

深入

雨阳隆春 等 编著

JSP

网络编程



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

深入JSP网络编程

雨阳隆春 等 编著

清华大学出版社

(京)新登字 158 号

内 容 摘 要

本书较为深入、详实地讲述了怎样利用 JSP 构建完整的虚拟网站的全部技术与过程。全书共分为五个部分:第一部分是相关知识,包括 JSP 起源与构建 JSP 环境。第二部分是支持技术,包括 HTML 与 Dreamweaver。第三部分是继承者,包括指南、Java 基础、JSP 基本语法、Servlet 及其 API、内部对象、JSP Container、JSP 核心 API。第四部分是集成者,包括 JDBC、JavaBeans、Applet、XML。第五部分是综合应用。

本书语言简洁,由浅入深,既可作为广大 JSP 网络编程人员的参考指南,也可作为资深网站编程人员理想的参考用书。

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。

图书在版编目(CIP)数据

深入 JSP 网络编程/雨阳隆春编著,一北京:清华大学出版社,2001.6

ISBN 7-302-04336-1

I. 深... II. 雨... III. 计算机网络—程序设计

IV. TP393.092

中国版本图书馆 CIP 数据核字(2001)第 034827 号

出 版 者:清华大学出版社(北京清华大学学研大厦 A 座,邮编 100084)

责任编辑:宋 韬

印 刷 者:北京市清华园胶印厂

发 行 者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:32.75 字数:756 千字

版 次:2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

书 号:ISBN 7-302-04336-1/TP·2680

印 数:0001-5000

定 价:52.00 元

前 言

亲爱的读者,非常高兴您注意到本书。JSP 这种新兴的编程技术,说它新,真是当之无愧。1998 年 4 月, Sun 公司发布 JSP 0.9 规范,到现在,才有 JSP 1.2 草案。历史很短,然而成就却是巨大的。到网上看看,就知道它的影响有多大了。本书中前言部分涉及到的内容包括本书主要内容、特点、符号约定、适用对象,另外还有编写分工、致谢和参考文献。

主要内容

本书的书名是《深入 JSP 网络编程》,从概括书的内容、结构和特点这些角度看,我们不妨为本书拟定的另一个戏说书名——《三“心”二“意”不懂四“书”五“经”学 JSP 构建虚拟网站》,更能体现本书的主要内容、结构和特点。

那么,何谓三“心”二“意”,四“书”五“经”?

所谓三“心”就是 JSP 技术的三个核心概念,即:从整体技术角度把握,JSP 是服务端动态页面技术;从组成结构角度把握,JSP 是页面应用程序(Web Application);从语义实现角度把握,JSP 是一个容器(JSP Container)。即这三“心”的关系是从整体到局部,层层递进的关系。前面说的仅仅是概念上的三“心”,而基本语法、Servlet 和内部对象则是内容上的三“心”。二“意”是指 JSP 在内容上,既间接继承了 Java 的语法、类库,又直接继承了 Servlet 的类库、容器。事实上,JSP 与 Servlet 关系是如此的紧密,以至于可把 JSP 看成 Servlet 的特定简化实现。

四“书”是指 JSP 主要集成了四种技术:首先,JSP 是三层架构 C/S(客户/服务)技术,集成了 JDBC;其次,JSP 支持组件模型,集成了 Java Beans;再次,JSP 支持客户端动态页面技术,集成了 Applet;最后,JSP 支持开放技术,集成了 XML。说 JSP 是一个集大成者一点也不过分。五“经”是指本书共分为五个部分:第一部分是相关知识,包括 JSP 起源与构建 JSP 环境;第二部分是支持技术,包括 HTML 与 Dreamweaver;第三部分是继承者,包括指南、Java 基础、JSP 基本语法、Servlet 及其 API、内部对象、JSP Container、JSP 核心 API;第四部分是集成者,包括 JDBC、JavaBeans、Applet、XML;第五部分是综合应用。

JSP 内容众多,本书主要讲解构建网站的关键技术,不作面面俱到的讲解。重点讲了 JSP 语法、内部对象 Servlet、JDBC 和综合应用内容。

相信本书会使读者对 JSP 也有了一个比较清晰的整体印象,并学到非常实用的编程知识和技巧。

特点

入门要求低:阅读本书不要求一定会 HTML、Dreamweaver 和 SQL,本书用较少的篇幅系统地、结构化地介绍了上述内容。在不了解上述内容的情况下,读者会在阅读中不知不觉

已经循序渐进地掌握了上述知识。

抓住主要矛盾:本书抓住主要矛盾体现在两个方面。一是决不纠缠于与 JSP 不密切的知识。比如 JSP 环境构建、数据源的建立,如果在学习 JSP 之前还要求掌握 Linux 的操作,或者要求使用 Oracle 建立一个纯粹的 Java 驱动程序(JDBC),那就没有抓住主要矛盾。事实上,在哪个平台、哪个数据库支持下并不重要,重要的是怎样在该平台、数据库下使用 JSP。抓住主要矛盾的另一个方面是对主要内容绝对不惜笔墨,而对次要内容惜墨如金,如对 Servlet、JDBC 不惜一切代价争取写得更深入一些,而对 HTML 只讲述了表单、表格和框架,对 Dreamweaver、SQL 也是详略得当。这一点请读者明鉴。

完整性:本书基本上实现一个完整的网站所需知识,包括动态广告、时间日期服务、新闻、BBS、电子商务、留言板等等。附录 2、附录 3 包括了完整的 Servlet API 包,这对 JSP 的开发是非常有帮助的。JDBC API 也基本上做到了完整,包括所有的重要的类和界面。

概括性:本书每章的标题不仅对该章内容具有高度概括性,而且,内容也极具代表性和概括性。

符号约定

本书一共使用了三种符号,它们代表三个层次,它们是:

▼:代表顶层,有顶层目录的意味。

▽:代表中间层,有子目录的意味。

▶:代表底层,有文件的意味。

如:

▼元素(Elements)

▽脚本元素

▶声明

▶脚本片段

▶表达式

▽注释

▶输出注释

▶隐藏注释

.....

代表元素分为脚本元素、注释等几类,而脚本元素又分为声明、脚本片段和表达式三类。其余类推。另外,本书采用 Word 排版,有“注意”、“技巧”和“提示”等图文表示,并且对 JSP 语法、命令和程序代码用不同的底纹表示,请读者注意。

适用对象

本书不要求一定要了解 HTML、Dreamweaver 和 SQL,但最好对 Java 程序设计语言有相当的了解,熟悉其主要的类库(包)。了解 Java 是本书对读者的惟一要求,同时将发现了

解 Java,对学习、使用 JSP 是大有裨益的,用“如鱼得水,如虎添翼”来形容再恰当不过了。并且,也将看到 Java 这样的程序设计语言要在四五十页内讲清是根本不可能的。

编写分工

本书第 7 章、第 11 章和第 14 章聊天室部分由李云集编写,其余章节由雨阳隆春编写。全书由雨阳隆春统稿。此外,参加本书编写的还有李长传、李东阳、李义、王东、王三高、王品、张择、张阳雪、李春雨、杨新阳、张光明、赵问、赵长志、赵虚、许刚、许强、陈阵、陈加目、赵烁、陈金、陈非、孙拟、孙引元、崔飞、崔函、伍姓、伍冬夏、伍飞雪、伍欲和伍是等,在此一并致谢。

感谢博嘉科技的王松老师。首先感谢他的理解和支持,使本书能在较短的时间内以较高的质量面向读者。

感谢博嘉科技的刘清松等,是他们使本书内容更精致、版式更合理。

由于我们的水平和经验有限,书中难免有不准确、不贴切甚至错误的地方,欢迎读者批评并提出宝贵意见和建议。如有意见或建议,欢迎与我们联系:◆电话:(028) 5404228 ◆E-mail: bojiakeji@163.net。我们的通讯地址是:四川大学西区建筑学院成都博嘉科技资讯有限公司,邮编:610065。

雨阳隆春

2001 年 4 月

目 录

第 1 章 JSP 起源、构建 JSP 环境	1
1.1 JSP 起源	1
1.1.1 JSP 的含义	1
1.1.2 JSP 起源与 Web 发展趋势	1
1.1.3 JSP 的设计目标	3
1.1.4 JSP 与三层结构模型关系	4
1.1.5 JSP 与 ASP 的比较	5
1.1.6 JSP 与 PHP 的比较	8
1.2 构建 JSP 环境	8
1.2.1 什么是 JSP 环境	8
1.2.2 为什么构建 JSP 环境	8
1.2.3 怎样构建 JSP 环境	9
1.3 本章小结	17
第 2 章 HTML 与 Dreamweaver	18
2.1 HTML	18
2.1.1 TABLE(表格)	18
2.1.2 FORM(表单)	23
2.1.3 FRAMESET(框架)	30
2.2 Dreamweaver	34
2.2.1 表格	34
2.2.2 表单	37
2.2.3 框架	39
2.2.4 用图层排版页面	41
2.3 构建虚拟网站	42
2.3.1 构思 JSP 虚拟网站	43
2.3.2 构建 JSP 虚拟网站	45
2.4 本章小结	47
第 3 章 JSP 指南	48
3.1 Hello World	48
3.2 从组成的角度看 JSP	51
3.2.1 JSP 语法一览	51
3.2.2 JSP 是 Web Application	52

3.3 本章小结·····	53
第4章 Java 基础 ·····	54
4.1 Java 程序设计基础·····	54
4.1.1 Java 应用程序的组成·····	54
4.1.2 Java 程序设计基础·····	55
4.2 Java 面向对象程序设计·····	61
4.2.1 对象·····	62
4.2.2 类·····	62
4.2.3 打印杨辉三角·····	65
4.3 本章小结·····	67
第5章 JSP 基本语法 ·····	68
5.1 元素·····	69
5.1.1 脚本元素·····	69
5.1.2 注释·····	74
5.1.3 指令·····	75
5.1.4 行为·····	87
5.2 Template Data·····	96
5.2.1 Template Data·····	96
5.2.2 Template Text·····	96
5.2.3 引用与转义·····	97
5.3 为虚拟网站加第一块砖·····	100
5.3.1 连接页头、主体和页脚·····	100
5.3.2 广告轮显·····	102
5.4 本章小结·····	105
第6章 Servlet ·····	106
6.1 Servlet 概述·····	106
6.1.1 什么是 Servlet·····	106
6.1.2 为什么要使用 Servlet·····	107
6.1.3 Servlet 与 CGI 相比有哪些优点·····	107
6.1.4 Servlet 与 JavaServer 体系结构的关系·····	108
6.2 Tutorial·····	110
6.2.1 编写 Servlet·····	110
6.2.2 编译 Servlet·····	114
6.2.3 运行 Servlet·····	114
6.2.4 Servlet 的基本执行流程·····	116

6.3 与客户端交互	117
6.3.1 request 和 response	117
6.3.2 处理 GET 和 POST 请求	118
6.4 Servlet 的生命周期	122
6.4.1 初始化 Servlet	122
6.4.2 与客户端交互	124
6.4.3 销毁 Servlet	124
6.4.4 Servlet 结束时处理 Service 线程	125
6.5 存储客户端状态	128
6.5.1 Session 跟踪	128
6.5.2 Cookies	132
6.6 Servlet 的通信	137
6.6.1 通过 RequestDispatcher 对象使用服务器上的其它资源	137
6.6.2 在 Servlet 间共享资源	140
6.6.3 从 Servlet 中调用其它 Servlet	142
6.7 运行 Servlet	143
6.7.1 在浏览器地址栏中直接键入 Servlet 的 URL	143
6.7.2 从 HTML 页面调用 Servlet	143
6.8 本章小结	146
第 7 章 内部对象	147
7.1 内部对象概述	147
7.1.1 内部对象的功能	147
7.1.2 内部对象的作用域	148
7.2 JSP 内部对象详解	149
7.2.1 Resquest 对象	149
7.2.2 Rresponse 对象	158
7.2.3 Out 对象	165
7.2.4 Session 对象	169
7.2.5 Application 对象	179
7.2.6 其它内部对象	183
7.3 还想多了解点吗	188
7.4 本章小结	190
第 8 章 JSP Container	191
8.1 编写支持实例	191
8.2 命名约定	200
8.3 编译	201

8.3.1 编译	201
8.3.2 预编译	201
8.4 调试和错误处理	202
8.4.1 调试	202
8.4.2 错误处理	202
8.5 翻译执行	202
8.6 容器	203
8.6.1 什么是 JSP 容器	203
8.6.2 JSP 页面与 JSP 容器的关系	203
8.6.3 JSP 页面实现类	204
8.6.4 JSP 容器的行为	205
8.7 留言板	213
8.7.1 留言板的说明	213
8.7.2 留言的处理	213
8.7.3 查看留言	217
8.7.4 错误处理	218
8.8 本章小结	220
第 9 章 JSP 核心 API	221
9.1 内部对象	221
9.1.1 PageContext	221
9.1.2 JspWriter	227
9.1.3 一个实现实例	233
9.2 Exceptions	234
9.2.1 JspException	234
9.2.2 JspTagException	235
9.3 JSP 页面实现对象与容器的联系	236
9.3.1 JspPage	236
9.3.2 HttpJspPage	237
9.3.3 JspFactory	238
9.3.4 JspEngineInfo	239
9.4 计数器	240
9.4.1 计数器的实现	240
9.5 本章小结	242
第 10 章 JSP 对 JDBC 的集成	243
10.1 关系数据库标准语言 SQL	243
10.1.1 SQL 概述	243

10.1.2	数据定义	244
10.1.3	数据查询	247
10.1.4	数据更新	252
10.2	JDBC 概述	255
10.2.1	JDBC 是什么	255
10.2.2	为什么有 JDBC	255
10.2.3	JDBC 与 ODBC 的比较	256
10.2.4	JDBC 的功能模型	257
10.2.5	JDBC 驱动程序的类型	258
10.3	Tutorial	259
10.3.1	建立数据源	259
10.3.2	加载驱动程序	260
10.3.3	建立连接	260
10.3.4	建立语句对象	260
10.3.5	添加数据到数据库	261
10.3.6	获取结果集合	264
10.3.7	数据处理	264
10.3.8	获得元数据	265
10.3.9	将处理结果写回数据库	267
10.3.10	关闭对象	269
10.3.11	处理异常和警告	270
10.4	建立数据源	271
10.5	Driver	273
10.6	DriverManager 和 DataSource	276
10.6.1	DriverManager 类	276
10.6.2	DataSource	278
10.7	Connection 和 PooledConnection	281
10.7.1	Connection	281
10.7.2	PooledConnection	284
10.8	Statement, PreparedStatement 和 CallableStatement	290
10.8.1	Statement	290
10.8.2	PreparedStatement	295
10.8.3	CallableStatement	301
10.9	ResultSet	308
10.10	ResultSetMetaData	321
10.11	SQLException 和 SQLWarning	323
10.11.1	SQLException	323
10.11.2	SQLWarning	325

10.12 新闻	327
10.12.1 新闻显示	327
10.12.2 新闻发布	331
10.13 本章小结	333
第 11 章 JSP 对 JavaBeans 的集成	334
11.1 JavaBeans 概述	334
11.1.1 JavaBeans 的属性	335
11.1.2 JavaBeans 的方法	335
11.1.3 JavaBeans 的事件	336
11.2 在 JSP 页面中使用 JavaBeans	336
11.2.1 <jsp:useBean... /> 标记	336
11.2.2 <jsp:setProperty> 标记	337
11.2.3 <jsp:getProperty> 标记	339
11.2.4 编写自己的 Bean	345
11.2.5 通用数据库 Bean	345
11.2.6 购物车 Bean	347
11.3 本章小结	350
第 12 章 JSP 对 Applet 的集成	351
12.1 <jsp:plugin> 行为	351
12.2 时钟、日期 Applet 的实现	353
12.2.1 时钟 Applet	353
12.2.2 日期 Applet	356
12.3 本章小结	358
第 13 章 JSP 对 XML 的集成	359
13.1 JSP 页面的 XML 语法	359
13.1.1 XML 的几个基本概念	359
13.1.2 JSP 页面的 XML 语法	360
13.1.3 实例	364
13.2 标记扩展	365
13.2.1 taglib Directive	365
13.2.2 标记库描述器及其格式	366
13.3 本章小结	367
第 14 章 网站建设	368
14.1 BBS	368

14.1.2	注册	370
14.1.3	登录	375
14.1.4	查询	378
14.1.5	版面显示	383
14.1.6	文章标题显示	385
14.1.7	文章显示	387
14.2	电子商务	389
14.2.1	首页	389
14.2.2	进入书屋	395
14.2.3	购物	396
14.2.4	购物车显示	406
14.2.5	退货系统——部分退回	410
14.2.6	全部退回	412
14.2.7	结帐系统	414
14.2.8	数据库更新	419
14.3	聊天室	421
14.3.1	聊客信息管理	421
14.3.2	聊天室的实现	431
14.4	本章小结	442
附录 1	Tomcat 安装汇总	443
1.1	资源下载	443
1.2	Tomcat 在 Windows NT 4.0, Windows 2000 下的安装	443
1.3	Tomcat 在 Redhat 下的安装并与 apache 相连	444
1.4	Tomcat 在 Unix 下的安装	445
附录 2	Servlet API——javax.servlet 包	447
2.1	javax.servlet Class GenericServlet	447
2.2	javax.servlet Interface RequestDispatcher	448
2.3	javax.servlet Interface Servlet	450
2.4	javax.servlet Interface ServletConfig	452
2.5	javax.servlet Interface ServletContext	453
2.6	javax.servlet Class ServletException	460
2.7	javax.servlet Class ServletInputStream	461
2.8	javax.servlet Class ServletOutputStream	462
2.9	javax.servlet Interface ServletRequest	463
2.10	javax.servlet Interface ServletResponse	469
2.11	javax.servlet Interface SingleThreadModel	473

2.12	javax.servlet Class UnavailableException	474
附录 3	javax.servlet.http 包	476
3.1	javax.servlet.http Class Cookie	476
3.2	javax.servlet.http Class HttpServlet	480
3.3	javax.servlet.http Interface HttpServletRequest	486
3.4	javax.servlet.http Interface HttpServletResponse	493
3.5	javax.servlet.http Interface HttpSession	502
3.6	javax.servlet.http Class HttpSessionBindingEvent	507
3.7	javax.servlet.http Interface HttpSessionBindingListener	508
3.8	javax.servlet.http Interface HttpSessionContext	508
3.9	javax.servlet.http Class HttpUtils	509

第1章 JSP 起源、构建 JSP 环境

1.1 JSP 起源

1.1.1 JSP 的含义

JSP 是“Java Server Pages”的缩写,直译过来是“基于 Java 的服务端动态页面技术”。这是 Sun 公司近年来的力作。自 1990 年开始设计 Java,其后推出、应用、推广到现今不过 10 年光景,Java 已经获得极其崇高的地位,Java 被称为一种划时代的程序设计语言便是最有力的证明。事实上,Java 早已以其简单、面向对象、平台无关等特性征服了世界。在 Internet 应用上,更是如日中天。Sun 也不断推出基于 Java 的新技术巩固其地位与优势,先后推出了 Java Applet、JavaScript、Servlet 等技术及其产品。JSP 技术正是这种战略的延伸,是 Sun 保持 Java 地位与优势的需要。基于 Java 是 JSP 一切优秀品质的根本物质基础。因为 Java 是编译(在 JSP 中称翻译更恰当)解释执行的,所以 JSP 也就摆脱了 VBScript、Perl 等脚本语言纯粹的解释执行带来的低效率问题。尽管在 JSP 中 Java 仍被称为脚本语言。因为 Java 是平台无关的,所以 JSP 是跨平台的。基于 Java 是 JSP 区别 ASP、PHP 等服务端动态页面技术的重要特征。JSP 是服务端技术,所以 JSP 文档要先经服务端翻译、解释、执行,才能得到客户端浏览器能识别的 HTML 文档。这是当前 Web 技术的潮流,是解决客户端浏览器兼容性问题的客观需要。服务端技术是 JSP 区别于 JavaScript、VBScript 等客户端动态页面技术的重要特征。JSP 是一种页面技术,而现在 JSP 只实现了 HTTP 协议,只能应用于 Web 页面。这是 JSP 区别于 Servlet 的重要特征。JSP 是一种动态技术,这是 JSP 基于 Java 的自然延伸。我们为什么在页面中引入程序设计语言,为的就是实现动态交互,这是我们孜孜以求的目标。因此,JSP 就是基于 Java 的服务端动态页面技术。

1.1.2 JSP 起源与 Web 发展趋势

JSP,诞生于名门望族 Sun 家中。说起其身世,就不得不说说江湖中的一段恩怨——“嫁衣”(浏览器标准)之争。十年前,Web 诞生。当时这个小姑娘样子还是相当难看的,只有头和脚(C/S 两层架构),而且目光呆滞(静态),粗布白衣(文本)。即便是这样,她仍然获得了不少人的青睐,很快就鹤立鸡群,其他的前辈、兄长只有艳羡的份。小姑娘目光呆滞,这可是大问题。想想吧,跟她打招呼,她却不理睬你,于是就有了 CGI 脚本。还有粗布白衣,日久生厌。怎么办呢?到了 1993 年,小姑娘有了第一件漂亮的衣裳——图形

用户界面的 Web 浏览器,一下子吸引了全球人的目光,也吸引了各路英雄豪杰(IT 厂商)。小姑娘长大了,花(标准)落谁家呢?当然要看谁做的“嫁衣”功能更强大,外表(界面)更漂亮(友好)。各路英雄豪杰为争夺“嫁衣”,文争武斗,搅得江湖浩劫丛生。先是 Netscape 凭 Navigator 独领风骚两三年。Navigator 功能强大,支持更多的 HTML 扩展标记,成为当时事实上的标准。然而随着 Microsoft 的强势杀人,其 IE 支持比 Navigator 更多、更好的扩展 HTML 标记,与 Netscape 分庭抗争。此时,二者的 HTML 部分标记是不兼容的。Netscape 深感压力,联合了 Sun,Java 阵营渐成。Sun 大有来头,Java 就是其镇家之宝,当年也是春风得意。二者将 Sun 的独门武功 Java 集成于 Navigator 之中,使其支持 Java Applet。此时浏览器有了一定的交互能力,界面也更友好。Microsoft 不愧一代枭雄,独力对抗 Java 阵营亦略占上风。不久便力推 XML 以制衡 Sun 的技术优势。Microsoft 原想利用 XML 这门绝世武功将 Sun 打发出局,然而意想不到的是这两门武功并不相生相克,反而相辅相成。二者没分出高下,自是不会善罢甘休。不久,Sun、Netscape 宣称自己的浏览器支持 JavaScript。而此时 Microsoft 也没有闲着,他想要使本门的武功发扬光大,必须发展创新。那么哪门武功能够和值得去创新呢?很快 VB 进入了 Microsoft 的视线,VB 是解释型的语言,非常适合网络上灵活多变的请求,于是全情投入 VBScript 的修炼。寒暑易节,江湖风传 JavaScript 如何厉害,让 Microsoft 大为震惊,一边加快了 VBScript 的修炼,一边找来 JavaScript 研究,这样,到 VBScript 大功告成时,JavaScript 功力亦相当深厚。二者明争暗斗,你来我往,互有攻守。一般情况下,Sun 总能在技术上快 Microsoft 半分,但 Microsoft 总能将界面做得更友好,因此,二者大比武总是不能决出胜负。这可苦了芸芸众生,由于二者不和,我等只好今天学这个,明天又学那个,谁解其中味! Microsoft 和 Sun 似乎都意识到这点,同时也看到 C/S 两层架构的弊端,二者各自根据自己的武功套路,分别演化出 ASP 和 Servlets。同样二者互有优劣,ASP 更简易,Servlet 功能强大但较复杂,Microsoft 以亲民政策略占上风。Sun 不久就意识到问题之所在,毕其功于一役,推出 JSP。JSP 支持绝大多数服务器,同时改善了人机界面,生成动态网页更简易,大有一统江湖之势,不知这次 Microsoft 以何为对?

上文“戏说”,只是想让读者了解这段发展历史,从中感受发展趋势。事实上,下面的推论也并不需要了解太多史实细节。推论如下:

(1) 当今时代,谁也不能在技术上领先对手很长时间。在 IT 业,6~18 个月是一个比较客观的平均技术代时间。谁落后对手太长时间,通常他将会被淘汰。这与几千年前是不同的,想想我们的四大发明领先其它国家多长时间。主要原因是现在与以前的资讯传播方式不一样了。

(2) 既然谁也不能领先,那么其它方面的因素,如人机界面等将会起到决定性的作用。在人机界面方面,Sun 有很大改进,但仍需加强。当您为运行 JSP、Servlet 配置环境的时候,也许就会有些感受。当您直接利用服务器调试程序的时候,您的感受可能会更深一些。

(3) 竞争从最初的浏览器之争全面升级为当今先进技术综合应用的竞争,从可视化(Visual)到组件技术(COM)再到 C/S 体系架构。当今先进技术无一不在互联网上一展身手,这也表明了互联网的勃勃生机,预示着互联网是当今最有前途的技术之一。

(4) 对同一种技术,有很多实现手段(方法)。如客户端动态页面技术,Sun 以 JavaS-

cript 实现,Microsoft 以 VBScript 实现。又如服务端动态页面技术,Sun 以 Servlets、JSP 实现,Microsoft 以 ASP 实现。当然还有其它 CGI 技术,如 PHP。

(5) 虽然有多种实现手段,但它们是有点共同的。对客户端动态页面技术,Sun 和 Microsoft 都使用了脚本语言,即解释型的语言。对服务端动态页面技术,都采用 C/S 三层体系架构,在底层都封装(Sun 使用 JDBC,Microsoft 使用 ODBC)了数据源,在中间层都使用了组件技术(Sun 使用 JavaBeans,Microsoft 使用 COM)封装了实现逻辑,以及提供事务处理环境以支持高伸缩性、分布式事务处理,等等,不一而足。“共同”就意味着“必然”。

(6) 上一点的推论,共同是有时间烙印的。时代决定了可用的资源,当前技术不能超出可用的资源。

(7) 因为有多种实现手段,所以通常互不兼容,但又兼收并蓄。例如,微软的 IE 既支持 VBScript 又支持 JavaScript(JScript)。

(8) 解决兼容性问题推动当前浏览器乃至其它技术的重要动力。

我们从更高角度看,抛开二者的商业竞争因素、技术上的个体差异,事实上,每一项新技术的应用,都有一定的目的,即都要解决一个矛盾。在 Web 产业中,有三个基本的矛盾:一是性能(功能)问题,二是界面问题(简易性),三是兼容性问题。

Web 发展至今,请容许我们大胆的把它分为三个阶段。第一阶段是二层架构服务端页面技术,目标是解决基本的交互问题,特征是使用的人比较少,界面不够友好,但有能够满足需要的交互能力;第二阶段是二层架构客户端动态页面技术,目标是解决界面不够友好的问题,特征是使用的人增多,客观上要求界面更加友好;第三阶段是三层架构服务端动态页面技术。第一、二阶段的兼容性问题暴露出来了,这时兼容性上升为主要矛盾。

1.1.3 JSP 的设计目标

JSP 技术让动态网页更易编写,功能更强,可移植性更好。概括起来,JSP 的设计目标主要有以下几点:

1. 一次编写,处处运行

JSP 技术是完全的与平台无关的设计,包含它的动态网页与底层的服务组件设计。可以在任何平台下编写 JSP 网页并且在任何支持 JSP 的系统上执行。

也可建立自己的组件,并在 JSP 中使用。目前主要是 JavaBeans 和 Java Servlet,而它们都是跨平台的。

2. 加强组件能力

JSP 技术以 Java 的组件模型 JavaBeans 加强了组件的使用能力。

这不但省去了开发的时间,而且还可以将网页页面设计和商业逻辑分开,有助于快速开发和简单维护。

3. 作为 Java 企业平台的门户

JSP 高度整合了 Java 企业平台部分(Java 专注于企业应用方面的平台技术,如 JDBC、