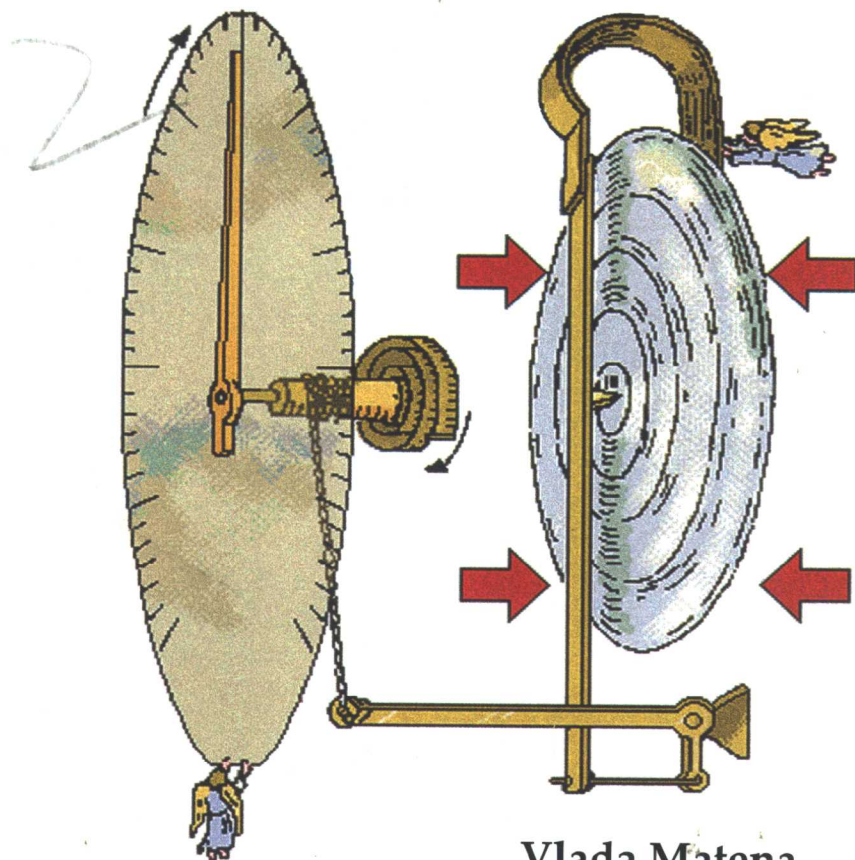


Applying Enterprise JavaBeans
Component-Based Development
for the J2EE Platform

Sun 公司核心技术丛书



J2EE 平台上的 EJB 组件开发



(美) Vlada Matena 著
Beth Stearns

瞿裕忠 陆海涛 彭晓晖 尚云飞 等译



机械工业出版社
China Machine Press



Addison-Wesley

Sun 公司核心技术丛书

J2EE 平台上的 EJB 组件开发

(美) Vlada Matena 著
Beth Stearns

瞿裕忠 陆海涛 彭晓晖 尚云飞 等译



机械工业出版社
China Machine Press

J2EE 作为一个标准的体系结构,为使用 Java 进行基于 Web 的企业应用的开发与部署提供了无与伦比的支持。本书重点讨论 J2EE 平台的主干 EJB 的体系结构,描绘如何使用 EJB 组件体系结构开发与部署企业应用,并以示例阐述了在企业应用开发中如何应用 EJB 体系结构及其特性。

本书适合计算机专业研究生、企业应用开发部署人员及对 EJB 技术感兴趣的其他人员阅读。

Vlada Matena, Beth Stearns: Applying Enterprise JavaBeans: Component-Based Development for the J2EE Platform.

Original edition copyright © 2001 by Sun Microsystems, Inc.

Chinese edition published by arrangement with Addison Wesley Longman, Inc. All rights reserved.

本书中文简体字版由美国 Addison Wesley 公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2001-0841

图书在版编目 (CIP) 数据

J2EE 平台上的 EJB 组件开发 / (美) 麦特纳 (Matena, V.), (美) 斯特恩 (Stearns, B.) 著; 瞿裕忠等译. 北京: 机械工业出版社, 2001.8

(Sun 公司核心技术丛书)

书名原文: Applying Enterprise JavaBeans: Component-Based Development for the J2EE Platform

ISBN 7-111-09090-X

I. J... II. ①麦...②斯...③瞿... III. JAVA 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 045638 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 刘立卿

北京市密云县印刷厂印刷·新华书店北京发行所发行

2001 年 8 月第 1 版第 1 次印刷

787mm×1092mm/16·20.5 印张

印数 0 001-5 000 册

定价: 38.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

随着 Web 和 Internet 的发展,越来越多的企业开始开发与部署基于 Web 的企业内部网和外部网应用。J2EE 为基于 Web 的企业应用提供了无与伦比的支持。J2EE 是一个标准的体系结构,它特别面向使用 Java 程序设计语言进行基于 Web 的企业应用的开发与部署。J2EE 平台的主干是 EJB 体系结构,它规范了如何用 Java 程序设计语言开发与部署服务器端应用业务逻辑组件。J2EE 平台的其他部分,例如 JSP 和 servlet,是对 EJB 体系结构的补充,提供呈现逻辑以及客户交互控制逻辑等功能。一个着眼于创建与销售可重用的基于 EJB 的业务组件的技术已经开始广泛应用。

本书提供了 EJB 体系结构的深层次技术,描绘了如何使用 EJB 组件体系结构开发与部署企业应用。全书共 9 章,第 1 章描述了 EJB 和 J2EE 体系结构的当前状态,以及它们如何满足当今企业计算的需求。第 2 章给出了 EJB 体系结构的简明概要。第 3 章描绘了企业应用开发过程中涉及的角色和任务。第 4 章和第 5 章重点介绍会话 bean,分别阐述了会话 bean 的应用开发以及 EJB 容器为会话 bean 提供的支持和服务。其中的一个福利登记应用示例阐明了会话 bean 的关键点。第 6 章和第 7 章着眼于实体 bean,它们给出了实体 bean 编程的基本概念,并讨论了容器为实体 bean 提供的服务,并展示了一个实体 bean 应用示例。第 8 章和第 9 章分别描述了 EJB 体系结构对事务处理的支持以及 EJB 安全环境。本书适合于计算机专业的研究生、企业应用开发者、信息技术经理以及其他对 EJB 技术感兴趣的读者。我们衷心期望本书能够有益于我国 EJB 企业应用开发。

本书第 1、2、3 章由瞿裕忠翻译,第 4 章由尚云飞翻译,第 6、7 章分别由陆海涛和彭晓晖翻译,第 5、8、9 章由东南大学 XObjects Group (<http://xobjects.seu.edu.cn>) 的几位成员共同翻译。全书由瞿裕忠审稿。有关 EJB 技术及其在电子商务中应用的技术资料和应用示例,请访问东南大学电子商务研究所的 Web 站点 (<http://ebi.seu.edu.cn>)。在翻译过程中,译者参阅了大量的国内外资料。本书的翻译工作得到了机械工业出版社的支持,也得到了东南大学 XObjects Group 成员的大力支持,在此,我们深表谢意!

限于译者的水平及时间,本书尚有许多不足之处,恳请读者指正。

2001 年 3 月于东南大学

序

自 19 世纪 80 年代以来，还从没有像 20 世纪 90 年代后期那样，能有如此快速而持续的创新和知识进步。新型商务和通信正在不断创建，新兴公司主导着工业界，冲破了地理边界，迫使政府和其他法律机构行动起来面对这个变化。这一变化的核心是 Web（World Wide Web，万维网）及相关技术，如 Java，这些技术使得公司能够利用这一范型的转变。

几年前，Web 主要还是科学家们用来共享文档的小型网络。今天，它已经发展成为一个大型的网路，将上百万的用户连接到全世界的成百万的计算机上。现在，它不仅仅用来共享文档，它也许是世界上最大的最新信息的存储库。它被用来运行业务作业并处理金融交易，它为金融和商务交易创造了一个急速增长的新市场。Internet 提供了一个将大量不同计算机连接起来的途径。

世界新经济导致公司被快速地收购或合并，它们中的大部分有着全球范围的业务，其员工队伍越来越要求适合流动工作，主要因为手持设备越来越可用，提供了安全地存取公司信息资源的能力，从而能够按需完成工作。不仅仅员工，而且客户和供应商也必须具备在需要时能够连接到公司网络以得到所需的服 务的能力。赢家和失败者的区别就在于公司信息技术部门是否能够提供一个可靠的解决方案，它能够无缝地跨越建立在不同平台上的企业信息系统以及异构的数据中心。公司信息技术人员面临的最大问题是适合于新时代商务的新型应用体系结构到底是什么？

自从 1995 年问世以来，Java 程序设计语言以及 Java 平台技术在信息技术发展中扮演着越来越重要的角色。Java 程序设计语言开始作为写作可移植的客户应用（applet）的一种方式，这些客户应用运行于 Web 浏览器以提供动态内容。然后，它用来写作独立的客户应用，采用 JDBC 作为连接公司数据库服务器的方式。servlet 技术使服务端应用的创建成为可能。applet 和 servlet 技术的组合可支持这种客户/服务器应用的开发，它以一个比 CGI 技术更加可伸缩的可靠的方式支持动态内容的服务和安全事务处理。

运行 applet 的 Web 浏览器，这个体系结构的客户端是低成本的和可广泛使用的。它们有一个内置的安全模型以防止病毒的侵入。applet 在使用之后被丢弃，并且通过一个可信发起者进行数字签名之后是可信的。这些客户/服务器应用的部署是简单方便的。applet 和 servlet 能够驻留在支持 Java 程序设计语言的 Web 服务器上，并通过一个统一资源定位器（URL）来简单地定位。大量的新的 Java 中间件厂商不断涌现，形成了 Java 程序设计语言应用服务器市场。

即使对于这一新的客户/服务器应用范型，公司信息技术以及应用平台提供商也面临巨大挑战：HTTP（Hypertext Transfer Protocol，超文本传输协议）是一个传统的协议，并不适合使用在高性能客户/服务器应用中。公司面临对快速自动化新业务过程的需求，通常需要利用和扩展那些在 Internet 时代之前构建的企业应用系统中的有价值的信息资源。这些老系统的一个显著特性是它们已经被精心调整过，并且有着多年的健壮性，可靠地提供了安全的、高性能

的、可伸缩的以及事务处理的能力。另外，公司长期以来已经精心构建了他们与现有提供商的可信关系，相互之间有着既得利益。替换这些老系统是不可取的，利用已有的企业信息基础设施是很重要的。

程序员生产效率来自：面向对象技术，新型客户/服务器应用的快速面市和高效部署，基于通用 UNIX 接口的跨越不同提供商系统的无缝集成，但这些还远远没有实现。每一个 Java 程序设计语言应用服务支持它自己专用的组件模型，以它自己专用的方式与企业信息系统集成。诚然，Java 程序设计语言已经证实了它极大地增强了程序员生产效率。工业界对 JavaBeans 技术的广泛支持为 Java 程序设计语言建立了一个合适的组件模型，虽然这一技术的焦点在于图形用户界面的组件。有关开发业务逻辑组件的标准的需求逐渐明确。

Java 平台的一个基本原则是“Write Once, Run Anywhere”。这完全是可能的，因为提供商已经集成了 JVM (Virtual Machine, Java 虚拟机)，它在操作系统之上为应用提供了一个通用的运行环境。但是，分布式基础设施服务，比如数据库、事务处理会话管理和消息服务，通常与内在的操作系统分离。一个典型的公司也许使用来自多个提供商的许多不同的产品和配置以实现他们的基础设施服务。集成这些异构的系统通常是公司信息技术部门的职责。这种代价是很昂贵的，因为这需要专门化的稀有的编程技术，而且很难及时完成。为员工、客户、供应商和业务伙伴提供一个无缝的服务驱动的应用网络必须由提供商来完成，而不是公司信息技术人员。公司应该将精力只集中在创建业务逻辑来支撑不断产生的业务过程。

Internet 已经为伸缩概念增加了一个新的维度。Internet 应用很可能必须支持千百个（也许是几百万个）并发用户，他们通过具备各种能力的客户设备连接到这个服务驱动的网络，并不是所有的设备都具备支持浏览器的完全特性的能力。不管是通过一个蜂窝电话、一个掌上设备、一个膝上型电脑连接，还是长久地通过一个桌上型电脑连接，用户需要通过合适的用户界面来访问企业信息服务。公司应该能够开发适合于各种设备的强制性的用户交互接口，而且应该快速地调整应用以支持最新部署的设备。

在 1997 年后期，来自 Sun Microsystems、BEA、IBM、Oracle、Sybase 的一些 Java 程序设计语言的爱好者，开始培育一个基于 Java 技术的企业计算的想象，这是一个具有竞争力的想象，因为它的焦点在于业务利益，而且是一个具有说服力的执行策略。我们的目标是创建一个 Java 平台的标准，在这个平台上，企业系统提供商能够实现兼容产品，同时也为各自的的不同点提供扩展空间；这样，公司信息技术人员能够很容易地从可移植的组件技术中得益，将注意力集中到支撑他们的业务过程需求，而不是解决分布式基础设施产品的集成问题；这样，独立软件提供商能够被吸引到这个辉煌的组件产业中。

我们努力创建一个强大的业务伙伴联盟，动员了独立软件提供商、企业软件提供商、应用系统供应商和主要的计算机系统提供商。这个业界合作导致了一组技术，这些技术构成了 J2EE。

J2EE 包括对基本应用编程接口 (API) 的支持。

- Enterprise JavaBeans (EJB)，它是一个服务端组件模型，提供了跨越应用服务器的可移植性，并代表应用组件实现了复杂系统逻辑，例如事务处理和安全等。
- JavaServer Pages (JSP) 和 servlet，它提供了对动态内容生成和浏览器客户会话管理的

支持。

J2EE 中的其他 API 使得 Java 应用通过一组标准的编程接口来存取企业基础设施服务，这些 API 包括下列编程接口。

- JDBC，用于统一地存取关系数据库，如 Oracle、Sybase、Informix、DB2。
- Java Transaction Service (JTS)，用于存取分布式事务处理服务，基于 CORBA 的对象事务处理服务 (OTS)。
- Java Transaction API (JTA)，提供一个应用内的事务处理划分。
- Java Message Service (JMS)，用于存取各种异步的可靠的消息系统，包括可靠队列、发布与订阅服务。
- Java Naming and Directory Interface (JNDI)，用于存取各种命名和目录服务，包括 DNS、LDAP 以及 CORBA 命名服务。
- Remote Method Invocation (RMI) over Internet Inter-ORB Protocol (IIOP)，提供在 CORBA IIOP 协议之上的远程方法激活。

在 1998 年的 JavaOne 大会上，EJB 与企业 Java 平台 (JPE) 的宣言一起发布，JPE 后来被重新命名为 J2EE。在发布之时，至少有 15 家提供商宣布了在他们的产品中对这个技术的支持，包括关系数据库系统、面向消息的中间件系统、应用服务器、工具、集成开发环境等产品的供应商。从此以后，产业界的支持无疑在继续，事实上在加速，更多的产品可用来支持 EJB，并且已经有许多部署在产品中的 EJB 应用，它们解决了业务的关键需求。一个着眼于创建与销售可重用的基于 EJB 的业务组件的产业已经形成。体系结构、技术和产品很方便地帮助企业信息技术人员实现他们的目标——那就是应用的快速面市以支持新的业务需求，在继续利用他们在企业信息系统中现有投资的同时使用可移植的业务逻辑组件。

在本书中，Vlada Matena —— EJB 的首席设计师，和 Beth Stearns 神奇地解释了 EJB 的动机、目标和功能。他们通过易读的文本，辅之以真实的示例，使得无论是一个信息技术经理还是一个应用开发者，都能很容易理解这项技术，了解应用编程接口的亮点、以及应用编程接口的合理使用。

我很高兴曾经与 Vlada、Beth 以及其他众多的 EJB 技术的重要贡献者一起工作。很荣幸有机会为本书作序！

Mala Chandra

Firedrop 公司副主席，工程师

(曾任 Sun Microsystems 公司企业 Java 高级主管)

2000 年 5 月 29 日，加州

前 言

本书提供 EJB (Enterprise JavaBeans, 企业 Java 组件模型) 体系结构的深入介绍, 描绘如何使用 EJB 组件体系结构开发与部署企业应用。

本书读者应该熟悉 Java 程序设计语言, 并且对企业 bean 和 EJB 体系结构有所了解。虽然我们将简单地介绍 EJB 体系结构的基本知识, 但本书决不是一本面向那些刚入门企业 bean 的指南。相反, 本书提供 EJB 体系结构的深层次技术, 适合于企业内部实现应用的信息技术人员或者面向企业开发一般应用的独立软件提供商 (ISV)。

EJB 体系结构为企业应用定义了一个组件模型, 它描述了:

- 如何作为一个组件集合设计应用。
- 组件之间如何交互。
- 这些组件如何与它们的 EJB 容器交互。

EJB 体系结构以“约定”形式定义这些交互, 从而使得应用能够使用不同来源的组件。因为 EJB 组件必须遵循这些约定, 一个应用能够由来自不同提供商的软件组件构成。

EJB 1.1 规范主要从容器提供商的角度定义了这个体系结构约定。相反, 本书主要从 EJB 应用开发者的角度介绍这个体系结构。

有关两个企业应用开发的详细描述构成了本书的主线。尽管这些应用示例相对来说是简单的, 但是它们展示了企业应用开发中遇到的许多典型问题。我们使用这些示例展示 EJB 体系结构是如何帮助开发者解决这些问题的。

第一个应用示例是由一个 IT 部门内部开发的一个福利登记应用。这个应用对于解释一个会话 bean 怎样工作以及阐述开发者如何使用会话 bean 来说是一个很好的工作示例。

第二个示例采用第一个示例中的福利应用 (它是内部开发的), 将它变成一个由独立软件提供商 (ISV) 开发的一个应用。与一个内部 IT 部门相比, 一个 ISV 有着不同的设计目的。ISV 必须将应用设计成能够很容易地部署到许多不同客户的作业环境。因为每一个客户都有着独特的作业环境, ISV 必须面对一些挑战。另外, 一个 ISV 通常需要将应用设计成能够由一个客户或者集成商来扩展。我们阐述实体 bean 体系结构是如何帮助 ISV 克服这些困难的。

这两个示例阐述了在企业应用开发中将 EJB 体系结构应用到这些特定问题的许多技术。除了这两个应用示例, 我们描述了 EJB 体系结构的个别特性, 并讨论何时以及如何在应用中使用这些特性。

虽然典型的应用开发者不必知道 EJB 容器是如何工作的, 但是我们还是阐述了 EJB 容器的一些内幕。我们这样做主要是为了向读者展示 EJB 容器能完成许多任务来满足 EJB 应用的要求。

其他信息资源

你应该参考与 Java2 平台企业版应用体系结构 (J2EE) 相关的出版物。下列书籍对于那些开

发除了企业 bean 以外的其他应用组件的开发者来说特别有益。

- *Java™ 2 Platform, Enterprise Edition Specification, Version 1.2* copyright 1999, Sun Microsystems, Inc. Available at <http://java.sun.com/j2ee/docs.html>.
- *Java™ 2 Platform, Enterprise Edition, Connector Specification*, copyright 2000, Sun Microsystems, Inc.
- *Java™ 2 Platform, Enterprise Edition, Platform and Component Specification*, Shannon, Hapner, Matena, Davidson, Pelegri-Llopert, Cable, Enterprise Team, copyright 2000, Sun Microsystems, Inc.
- *Enterprise JavaBeans™ 1.1 Specification, Final Release*, copyright 1999, 2000, Sun Microsystems, Inc.
- *Enterprise JavaBeans™ 2.0 Specification, Public Draft*, copyright 2000, Sun Microsystems, Inc.
- *Developing Enterprise Applications with the Java™ 2 Platform, Enterprise Edition, Version 1.0*, Kassem, Enterprise Team. copyright 2000, Sun Microsystems, Inc.
- *RMI over IIOP 1.0.1 Specification*, copyright 1999, Sun Microsystems, Inc. Available at <http://java.sun.com/products/rmi-iiop>.

关于应用示例

值得注意的是，第 4 章中的应用示例和第 7 章中的实体 bean 应用的书写并没有使用交互式开发环境。通常，企业应用开发者使用一个商用 IDE 来开发 EJB 应用。一个 IDE 生成很多 JDBC 和其他的数据库存取代码，手工书写这些代码通常是很艰难的。应该认识到并记住 EJB 应用开发比我们的一些示例代码容易，也许是因为实践中大量的代码是由 IDE 自动生成的。

由于我们的目标是阐明 EJB 体系结构的使用，因此我们希望保持代码的相对简单。从而，我们并不总是表述什么样的开发者被认为是企业应用的最佳编程实践者。假如包含这种代码以显示这样的经验，那么我们会模糊 EJB 的阐述。为了简化示例代码起见，我们有时并不很好地处理代码中抛出的异常。

本书内容

本书以叙述 EJB 体系结构的好处作为开篇。第 1 章讨论不同的企业应用体系结构，以及它们是如何发展的，特别是随着 Web 的发展而发展。这一章描述了 EJB 和 J2EE 体系结构的当前状态，以及它们是如何适应于满足当今企业计算的需求。

第 2 章给出了 EJB 体系结构的一个简明的概要。对于那些尚未熟悉 EJB 体系结构的读者，这一章是一个很好的起点，因为它定义了 EJB 术语和企业 bean 的结构。它定义并描述了 EJB 应用和一些基本概念，例如业务实体和业务过程。它给出了各种类型企业 bean 的概况、构成企业 bean 的部件、以及如何使用企业 bean 来建模企业应用的业务逻辑。

EJB 应用开发可以用需要执行的任务来加以思考。第 3 章最终描绘了应用开发过程中涉及

的角色和任务。

在这个舞台搭建以及入门资料解释之后，我们将注意力集中到会话 bean 和实体 bean。有两章集中在会话 bean，另外两章集中在实体 bean。第 4 章介绍适合于那些使用会话 bean 的应用的典型编程风格。这一章对于实现会话 bean 的开发者和开发会话 bean 的客户端的应用编程者有所裨益。对于会话 bean 的开发，它描述了实现会话 bean 的方法的最佳途径。对于开发会话 bean 的客户端的应用编程者，它展示了如何正确使用会话 bean 的 home 和 remote 接口。一个福利登记应用示例阐明了会话 bean 的关键点。

第 5 章描述了一个 EJB 容器为一个会话 bean 提供的支持和服务。当会话 bean 被部署与定制到一个特定的作业环境中时，以及在运行时一个客户端应用激活这个会话 bean 时，容器通常为这些会话 bean 提供服务。虽然这些容器服务对于 bean 开发者和客户端应用编程者是隐蔽的，但是这些服务对于简化 bean 和客户端应用的开发大有帮助。这一章给出了这些服务的许多内幕。

在结束会话 bean 的讨论之后，本书将注意力转移到实体 bean。实体 bean 与会话 bean 有很大的不同。第 6 章从 bean 和客户端开发者两种角度，给出了实体 bean 编程的基本概念并讨论了容器为实体 bean 提供的服务。这一章类似于第 5 章对会话 bean 的讨论。这一章还详细描述了管理实体对象状态的策略。第 7 章利用那个用在会话 bean 中的福利登记应用示例，展示如何使用实体 bean 书写同样的应用。这个示例阐明了有关实体 bean 的许多开发技术，还阐述了实体 bean 如何被 ISV 用来制作他们的应用，以便能够跨越众多客户端的作业环境。

使用企业 bean 的所有应用事实上都依赖于事务处理。第 8 章描述了 EJB 体系结构对事务处理的划分方法。它包含了应用开发者必须明白的有关事务处理的本质。

安全是企业应用的另一个关键领域。EJB 体系结构宣布了对安全管理的支持。第 9 章描述了 EJB 安全环境，特别从应用开发者的角度进行了阐述。

本书最后包含一个有关 API 参考的附录，一个有关支撑类代码示例的附录，以及一个术语表。参考部分包含 EJB 体系结构定义的所有接口，以及每一个接口中的方法。

目 录

译者序	
序	
前言	
第 1 章 EJB 体系结构的优势	1
1.1 从二层体系结构到 J2EE 体系结构	1
1.1.1 二层应用体系结构	1
1.1.2 传统的三层应用体系结构	2
1.1.3 早期的基于 Web 的应用体系结构	3
1.1.4 J2EE 应用体系结构	4
1.2 EJB 体系结构的优势	5
1.2.1 对于应用开发者的好处	6
1.2.2 对于客户的好处	7
1.3 小结	7
第 2 章 EJB 体系结构概况	9
2.1 什么是 EJB 应用	9
2.2 业务实体、过程和企业 bean 的类型	11
2.2.1 业务实体	11
2.2.2 业务过程	11
2.2.3 业务规则的实现	12
2.2.4 企业 bean 的类型	12
2.2.5 实体 bean 和会话 bean 的选择	13
2.2.6 何时使用依赖对象	14
2.3 企业 bean 的结构	14
2.3.1 企业 bean 的 home 接口	16
2.3.2 企业 bean 的 remote 接口	18
2.3.3 企业 bean 类	19
2.3.4 部署描述符	21
2.4 容器工具和服务	22
2.4.1 容器元素	23
2.4.2 容器运行时服务	23
2.5 小结	27
第 3 章 EJB 角色	28
3.1 EJB 角色	28
3.1.1 bean 开发者	29
3.1.2 应用组装者	30
3.1.3 部署者	31
3.1.4 系统管理员	31
3.1.5 EJB 容器提供者	32
3.1.6 EJB 服务器提供者	32
3.2 工具	33
3.3 小结	33
第 4 章 会话 bean 应用开发	35
4.1 何时使用会话 bean	35
4.1.1 在 Web 应用中使用会话 bean	35
4.1.2 在传统的三层业务应用中使用 会话 bean	36
4.2 理解会话对象的状态	37
4.2.1 有状态和无状态的会话 bean 的比较	37
4.2.2 理解交互状态	38
4.3 应用示例概述	40
4.3.1 应用的用户视图	40
4.3.2 应用程序的主要部分	40
4.3.3 福利登记业务过程	43
4.4 EnrollmentEJB 有状态的会话 bean 的细节	45
4.4.1 EnrollmentEJB 会话 bean 的主要部分	45
4.4.2 会话 bean 类 EnrollmentBean 的细节	55
4.4.3 客户开发者的观点	82
4.5 PayrollEJB 无状态的会话 bean	87
4.5.1 PayrollEJB 无状态的会话 bean 的组成	88
4.5.2 客户开发者的观点	95
4.6 数据库模式	96
4.6.1 EmployeeDatabase 的数据库模式	96
4.6.2 BenefitsDatabase 的数据库模式	97

4.6.3 PayrollDatabase 的数据库模式	98	7.3.2 EnrollmentEJB 会话 bean	169
4.7 容器提供的服务	98	7.3.3 EmployeeEJB 实体 bean	181
4.8 小结	99	7.3.4 SelectionEJB 实体 bean	184
第 5 章 容器中的会话 bean	101	7.3.5 WrapperPlanEJB 实体 bean	198
5.1 容器元素	102	7.3.6 EnrollmentWeb Web 应用	213
5.2 容器在运行时怎样管理会话 bean	102	7.3.7 BenefitsAdminWeb Web 应用	214
5.2.1 EJB home 接口的查找	103	7.3.8 BenefitsDatabase	214
5.2.2 会话对象的创建	104	7.3.9 各个部件的打包	215
5.2.3 业务方法的调用	105	7.4 Premium Health 公司开发的部件	217
5.2.4 会话 bean 的活化与钝化	107	7.4.1 概况	217
5.2.5 会话对象的删除	108	7.4.2 PremiumHealthPlanEJB 实体 bean	218
5.2.6 会话 bean 的超时	110	7.4.3 HelperEJB 会话 bean	227
5.3 小结	110	7.4.4 PremiumHealthAdminWeb Web 应用	227
第 6 章 理解实体 bean	111	7.4.5 PremiumHealthAdminApp	227
6.1 实体 bean 的客户视图	111	7.4.6 PremiumHealthDatabase	227
6.1.1 home 接口	112	7.4.7 打包	229
6.1.2 remote 接口	114	7.5 Providence 公司开发的部件	231
6.1.3 主键和对对象标识	116	7.5.1 概况	231
6.1.4 实体对象生命周期	117	7.5.2 ProvidencePlanEJB 实体 bean	231
6.1.5 实体 bean 句柄	118	7.5.3 ProvidenceServiceWeb Web 应用	248
6.2 实体 bean 的 bean 开发者视图	119	7.5.4 打包	248
6.2.1 实体对象的持久性	119	7.6 集成技术总结	249
6.2.2 实体 bean 类的方法	124	7.6.1 bean 提供者发布 EJB 客户 - 视图接口	250
6.2.3 实体 bean 实例的生命周期	127	7.6.2 客户提供者发布 EJB 客户 - 视图接口	250
6.2.4 使用 ejbLoad 和 ejbStore 方法	143	7.6.3 容器管理持久性的使用	251
6.2.5 设计实体 bean 的 remote 接口	150	7.7 小结	252
6.2.6 对一个实体对象的同时调用	153	第 8 章 理解事务	253
6.2.7 与现有数据共同使用实体 bean	154	8.1 声明性事务划分	254
6.3 小结	155	8.1.1 事务属性	254
第 7 章 实体 bean 应用示例	156	8.1.2 事务属性值	254
7.1 应用概况	157	8.1.3 示例应用程序的事务属性	257
7.1.1 问题描述	157	8.2 事务的编程划分	258
7.1.2 应用的主要部分	157	8.2.1 通过客户划分事务	259
7.1.3 分布式部署	160	8.2.2 通过会话 bean 划分事务	262
7.2 Star Enterprise 公司的已有部件	161	8.2.3 使用事务的编程划分的缺陷	265
7.2.1 员工数据库	161	8.3 小结	266
7.2.2 薪水系统	162		
7.3 Wombat 开发的部件	168		
7.3.1 Wombat 部件的概貌	168		

第 9 章 管理安全	267	9.3.5 方法许可	272
9.1 系统管理员的职责	267	9.3.6 可编程安全 API	272
9.1.1 管理用户和安全要点	268	9.3.7 安全应用示例	273
9.1.2 管理要点映射	268	9.4 部署者的职责	276
9.2 容器提供者的职责	269	9.4.1 部署 Wombat 的企业 bean	276
9.2.1 鉴别和授权任务	269	9.4.2 部署 Premium Health 的 企业 bean	277
9.2.2 管理多个应用程序和域	270	9.4.3 部署 Providence 的企业 bean	277
9.3 应用程序提供者的安全视图	270	9.5 小结	277
9.3.1 客户鉴别	270	附录 A API 参考	278
9.3.2 授权	270	附录 B 代码示例	295
9.3.3 声明性安全机制	271	术语表	309
9.3.4 安全角色	272		

第 1 章 EJB 体系结构的优势

EJB 是一个在产品环境下开发、部署和管理可靠企业应用的时尚体系结构。本章阐述对于企业应用使用 EJB 体系结构的好处。

本章讨论企业应用体系结构的发展。显然，这样的体系结构必须要发展，因为其下面的计算机支撑和传输系统已经发生巨大的变化，并将在未来继续会变化。随着 Web 和 Internet 的发展，现在越来越多的企业应用是基于 Web 的，包括企业内部网和外部网应用。Java 2 平台企业版 (J2EE) 与 EJB 体系结构一起为基于 Web 的企业应用提供了无与伦比的支持。

使用 EJB 体系结构有许多好处。本章描述 EJB 体系结构的优势以及对于应用开发者和客户的好处。

1.1 从二层体系结构到 J2EE 体系结构

企业应用体系结构已经经历了巨大的发展。第一代企业应用是集中式的大型机应用。在 20 世纪 80 年代后期和 90 年代早期，几乎所有新的企业应用采纳二层体系结构（也称之为客户/服务器体系结构）；后来企业应用体系结构发展到三层结构；然后是到基于 Web 的体系结构；目前，发展到 J2EE 应用体系结构。

本节从二层体系结构开始讨论企业应用体系结构的发展。我们并没有描述体系结构如何从集中式主机发展到二层体系结构，因为这与本书的内容关系不大。

1.1.1 二层应用体系结构

一个二层应用体系结构，业务应用作为一组操作系统层面的应用进程来构成，这些进程运行在客户机上。这个客户机通常是企业内的 PC 机。每一个这样的应用实现了一个或几个业务过程，以及代表用户与业务过程之间交互的 GUI（图形用户界面）（一个业务过程是一个用户与一些企业信息交互的封装）。这个运行在客户 PC 机上的应用通过网络与存储企业数据库的数据库服务器通信。这个数据库服务器存储企业数据，而客户应用通常借助 SQL（结构查询语言）语句存取这些数据。如图 1-1 所示。

二层应用体系结构适合于 Web 出现之前的大多数应用。它的主要好处是很容易开发二层应用，特别是因为呈现逻辑与业务逻辑驻留在同一个进程中，开发者不必处理分布式应用的复杂关系。

但是，它的不利之处要比好处更多。二层体系结构的主要不利之处是其业务逻辑没有从呈现逻辑中分离开来。程序员不能在二层结构的应用

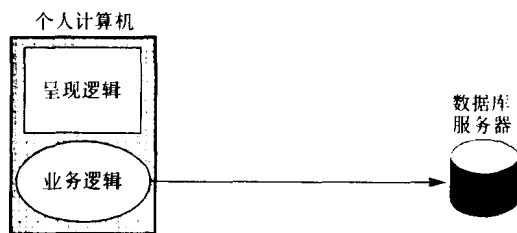


图 1-1 二层应用体系结构

中清楚地将业务逻辑从呈现逻辑中分隔出来。这就导致一系列问题：容易危及数据库的完整性、难以管理、难以维护、安全侵害、伸缩性的局限、客户体系结构的限制、局限于一种呈现类型。

- 容易危及数据库的完整性。因为每一个客户程序嵌入业务逻辑，而这个客户程序中一个错误或一个臭虫就会很容易地危及企业数据库的完整性。
- 在大型企业中难以管理。在这个体系结构中，应用部署在客户机上，企业的 IT 部门必须维护这些应用。如果一个业务过程改变了，那么 IT 部门必须用这个应用的新版本替换所有的老版本。这对于一个具有几万台 PC 机的企业来说是一个艰巨的任务，尤其是当有着许多膝上型机器时。
- 难以维护代码。二层体系结构并不支持模块程序设计，这使得应用程序的代码难以维护。对于更大的组织来说，维护的困难也呈指数级增长，因为大型企业通常使用更多的程序员来编程并维护这些应用。
- 其应用容易受到安全侵害。一个富有技巧的程序员也许会侵害安装在 PC 机上的应用，修改这个应用实现的业务过程。
- 其伸缩性的局限，难以应付大数目的用户。每一个运行的应用通常需要连接到企业数据库。打开的连接数通常受到数据库产品的限制，所有的用户就不能在同一时间运行这个应用。
- 需要一个同构的客户体系结构。在 Java 语言出现之前，二层体系结构通常需要客户机是同构的。例如，通常需要所有的客户机是运行相同类型操作系统的 PC 机。
- 将一个应用绑定于一种特定的呈现类型。因为同一个应用不仅实现了业务过程，而且实现了呈现逻辑，因此不太可能使用不同的呈现类型来重用业务逻辑的实现，例如一个浏览器或者智能蜂窝电话。

虽然企业能够在 Web 出现之前忍受二层体系结构的局限性，但是 Web 的问世与成长改变了一切。上述这些局限性本质上与 Web 完全不兼容。这主要是因为 Web 客户机固有地缺乏智能，并且有着大量的这样的客户机。其结果是应用开发者和它们的客户寻求另一种应用体系结构。

1.1.2 传统的三层应用体系结构

传统的三层体系结构克服了二层体系结构的一些限制。这个三层体系结构将呈现逻辑从业务逻辑中剥离开来。它将业务逻辑放置在服务器端，只有呈现逻辑部署在客户 PC 机上。如图 1-2 所示。

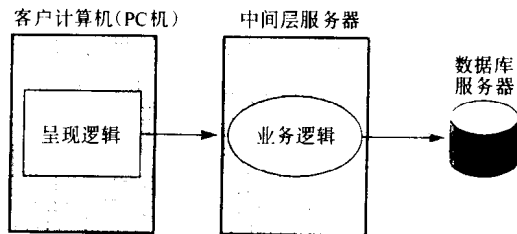


图 1-2 三层应用体系结构

三层体系结构有了一些改进。中间层服务器通过重用那些跨越多个客户的昂贵资源来改进可伸缩性，例如数据库连接。可伸缩性的改进带来了性能的改进。这个体系结构也改进了安全与应用管理。三层体系结构已经使用在大多数的企业资源规划（ERP）系统和大容量事务处理的特殊系统（例如 CICS 和 Tuxedo）中。

虽然三层体系结构消除了二层体系结构中的一些缺陷,但是它也有不足之处:复杂性、应用缺乏可移植性、提供商的不兼容性、采用的有限性、与 Web 的不兼容性。

- 复杂性。开发一个三层应用比开发一个二层应用更复杂。例如,程序员必须处理分布、多线程、安全等事项。分布式应用引入了实质性的开发者必须处理的系统级程序设计复杂性。另外,分布式应用需要客户 IT 部门来补偿应用部署与管理支撑的缺乏。为了降低分布式应用的复杂性,提供商借助于应用框架的使用来解决问题,例如事务处理(TP)监视器。ERP 提供商使用应用服务器这个概念。这个应用框架的用意在于将程序员从处理这些复杂关系中解脱出来。今天,应用服务器是一个适用于应用框架的最常用的术语。
- 应用缺乏可移植性。因为适用于三层平台的一个应用框架的各个提供商在他的框架中使用不同的应用编程接口,因此独立软件提供商(ISV)就不可能写出那些能够部署在由其他提供商提供的应用服务器的应用程序。
- 提供商的不兼容性。难以集成来自不同提供商的应用,因为各个提供商使用一组不同的协议,而在这些协议之间没有标准的互操作性。
- 采用的有限性。ISV 并没有热情去为多个尚未被广泛采用的框架写应用程序。虽然有一些工具支持分布式应用,这些工具只能在它们被开发时所面向的框架上工作。许多工具不能跨越大多数框架。其结果是,只有对一组相容的工具的有限的支持。同样,程序员对于工具和框架的知识也是很有限的。
- 与 Web 的不兼容性。这个传统的三层体系结构不能直接与 Web 一起工作。这个三层体系结构对于客户与运行在服务器上的应用之间的通信使用一个专用的协议,而这个协议不能与 Web 一起工作。虽然许多应用框架提供商在他们的前端产品上增加了对 Web 客户的支持,但是其结果依然受制于上述列出的缺陷。

1.1.3 早期的基于 Web 的应用体系结构

Web 的引入与成长改变了一切。因为二层和传统的三层体系结构都不支持 Web 应用的开发,因此早期的 Web 应用开发者不得不想出一个新的途径。它们使用各式各样的 Web 服务器插件扩充。这些扩充激活服务器上的程序来从存储在企业数据库中的信息动态地构造 HTML 文档。同样,这种 Web 服务器的扩充也从提交的 HTML 表单中将信息加入到企业数据库中。

这种扩充的一个例子是 CGI (Common Gateway Interface) 脚本 (CGI 是开发 HTML 页面和 Web 应用的一个接口, CGI 应用通常也称为 CGI 脚本)。虽然 CGI 脚本和类似的机制允许企业应用开发者构建简单的 Web 应用,但是 CGI 方式并不适合于更加复杂的企业应用,原因如下:

- CGI 脚本没有提供内在业务过程或者一个业务实体的结构良好的封装。
- CGI 脚本难以开发、维护和管理。高层的应用开发工具没有为 CGI 脚本的开发提供良好的支撑。
- CGI 脚本将业务过程的实现和呈现逻辑的实现缠绕在一起。当需要改变实现中的一部分(例如一个业务过程)时,就会有意外改变其他部分的危险。
- CGI 脚本实现并不利于业务规则完整性的维护。企业业务规则的实现被分散到多个 CGI

脚本程序中，它们被部署到企业中的几个 Web 服务器上，因此，企业维护其业务规则的完整性是困难的。

1.1.4 J2EE 应用体系结构

J2EE 是一个标准的体系结构，它特别面向使用 Java 程序设计语言进行基于 Web 的企业应用的开发与部署。ISV 和企业能够使用这个 J2EE 体系结构进行内部网应用的开发和部署，从而有效地替代二层和三层模型，对于 Internet 应用的开发，有效地替代基于 CGI 的方式。图 1-3 展示了基于 Web 应用的 J2EE 应用编程模型。

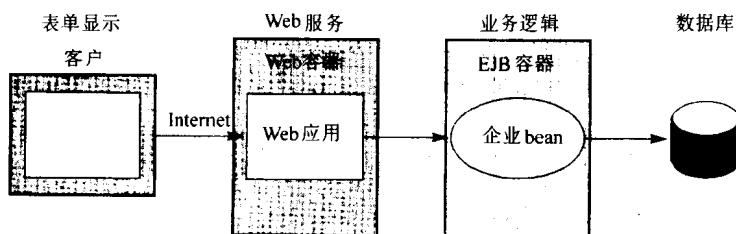


图 1-3 基于 Web 应用的 J2EE 应用编程模型

J2EE 也提供了对于二层和三层应用的支持。图 1-4 展示了对于二层应用的支持（注意：应用 - 客户容器也就是 Java 2 标准版 [J2SE] 编程环境）。

图 1-5 展示了对于三层应用的支持。

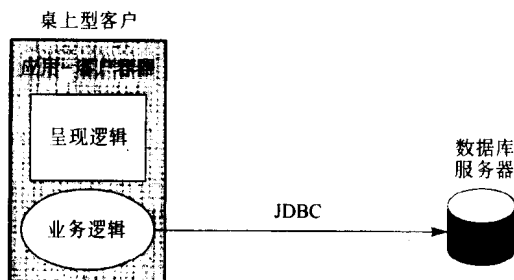


图 1-4 二层应用的 J2EE 应用编程模型

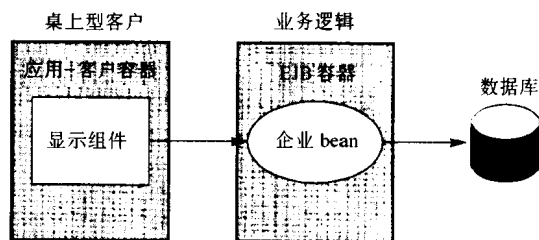


图 1-5 三层应用的 J2EE 应用编程模型

J2EE 平台也提供对于 Java applet 的支持，它是从 Web 容器装载到浏览器的小程序，如图 1-6 所示。

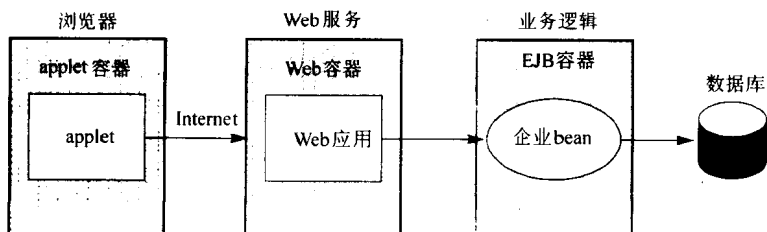


图 1-6 基于 Web 的 applet 的 J2EE 应用编程模型