

国外计算机科学经典教材

J2EE 1.4 Essentials

J2EE 1.4 基础教程

(美) Aaron E. Walsh 著

李化潇东 译



清华大学出版社

内 容 简 介

本书介绍 J2EE 1.4 平台最基本的要素,重点阐述了 J2EE 1.4 的新特性和新功能。全书包括 12 章,分为三个部分。第 I 部分对 J2EE 平台提供了一个概述,并解释了什么是“Web 服务”以及 J2EE 1.4 如何支持这种分布式计算形式。第 II 部分讲述 J2EE 多层体系结构的 Web 层和业务层中的组件——servlet、JSP 和 bean。第 III 部分分析由 J2EE 1.4 支持的基于 XML 的新型 Web 服务功能。

本书适合于作为高等院校和软件学院计算机软件专业讲授 J2EE 的教材,也适合有一定经验的 Java 程序员阅读。

EISBN: 0-7645-2615-4

Aaron E. Walsh

J2EE 1.4 Essentials

Original English language edition Copyright © 2003 by Wiley Publishing, Inc., Indianapolis, Indiana. All rights reserved including the right of reproduction in whole or in part in any form. This translation published by arrangement with Wiley Publishing, Inc.

本书之英文原版由 Wiley Publishing, Inc., Indianapolis, Indiana 于 2003 年出版。版权所有,未经书面许可,不得以任何形式复制本书的全体或部分内容。本中文翻译版由 Wiley Publishing, Inc. 授权清华大学出版社出版。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号 图字:01-2003-2671

图书在版编目(CIP)数据

J2EE 1.4 基础教程/(美)沃尔什(Aaron E. Walsh)著;李化,满东译.—北京:清华大学出版社,2003

书名原文:J2EE 1.4 Essentials

(国外计算机科学经典教材)

ISBN 7-302-07407-0

I. J… II. ①沃…②李…③满… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 092467 号

出 版 者:清华大学出版社

地 址:北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

客户服务:010-62776969

组稿编辑:曹康

文稿编辑:陈宗斌

封面设计:康博

版式设计:康博

印 刷 者:北京嘉实印刷有限公司

装 订 者:北京密云县京文制本装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:20.5 字数:524 千字

版 次:2003 年 11 月第 1 版 2003 年 11 月第 1 次印刷

书 号:ISBN 7-302-07407-0/TP·5472

印 数:1~4000

定 价:42.00 元

译者序

自从 1995 年 Sun 公司正式推出 Java 以来, Java 技术迅猛发展。目前, 全球有 300 多万开发人员采用 Java 技术, 有 90% 以上的应用服务器基于 Java 平台。Java 的“火热”程度大家有目共睹。而 J2EE 1.4 平台的推出, 更是为 J2EE 添辉增艳。

本书是站在一个很高的角度来介绍 J2EE 1.4 的, 全书系统而全面地阐述了 J2EE 1.4 中的要点与精华。因此, 读者可以从整体上了解 J2EE 1.4 的功能, 对 J2EE 1.4 中的各种技术和元素之间的关系有一个清晰的了解。另外, J2EE 1.4 中对于 Web 服务的支持是其主要的新增特性, 本书使用了一个完整的部分对 J2EE 1.4 的 Web 服务支持进行详细的介绍。

全书中的主题是按照功能的顺序进行安排的, 读者可以按照 J2EE 1.4 的关键特性的构建和部署方式来学习这些关键特性, 本书独特的组织方式有助于读者快速掌握对于构建分布式应用和 Web 服务所必要的关键组件。

另外, 书中在恰当的位置给出了醒目的内容, 以引起读者注意, 比如“注意”、“Web 资源”、“提示”、“警告”、“交叉引用”。这些特定的内容为读者提供附加的指导信息, 使得读者能更充分地利用本书, 关注需要引起注意和应当谨慎的地方, 同时, 本书提供了大量的“Web 资源”, 使得本书得到扩延, 大大突破本书内容的局限。读者可以根据这些“Web 资源”更加深入和全面地了解 J2EE 1.4 平台。本书具有及时、实用和可信的特点, 是一本不可多得的 J2EE 参考手册。

原书行文流畅清晰, 技术含义表达明确易懂。我们在翻译的过程中, 也尽量保持原书的这一特色, 在译文中融入我们自己的开发经验, 运用了一些翻译技巧, 力求以最便于读者理解的方式表达出来。在翻译过程中, 本着对读者负责的态度, 我们将原书中的错误进行了修正。

本书由李化、潇东翻译, 参加审校的人员有陈大峰、栾景美、郭卫锋、周卓、梁列权等, 全书最后由肖国尊统稿。由于时间紧迫, 书中难免存在一些不足和错误之处, 敬请广大读者批评指正, 我们的电子邮件地址是 csh999@263.net。

译者

2003 年 5 月

前言

欢迎阅读本书。由于本书是专门为有经验的 Java 程序员编写的,因此,本书在向您介绍 Java 2 Enterprise Edition(J2EE)平台最基本的要素时,着重介绍了 J2EE 1.4 的新特性和功能。同时,本书假定您知道如何使用 Java 编程,但对 J2EE 平台和“分布式多层应用”的概念不熟悉(J2EE 使您能够开发、部署和管理分布式多层应用)。

本书组织结构

全书由 12 章组成,分三个部分:

- 第 I 部分: J2EE 1.4 与企业 Web 服务导论。这一部分提供了 J2EE 平台的一个高级概述,并解释什么是“Web 服务”,以及 J2EE 1.4 如何支持这种新的基于标准的分布式计算形式。这一部分中的章节向您讲述基本的 J2EE 概念、术语和大体结构,从而为本书后两部分介绍的主题提供了一个基本的框架。
- 第 II 部分: 使用 servlet、JSP 和 bean 实现从客户到服务器的通信。这一部分介绍 Java servlet 和 JavaServer Page(JSP)技术, JSP 驻留在 Web 层上,为支持 Web 的客户方程序提供访问服务器方 J2EE 应用组件的能力。随后,您将学习(驻留在服务器方业务层(business tier)上的)Enterprise JavaBean(EJB)如何封装企业的业务逻辑,以及如何推动与驻留在后端 EIS(企业信息系统)层上的数据库和现有企业系统之间的交互。
- 第 III 部分: 使用 XML 创建 Web 服务。这一部分分析由 J2EE 1.4 支持的基于 XML 的新 Web 服务功能。本书的这一部分对组成 J2EE 1.4 的核心 XML API 进行了概述,同时将重点放在每一个(支持 J2EE 应用基于 XML 标准如 SOAP、WSDL、UDDI 和 ebXML 来创建、发布和使用 Web 服务的)基本特性上。

除了书中看到的内容外,本书还有另外的两章,以电子版本形式通过本书的 Web 支持站点以在线方式提供。这两章提供了在书本中没有讨论的其他 J2EE 1.4 特性和功能的一个概述,同时展望了 J2EE 平台的未来前景。

Web 支持站点

作者为本书提供了一个 Web 支持站点,您可以在这个 Web 站点下载本书中每个带有编号的源代码,以及作为本书补充的另外两章的内容,同时可以访问机密的 Web3D Web 源代码和材料,以及免费的 J2EE 1.4 相关资源,包括文章、教程和其他代码示例。

在本书中出现的所有带有编号的清单全部包含于 WinZip 文件 J2EE1.4Essentials.zip 中,您可以通过访问 www.wiley.com/compbooks/walsh 来下载此文件,在您下载 J2EE1.4Essentials.zip 文件之后,您可以通过双击该文件来解压缩它包含的源代码(如果您还没有在自己的系统上安装

WinZip, 那么, 您可以从 www.winZip.com 下载 WinZip 的一个评估版本)。

在解压缩文件时, 使用 WinZip 的默认选项(确认选中了 Use Folder Names 选项)并对 J2EE1.4Essentials.zip 文件进行解压缩, 至少需要占用 1MB 的可用空间。解压缩进程会创建一个名为 J2EE1.4Essentials 的文件夹。只要在 Extract 对话框中选中 Use Folder Names 选项, 在 J2EE1.4Essentials 文件夹里面就会创建整个文件夹结构。您将看到按章号安排的文件夹, 在每一章对应的文件夹中, 可以看到该章中的源代码清单(注意, 在此.zip 压缩文件中只包含可以编译和运行的完整代码清单。例如, 不包含在框注中出现的代码片断)。

除了源代码外, 这个 Web 支持站点包含额外的两章, 您可以阅读这两章来学习 J2EE 1.4 引入的新的特性, 并了解 J2EE 平台未来的前景。在这个 Web 支持站点上, 您可以对本书中介绍的所有在线资源通过点击的方式来进行访问, 这个支持站点还包含许多其他内容。

可以参见附录“J2EE 1.4 精华资源”来了解有关本书的 Web 支持站点的更多信息。这个附录包含与支持站点有关的信息和大量用于延伸本书内容的重要资源。

本书中使用的标识

在阅读本书时, 您会遇到下面的标识:

Web 资源

“Web 资源”标识向您提供用于补充当前讨论的 Web 站点和页面的链接。

注意:

“注意”标识使您对重要问题或与当前讨论有关的概念引起注意, 当实现自己的 J2EE 应用时, 您应当注意这些问题或概念。

提示:

“提示”标识提供一些方法, 在编写 J2EE 代码或使用 J2EE 产品时, 您可以根据此提示来节省时间, 工作和/或金钱。

警告:

“警告”标识包含相应的建议, 这些建议有助于您避免错误或避开可能导致 J2EE 应用失败和/或出现严重问题的状况。

交叉引用

“交叉引用”标识用于指引您到本书别的章节, 从而发现与特定主题有关的更多信息。

展望未来

J2EE 1.4 是全新的。在写作本书的时候, 正式的 J2EE 1.4 规范还处于 Sun Microsystems、

它的业务伙伴和其他参与者的最后开发阶段。

为了阐述 J2EE 1.4 引入的大量重要概念和特性, 本书提供了许多代码, 这些代码包括来自 Sun 自己的标志性 J2EE 应用 Java Pet Store 的版本 1.3 的代码, 还包括 Sun 的 J2EE Tutorial 示例, 乃至来自作者本人参与的 Web3D Web 数字媒体网络项目的代码示例。

通过对本书后面的章节、其中包含的代码, 加上本书支持站点上额外的两章以及通过支持站点(参见附录)提供的资源的学习, 我相信您会拥有学习目前的 J2EE 1.4 所必需的所有信息。我希望您对本书和增补的材料感兴趣(就象我在准备它们时所表现出来的兴趣那样), J2EE 1.4 的引入标志着您开始进入一个美丽的新世界。

致 谢

要感谢 Wiley Publishing, Inc. 和 Sun Microsystems, Inc., 是它们使得本书的出版成为可能。特别地, 我对 Wiley 的责任编辑 Chris Webb 表达我个人的感谢, 因为他准许本书的出版并对本书从概念构思到完稿定形提供了帮助。我还要感谢 Wiley 高级项目编辑 Jodi Jensen, 他与我天天在一起工作, 确保我保持进度并以最佳状态写作。

我还要感谢文稿编辑 Kezia Endsley 的大量工作, 他消除了书稿中常见的问题, 同时要感谢图片设计师 Rashell Smith 和 Karl Brandt, 他们将我的草图转换成了专业的插图。特别感谢技术编辑 Nick Jacobs, 他提供了非常宝贵的见解与反馈信息。

最后, 我要感谢 Web3D Web 的成员对本书的细心检查并为本书提供代码和内容。特别感谢 Nick Nagel、Dean Hantzis、Milena Mejia、Sue Cavanaugh 和 Ian Lamont 提供的代码和内容。同时, 还要感谢 Web3D Web 的成员——Bill Wellington、Raja Krishnasamy、Gerardo Quintieri、Nicholas Quintieri 和 Carol Astone。

我要将我的谢意献给所有读者, 我希望您们喜欢我们一同创作的这本书。

关于作者

Aaron E. Walsh 是 Mantis Development Corporation 的主席, 该公司位于波士顿, 专门从事高级媒体和网络技术工作与 Web3D Web 的建设。

作为一名国际知名的技术工作者, Aaron 已经编写了十几本编程书籍, 包括 Java For Dummies、Java Bible, Java 2 Bible 和 Foundations of Java Programming for the World Wide Web(全部由 Wiley 公司出版)。作为 Internet 标准团体的一个积极分子, Aaron 管理一个相关的专题研究组, 并在行业会议上发表演讲, 同时在波士顿学院讲授 Java 编程和 Internet 软件开发。

目 录

第 I 部分 J2EE 1.4 与企业 Web 服务导论

第 1 章 J2EE 概述	1
1.1 在 J2EE 中以“企业”作为目标	1
1.1.1 定义“企业”	2
1.1.2 企业应用开发的考验与困难	4
1.2 J2EE 如何解决企业开发中的难点	9
1.2.1 简单而统一的基于组件的结构	9
1.2.2 内建的可伸缩性与安全性支持	13
1.2.3 提供稳定和可靠的基础	13
1.2.4 与企业信息系统和遗留系统的集成	14
1.2.5 Web 服务的内建支持	16
1.2.6 独立于厂家和平台的可移植应用	18
1.3 小结	20
第 2 章 J2EE 1.4 的结构、术语和特性	22
2.1 J2EE 的分布式多层应用模型	22
2.2 客户、组件、模块、容器和服务	25
2.2.1 客户	25
2.2.2 组件与模块	31
2.2.3 容器与服务	42
2.3 平台角色	45
2.4 J2EE 1.4 中的新特性	46
2.5 小结	47
第 3 章 平台服务与 Web 服务	49
3.1 平台服务	49
3.1.1 使用 HTTP 和 HTTPS 的网络服务	50
3.1.2 使用 JTA 和 JTS 的事务服务	50
3.1.3 使用 RMI-IIOP 和 Java IDL 的 CORBA 服务	52
3.1.4 使用 JDBC API 的数据库服务	53
3.1.5 使用 JMS 的消息传递服务	53
3.1.6 使用 JNDI 的命名服务和目录服务	54
3.1.7 使用 JavaMail 和 JAF 的邮件服务	55
3.1.8 使用 JAAS 和 JACC 的安全服务	56
3.1.9 使用 JAXP 的 XML 解析服务	57

3.1.10 使用 Java API for XML 的 Web 服务	58
3.1.11 使用 J2EE Connector 结构的 EIS 连接服务	64
3.1.12 部署服务与管理服务	66
3.2 平台应用程序编程接口(API)	67
3.2.1 J2SE 要求的 API	67
3.2.2 J2SE 可选用的 API	68
3.3 小结	69
第 4 章 实例分析与示例场景	70
4.1 Java Pet Store 和 Java Smart Ticket	70
4.1.1 Java Pet Store 示例应用	71
4.1.2 Java Smart Ticket 示例应用	74
4.2 Web3D Web	76
4.3 网格计算、在线购物和搜索服务	78
4.4 小结	79

第 II 部分 使用 servlet、JSP 和 bean

实现从客户到服务器的通信

第 5 章 使用 Java servlet 扩展与增强 Web 服务器	80
5.1 Java servlet 技术集成到 J2EE 中的方式	81
5.1.1 servlet 支持 J2EE Web 层	81
5.1.2 servlet 同流行的 Web 服务器扩展技术之间的对比	83
5.1.3 Web 组件、Web 应用和 Web 容器	85
5.1.4 servlet 生命周期	89
5.2 剖析 servlet 的主要特性	92
5.2.1 使用请求对象和响应对象来访问和应答客户请求	94
5.2.2 使用过滤器来处理 and 转换请求和响应	99
5.2.3 使用 cookie 来跟踪和管理用户会话	100
5.3 小结	101
第 6 章 使用 JSP 创建动态 Web 页面	102
6.1 分析 JSP 技术是如何集成到 J2EE 中的	102
6.1.1 JSP 提供 MVC 结构中的视图	107
6.1.2 JSP 生命周期	109
6.2 剖析 JSP 的主要特性	111
6.2.1 使用 JSP 元素来创建动态内容	115
6.2.2 Web3D Web Media Submission 应用示例	119

6.3	使用指令同容器进行通信	124
6.3.1	使用 page 指令配置页面	125
6.3.2	通过 taglib 指令来利用定制标记	126
6.3.3	使用 include 指令来创建复合页面	128
6.4	使用动作和定义脚本变量	128
6.5	使用脚本激活页面	131
6.6	JSP 和 XML	134
6.7	小结	135
第 7 章	使用 EJB 构建服务器方业务组件	136
7.1	分析 EJB 技术是如何集成到 J2EE 中的	136
7.1.1	可重用分布式业务组件	136
7.1.2	可移植的和可互操作的业务组件	138
7.1.3	业务层的 Web 服务	140
7.2	bean 类型: 会话 bean、实体 bean 和消息驱动 bean	140
7.2.1	会话 bean	141
7.2.2	实体 bean	147
7.2.3	消息驱动 bean	148
7.3	通过接口定义入口	150
7.3.1	使用 home 接口来访问 bean 的组件接口	151
7.3.2	定义本地客户视图、远程客户视图和 Web 服务客户视图	152
7.4	为 Points Calculator bean 示例创建客户	161
7.4.1	使用 JNDI 访问 home 接口	162
7.4.2	使用 JSP 对 Web 客户进行编码	164
7.4.3	对 Java 应用客户进行编码	168
7.5	打包和部署 EJB	170
7.6	小结	173
第 8 章	深入分析事务并略谈数据库	174
8.1	深入分析事务	174
8.1.1	原子的、一致的、隔离的和持久的(ACID)事务	175
8.1.2	容器管理的事务	178
8.1.3	bean 管理的事务	185
8.2	使用实体 bean 来访问数据库	186
8.2.1	使用主键来惟一确定实体	189
8.2.2	通过 home 接口来使用 finder 方法定位实体	192
8.2.3	容器管理的持久性	195
8.2.4	bean 管理的持久性	198
8.3	小结	199

第III部分 使用 XML 创建 Web 服务

第 9 章 使用 JAXP 处理 XML	200
9.1 分析 XML 和 JAXP 是怎样集成到 J2EE 结构的	200
9.2 使用 JAXP 处理、转换和样式化 XML 文档	203
9.2.1 使用 SAX 和 DOM 解析 XML 文档	204
9.2.2 使用 XSLT 转换和样式化 XML 文档	233
9.3 小结	239
第 10 章 用 JAX-RPC 创建 Web 服务端点和客户	241
10.1 分析 JAX-RPC 是如何集成到 J2EE 的	241
10.1.1 最小平台要求	242
10.1.2 JAX-RPC API	243
10.1.3 理解术语“Web 服务”	244
10.2 实现 Web 服务端点和客户	246
10.2.1 JAX-RPC 支持的数据类型	246
10.2.2 用 JAX-RPC 实现 Web 服务端点	247
10.2.3 产生 stub、tie 和 WSDL 描述	253
10.2.4 打包和部署服务	257
10.2.5 用 JAX-RPC 实现 Web 服务客户	260
10.3 小结	267
第 11 章 用 SAAJ 发送和接收 XML 消息	268
11.1 分析 SAAJ 是如何集成到 J2EE 的	269
11.2 SAAJ 概述	270
11.2.1 SOAP 消息结构	270
11.2.2 通过 SOAP 连接发送消息	275
11.3 使用 SAAJ 创建、发送和处理 SOAP 消息	276
11.3.1 创建和发送 SOAP 消息	278
11.3.2 处理 SOAP 消息	287
11.4 小结	289
第 12 章 用 JAXR 访问 XML 注册库	290
12.1 分析 JAXR 是如何集成到 J2EE 的	290
12.1.1 注册库提供者、JAXR 提供者和 JAXR 客户	291
12.1.2 JAXR 1.0 API 包	292
12.2 创建、打包和部署 JAXR 客户	294
12.2.1 实现 JAXR 客户	295
12.2.2 打包和部署 JAXR 客户	307
12.3 小结	309
附录 J2EE 1.4 精华资源	310

第 I 部分 J2EE 1.4 与企业 Web 服务导论

第 1 章 J2EE 概述

本章内容包括：

- Java 2 Platform, Enterprise Edition(J2EE)导论
- 定义 J2EE 的目标——“企业”
- 了解 Web 服务的功能
- 将应用功能跨层进行逻辑分布与物理分布
- 分析企业应用开发的要求
- 了解 J2EE 如何解决企业开发中的难点

毫无疑问，自从 J2EE 这项革新技术最先由 Sun Microsystems 公司于 1999 年推出以来，J2EE 1.4 版本是最重要的 J2EE 版本。在过去的这些年中，J2EE 发展十分迅速，它是当今用来开发企业应用和提供计算解决方案的事实标准。

J2EE 1.4 在 Java 2 Platform Standard Edition(J2SE)之上添加了大量面向企业的特性和功能，而事实上，J2EE 本身在大约 5 年前就已经对传统的软件开发进行了革命。J2EE 1.4 引以为荣的是拥有大量重要的和令人兴奋的新技术，而这些技术是在 J2EE 的以前版本中所没有提供的，它们包括对可扩展标记语言(Extensible Markup Language, XML)和 Web 服务的广泛支持。

1.1 在 J2EE 中以“企业”作为目标

J2EE 是分布式多层应用的开发与部署平台，它直接面向企业。如图 1-1 所示，J2EE 在 J2SE 平台之上增加了大量企业级特性。换句话说，J2EE 将企业特性加入到 J2SE 平台中，而 J2SE 平台的重点则放在传统的桌面应用开发与部署上。

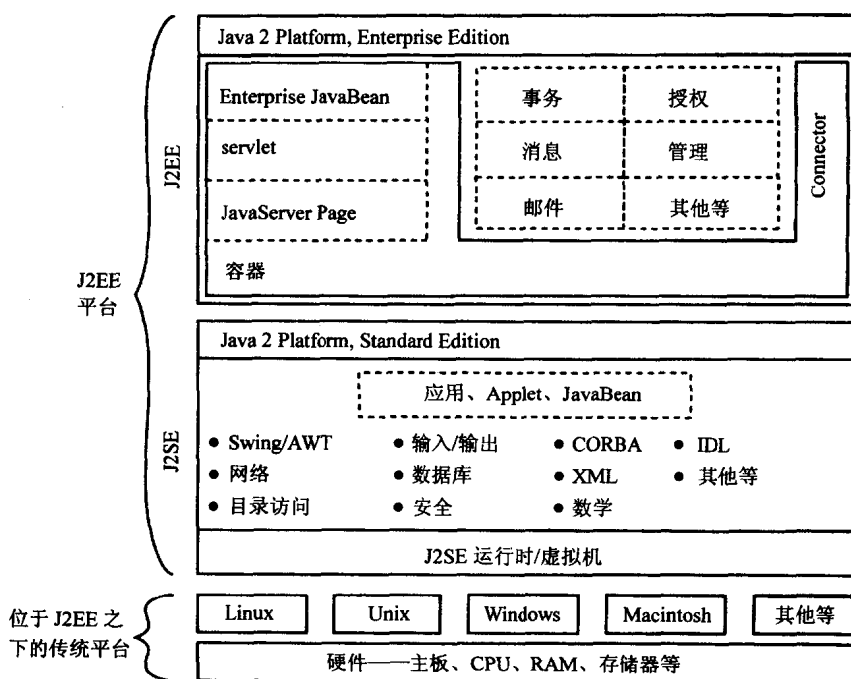


图 1-1 J2EE 在 J2SE 之上添加了多种企业级特性

1.1.1 定义“企业”

传统上, 术语“企业(enterprise)”描述各种形式和规模的业务环境, 范围涉及小商店到大型公司。然而, 从一般的含义上讲, 这个术语所描述的是依照一组公共规则来进行运作的任何组织(在传统企业环境中, 这组规则就是所谓的“业务规则”)。大学、非赢利组织、宗教组织以及其他实体只是 J2EE 可以构建的许多形式的非传统“企业”中的少数几个实体。

实质上, J2EE 应用只是特定于企业的软件程序, 它基于控制组织的一般规则中的某一部分来提供某种形式的功能。例如, 公司企业可能使用 J2EE 创建自动化的购货订单处理应用, 而大学则更有可能使用 J2EE 来构建校园内的评级与成绩单系统。同时, 非赢利组织和宗教组织可能使用 J2EE 来创建方便进行捐赠和跟踪捐赠活动的应用。另一方面, 医疗组织更有可能构建病人档案跟踪系统。在所有这些情况下, 特定于某个组织的运作规则都会在其企业应用中反映出来。

J2EE 平台可管理软件基础结构和支持 Web 服务, 这使得开发人员可以创建安全的、分布式的以及可互操作的企业应用。因为由 J2EE 构建的这种应用具有分布式的、多层的和基于组件的结构, 所以, J2EE 促进了可伸缩应用的开发, 这种应用的各种工作部件在物理上分散于网络中。正如您将看到的那样, 这种应用开发方法不仅使得应用本身具有高度的可伸缩性, 同时, 由于这种方法采用一种自然的基于团队的编程方式, 开发流程同样具有可伸缩性。

实际上, 术语“企业应用(enterprise application)”与“分布式应用(distributed application)”的含义是相同的, 这是因为, 几乎每一个为企业构建的实际应用都是以某种方式分布在网络上的。换句话说, “企业应用”只是另一种用于描述分布式多层应用的方法, 这种应用分散在组

织机构的网络基础结构的不同级别(也就是不同层)上(参见图 1-2)。当然,企业应用与其他形式的分布式应用之间的区别在于:企业应用通常横跨“绑定”它们服务的各个部门和工作单位。

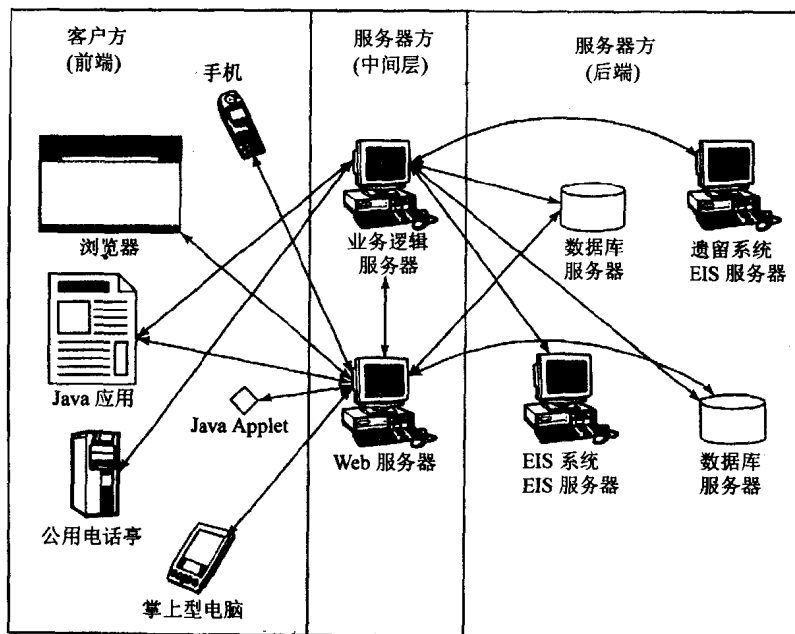


图 1-2 通常，企业应用是分布式多层应用，构成这种应用的各个程序在通过网络连接的不同计算机上执行

多层结构

多层结构是 J2EE 的核心，在多层结构中，组成应用的各种组件在逻辑上是分离的或者是分布式的，跨越网络计算环境的各个级别或各层(这种结构也称为 n 层结构)。换句话说，其中的“层”代表功能的逻辑层或逻辑级别：多层应用在逻辑上对应用组件进行分组，分组的依据是这些应用组件在整个应用结构中执行的功能。

值得注意的是，多层应用的各个层表示功能的逻辑分离，而不是表示物理分离。例如，组成 J2EE 应用的组件可能在逻辑上分布在客户层、Web 层、业务层和 EIS 层上，而这些层在物理上可以分布在任何数量的计算机系统中，如图 1-2 和图 1-3 所示。

企业应用比较复杂，在构建与维护企业应用时需要谨慎，因为企业应用如同生活在异类环境中的麻烦的野兽。典型情况下，企业应用是由分布在各种不同的硬件上并在这些硬件上执行的大量软件程序或组件组成的。这些硬件可以是高端大型机，也可以是小型机和工作站，或低端桌面 PC。

组成这些应用的软件组件通常是用几种不同的编程语言编写的，并且这些软件通常使用许多协议在彼此之间进行通信。组成企业应用的组件可能分布在局域网(LAN)上，也可能分布在广域网(WAN)上，甚至有可能分布在 Internet 上。例如，使用 J2EE 创建的企业应用在物理上可能分布在许多计算机系统和网络拓扑中，由此，可以通过 J2EE 来产生具有高度的可伸缩性和高度可用的应用。

注意:

尽管企业应用很复杂,与传统独立的桌面应用比较时尤其如此,但是,企业应用实际上只是通过网络来与另一个软件程序集合进行交互的软件程序集合。

分布式多层应用并不是什么新事物。在过去的几十年当中,我们已经能构建在联网的不同计算机上运行的应用(该应用的各个可执行部分分散在这些计算机上执行)。然而,真正新的方面在于存在这样一种趋势:使用基于开放的和可互操作的 Internet 标准和 Web 标准的综合性分布式开发平台(即框架)来构建这些应用,并且这种趋势越来越强烈。

例如,J2EE 平台是一种完整的端到端解决方案,开发人员可以使用这种解决方案来构建软件组件,并在企业应用中的每一层中部署这些软件组件。这样,J2EE 可以用于开发广泛的各种不同的前端客户、(支持组织的 Web 连接和业务逻辑的)中间件、后端数据库以及企业信息系统(Enterprise Information System, EIS)集成解决方案。同时,J2EE 支持开放的 Internet 标准和 Web 标准,这些标准包括:超文本标记语言(Hypertext Markup Language, HTML)、可扩展标记语言(Extensible Markup Language, XML)、超文本传输协议(Hypertext Transfer Protocol, HTTP)以及简单对象访问协议(Simple Object Access Protocol, SOAP)。

注意:

J2EE 平台是用于开发各种形式的分布式多层应用的综合性端到端(客户方与服务器方)解决方案。在 J2EE 中,术语“组织(organization)”容易作为“企业”的代名词。然而,通过比较,“企业”这个术语的适用范围更加广泛。

1.1.2 企业应用开发的考验与困难

开发企业应用需要相当高的技能。事实上,在进行企业应用的开发时,需要进入一种复杂的和具有挑战性的最高境界。在出现最糟糕的情况时,企业应用开发项目的难点可以卡住单个的程序员和整个团队。总而言之,如果软件开发项目的主体部分超过了预算、过了截止日期或者缺少应当提供的特性(或者这些特性根本就未能在开发过程中出现),那么,毫无疑问,企业项目的复杂性和整个开销就会增加,从而导致这个企业项目成为风险业务。

企业应用开发人员面对许多的困难,J2EE 被专门设计用于解决这些困难。在下一节中,将向您一一介绍摆在企业应用开发人员面前的主要困难,随后,将向您仔细解释 J2EE 是如何解决这些问题的。

1. 争取四个目标:稳定性、可伸缩性、简单性和安全性

典型情况下,企业应用是关键任务型应用。在没有恰当理由的情况下,组织是不会开发企业应用的。在大多数场合下,开发企业应用的理由是明确的——推进组织的目标。

简而言之,构建企业应用的目的是支持或推进业务目标或组织的目标。企业应用可能帮助企业更加有效地进行竞争,使操作成为流线型的操作,省钱或挣钱,还有可能通过许多其他方法来支持或推进组织的目标。然而,如果要有有效地做到这一点,需要涉及到构建满足前面提到的四个目标(即稳定性、可伸缩性、安全性和简单性)的应用。下面对每个目标作相应的描述。

注意:

正如您将在后续章节中看到的那样, J2EE 应用指的是任何可部署的 J2EE 功能单元, 在 J2EE 应用中, 除了包括完整的分布式应用(它由任何数量的模块组成)外, 还可以包括单个的模块和模块组。模块指的是软件单元, 由一个或多个 J2EE 组件组成。组件是指代表独立的程序功能单元的 Java 类。尽管 J2EE 组件只不过是 Java 类, 但是, 与传统的类相比, J2EE 组件确实存在三大不同之处。具体来讲, J2EE 组件: 1) 必须组装到应用中; 2) 在部署之前必须进行验证, 以便确保它们具有良好的组成并符合 J2EE 规范要求; 3) 必须在 J2EE “容器” 里面执行, 并由 J2EE “容器” 管理。这些容器为每一个组件提供了访问特定的和标准化的服务的能力。

(1) 稳定性

稳定性是一种稳定性质, 表示应用是按照预期的方式执行的, 不存在崩溃、冻结现象, 也不会显示通常与多 bug 的、不完善的或不稳定的产品有关的那些特征。对于要么直接影响要么间接影响人类生活的关键任务应用(如医疗应用和航空应用)而言, 稳定性显得尤为重要。不稳定的应用可能对组织产生负面影响, 因为不稳定的应用可能消耗额外的资源, 并且由于额外的技术支持, 或者为了消除存在问题的系统所带来的不好的公众影响, 而需要花费额外的钱。特别是, 不可靠的应用容易失去潜在的客户, 或者为现有的客户提供一个充分的理由来考虑另外的可选解决方案。尽管稳定性是很重要的, 但这个目标并不容易达到。某些专家认为, 即使是有经验的程序员在他们编写的每 10 行代码中也至少会出现一个 bug 或一个逻辑错误。对于分布式应用而言, 这个比率甚至还要增大, 因为构建这些应用的复杂性会增加。尽管这样的统计无法得到验证, 但是众所周知的一个事实是: bug 和逻辑错误属于软件开发的一部分。消除已知的 bug 和其他编程错误可以直接提高整个应用的稳定性, 但这不是一项容易实现的任务——特别是未知问题通常潜伏在代码中, 而只有在应用部署之后和在实际运行的考验下才展现出来。

(2) 可伸缩性

可伸缩性是指应用为了与加在它上面的需求保持同步而进行增长或伸缩的能力。可伸缩性与可用性(availability)关系密切, 可用性描述应用的可用频率与可用程度。在应用不能伸缩的情况下, 那么, 它可能在大量使用的场合中都变得不可用。例如, 当有太多的用户同时与该应用进行交互时, 它就有可能变得无法使用, 因为这样会使得该应用的各种组件的负荷过大。如果该应用能够动态伸缩, 适应这样的负荷, 那么, 即使存在大量需求, 该应用也容易变得可用。与稳定性一样, 可伸缩性不会如魔法般地出现, 它必须有意识地在代码级以及平台级将其结合到应用中(如果应用的运行时执行环境不可伸缩的话, 应用本身也许不能完全伸缩来满足使用需求)。特别是, 企业应用存在许多瓶颈(choke point), 在这些位置可能存在点可伸缩性瓶颈(point scalability bottleneck)。开发人员不仅必须重点考虑应用的每个组件的可伸缩性, 还必须考虑自己的可伸缩性问题, 开发人员自身的可伸缩性问题涉及到网络及分布式应用结构中的各个层。

(3) 安全性

安全性是应用受到保护的程度或者容易被人擅自使用的程度。尽管在过去 10 年中安全的重要性有所增加, 但是, 在 2001 年 9 月 11 日, 安全问题立刻成为整个信息技术(Information Technology, IT)业的焦点。在那一天, 美国的国家安全提升至最高的可能级别, 导致技术业和世界范围的直接连锁反应。以往(与修补或修复所需的时间和成本相比)认为是可以接受的应用安全弱点, 现在对于许多组织和个人而言变成了极为重大的问题。于是, 为充分解决安全问题

而需要花费的时间和工作越来越多地被事先安插到了软件设计最初阶段，安全问题归属到了软件设计的最初阶段。现在，许多企业从一开始就将安全问题放在第一位，而不是在系统实际运行过程中发现弱点时才发布补丁和进行修复。然而，这样做，企业需要增加自己的开发预算和延长交付截止日期。在系统长时间地运转之后，这些预先的投资会收获相当可观的回报，因为安全性的提高可以增加客户的信心和满意度，同时减少由容易受到攻击的软件所带来的额外的技术支持与专门的开发工作。当软件是可靠的和安全的时候，就不需要为了提供技术支持而安排另外的电话接线员，不需要拨打电话来平息客户的抱怨，也不需要开发补丁和修复程序。

(4) 简单性

简单性指的是：从应用的最终用户的角度和构建与维护该应用的开发人员的角度，来看应用的简易程度。对于最终用户而言，应用的简单性主要反映在他们所交互的用户界面中，其次是反映在他们可以使用的界面的多样性中。如果应用简单易用，并且具有从多个不同的最终用户配置(例如，通过标准的 Web 浏览器、客户前端应用、可移动的无线计算设备等)进行访问的额外好处，那么用户将更有可能接受该应用。然而，从软件开发人员的角度来看，简单性指的是在维护整个应用的稳定性、可伸缩性和安全性的同时快速和有效地构建或增强应用的能力。对于程序员而言，简单性最终转化成程序员的生产率，程序员可以用较少的时间和工作来做更多的事情。对于企业应用而言，简单性尤其重要。由于企业应用固有的复杂性，引入到开发过程中的任何简单性都可以获得巨大的收获，这些收获与程序员的生产率和他们对于他们所做的工作的感受有关。程序员构建和部署复杂应用越快和越容易，他们就越有可能觉得工作愉快，对于雇主而言所获得的回报形式就是雇员跳槽率变低，生产率增加。

注意：

4S 表示 Stability(稳定性)、Scalability(可伸缩性)、Security(安全性)和 Simplicity(简单性)。从这 4 个需求出发，我建立了 Web3D Web(<http://web3Dweb.com/>)——一个新兴的数字娱乐平台和网络，4S 描述应当推动每一个分布式软件应用的 4 个最重要的需求。对于不稳定的、不可伸缩的、不安全的和不容易使用与升级的企业应用而言，几乎可以肯定的是，这些应用的寿命较短。

2. 提高程序员生产率

现在，企业越来越需要做更多的事情。企业必须从它们现有的(或收缩的)软件开发人员中榨取越来越多的剩余价值，同时满足客户要求，并顶住激烈的竞争压力。

由于预算收缩、失业以及激烈的竞争等原因，程序员在工作时要比以前更加努力，才能及时地将其产品投放到市场，且不超过预算成本。

程序员生产率的需求空前高涨。那些不满足要求的程序员容易被有经验的老手所代替，而在几年前并不是这样。不久以前，跳槽是常有的事，因为程序员在令人兴奋的新公司中寻找自己感兴趣的和有利的机会不会太难。现在，这种情形发生了变化。由于有如此多有经验的软件开发人员寻找工作，对于有能力雇佣新员工的公司而言，发现和留住合格的职员比较容易。对于没有能力雇佣新雇员的公司而言，他们必须使得现有的(也许是收缩的)程序员的潜能发挥至极致，这样就使得多产的程序员具有更高的价值。这样看来，是什么东西能够使得程序员更具有生产率呢？

自然, 最重要的因素是经验和才能。如果程序员在他们期望工作的领域没有经验, 并且不能快速学习所需要的新技术, 那么, 生产率就无从谈起。

假定程序员胜任他的工作, 在获取高生产率的时候, 会有大量其他因素起作用。其中, 最重要的因素之一是: 企业(和企业的程序员)是否遵守统一编程模型(Unified Programming Model)。在这个模型中, 软件组件以一种统一的和一致的方式进行开发、测试、部署和维护, 理想情况下, 可以使用一个标准化的工具包来帮助自动化和对这个过程流线性化。

当您考虑分布式软件应用可以使用非常离散的多种技术的组合来创建时, 统一的编程模型尤其重要。前端可以容易地由 Web 页面组成, 这些 Web 页面是使用 HTML、XHTML、XML、多种脚本语言(JavaScript、Jscript、VBScript 等)以及嵌入式插件(如 Flash 或 QuickTime 等)的各种不同的组合进行构建的。

通过使用公共网关接口(Common Gateway Interface, CGI)程序和脚本, 可以在服务器端支持这些客户端技术, 而应用的中间层可能捕获(用 C、C++、C#、Java 和其他语言编写的)程序形式的业务逻辑。在加上后端数据库技术和相关的查询语言(如 SQL、XQuery 等)之后, 您会使自己陷入大量的技术杂烩当中。

如果缺少单一的、统一的编程模型, 程序员容易由于复杂性而受到打击。而统一的编程模型可以使如何构建分布式应用变得清楚易懂, 同时还可以定义用于参与的开发成员进行交流的标准和常用的词表(这个常用的词表有助于在程序员彼此交谈的时候消除模糊之处, 并有助于帮助整个团队彼此进行交流)。由此, 这样的模型有助于使设计和开发应用更加容易, 同时还顾及到了工作的明确分工: 每个人都知道如何各行其事, 从而促进生产率的提高。

除了遵循单一的、统一的编程模型(如在 J2EE 中引入的编程模型)外, 建立和遵循标准要比彻底重做或采用专有解决方案生产率更高。当开发人员通过一种标准方式来访问他们正在开发的应用所支持的各种服务时, 他们都遵循同一个考虑周全的解决方案, 而不是采用他们各自不同的解决方案。由此, 这样做可以产生每个人都可以使用的标准代码存储库, 从而实现软件重用。此外, 使用可重用代码来组装应用也可以提高程序员的生产率。

当采用众所周知的开放标准而不是采用专有解决方案或不太知名的标准时, 通常可以提高生产率。因为标准清晰地定义了对于执行特定任务或操作所必要的正确规则或流程, 标准可以推动产品的互操作性(例如, 标准支持业界范围内的内容交换格式), 同时使得程序员的工作更加容易和更加清晰。大多数众所周知的标准(代码)都具有开放性、可免费获得、或者低成本实现等特点, 程序员可以利用这些标准, 而不是被迫编写他们自己的代码。

此外, 开放标准有助于防止出现厂家锁定(vendor lock-in)的情况。当应用依赖于一个厂家或有限数量的厂家专有地提供或拥有的实现、产品或服务时, 就会出现厂家锁定的情况。相比之下, 开放标准(open standard)通常受到大多数厂家的支持, 因此, 与专有的解决方案相比, 几乎不会出现厂家锁定的情况。开放标准鼓励厂家之间的制衡, 因为这些标准可以由任何人实现, 典型情况下, 当涉及到实现和平台的时候, 可选择的范围较广。

通常, 基于开放标准的应用还具有高度的可移植性, 并且, 在典型情况下, 这样的应用在广泛的计算机平台上执行, 而只需极少的移植工作, 或者不需要移植工作。J2EE 就属于这种情况, J2EE 定义一个分布式的应用开发平台——基于开放标准(如 Java、XML、HTTP、SOAP 等)的平台。