean 和会话 Bean.....	158
7.10	优化 Entity Bean 七原则.....	158
7.11	开发实例.....	162
7.12	小结.....	185
第 8 章	RMI 规范.....	186
8.1	Java 分布式对象模型.....	186
8.2	远程对象.....	190
8.3	客户机接口.....	194
8.4	服务器接口.....	196
8.5	接口.....	206
8.6	激活.....	209
8.7	stub 和 skleton 所用的接口和类.....	228
8.8	垃圾收集器接口.....	233
8.9	RMI 通信协议.....	236
8.10	小结.....	242
第 9 章	消息驱动 Bean 应用开发.....	243
9.1	JMS 简介.....	243
9.2	消息驱动 Bean.....	251
9.3	消息驱动 Bean 的基本结构.....	253
9.4	消息 Bean 的配置和部署.....	254
9.5	一个消息驱动 Bean 实例.....	257
9.6	开发实例.....	263
9.7	小结.....	268
第 10 章	事务.....	269
10.1	何谓事务.....	269
10.2	使用事务的原因.....	270
10.3	数据库操作的事务管理.....	271
10.4	CORBA OTS.....	272
10.5	事务的 4 个特征 (ACID).....	273
10.6	JTA 和 JTS.....	275
10.7	声明的事务管理.....	277
10.7.1	事务范围.....	278
10.7.2	理解事务属性.....	278
10.7.3	事务属性.....	279
10.7.4	事务属性用法.....	282
10.7.5	事务属性的总结.....	283
10.8	事务的编程划分.....	283

10.9 Bean 自管理的事务处理.....	286
10.10 事务与实体 Bean.....	288
10.11 事务模型.....	289
10.12 事务隔离性.....	292
10.13 会话同步接口.....	295
10.14 加入和使用事务.....	296
10.15 开发实例.....	297
10.16 小结.....	304
第 11 章 安全管理	305
11.1 EJB 安全模型概述.....	305
11.2 EJB 的安全机制和安全管理.....	306
11.2.1 验证.....	307
11.2.2 授权.....	308
11.2.3 安全通信.....	309
11.3 EJB 6 种角色在安全管理中的职责.....	309
11.4 开发实例.....	313
11.5 小结.....	314
第 12 章 Web 组件和 EJB	315
12.1 Applet 与 EJB.....	315
12.2 Servlet 与 EJB 交互.....	335
12.2.1 Servlet 概述.....	335
12.2.2 使用 Servlet 与 EJB 进行交互.....	336
12.3 JSP 与 EJB 交互.....	340
12.4 小结.....	340
第 13 章 EJB 的综合应用实例	341
13.1 EJB 应用决策.....	341
13.2 入门实例.....	344
13.2.1 JBuilder 的安装与配置.....	344
13.2.2 EJB 的典型示例.....	348
13.2.3 Entity Bean 的典型示例.....	358
13.2.4 部署文件参考.....	370
13.3 综合实例.....	371
13.4 小结.....	386
附录 A EJB API	387
A.1 javax.ejb 包.....	387
A.2 javax.ejb.spi 包.....	391
附录 B 词汇表	392
B.1 J2EE 所包括技术列表.....	392
B.2 其他术语列表.....	392
附录 C 优化 Entity Beans 的七条守则源码参考	396

第 1 章 EJB 综述

本章重点

- EJB 基本概念
- J2EE 体系结构
- EJB 体系结构
- EJB 规范
- EJB 编程模型

1.1 EJB 基本概念

Sun 公司对 EJB (Enterprise Java Bean) 的定义是: EJB 是用于开发和部署多层结构的、分布式的、面向对象的 Java 应用系统跨平台的构件体系结构。采用 EJB 可以使开发商业应用系统变得更容易, 应用系统可以在一个支持 EJB 的环境中开发, 开发完之后部署在其他的环境中, 随着需求的变化, 应用系统可以不加修改地迁移到其他功能更强、更复杂的服务器上。首先介绍 EJB 的几个基本概念。

1.1.1 分布式对象

1. 分布式对象产生的背景

分布对象技术是计算机网络技术和面向对象技术相互协调、相互促进而发展起来的。以前的计算机系统多是单机系统, 多个用户是通过联机终端来访问的, 没有网络的概念。网络出现后, 产生了客户机/服务器的计算服务模式, 多个客户端可以共享数据库服务器和打印服务器等。随着网络的更进一步发展, 许多软件需要在不同厂家的网络产品、硬件平台、网络协议异构环境下运行, 应用的规模也从局域网发展到广域网。在这种情况下, 客户机/服务器模式的局限性也就暴露出来了, 于是中间件应运而生。中间件是位于操作系统和应用软件之间的通用服务, 它的主要作用是用来屏蔽网络硬件平台的差异性和操作系统与网络协议的异构性, 使应用软件能够比较平滑地运行于不同平台上。同时中间件在负载均衡、连接管理和调度方面起了很大的作用, 使企业级应用的性能得到了大幅提升, 满足了关键业务的需求。但是在这个阶段, 客户端是请求服务的, 服务器端是提供服务的, 它们的关系不对称。随着面向对象技术的进一步发展, 出现了分布式对象技术。可以这么说, 分布式对象技术是随着网络和面向对象技术的发展而不断地完善起来的。90 年代初 CORBA (Common Object Request Broker Architecture, 公共对象请求代理体系结构) 1.0 标准的颁布, 揭开了分布式对象计算的序幕。

分布对象计算中, 通常参与计算的计算体 (分布对象) 是对称的。分布对象往往又被称为组件 (Component), 组件是一些独立代码的封装体, 在分布计算的环境下可以是一个简单的对象, 但大多数情况下是一组相关的对象复合体, 提供一定的服务。分布环境下,

组件是一些灵敏的软件模块，它们可以位置透明、语言独立和平台独立地互相发送消息，实现请求服务。

2. 分布式对象模型与标准

分布式对象技术采用面向对象的多层次客户机/服务器的计算模型，该模型将分布在网络上的全部资源（无论是系统层还是应用层）都按对象的概念来组织，每个对象都有明确定义的访问接口。创建和维护分布对象实体的应用称为服务器（或者为容器），按照接口访问该对象的应用为客户。服务器中的对象不仅能够被访问，而且自身也能作为其他对象的客户。因此，在分布式对象技术中，客户与服务器角色的划分是相对和多层次的。支持客户访问异地分布对象的核心机制称为 ORB（Object Request Broker，对象请求代理），ORB 处于分布式对象技术的核心位置。这里的 ORB 如同一条软件总线把分布式系统中的各类对象和应用连接成相互作用的整体。目前分布式对象技术已经成为建立应用框架和软构件的核心技术，其中有 3 种具有代表性，下面分别介绍。

（1）公共对象请求代理体系结构 CORBA。CORBA 是对象管理组 OMG（Object Management Group）设计的一个标准。CORBA 标准主要分为 3 个层次：对象请求代理、公共对象服务和公共设施。最底层是对象请求代理 ORB，规定了分布对象的定义（接口）和语言映射，实现对象间的通讯和相互操作，是分布对象系统中的“软总线”；在 ORB 之上定义了很多公共服务，可以提供诸如并发服务、名字服务、事务（交易）服务、安全服务等各种各样的服务；最上层的公共设施则定义了组件框架，提供可直接为业务对象使用的服务，规定业务对象有效协作所需的协定规则。总之，CORBA 的特点是大而全，互操作性和开放性非常好。目前 CORBA 的最新版本是 2.5。CORBA 3.0 也已基本完成，增加了有关 Internet 集成和 QoS 控制等内容。CORBA 的缺点是庞大而复杂，并且技术和标准的更新相对较慢，CORBA 规范从 1.0 升级到 2.0 所花的时间非常短，因此再往上的版本发布就十分缓慢了。

（2）组件对象模型 COM/DCOM。Microsoft 的 COM 平台效率比较高，同时它有一系列相应的开发工具支持，应用开发相对简单。但它有一个致命的弱点是 COM 的跨平台性较差，如何实现与第三方厂商的互操作性始终是它的一大问题。

（3）Java 标准。Java 的优势是纯语言的，跨平台性非常好。Java 分布对象技术通常指远程方法调用（RMI）和企业级 JavaBean（EJB）。RMI 提供了一个 Java 对象远程调用另一 Java 对象的方法的能力，与传统 RPC 类似，只能支持初级的分布对象互操作。Sun 公司于是基于 RMI 提出了 EJB。基于 Java 服务器端组件模型，EJB 框架提供了诸如远程访问、安全、交易、持久和生命期管理等多种支持分布对象计算的服务。

3. 基于分布式对象的软件开发

在面向对象技术给解决软件危机带来曙光之时，分布式网络计算的巨大压力又给软件开发提出了许多新的难题，使软件开发仍处于高风险状态。新的分布式网络计算要求软件实现跨空间、跨时间、跨设备、跨用户的共享，导致软件在规模、复杂度、功能上极大增长，迫使软件向异构协同工作、各层次上集成、可反复重用的工业化道路上前进。为适应软件的这种需求，新的软件开发模式必须支持分布式计算、浏览器/服务器结构、模块化和组件化集成，使软件类似于硬件，可用不同的标准组件拼装而成。具体地说，可实现下列

3 点要求:

- 提供一种手段, 使应用软件可用预先编好的、功能明确的产品部件定制而成, 并可用不同版本的部件实现应用的扩展和更新;
- 利用模块化方法, 将复杂的难以维护的系统分解为互相独立、协同工作的部件, 并努力使这些部件可反复重用;
- 突破时间、空间及不同硬件设备的限制, 利用客户和软件之间统一的接口实现跨平台的相互操作。

1.1.2 服务器端组件模型

当今最流行的服务器端组件模型来自于微软、Sun 和对象管理组织 OMG。下面简要地描述这些结构的轮廓。

1. J2EE 体系结构

J2EE 平台提供了一个基于组件的方法, 用于设计、开发、装配及部署企业应用程序, 是 SunONE 软件构架的基础组成部分。J2EE 平台提供了多层的分布式应用模型、组件重用、一致化的安全模型以及灵活的事务控制, 同时保证用户平台独立的、基于组件的 J2EE 解决方案不会被束缚在任何一个厂商的产品和 API 上。J2EE 技术获得了各大开发平台提供商的积极支持, 并已成为当今企业应用开发的主流平台。最近 Sun 公司发布的 J2EE v1.3 beta 版包含了许多引入注目的功能, 具体如下。

Java 消息服务: 定义了一组支持 nKs 消息 API 实现。

J2EE 连接器技术: 定义了扩展 J2EE 服务到外部非 J2EE 应用程序的标准。

XMLAPI: 提供了一组用于 XML 解析器的 JavaAPI。

Java Servlet: 提供了过滤器的实现并提出了一个轻量级的架构来实现请求/响应的传送。

JSP: 改进了对标记库的实时支持。

EJB 2.0: 引入了 MessageDriverBean (消息驱动 Bean) 和一个全新的 CMP 组件模型。

2. OMA/CORBA

对象管理组织 OMG 是一个国际性的软件行协会, 主要宗旨是促进面向对象的方法在软件工程中的应用, 以及在面向对象的软件工程方法学的基础上, 为大规模并行系统软件的开发与应用制定软件体系结构模型和通用接口规范。

OMG 于 1990 年末提出了一个对象管理结构的基础结构 OMA。OMA 主要涉及用于面向对象语言、系统、数据库以及应用程序框架的统一术语体系, 面向对象软件系统的抽象框架和基于面向对象技术的分布式软件系统参考模型等。为了构造上述参考模型, OMA 规定了下面 4 个方面的标准。

对象请求代理 ORB: 即关键通讯单元, 在不同的应用程序对象之间以高度协同的方式发送消息。

对象模型: 即独立设计的、可移植的抽象模型, 可以与其他符合 OMA 规范的面向对象系统进行通讯。

对象服务: 利用 ORB 实现基本对象功能, 确定对象的逻辑模型和物理存储方式。

通讯组件: 包含了适用于很多应用领域的基本功能, 这些功能可以通过符合 OMA 规

范的类接口获得。

OMG 于 1991 年末提出的 CORBA 是 OMA 参考模型中的 ORB 接口技术规范。这一标准规定了如何定义、创建、调度、引用对象，以及对象之间如何通讯。符合 CORBA 规范的 ORB 是典型的中间件，允许客户机对象向服务器对象发出请求。

CORBA 规范的基本组成如图 1-1 所示。

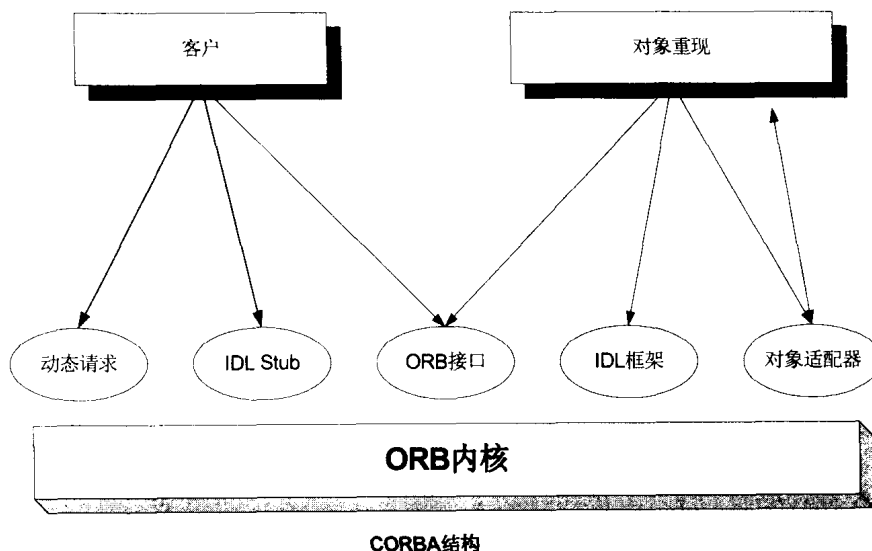


图 1-1 CORBA 结构

对象请求代理 ORB 是 CORBA 的核心，作为一个软件组件，对象可以通过 ORB 发出请求并接收响应。在 CORBA 中，所有的通讯都通过 ORB 进行，因而无论一个对象是本地的还是远程的，对于与之进行通讯的其他对象来说都是等价的。CORBA 并没有规定如何具体实现一个 ORB，几乎每一个不同的产品都有其具体的 ORB 实现。

接口定义语言 IDL (Interface Definition Language) 是用于描述对象接口的一种高级符号语言。IDL 不涉及任何接口的实现细节，所有 CORBA 系统都根据 IDL 用某种特定语言来实现接口。CORBA 为 C、C++、Smalltalk 和 Java 制定了规范，ADA95 和 COBOL 的映射规范也正在制定中。IDL 描述通常以接口库 (Interface Repository) 的方式进行存储。

接口库 (Interface Repository) 中包括了所有的服务对象属性、操作、自定义数据类型及异常处理的 IDL 定义。

基本对象适配器 BOA (Basic Object Adapter) 描述了 ORB 和服务器端应用程序之间的接口，BOA 负责调度服务器端应用程序维护的对象，并与服务对象交换消息。

静态请求接口 SII (Static Invocation Interface) 假设在编译时刻客户对象能够明确了解服务对象的接口，即客户与服务器之间的关系应当是可知、固定的，服务接口的任何改变都会导致软件系统的重新编译。SII 是由 IDL 描述的。

动态请求接口 DII (Dynamic Invocation Interface) 是一组与服务器无关的通用接口。与新服务对象及其行为相关的信息可以通过接口库获得，客户程序根据这些信息可以在进行时刻动态地构造请求。DII 使得开发智能化即插即用的软件系统成为可能，但需要涉及大量的 CORBA API 调用，增加了软件的复杂度。

总之, CORBA 提供了一个集成的框架, 应用程序只要给出用接口定义语言 IDL 书写的界面, 就可插入框架, 与其他对象协同工作, 为在分布式环境下实现不同应用程序即插即用的集成提供了有力支持。

3. Microsoft DNA

在基于 Web 的开发模型中, 应用程序是分布式的, 也就是说: 程序的一部分运行在客户机上, 一部分运行在对象服务器上, 还有一部分运行在数据库服务器上。这种结构有很多优点, 包括伸缩性、易于配置和扩展, 但是它们的有效性依赖于正确的实现。因此, 一个特殊的平台脱颖而出: Microsoft Windows DNA。Windows DNA 的结构涉及到一个 Web 服务器和一个事务处理服务器, 以及其他几个产品和服务。这些服务器经常被单独地称为 IIS (Internet Information Server) 和 MTS (Microsoft Transaction Server)。

这些东西共同形成了位于 Microsoft Windows 顶部的一个软件平台层, 它们开放地紧密集成起来, 并提供丰富的应用程序服务。MTS 和 Microsoft Windows 2000 下的强大服务结合在一起, 形成了现在大家所知道的 COM+, 即: $MTS+COM=COM+$ 。

Windows DNA 结构使开发者把精力集中在最需要的开发领域以满足业务需求, 而不用对它的支撑层操心过多。Windows DNA 提供内部结构和服务能方便地实现安全性、事务管理以及已有系统的互操作性。采用 Windows DNA 结构意味着能够一步一步地开始使应用程序支持 Web, 而不是从一无所有做起。通过支持最新的可扩展标记语言 XML 和简单对象访问协议 SOAP 技术, 用户可以确信应用程序能够很好地工作于现在和未来。

对于基于 Windows DNA 的应用程序, 客户应用程序可能就像在远程工作站上运行的浏览器那样简单。服务器端应用程序是 Web 服务器主持的 Active Server Page (ASP) 或 Internet 服务器应用程序编程接口 (ISAPI) 应用程序, Web 服务器可能执行脚本。脚本调用执行业务逻辑的对象, 然后业务对象再操作数据库服务器上的数据。

服务器端的所有操作都可以发生在一个或多个服务器上。多层的方法具有将负载分布到多个服务器的优点, 它比单服务器解决方案具有更好的可扩展性。

COM 对象允许开发人员封装业务功能。而且, 因为组件代表的是分散的任务, 所以也就更容易开发、测试和重复利用。将其应用于上述 Web 应用程序体系结构时, 在 Web 服务器上执行的脚本会调用实现业务逻辑的 COM 对象。COM 对象用三种方式协助创建可扩展的应用程序, 具体描述如下:

- 它们从表现和数据层分离多余的业务逻辑, 将每层专用于特定的任务;
- COM 对象可以共享资源, 例如数据连接;
- COM 对象可以分布到多个服务器上, COM 对象允许多个服务器执行相同的任务, 从而提高应用程序的可用性。

另外, Windows DNA 支持异步消息传递、可扩展的工作流体系结构、数据库复制、分布式数据库及 Windows 群集的故障转移群集解决方案。

使用 Windows DNA 结构, 能够建立可伸缩、可扩展的电子商务解决方案。对于电子商务应用程序将如何与其他电子商务程序通讯, 这些应用程序究竟在哪里运行, Microsoft 亮出了它的最新一代操作平台: Microsoft.NET。

4. Microsoft.NET

通过打开不同的 Web 应用程序并把它们呈现为 Web 服务, Microsoft.NET 平台方便了它们之间的集成。通过支持开放的 Internet 协议, .NET 平台允许电子商务应用程序以及其他基于 Web 的应用程序相互通讯、交换数据并实现了真正的互操作性。Microsoft 还开发了一套企业级服务器(共同称为 .NET Enterprise Server), 包括数据库服务器、商务到商务(B2B)管理服务器以及便于加强 Web 站点安全性的服务器。

至于 COM 组件, ASP.NET 将采用封装的形式使得原有的 COM 组件仍旧能够运行, 但这些 COM 组件将运行在 CLR 受管理的环境之外, 而且受管理以及非受管理环境之间的切换会牺牲一定的性能。因此, 很多用户可能会决定重新把 COM 组件编写成 COM+2.0。

值得提及的是, 为当前所开发的 DNA 应用的未来而担心仍旧是没有必要的。Microsoft 发布 .NET 最终版本的时间在 2001 年后, 而且基于 Microsoft DNA 体系的“老式”应用将仍能在装备了 .NET 工具的 Windows 2000 服务器上运行, 只是这些应用将无法利用任何 .NET 的优势。由此, 对于当前和未来的基于 DNA 的应用, 可以认为短期内不会受到任何影响或者威胁。

1.1.3 应用服务器

应用服务器在市场上非常热门。它的设计目标是管理业务系统的诸多方面, 包括许多组件, 如 EJB 服务器、HTTP 服务器、安全 HTTPS 服务器等。不同应用程序服务器厂家提供各种不同的应用服务器特性, 例如, 应用服务器包括 HTTP 服务器, 从而不必去部署应用程序时购买单独的 Web 服务器。同样, 应用服务器通常提供安全 HTTPS 服务器, 使应用程序可以使用安全的 Web 页面。

这种应用服务器竞争导致不同厂家实现的应用程序之间缺乏可移植性。服务器为通过 Sun 公司的兼容性证书测试, 需要进行 6000 多种测试。下列应用程序服务器已经通过 Sun 公司的兼容性测试: IBM Websphere, BEA Weblogic, Iplanet 和 SilverStream。最新更新信息见 Sun 公司的 Web 站点 (<http://java.sun.com>)。

1.2 J2EE 体系结构

在当今竞争激烈的市场中, 各种机构正面临着一种艰难的挑战: 既要在业已缩短的开发周期内降低成本, 又要提供广泛的电子商务服务。为了解决这个问题, 许多公司都在研究如何使用 Java 在中间层实施商业逻辑。Java 的使用者常常发现: 开发变得简单了, 部署的速度也变快了。此外, 分布式 Java 应用的实施还可以提高可伸缩性和可靠性。

过去, 选择 Java 的公司曾经遭遇安全和性能问题。在许多地方, 这些问题降低了 Java 在企业内的总体价值。Sun 公司已竭尽全力来解决这些问题, Java 2 企业版(J2EE)软件开发工具包即是这种努力的成果。

1. J2EE 概念描述

J2EE 是美国 Sun 公司刚刚推出的一种全新概念的模型, 与传统的互联网应用程序模型相比有着不可比拟的优势。当今许多企业都需要扩展他们的业务范围, 降低自身经营成本, 缩短他们和客户之间的响应时间, 这就需要企业、合作伙伴和雇员之间有一种简捷、快速

的服务。典型地说,提供这些服务的应用软件必须同企业信息系统(EIS)相结合,并提供新的能向更多的用户提供的服务。这些服务要具备以下的特点:

- 高可用性,用来满足现在的全球商业环境;
- 安全性,保护用户的隐私和企业数据的安全;
- 可依赖性和可扩展性,保证商业交易的正确和快捷。

通常这些服务是由分布的应用程序组成的,包括前端数据端和后端数据源以及它们之间的一层或几层,这些中间层提供了把商业功能和数据与 EIS 相结合的功能。这些中间层将客户端从复杂的商业逻辑中分离出来,利用成熟的 Internet 技术使用户在管理上所花费的时间最小化。J2EE 正式降低了开发这种中间层服务的成本和复杂程度,从而使得服务可以被快速地展开,并能够更轻松地面面对竞争中的压力。J2EE 通过定义一种标准的结构来实现它的优势,具体如下。

J2EE Application Programming Model: 一种用于开发多层次、瘦型客户用户程序的标准设计模型。

J2EE Platform: 一个标准的平台,用来整合 J2EE 的应用程序,指定一系列的接口和方法。

J2EE Compatibility Test Suite: 一套兼容测试组件,用来检测产品是否和 J2EE 平台兼容。

J2EE Reference Implementation: 用来示范 J2EE 的能力。

2. J2EE 应用程序模型

J2EE 被设计为顾客、雇员、供应商、合作者提供企业级服务,这样的应用程序天生具有复杂性,它们要访问各种类型的数据并向大量的客户端分发。为了更好地控制、管理这些应用程序,支持在中间层引入各种各样用户的商业功能,中间层描述了一个被企业的信息技术部门紧紧控制的环境。J2EE 应用程序依靠于 EIS 层来存储企业的商业数据。这些数据和用来管理它的系统是企业的核心。最初,双层结构(客户机-服务器)应用程序模型许诺将提供伸缩性与更多的功能,但不幸的是,直接向用户提供 EIS 服务的复杂性和在每台用户计算机上安装和维护商业逻辑所引起的管理上的问题成了很大限制。而这些双层结构中的限制在三层结构中已不存在了。

如今要求既开发服务的商业功能,又开发访问数据库和其他资源的代码,这是很复杂的技术问题,因为每一种多层结构的服务器都有它自己的应用模型,因此组建一支富有经验的开发队伍也将是困难的。另外,随着服务规模的扩大,为了降低开支和加快响应速度,也要经常对其基层代码进行修改。J2EE 应用模型定义了一种建筑模型来让多层应用程序实现服务,并消除了以上的问题,提供了可伸缩的、易访问的、易于管理的方法。

J2EE 应用模型把实现多层结构服务的工作划分为两部分:开发者实现商业和表达逻辑,以及由 J2EE 平台提供的标准系统服务。开发者可以依赖于这个平台为开发中间层服务中遇到的系统级硬件问题提供解决方案。J2EE 应用模型为中间层应用程序提供了“编译一次,任意运行”的特点,这种标准模型使培训开发人员的费用最小化。J2EE 应用程序模型通过在建立多层应用程序中使其复杂程度最小化,为简化和加速应用程序的开发迈出了重要的一步。

3. 安全性

当其他的应用程序模型需要在每个程序中有特定平台的安全性时,J2EE 平台定义了一