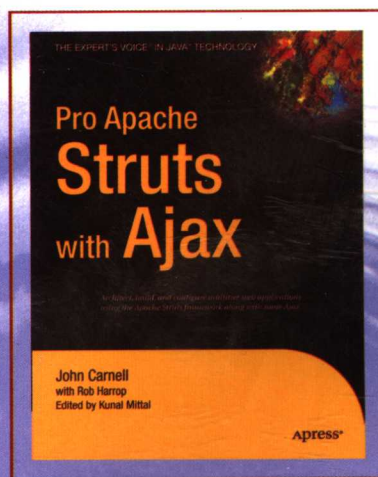


Pro Apache Struts with Ajax

Struts 与 Ajax 高级程序设计

[美] John Carnell 著
Rob Harrop
Kunal Mittal
龚波 徐雅丽 等译

- 经典 Struts 著作，涵盖 Ajax
- 深入剖析 Web 编程思想和 Struts 各种高级主题
- 注重实践，提升你解决实际问题的能力
- *Pro Spring* 作者新著



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Java系列

Struts与Ajax高级程序设计

Pro Apache Struts with Ajax

[美] John Carnell
Rob Harrop 著
Kunal Mittal

龚波 徐雅丽 等译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

Struts 与 Ajax 高级程序设计 / (美) 卡内尔 (Carnell, J.), (美) 哈罗普 (Harrop, R.), (美) 米托 (Mittal, K.) 著; 龚波等译. —北京: 人民邮电出版社, 2008.2

(图灵程序设计丛书)

ISBN 978-7-115-17227-3

I. S… II. ①卡…②哈…③米…④龚… III. ①软件工具—程序设计②计算机网络—程序设计 IV. TP311.56 TP393.09

中国版本图书馆 CIP 数据核字 (2007) 第 179497 号

内 容 提 要

本书详细介绍了 Struts MVC Web 框架的使用方法, 可以帮助读者解决在日常 Web 应用程序开发中遇到的各种难题。不合理的设计会导致对 Struts 框架的长期依赖, 这会使代码复用变得非常复杂。本书基于 Struts 1.2 版对此类问题给出了专业的解决方案。本书的重点在于如何发现常见的设计错误 (反模式) 以及如何使用 Struts 来解决这些问题。

本书内容丰富, 信息全面, 主要内容包括模式和反模式、Struts 基础及其高级知识、Tiles 框架、日志和调试、velocity 模板引擎、Struts 和 Ajax 等。通过本书, 读者不仅能够深入地了解 Struts 的工作原理, 而且还能掌握处理日常开发中遇到的问题的有效方法。

本书适合于所有具备一定开发经验并掌握 JSP、servlet 以及 Struts 基础知识的高级读者阅读。

图灵程序设计丛书

Struts 与 Ajax 高级程序设计

-
- ◆ 著 [美] John Carnell Rob Harrop Kunal Mittal
 - 译 龚 波 徐雅丽 等
 - 责任编辑 傅志红 杨福川
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子函件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - 新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
 - 印张: 26.5
 - 字数: 626 千字 2008 年 2 月第 1 版
 - 印数: 1—4 000 册 2008 年 2 月河北第 1 次印刷
 - 著作权合同登记号 图字: 01-2007-2469 号

ISBN 978-7-115-17227-3/TP

定价: 59.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Pro Apache Struts with Ajax* by John Carnell, Rob Harrop, Kunal Mittal, published by Apress L.P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA.

Copyright © 2006 by John Carnell, Rob Harrop, Kunal Mittal. Simplified Chinese-language edition copyright © 2008 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P.授权人民邮电出版社独家出版。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。

致我的妻子 Janet: 谢谢你在我编写本书 (以及以前我编著其他图书) 期间为我付出的爱、耐心和时间。如果没有你的爱和智慧, 我的生活将一片黑暗! 你是我的一切!

致我的儿子 Christopher: 我的所有经历都无法与我第一次把你抱在怀中的感觉相比。每天都沉醉在你的奇迹中!

——John Carnell

谨以本书献给我的高中英语老师 Neville McGraw, 感谢您激发了我对文学持久的兴趣, 并教我明白把想法诉诸笔端的重要性。

——Rob Harrop

译者序

Struts 是 Apache Jakarta 项目的组成部分，目标是为 Java Web 应用程序提供一个开源框架，实现应用程序逻辑的分离，最终使应用程序具有更好的可扩展性和可维护性。Ajax 技术是这两年的热门技术之一，代表未来 Web 应用程序开发的发展方向。Ajax 致力于改善用户体验，也就是说，它是一种表现层技术。

本书的作者 John Carnell 与 Rob Harrop 对 Struts 非常熟悉，多年来一直参与 Apache 项目，而且两位的文笔不错，著作颇丰。

本书是一本有关 Struts 框架编程的优秀图书，倾向于使用案例来介绍原理，并在介绍原理的过程中引入更多参考资源，由浅入深，娓娓道来。本书并没有一味进行理论说教，作者希望读者能够掌握 Struts 的精髓和原理，指导开发者纠正以前的错误，避免未来出现类似的问题，这也是本书在第 1 章以及其后每个章节都会多次提到反模式的原因。

我们非常荣幸能够参与本书的翻译工作，并向国内读者引荐这本优秀图书。在本书的翻译过程中，我们竭尽全力，唯恐辜负广大读者的期望，经常为了一个单词或一个句子，要斟酌好多天，要上网查询很多资料，要咨询很多人。不过，为了能让本书的中文版面世，再苦再累也不算什么，也能对那些给我们提供帮助的人有所交代。

本书由龚波和徐雅丽主译，具体分工如下：龚波翻译前言和第 1 章，以及全书的审校；徐雅丽翻译第 2 章～第 7 章；葛丽翻译第 8 章；田振中翻译第 9 章；龚志翔翻译第 10 章；王高翔翻译第 11 章；白红利翻译第 12 章；田丽韞翻译附录。李志和罗贤锋负责全书的排版；田野、牛献忠、张巧莉、王冬冬等对本书的翻译工作提供技术支持，并帮助处理翻译文本的审校工作，在此对他们表示诚挚感谢。

译者

2007 年于北京

致 谢

当人们谈论一本书时，经常只会考虑作者为编写本书所付出的辛勤劳动。但是，实际上任何图书的出版都是团队努力的结果。首先，我非常感谢 Gary Cornell，他给予了我充分的信任，并建议我继续编写本书的第 2 版。以他的工作背景和丰富经验，他的信任对我是莫大的鼓励。

我还要感谢下面的一些人：

- Beth Christmas——Apress 的项目编辑，她始终不知疲倦地推进本书的进度。
- Ami Knox——本书的文字编辑，她对细节非常关注，使我成为一位智慧、规范的作者。
谢谢你，Ami！
- Jan Machacek——本书的技术编辑。你深入审校本书，使我严谨、求实，使本书卓尔不群。
- Rob Harrop——本书的合著者。Rob 对本书的编写给予了很多支持，他的深入分析和辛勤劳动让我非常感激！

John Carnell

出版一本书，并不像表面看起来那么容易，这一点很多人没有意识到。首先，我非常感谢本书的合著者 John Carnell，他的沟通能力非常强，甚至能够让菜鸟明白最难理解的话题。同样，要感谢本书的技术审校者和我的同事 Jan Machacek，他毫无疑问是当今世界上最好的 Struts 程序员之一。多谢 Apress 的每个人，尤其是 Beth Christmas 和 Ami Knox。如果没有这个伟大团队的背后支持，完成这本书绝对是白日做梦。最后，我还要感谢我的女朋友 Sally，感谢她在我编写本书的漫长日子中，每天晚上陪我坐在计算机旁，倾听“枯燥”的 Struts 故事。

Rob Harrop

我非常感谢 John、Rob 和整个 Apress 团队，感谢他们把编辑本书的机会交给我。在本书的完成过程中，Steve、Elizabeth、Lori、Bill 和其他很多人都是幕后英雄。真正的功劳应该属于他们！我还要感谢我的妻子 Neeta，以及可爱的小狗 Dusty 和 Snowie，在我伏案工作的那些周末，他们给予了我充分的理解。

Kunal Mittal

本版前言

即使面临着 JavaServer Faces (JSF)、Spring MVC、WebWork、Wicket 以及其他 API 和框架的挑战, Apache Struts 1.2.x 仍旧是基于 MVC 的 Java Web 框架的事实工业标准。

从本质上来讲, 本书是以前出版的 *Pro Jakarta Struts, Second Edition* 的修订版本, 关注开源的 Apache Struts MVC Web 框架的变化, 主要包括:

- 本书的 Struts Web 框架基于最新的 Struts 1.2.x。
- 本书介绍了 Struts 在 Apache 软件基金会内从 Jakarta 到 Apache 的进化过程。
- 本书新增了一章, 讲解如何集成 Ajax (Asynchronous JavaScript and XML) 和 Apache Struts。

尽管本书讲解了如上的内容, 但是并没有涉及正在快速发展的 Apache Shale 和 Struts 2.0, 也就是常说的 Struts Action Framework 2.0, 后者结合了 Struts 2.0 和 WebWork。当然, Apress 后续的图书会讲解这些内容。

本版编辑

前 版 前 言

(*Pro Jakarta Struts*, 第 2 版)

当人们发现我写过书的时候，经常问我的一个问题就是“你为什么愿意写作？”尽管这是一个简单得不能再简单的问题，但其答案却并不是那么清晰和简单。

如果必须使用一句话来概括我编写本书的理由，那就是我热爱技术，喜欢用技术来实现期望的东西。我从 12 岁就开始编程。我使用过很多种技术，最近四年来我有机会使用多个开源项目来开发企业级软件。

我对这些技术的质量和功能印象很深。我最喜欢的开源技术之一是 Apache Group 的 Struts 开发框架。Struts 框架是一个非常强大的开发框架，允许 Java Web 开发者将注意力集中在应用程序创建上，而不必太关注基础架构。

当着手编写本书的第 1 版时，我心中有两个目标。首先，我希望本书能够向读者介绍 Struts 开发框架，但是不会向他们灌输 Struts 应用程序开发的编程细节。就我个人看来，对于大多数人，甚至是高级开发者来说，最好是从动手实践中学习，而不是靠阅读大量细节来学。

其次，我希望读者能够了解如何使用 Struts 解决自己编写 Web 应用程序时碰到的很多问题。因此，本书着重讨论了如何发现常见的设计错误（也称为反模式）和如何使用 Struts 解决这些问题。

但是，本书始终坚持一个核心原则：框架无法完全代替开发者设计应用程序。像很多其他框架一样，Struts 框架是个工具，不正当使用反而会带来麻烦。因此本书强调良好设计的重要性，即使使用 Struts 框架也要重视它。良好的代码是从来不会意外地产生的，事先必须经过周密和清晰的设计，才可能实现。

本书的目标读者群是中高级开发者。本书中提供的应用程序范例非常简单，任何具备 JSP 和 Servlet 基础知识的读者都可以很容易地理解。但是，本书的作者希望读者时刻注意，任何设计决策和设计模式对于应用程序长期的可扩展性都会产生重要影响。

在本书的第2版中，我们根据Struts 1.1更新了所有资料，并使用单独的几章来介绍Struts 1.1新增的特性（如 Tiles 和 Validator 框架）。除此之外，还讨论了其他开源技术，如 ObjectRelationalBridge、Lucene和Velocity，将这些开源技术和Struts结合起来使用，可以大大减少实现应用程序功能所需的时间和工作量。

最后，我想本书决不是只让读者听我单方面讲故事。相反，本书是双向的交流互动，这种互动自从我第一次接触 Commodore 64 就已经开始。因此，我非常希望得到读者的反馈（不管是正面的，还是页面的），你若有问题、意见，或只是想发泄，请与我联系，我的邮件地址是 john.carnell@netchange.us。我希望你能够喜欢本书，同时我期望听到你的声音。

John Carnell

目 录

第 1 章 我们做错了什么: Web 反模式解释..... 1

- 1.1 本书主要内容 2
- 1.2 本章主要内容 3
- 1.3 Web 应用程序开发面临的挑战 4
 - 1.3.1 企业服务 5
 - 1.3.2 应用服务 6
- 1.4 模式和反模式介绍 7
 - 1.4.1 Web 应用程序反模式 8
 - 1.4.2 Concern Slush 9
 - 1.4.3 Tier Leakage 11
 - 1.4.4 Hardwired 13
 - 1.4.5 Validation Confusion 15
 - 1.4.6 Tight-Skins 17
 - 1.4.7 Data Madness 18
- 1.5 反模式、JOS 框架以及经济学 20
- 1.6 小结 24

第 2 章 Struts 基础..... 25

- 2.1 JavaEdge 应用程序架构 26
 - 2.1.1 设计 26
 - 2.1.2 使用 Struts 实现 MVC 模式 28
 - 2.1.3 起步: JavaEdge 源代码树 30
 - 2.1.4 Command 模式的作用 40
- 2.2 构造表示层 44
 - 2.2.1 JavaEdge 主页 45
 - 2.2.2 Bean 标签 48
 - 2.2.3 逻辑标签 57
 - 2.2.4 迭代式标签 58
 - 2.2.5 条件式标签 58
 - 2.2.6 移动式标签 61
- 2.3 小结 62

第 3 章 使用 Struts 的表单表示和验证机制..... 64

- 3.1 表单验证的问题 64
- 3.2 使用 Struts 进行表单验证 65
- 3.3 使用 Struts 实现表单验证 67
 - 3.3.1 struts-config.xml 文件 68
 - 3.3.2 Struts ActionForm 类 69
 - 3.3.3 使用数据预先填充 ActionForm 75
 - 3.3.4 其他预先填充技术 78
 - 3.3.5 正确地预先填充表单 79
 - 3.3.6 验证表单数据 80
- 3.4 Struts HTML 标签库 88
 - 3.4.1 建立 Struts HTML 表单 90
 - 3.4.2 使用文本和文本域输入字段 92
 - 3.4.3 下拉列表、复选框和单选框 93
 - 3.4.4 创建更加动态的 ActionForm 95
 - 3.4.5 ActionForm 和业务逻辑 101
- 3.5 小结 106

第 4 章 使用 Struts 管理业务逻辑..... 107

- 4.1 业务逻辑反模式和 Struts 107
 - 4.1.1 Concern Slush 和 Struts 108
 - 4.1.2 Tier Leakage 和 Struts 113
 - 4.1.3 分离业务逻辑和 Struts 115
- 4.2 实现设计模式 117
 - 4.2.1 实现 Business Delegate 模式 117
 - 4.2.2 实现 Service Locator 模式 122
 - 4.2.3 继续研究服务定位器 130
 - 4.2.4 EJB 和 Struts 134
- 4.3 处理 Action 类中的异常 140
 - 4.3.1 Struts 1.0.x 的异常处理 141

4.3.2 Struts 1.1 和以后版本的异常处理	143	6.4.5 使用 Tiles 定义的继承性	209
4.3.3 重新编写 ApplicationException 类	143	6.4.6 扩展 Tiles 定义	212
4.3.4 设置 struts-config.xml 文件	144	6.4.7 修改 template.jsp 文件	213
4.3.5 编写定制的 ExceptionHandler	146	6.4.8 给 tiles-defs.xml 文件添加一个新的定义	214
4.4 小结	149	6.4.9 修改 .homePage 定义	215
第 5 章 使用 ObjectRelational Bridge 实现数据访问层的架构	151	6.5 把 Tiles 定义映射到 Action Forwards	216
5.1 开发数据访问策略	152	6.6 小结	219
5.1.1 JavaEdge 应用程序的数据访问模型	153	第 7 章 动态表单和 Struts 验证框架	221
5.1.2 值对象	158	7.1 动态表单简介	222
5.1.3 JavaEdge 值对象	159	7.1.1 定义 postStoryForm Struts 表单 bean	222
5.1.4 使用 O/R 映射工具	162	7.1.2 实现 postStoryDynaForm.java 类	223
5.1.5 设置对象/关系映射	165	7.1.3 关于 BeanUtils 和上述代码的思考	228
5.1.6 使用哪个序列管理器	174	7.2 Jakarta Commons Validator 框架	231
5.2 使用 OJB	180	7.2.1 Validator 框架设置	231
5.2.1 数据检索: 一个简单的例子	185	7.2.2 实现必填字段验证	232
5.2.2 数据检索: 一个更复杂的范例	187	7.2.3 maxlength 验证规则	235
5.2.3 使用 OJB 存储数据	189	7.3 在 ActionForm 类中使用 Validator 框架	237
5.2.4 使用 OJB 删除数据	190	7.3.1 编写自己的验证规则	239
5.3 集成前面的代码	191	7.3.2 实现粗俗性规则	239
5.4 小结	194	7.3.3 向 validator-rules.xml 文件添加粗俗性检查规则	243
第 6 章 使用 Tiles 框架创建灵活的前端应用	196	7.3.4 Struts 验证和潜在的长期后果	245
6.1 Tiles 框架是什么	197	7.3.5 在表单中实现粗俗性规则	246
6.2 启用 Struts 1.1 版本以使用 Tiles	198	7.3.6 不使用 Java 的 ActionForm	247
6.2.1 配置 Tiles 插件	199	7.3.7 何时使用 Validator 框架	247
6.2.2 tiles-defs.xml 文件	201	7.4 小结	248
6.2.3 添加 Tiles TLD	201	第 8 章 使用 XDoclet 加速 Struts 开发	250
6.2.4 第一个 Tiles 模板	201	8.1 安装 XDoclet	251
6.3 Tiles 定义是什么	204	8.2 XDoclet 到底是什么	252
6.4 Tiles 定义: 基于 JSP 的方法	205	8.2.1 从 XDoclet 到源代码及其之间的步骤	255
6.4.1 覆写 Tiles 定义的属性值	207	8.2.2 可用的 XDoclet 标签	256
6.4.2 在 Tiles 定义中使用哑值	207	8.2.3 分析一个 XDoclet 标签	257
6.4.3 JSP Tiles 定义的劣势	208		
6.4.4 分析 tiles-defs.xml 文件	209		

8.3 集成 Ant 和 XDoclet	258	10.2.3 变量	313
8.4 XDoclet 和 Struts	262	10.2.4 访问变量	313
8.4.1 声明 Struts 表单 bean	262	10.2.5 变量值	313
8.4.2 声明 Struts 动作	263	10.2.6 JavaBean 属性	318
8.4.3 XDoclet 和 Java 继承	264	10.2.7 算术	318
8.4.4 声明应用程序异常	264	10.2.8 指令	319
8.4.5 使用<webdoclet.../>构建 struts- config.xml 文件	266	10.2.9 宏	327
8.4.6 XDoclet 和 Validator 框架	267	10.2.10 Struts 和 Velocity	328
8.4.7 从 Ant 生成 Validator 标签	270	10.3 使用 Velocity 的最佳实践	335
8.5 小结	272	10.3.1 使用宏	335
第 9 章 日志和调试	274	10.3.2 知道何时使用#parse 指令, 何时使用#include 指令	335
9.1 为什么使用日志	275	10.3.3 使用 JavaBean 属性名称	335
9.2 简单的 Web 应用程序日志	276	10.4 小结	335
9.2.1 ServletContext 提供的日志	276	第 11 章 扩展 Struts 框架	336
9.2.2 使用 Commons Logging	277	11.1 扩展 Action 和 ActionForm	336
9.3 Java 1.4 日志 API	281	11.1.1 向自己的动作提供通用服务	337
9.3.1 Apache log4j	284	11.1.2 干涉动作执行	341
9.3.2 使用 log4j 和 Commons Logging	285	11.2 扩展 RequestProcessor	343
9.3.3 日志继承	289	11.2.1 创建 RequestProcessor	343
9.3.4 日志性能	292	11.2.2 使用 RequestProcessor, 而不 使用 Filter	344
9.3.5 日志最佳实践	292	11.2.3 使用 RequestProcessor 验证 本地访问	347
9.3.6 JBoss 和 log4j	296	11.3 创建配置 Bean	347
9.3.7 在 JavaEdge 应用程序中集成 日志	297	11.4 再次研究 RequestProcessor	349
9.3.8 Web 层的日志	303	11.5 创建一个插件	353
9.3.9 使用 JBoss 和 Eclipse 调试 Struts 应用程序	305	11.5.1 新闻列表服务基础	353
9.3.10 调试 JavaEdge 应用程序	306	11.5.2 NewsletterManager	353
9.3.11 快速部署能力	307	11.5.3 NewsletterTask	357
9.3.12 调试 Struts 框架	308	11.5.4 NewsletterPlugIn	358
9.4 小结	308	11.5.5 配置插件	361
第 10 章 Velocity 模板引擎	309	11.6 小结	362
10.1 模板引擎是什么	309	第 12 章 Struts 和 Ajax	363
10.2 开始行动	310	12.1 Ajax 解密	363
10.2.1 Velocity 类和 VelocityContext 类	311	12.1.1 在 Google 中使用 Ajax	363
10.2.2 Velocity 引擎语言	312	12.1.2 在 Yahoo 中使用 Ajax	364
		12.2 在哪些地方应该使用 Ajax	364

12.2.1 Ajax 和 Web 2.0.....	364	12.4.2 GetCitiesNamesAction.....	370
12.2.2 Ajax 和 SOA	364	12.4.3 CitiesDAO.....	371
12.3 Ajax 内部机制	365	12.5 小结.....	372
12.3.1 Ajax 请求-答复循环	365	附录 A JavaEdge 设置和安装.....	373
12.3.2 XMLHttpRequest 对象	366	附录 B Struts 开发工具.....	380
12.4 Ajax 和 Struts.....	367	附录 C Struts 和 Stracks	407
12.4.1 Cities.jsp 文件	367		

第 1 章

我们做错了什么： Web反模式解释



世界上所有事物都逐渐走向混沌状态。本来有次序的变得没有次序了，原来创建的东西被毁坏了。在物理学领域，这个现象被长期关注，并赋予一个名称“熵”。

定义 《韦氏新世界词典》把熵定义为物质或者系统中无序程度的度量：在封闭系统中，熵不断增大，有效能量不断减少。

熵也是软件开发领域始终关注的一种现象。在很多项目中，最初的程序代码库是整洁和有组织的，符合个人的编码习惯、文档风格、指南和标准，但是随着应用程序的逐渐演进，过不了多久代码库就会变得非常混乱，难以维护。由于时间压力、应用程序升级以及错误修复等原因，你可能降低自己最初设定的标准。

无论是物理学，还是软件开发项目，在原理上，熵及后续的混乱其实是一样的。在软件开发项目中，应用程序及其代码库中的熵值越大，要想满足用户需求或者实现整体商业目标就越难。难以维护或者难以扩展的代码越多，开发者进行软件开发的有效时间就会越短，甚至会无意地修改最初正确的代码，增加出错的风险。

为什么软件开发工作如此容易从有序状态转变到无序状态？其中的原因很多，但是所有的原因都有一个根源——复杂性。其他常见的原因还包括：软件交付的时间压力、需求变更或者需求不清晰，项目管理者或者开发者的不良习惯。

编写应用程序代码的行为实际上是优化某些过程的结构和次序。这些过程可能是很常见的（诸如，判断某人的银行账户中是否有足够的钱来执行所请求的提款操作），也可能是非常复杂的（比如，导弹发射控制系统判断迫近的飞机是民用飞机还是军用飞机）。我们知道这要点儿想象力，不过你大体能想到了。

大多数开发工程师都知道代码所实现的过程是很难清晰定义的。这些过程经常是非线性的，不容易用离散的步骤进行描述。相反地，这些过程经常内含多个决策点，如果实现不当，可能导致完全不同的输出结果。

几乎没有一种软件是一成不变的。软件会经常变化，以满足终端用户不断变化的需求。一般的印象认为应用程序的功能很容易修改，同时不会影响整体的质量和完整性。

软件的非线性特性、始终变化的用户需求以及我们对软件可扩展性的单纯想法使得应用程序的复杂性难以避免。在软件开发项目中，熵和复杂性之间的关系可以总结如下：开发者面临的应用程序复杂性越大，应用程序的熵值会越高。这种复杂性会导致开发者花费在受雇要做的事务（即编写能够解决组织所面临特定问题的软件）上的时间更加少。

无法管理的复杂性会导致软件设计非常糟糕，使其充满各种错误，难以维护，更难以扩展和复用。负责维护代码库的开发团队只能对现有软件进行修修补补，直到源代码完全不可管理。在很多情况下，由于应用程序实现和维护的混乱状态，组织不得不抛弃原来编写的代码，而未曾尽享其利。

这时，所有的问题都牵涉如何保证软件质量，你甚至可能会反思自己涉足软件开发领域是否入错了行^①。实际上，并非所有的项目都是如此。我们中的很多软件开发工程师都已经成功交付能够给组织带来价值的应用程序。

但是，即使对于已经成功实现的应用程序，我们还会经常思考，其实有更好的软件实现方式能够在时间和资金要求的范围内提交质量更高的软件。但是为了实现这个目标，软件需要在更坚实的基础上实现。

编写软件就像盖楼，如果没有良好的计划和体系架构，很快就会被后续的开发压垮。但是，当与很多业务管理者和开发者谈论体系架构时，你从他们的表情中看到的只是痛苦和迷惑。体系架构是软件工程领域中最被滥用的词。

谈到体系结构，很多业务管理者会认为，软件开发团队（通常是成本非常高的团队）把工作时间花费在无用的东西上，不会给业务带来真正的价值。他们看到大量时间和资源被消耗，但是还没有得到很好的投资回报（ROI，Return On Investment）。

对于开发者而言，体系结构一词让他们既内疚又渴望。之所以内疚，是因为很多开发者意识到有更好的方法编写软件；之所以渴望，原因是如果拥有更多时间和资源，他们将能够组织出良好的开发框架，编写出更好的软件。

但是，事实是编写开发框架是项非常艰难的工作，需要高级开发人员花费专门的时间来完成。向组织中的业务管理者量化开发框架的价值更是一个巨大的挑战。

1.1 本书主要内容

本书演示了如何使用可以免费获取的Java Open Source（JOS）开发框架来创建和部署应用程序。我们将着重介绍来自Apache软件基金会（<http://apache.org>）和Jakarta（<http://jakarta.apache.org>）的JOS开发框架。

大多数图书非常强调代码解释，本书提供易实现的代码范例。本书作者希望提供一个路线图，辅助读者使用JOS开发工具来创建自己的应用程序。本书的目的不是详细阐述框架的每个细节，坦白地说，本书涉及的很多开发框架都完全可以用一整本书来介绍。

除XDoclet之外，本书还使用其他Apache技术来创建简单的应用程序，如下所示。

□ Struts Web开发框架：这是一个基于MVC（Model-View-Controller，模型-视图-控制器）

^① 本书的一位作者就曾经发出这样的感慨，只可惜他的法律水平不如计算机水平高。

的开发框架，使得开发者能够以可插拔和可扩展的方式组合应用程序。本书会讨论Struts 1.2框架的部分新特性。

- ❑ Tiles：一种新型的用户界面框架，允许开发团队把界面组件化，分解为多个易于创建和更新的代码段。
- ❑ 动态的ActionForms和Validator框架：一个新型的工具集，可以减轻诸如编写Web数据采集界面等单调工作。
- ❑ Lucene：一个强大的索引和搜索工具，可用于实现任何基于Web的应用的搜索引擎。
- ❑ Jakarta Velocity：一个模板框架，使开发团队可以轻松创建“可换肤的”应用程序，应用程序的“外观”可以很容易地被修改。
- ❑ ObjectRelationalBridge (OBJB)：一个对象/关系映射工具，可以显著地简化开发关系型数据库中数据访问代码的难度。借助于ObjectRelationalBridge，甚至在不必编写任何JDBC代码的前提下，开发团队就可以创建一个完整的应用程序。
- ❑ XDoclet：一个基于元标签的代码生成工具，使开发者不必像往常一样维护过多的J2EE配置文件（web.xml, ejb-jat.xml等）和Struts配置文件（struts-config.xml, validation.xml等）。请注意，XDoclet不是Apache技术，这一点很重要。但是，XDoclet能很好地支持Struts，并且本书会进一步讨论XDoclet技术。
- ❑ Ant：一个业界公认的Java构建工具，使得你可以创建复杂的应用程序和部署脚本。

除此之外，本书还会简要介绍异步JavaScript和XML（Ajax, Asynchronous JavaScript and XML）。Ajax是一种能够解决Web应用程序开发中常见问题的技术。下面使用一个示例来介绍它。^①

假设你拥有一个网站，可以收集客户的信息，诸如姓名、地址、电话号码，等等。你很有可能使用类似于下面的下拉列表域：State、City和Country。假设当客户选择好他们的Country值时，你希望能够根据所选的Country自动刷新State下拉列表域。一旦客户选择好State，你希望自动刷新City下拉列表域。在通常的Web应用程序设计中，上述效果的实现需要与Web服务器进行交互，并且使整个Web页面被刷新。根据这个页面的内容量大小不同，上述自动刷新过程也许会花费几秒钟时间。除此之外，你必须确定在这个阶段会执行哪个验证（很可能不需要，因为用户尚未单击Save按钮）。但是，如果使用Ajax技术，这种操作会在后台自动完成，或者说异步完成，能够避免页面刷新，并且能够提升应用程序的性能。只有所需的信息才会被传递给服务器，当前页面只从服务器接收所需的局部信息，并加载它。

如果你尚未完全理解，请不要担心，在本书最后我们会使用大量篇幅来讲解这个概念。

1.2 本章主要内容

本章不会讨论上面所列技术的详细信息。相反地，我们会讨论创建Web应用程序时所面临的一些挑战，探讨Web应用程序开发中常见的设计错误和缺陷。

应用程序开发的事实是：虽然所有开发者都喜欢从头编写新的应用程序，但是他们工作中多

^① 此处所说的范例也是Tier Leakage反模式的良好范例。——译者注