Ultrawise-IBM 教育学院
教育培养计划人才培育项目指定教材

基于Java的Web应用开发

谢 峰 梁云杰 主编

高等教育出版社
HIGHER EDUCATION PRESS

Ultrawise-IBM 教育学院

教育培养计划人才培育项目指定教材

基于 Java 的 Web 应用开发

谢 峰　梁云杰　主编

高等教育出版社

内容提要

本书是 Ultrawise-IBM 教育学院教育培养计划人才培育项目指定教材。

本书在讲述 HTML 和 J2EE 部分知识的基础上，重点讲述基于 J2EE 规范的 Web 编程基础。本书共 13 章，主要内容是：基于 J2EE 的 Web 应用简介，Web 开发基础 HTTP 与 HTML，J2EE 的 Web 容器，基于 Servlet 和 JSP 的 Web 应用，构建 Web 开发环境，运行第一个 Web 应用程序，Servlet 开发，Servlet 会话、上下文、协作，Servlet 过滤器，JSP 开发，JSP 自定义标签库，MVC 模型和 Struts 2.0 简介。本书所有描述都十分贴近实际应用，并在每章中列出了大量应用案例的示例程序供读者参考。

本书可作为应用型、技能型人才培养的各类教育 IBM 软件人才院校认证的教学用书，也可供计算机从业人员和爱好者参考使用。

图书在版编目（CIP）数据

基于Java的Web应用开发/谢峰，梁云杰主编. —北京：高等教育出版社，2009. 9

ISBN 978-7-04-028094-4

Ⅰ. 基… Ⅱ. ①谢… ②梁… Ⅲ. Java 语言-程序设计-工程技术人员-资格考核-教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 151665 号

策划编辑 冯 英 **责任编辑** 高瑜珊 **封面设计** 王凌波
版式设计 王 莹 **责任校对** 张 颖 **责任印制** 尤 静

出版发行	高等教育出版社	**购书热线**	010－58581118
社 址	北京市西城区德外大街 4 号	**咨询电话**	400－810－0598
邮政编码	100120	**网 址**	http://www.hep.edu.cn
总 机	010－58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经 销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印 刷	北京四季青印刷厂	**畅想教育**	http://www.widedu.com
开 本	787×1092 1/16	**版 次**	2009 年 9 月第 1 版
印 张	13.75	**印 次**	2009 年 9 月第 1 次印刷
字 数	270 000	**定 价**	50.00 元

物料号 28094-00

序言

蓝色巨人 IBM 是信息工业的一块活化石，同时又是这个时代最富有活力、最受人尊敬的公司之一。在技术研究上的巨大投入使得 IBM 成为世界上最具创新性的公司，IBM 的创新已经获得了超过 25 000 项美国专利，几乎是任何美国 IT 竞争对手同期总和的三倍，超过了惠普、戴尔、微软、Sun、Oracle、英特尔、苹果、EMC、Accenture 和 EDS 的总和。正是这些历史的积累和不断的锐意创新铸就了 IBM 后端庞大的知识体系，并使得其庞大的产品家族在业界强盛不衰。

面对市场在软件人才数量和结构方面的双重需求，IBM（中国）一直致力于帮助软件企业建立合理的人才架构和供求关系，为培养高素质、复合型人才创建健康的大环境。2002 年 4 月 3 日，IBM 公司宣布将培养 10 万软件生力军以满足中国市场对软件技术开发、软件市场化、软件企业经营管理等各类人才的需求。在此期间，将有 1000 家软件合作伙伴和 100 万人次的软件业界人才从中受益。IBM 教育学院就是这一战略的具体实践。IBM 教育学院成立于 2003 年初，面向国内所有初、中、高级的软件开发及 IT 管理人员，主旨是提供一个广泛的信息交流及技能培训的平台，帮助他们快速深入地掌握最新的软件技术及应用整合方案。因此，加入到 IBM 教育学院教育培养计划，循序渐进地获取知识与技能，是成为“随需应变”型优秀软件人才的第一步。

2003 年 4 月，IBM（中国）有限公司正式与上海智翔信息科技发展有限公司（Ultrawise）签署战略合作协议，由 Ultrawise 全面负责 IBM 教育学院教育培养计划人才培育计划项目的推广、实施工作。项目将依托 Ultrawise 对国内职业教育的丰富经验、遍布全国的授权教育体系以及 IBM 公司强大的知识积累和其全球领先的产品技术优势，从市场需求出发，全面整合当今主流技术，帮助合作院校度身定制专业课程，倾力帮助院校培养“专、职”（专业型、职业型）型软件应用、开发人才。

2004 年 5 月，在 IBM 教育学院教育培养计划的大力支持下，推出了 Ultrawise-IBM 教育学院教育培养计划人才培育项目（院校合作）系列课程。课程体系分为基础、技术和应用 3 个系列，既注重学生的理论基础、技术根基，又重视学生最终在职场上能直接获益的应用技能。通过基础课程的培养，使得学员能够迈过软件应用、开发的基础门槛；在技术课程的培训中，学员可以根据不同的职业角色（Job Role）循序渐进地学习该方向所必需的理论、技术，以形成牢固的知识框架和良好的技术素养；在最后的应用课程中，学员的学习主线将延伸到实践和应用层面，通过掌握几种业界领先厂商的相关产品，来实现对于所学技术和知识地运用，并能以此作为下一步求职的关键技能和敲门砖。另外，由于 Ultrawise-IBM 教育学院教育培养计划人才培育项目（院校合作）课程体系与 IBM 全球专业认

证体系有着承前启后的关系，学员可以继续根据其职业角色对感兴趣的IBM 软件产品进一步深入学习，获取 IBM 全球专业认证，成为最优秀企业争夺的 IT 技术专家，为自己的职业生涯添加一枚沉甸甸的砝码。

本书作为项目的基础课程，主要介绍基于 Java 的 Web 编程技术基础，同时还涉及了一部分互联网编程基础 HTML 和 J2EE 的基础知识。

在本书中所有描述的程序都十分贴近于实际工作中 Web 编程的基本应用。在 J2EE 的 Web 规范中一些不经常使用的规范或者一些高阶的应用，比如 JSP 标准 Tag、Web 图像处理、Javascript 动态网页技术等都不做介绍。目的在于使学员在学习教材中描述的书面知识的同时，也能以最快速度掌握 Web 应用开发的技巧和基础知识。

本书主要分为三大部分。

第一部分从第 1 章～第 6 章，主要介绍网络编程的基础知识，Web 编程开发环境的构建；

第二部分从第 7 章～第 11 章，主要深入地介绍 J2EE 规范中 Servlet 和 JSP 编程。

第三部分为第 12 章和第 13 章，主要介绍 Web 的高阶编程、MVC 模型以及 Struts 2.0 框架。

在每章都有大量的示例程序，这些示例程序都是典型的应用案例。此外，本书还有一个附录供读者在进行实际工作时予以参考。

由于时间仓促，缺点和错误在所难免，恳请专家和广大读者不吝赐教，批评指正。如果有表达不恰当的或者错误的地方，我们也欢迎读者给我们反馈，我们会尽快安排修订。最后，需要感谢本书主编谢峰、梁云杰，参编金怡冬、符泉麟、张志浩的辛勤工作，同时还要感谢蔡文琼对本书的编辑校对。希望这本书能给致力于从事 J2EE Web 编程的读者带来帮助。谢谢！

Ultrawise-IBM 教育学院

二〇〇九年八月

目录

第1章
基于J2EE的Web应用简介

学习目的

了解Web的基本应用

了解Web的基本应用架构

了解J2EE中Web的相关技术

1.1 Web应用发展

提到 Web 应用，就不得不提到 HTTP 协议以及 Internet 的发展。最初 Web 应用诞生的目的仅仅是科学家为了在大学校园内方便交换各自论文，而 Internet 和基于 HTTP 协议的 HTML 静态网页就成了这种方式的最好媒介。

起初这种简单的 Web 应用仅仅是一个个独立的站点，也称为 Web Site。HTML 页面称为超文本链接标记语言（HyperText Markup Language），是一种由简单的 HTML 命令组成的描述文本，HTML 命令可以说明文字、图形、动画、声音和表格等。页面与页面的交互是通过超链接完成的，而对于内容的修改，只能通过修改静态 HTML 网页来完成。从某种意义上说，这些网页只是电子形式的文本，在一处生成，内容固定，再发布到多处。

在最初阶段，对于大学院校来说，这种应用已经足够了。然而在企业界，很少有人发现这一新的渠道能带来什么“商机”。但不久之后，Web 用户就开始有新的要求，希望能得到更动态的网上体验。个人计算机成为企业不可或缺的资源，从个人宿舍、住家到办公室开始出现越来越多的计算机。随着 Windows 95 的问世，人们已经逐渐领教了 Corel WordPerfect 和 Microsoft Excel 丰富的功能，用户的期望也越来越高。Web 开始真正进入了动态时代。

CGI（Common Gateway Interface）首先问世，称为公共网关接口，是一种特殊的应用程序，部署在一台服务器上，供客户端也就是浏览器中的网页来访问。通俗地讲，CGI 就像一座桥把静态网页与服务器上执行的程序连接了起来，HTTP 协议就是这两者交互的媒介。CGI 程序分析 HTML 中的表单信息，然后进行运算处理，将动态处理后的 HTML 文本通过 HTTP 协议写回给浏览器，以此完成交互。CGI Web 应用架构如图 1-1 所示。

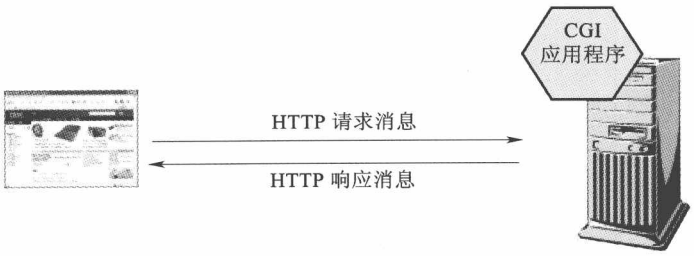

图 1-1 CGI Web 应用架构

CGI 应用程序为日后 Web 应用的发展奠定了基础，后来的任何一种 Web 应用模型都是在其基础上发展的。

虽然 CGI 可以实现动态交互，但是它所有的动态效果都需要在 CGI 程序中使用 print 语句输出。这样对于传统的编程方式而言，开发人员很难直观地看出最终输出的动态效果。程序设计人员与页面设计人员无法通过工

具进行配合使用。从安全角度讲，CGI 也存在一定的漏洞。最著名的漏洞就是在初期的 CGI 开发阶段，使用 C 语言等面向过程语言编写的 CGI 程序很容易受到缓冲区溢出攻击。

在 Web 应用程序还处于起步阶段的时候，应用软件已经步入了桌面应用时代，这时期最流行的就是以 Microsoft 公司的 VB、VC，Borland 公司的 Delphi、C++ Builder，以及 Sybase 公司的 Power Builder 三大阵营为主的应用软件。桌面应用软件结合数据库构成了当时大多数的企业级应用系统。桌面应用程序架构如图 1-2 所示。

图 1-2 桌面应用程序架构

CGI 应用程序在交互性和安全性上都与当时的桌面应用软件根本无法相比。在随后的几年中，虽然相继诞生了交互性很强的 Applet 和 ActiveX，可是由于种种原因都没有在 Web 应用中大面积使用。

进入 20 世纪 90 年代后期，Internet 的商业价值开始渐渐地被企业界所挖掘，各种商业的门户网站和购物网站开始渐渐地兴起。这其中一大部分功劳要归功于几种新型的 Web 服务器端应用技术：JSP、ASP 和 PHP，这三种 Web 应用技术至今仍然被大量地使用。

ASP 技术是微软公司开发的代替 CGI 程序的一种应用。ASP 是 Active Server Page 的缩写，即活动的服务端页面。ASP 在服务端运行，可以创建和运行动态网页，ASP 可以包含 HTML 标记、普通文本、脚本命令以及对一些特定微软应用程序的调用，比如 COM 组件，也可以包含一些交互式的内容，比如在线表单。通俗地说，ASP 就是由 HTML 文本与动态脚本程序相结合的特殊文本程序，在服务器端运行。运行时，ASP 服务器将 ASP 脚本翻译成静态的 HTML，返回给浏览器。由于特定的服务器限制，ASP 已经渐渐地退出了 Web 应用。

PHP（Hypertext Preprocessor）全称超文本预处理语言，完全基于开源代码的脚本式语言，与 ASP 采用相同的脚本技术，嵌入 HTML，但是又混入了 C、Java、Perl 以及其自创的一些语法。PHP 功能非常强大，几乎支持所有的数据库以及操作系统，并且得到广大开源社区的支持，这是 PHP 比 ASP 更加流行的主要原因。

JSP 技术是 Sun 公司于 20 世纪 90 年代末提出的基于 Java 语言的 Server 端脚本技术。在 Java 语言问世一年后，Sun 公司引入了 Servlet 这一概念，称为 Server 端小程序。与之呼应的是称为 Applet 的客户端小程序。Servlet 与 Applet 不同，Java 代码不再像 Applet 那样在客户端浏览器中运

行，而是在 Web 应用服务器中运行。从系统架构角度讲，Servlet 就是 Sun 公司推出的代替 CGI 程序的解决方案。同 CGI 一样，虽然 Java 语言本身强大的面向对象特性大大提高了 Servlet 的可读性，并降低了 Servlet 的开发难度，但是 Servlet 还是需要通过类似 print 语句输出动态网页内容，还不能将网页设计和程序设计剥离。于是 Sun 公司于 1998 年提出了 Java Server Page，简称 JSP，一种使用 Java 语法的动态脚本技术。JSP 强大之处在于它本身不仅仅是一种脚本语言，而且是一种企业级应用软件的 Web 开发标准。它的运行状态由各种 J2EE 应用服务器决定。这样开发人员在开发 JSP 时，许多底层的技术细节都由应用服务器厂商来解决。同时 JSP 有很强的开发性和可扩展性，尤其是 JSP 提供的 Tag Lib 这一新概念，使服务器端编程开始组件化、模块化。

进入 21 世纪，飞速发展的电子商务和 Internet 技术，推动着企业信息系统的构建和更新进程。随着业务规模的飞速膨胀，客户对于企业级应用软件的诸多期望始终无法得到完全满足。在这种市场需求下，要实现企业各个层次的集成，必然会导致软件在规模、复杂度和功能上的空前扩张。客户端处理业务逻辑的方式，使得客户端/服务器这种传统的计算模型难以重负，任何细微的业务需求变化都会给客户端的应用程序更新带来巨大的影响。

为了跟上这种新形势的发展，软件模型开始有了新的改进。Web 应用特有的 Thin Client 模式开始发挥作用，越来越多的信息化软件开始采用 Web 技术开发，JSP、PHP 以及后来的.Net 技术开始全面取代桌面信息管理软件。

不过基于 Web 开发的应用软件在操作性上始终无法给予客户非常舒适的体验。电子商务网站、门户网站和论坛等应用不需要太强的交互性，Web 应用完全可以胜任。可是对于实时交互性要求很强的应用系统，还是无法与桌面应用程序匹敌。这里不得不提一下 Web 应用开发的一个重要技术，那就是 DHTML，即动态网页技术。

DHTML 不是一个标准，而是网页开发几个最重要技术的结合，它们分别是 HTML、CSS Cascading Style Sheets（级联样式表）和 JavaScript。

CSS 是一种样式表语言，用于为 HTML 文档定义布局。CSS 涉及字体、颜色、边距、高度、宽度、背景图像和高级定位等方面。

JavaScript 是网景（NetScape）公司创建的一种脚本语言。最初的设计目的是为了帮助开发人员动态地修改页面上的标记，以便为客户提供更丰富的体验。在随后的发展过程中，JavaScript 被广泛地应用于创建各种各样的网站和 Web 应用系统。从技术角度讲，JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言，它的主要操控对象就是基于 HTML 元素的 DOM（Document Object of Model）对象，其采用的语法与 Java 语言类似。不过 JavaScript 是一种只能在浏览器运行的语言，因为只有在浏览器中，HTML 才能被翻译成 DOM 对象。JavaScript 现在已经成为 Web 应用开发中一个必不可少的环节。

DHTML 提供的动态特性使得提高 Web 应用交互性成为可能，开发人员可以像使用 VB、Delphi 等桌面开发语言一样使用 JavaScript 操控 HTML 元素，

但仍然无法从根本上改进 Web 交互的问题。这主要是因为 HTTP 协议的特殊性，每一次请求都需要和服务器交互一次，然后在得到响应后刷新网页，这就使很多操作都反复地刷新网页，从而使 Web 的可操作性下降。于是 Web 开发又将改进这一弊端的方案瞄准了 AJAX、Asynchronous JavaScript 和 XML。

AJAX 并不是什么新技术，它是利用特殊的 DOM 对象 XMLHttpRequest 代替 HTML 中 FORM 的提交和响应机制。XMLHttpRequest 对象实际上是一种利用 XML 作为 HTTP 协议传输媒介的封装对象，其最大特点就是支持异步传输。通过 JavaScript 操控的 XMLHttpRequest 对象，使在网页与服务端程序进行异步交互时，不需要反复地刷新页面，从而提高了页面的交互性，使用户在使用 Web 应用时达到几乎和桌面应用一样的效果。

现在，AJAX 技术被越来越多地采用，有许多新兴地使用 AJAX 的 Web 应用都把这种技术称为 RIA，也就是 Rich Client Internet Architecture，把它看成是 Web 应用和桌面应用的完美结合，甚至把它认为是 Web 2.0 的重要技术标准。

其实除了上面提的这些 Web 应用技术之外，还有很多其他的 Web 应用技术，比如 Flash 以及基于 Flash 的 Flex 框架，微软公司的 WPF、Sliver Light，Adobe 公司的 Applo 等都是新兴或者时下被广泛采用的 Web 应用技术。

从发展趋势上来看，Web 应用将更加趋于标准化、通用化。甚至有人提出了用 XML 代替 HTML 作为桌面表现层与 Web 表现层的统一描述性语言，比如 Mozilla 的 XUL 和基于 Java 开源的 XAMJ 等。

1.2 Web 应用架构

随着越来越多的企业开始用计算机来管理公司的核心业务，基于低级语言开发的单机应用软件这种应用已远不能满足行业的需求。越来越多的数据和业务信息都需要有专门的管理软件来集中管理，越来越多的终端客户要求参与业务管理，要求越来越高的用户操作友好性也需要更加丰富的界面图形来展现。于是便诞生了客户端/服务器模型。

对于桌面应用程序来说，客户端/服务器模型也称为胖客户端/服务器模型（Rich Client/Server），如图 1-3 所示。

所谓“胖客户端”，指的是利用本地硬件资源以及本地操作系统提供运算功能。在 Rich Client/Server 模型中，客户端可以由 N 台 PC 构成，每个客户端会运行一个客户端应用程序完成业务操作。服务器通常由一台或者几台高性能 PC 构成，主要是使用数据库为业务信息提供存储和查询的集中管理功能。“胖客户端”的优势在于它可以充分利用客户端的硬件资源，从而减轻服务器的压力，同时可以利用客户端的桌面资源提供丰富的客户体验；而其缺点也显而易见，每次程序更新升级都必须将更新程序在每台客户端做一次安装部署，从而导致了极大的维护代价，很难满足随需应变的企业级应用软件的要求。

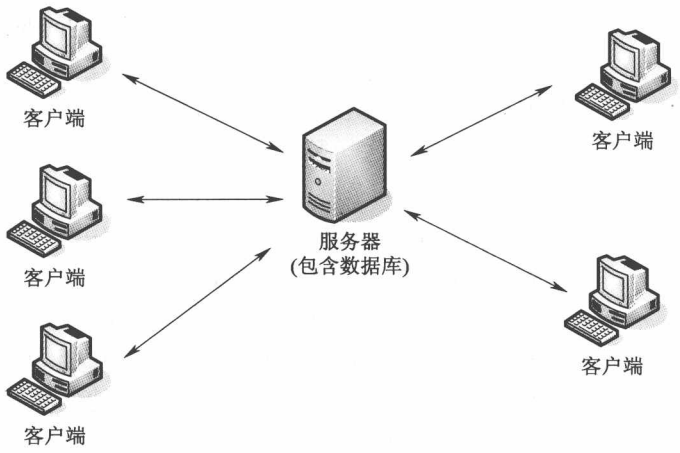

图 1-3 客户端/服务器模型

而“瘦客户端”（Browser/Server）却没有这个问题。“瘦客户端”指的是在客户端/服务器模型中一个基本无需应用程序的计算机终端。现在绝大多数 PC 的操作系统都有浏览器。而其展现形式又是通过 HTTP 协议传输的 HTML 文本，所以每次应用程序部署或者更新，无需在客户端执行，只需要更新服务端程序。图 1-4 所示的是 Browser/Server 的物理架构。

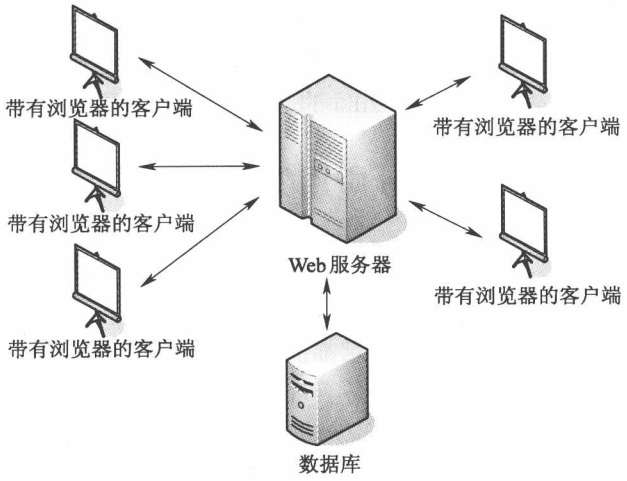

图 1-4 Browser/Server 物理架构

图 1-5 所示的是 Browser/Server 模型的逻辑架构。

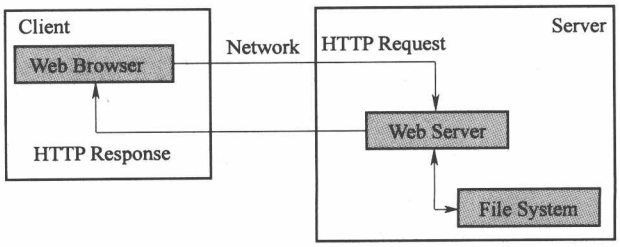

图 1-5 Browser/Server 逻辑架构

在 Browser/Server 模型中，浏览器和服务器的交互通过 HTTP 协议进行。所有的动态网页（如 JSP、ASP）、静态网页以及图片都发布在服务器上。当浏览器向服务器发出 HTTP 请求（HTTP Request）时，服务器将相应的动态和静态信息返回给 Client（HTTP Response），浏览器根据返回信息的 HTTP 信息正文翻译 HTML 元素并展现。例如，在网页中输入 www.google.cn这个域名，其实就是向 google 的 Web 服务器发送了一个 HTTP 协议请求，而在浏览器中看到 google 的主页，就是浏览器对 google 网页翻译的结果。

从某种角度来说，HTML 和浏览器之间的关系很像是 Java 语言和 Java 虚拟机之间的关系。浏览器和 Java 虚拟机都有一个容器的概念，也就是说 HTML 必须通过浏览器来解释运行。但是对于 HTML 而言，它并不关心运行在什么样的浏览器中，因为它是 W3C 的标准（有些特定的浏览器并不完全支持 W3C 的标准）。HTML 元素中所有的 DOM 对象都依赖于浏览器产生。而 Web 服务器只是提供 HTML 信息的一个渠道，对于 Web 服务器而言，它也不会关心所部署的 Web 资源何去何从，它只负责接受请求，并且响应请求。

1.3 J2EE 技术简介

当软件行业沉浸在 C/S 向 B/S 模型过渡的喜悦时期，Sun 公司于 1999 年 6 月又给软件行业带来了一个重量级成员——J2EE（Java 2 Platform Enterprise Edition）。它将传统软件的发展完全地推向了另一高度——分布式模型。

在介绍 J2EE 的 Web 应用之前，首先介绍一些 J2EE 的基本概念。

1．中间件

中间件处在操作系统和更高一级应用程序之间。它的功能是将应用程序运行环境与操作系统隔离，从而使应用程序开发者不必为更多系统问题忧虑，直接关注该应用程序在解决问题上的能力。

2．容器

容器是中间件的一种。在 J2EE 体系结构中，每一层的应用都需要有相应的运行环境，这些环境为基于标准的业务组件提供了基本的服务。在 J2EE 体系结构中，主要的容器有以下两种：

（1）Web 容器

给处于其中的应用程序组件（JSP、Servlet）提供一个环境，使 JSP 与 Servlet 直接和容器中的环境变量接口交互，不必关注其他系统问题。Web 容器主要由 Web 服务器来实现。

（2）EJB 容器

EJB（Enterprise Java Bean）也称为企业级 Java 对象，是 J2EE 体系结构中的基石。它提供给运行在其中的组件 EJB 各种管理功能。只要将满足 J2EE 规范的 EJB 放入该容器中，就立即被容器进行高效率地管理，且通过

现成的接口来获得系统级别的服务。例如邮件服务和事务管理。

Web 容器和 EJB 容器在原理上大体相同，区别主要是被隔离的外界环境。Web 容器更多的是与基于 HTTP 的请求打交道；而 EJB 容器主要与数据库和其他服务打交道。但它们都是通过与外界的交互实现，从而减轻应用程序的负担。

3．命名服务

Java 命名目录服务主要功能是提供一个目录系统，让其他各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能需要。

J2EE 分布式模型的划分是根据功能而不是根据物理方面进行的。在J2EE 的分布式结构中具体划分方式如图 1-6 所示。

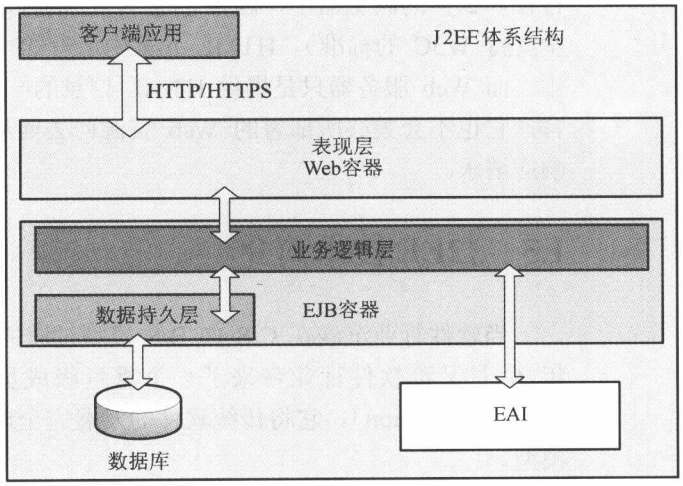

图 1-6 J2EE 分布式模型的划分

4．客户端应用

客户端用于与企业信息系统的用户进行交互以及显示根据特定商务规则进行计算后的结果。基于 J2EE 规范的客户端可以是基于 Web 的，也可以是不基于 Web 的独立应用程序。

在基于 Web 的 J2EE 客户端应用中，用户在客户端启动浏览器后，从 Web 服务器中下载Web 层中的静态HTML页面或由JSP、Servlet 动态生成的 HTML 页面。

在不基于 Web 的 J2EE 客户端应用中，独立的客户端应用程序可以运行在一些基于网络的系统中，比如手持设备、汽车电话等。同样，这些独立的应用也可以运行在客户端的 Java Applet 中。这种类型的客户端应用程序可以在不经过 Web 层的情况下直接访问部署在 EJB 容器（EJB Container）中的 EJB 组件。

5．表现层（Web 容器）

J2EE 规范定义的 Web 层由 JSP 页面、基于 Web 的 Java Applets 以及用于动态生成 HTML 页面的 Servlet 构成。这些基本元素在组装过程中通过打