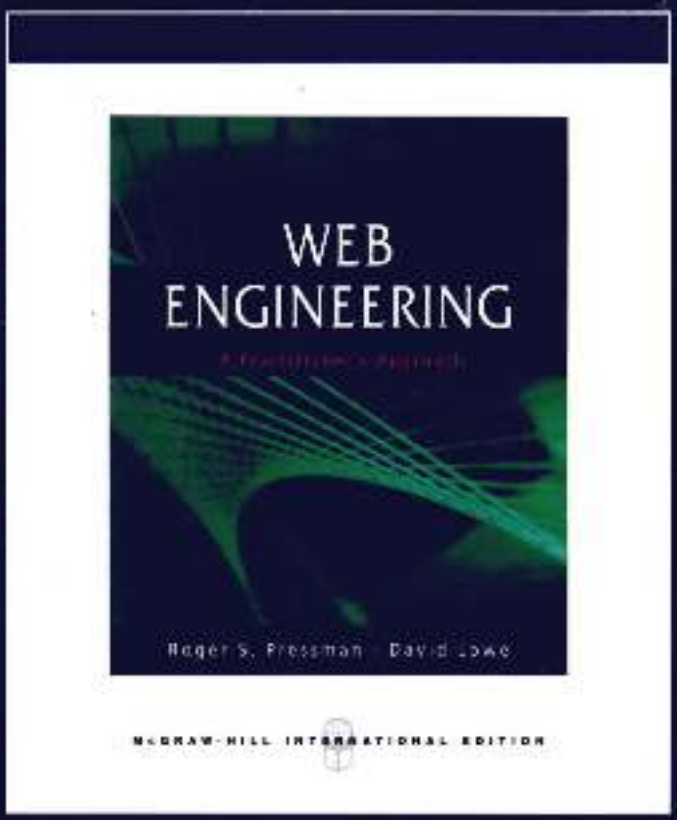
计 算 机 科 学 丛 书

Web工程
实践者的研究方法

Roger S. Pressman David Lowe 著 崔秋艳 闫兵 陈静玉 等译



Web Engineering
A Practitioner's Approach

机械工业出版社
China Machine Press

Web工程 实践者的研究方法

本书是最先讲述构建行业质量的Web应用程序所需要的过程的相关方法和技术的书籍之一。全书采用一个完整的案例，以整个Web应用系统开发的生命周期为主线，全面、系统地阐述了基于Web的工程化系统和应用的实践过程的各个方面，书中提供了大量生动翔实的对话场景，使于加深理解书中所述的理论及其应用。

本书特点

- 以一个完整的应用实例贯穿全书；
- 强调敏捷的过程和系统地使用一些已经在行业应用中证实的简单而实用的方法；
- 平衡概念、工具、技术、应用以及Web工程的各部分等各方面内容所占的比例。

作者简介

Roger S. Pressman



软件过程改善和软件工程技术方面国际知名的权威。30多年来，他作为软件工程师、管理人员、教授、作者及咨询顾问始终致身于软件工程领域。Pressman博士著有8部著作，撰写了很多技术文章，是多种行业期刊的国家编委人，曾生多种行业杂志的编委，并多年来一直担任《IEEE Software》杂志Manager专栏的编辑。Pressman博士是知名的演讲者，曾在许多行业会议上演讲。他还是美国计算机协会(ACM)、美国电气与电子工程师协会(IEEE)等组织的成员。

David Lowe



澳大利亚悉尼科技大学工程学院教学副院长、实时信息网络中心(CRIN)工任。Lowe教授在2001年获得澳大利亚工程教育协会麦格劳-希尔新工程教育家奖。他的研究兴趣包括Web开发和技术、软件工程和实时控制系统等。他著有3本著作，是许多Web会议委员会和杂志编辑委员会成员，并担任《Journal of Web Engineering》杂志主编。

客服热线：(010) 88378991, 88361066
购书热线：(010) 68326294, 88379649, 68995259
投稿热线：(010) 88379604
读者信箱：hzjsj@hzbook.com

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

Mc Graw Hill
www.mhedu.com

ISBN 978-7-111-28250-1

9 787111 282501

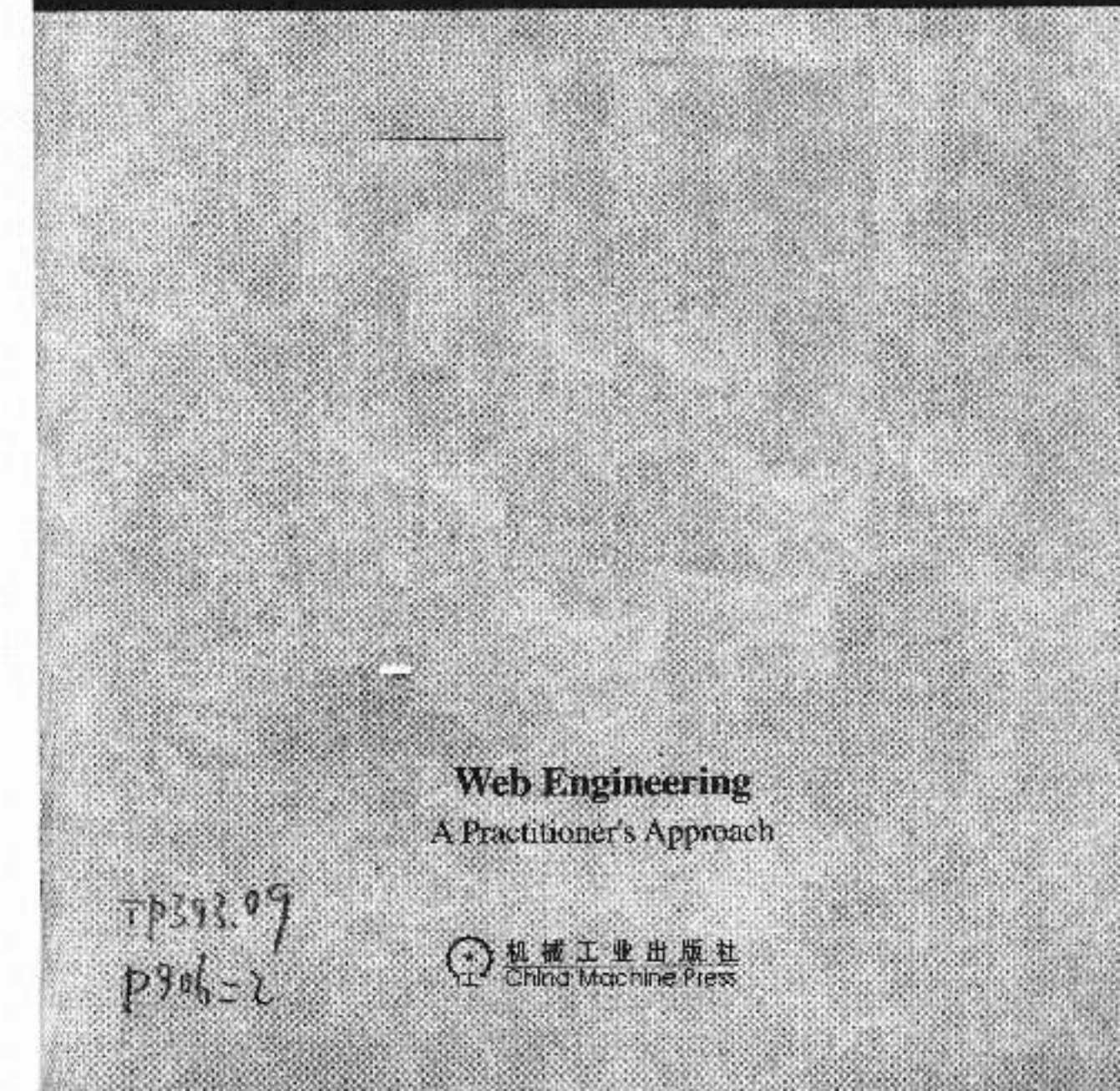
定价：49.00元

TP393.09
P906=2

机 科 学

Web工程
实践者的研究方法

Roger S. Pressman David Lowe 著 霍秋艳 闫兵 陈静玉 等译



TP393.09
P906=2

本书全面、系统地阐述了当前 Web 工程技术的各个方面，概括性地对基于 Web 的系统和应用的工程化实践过程进行了全面的介绍。主要内容包括：Web 工程过程、沟通、计划、建模活动、WebApp 分析建模、交互设计、信息设计、功能设计、构建和部署、设计模式、技术和工具、测试、变更管理和内容管理，以及 WebApp 的发展趋势。本书贯穿了一个应用实例，并以轻松诙趣的方式展现相关内容。

本书可作为计算机相关专业本科生、研究生的教材或参考书，也可供从事 Web 工程的相关人员参考。

Roger S. Pressman and David Lowe: *Web Engineering: A Practitioner's Approach* (ISBN 978-007-126377-1).

Copyright © 2009 by The McGraw-Hill Companies, Inc.

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育（亚洲）出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 McGraw-Hill 公司防伪标签，无标签者不得销售。

版权所有，侵权必究

本书法律顾问　北京市展达律师事务所

本书版权登记号：图字：01-2009-1620

图书在版编目（CIP）数据

Web 工程：实践者的研究方法/（美）普瑞斯曼（Pressman, R. S.）著；霍秋艳等译. —北京：机械工业出版社，2010.1

（计算机科学丛书）

书名原文：Web Engineering: A Practitioner's Approach

ISBN 978-7-111-28250-1

Ⅰ. W…　Ⅱ. ①普…　②霍…　Ⅲ. 主页制作-程序设计　Ⅳ. TP393.092

中国版本图书馆 CIP 数据核字（2009）第 161259 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：周茂辉

北京市荣盛彩色印刷有限公司印刷

2010 年 1 月第 1 版第 1 次印刷

184mm×260mm · 19.25 印张

标准书号：ISBN 978-7-111-28250-1

定价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了绝对性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅奠定了研究的范畴，还指示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与国际接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、翻译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大师名家的睿智与经典的光芒，跃然纸上。这套丛书的出版，也正体现了华章分社“面向教育，服务教育”的宗旨。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

华章科技图书出版中心

译 者 序

作为一本 Web 工程著作，本书全面、系统地阐述了当前 Web 工程技术的各个方面，并清晰地对基于 Web 的系统和应用的工程化实践过程进行了全面介绍。本书的问世，将对这个全新的工程学科产生重要的影响。

Web 工程是一门新兴学科，综合讲述的相关书籍寥若晨星。本书的焦点是 Web 工程，告诉我们关于基于 Web 的系统和应用的工程化实践过程，强调一个敏捷过程和一些已在行业应用中被证明过的简单而实用的方法。对于基于 Web 的特定工具和技术的讲解相对较少，因为已有数以千计的强调各种特定工具和技术的书籍、文献和基于 Web 的资源，但它们几乎都没有用联系的观点来看待 Web 工程。

本书是 Web 工程学科的及时雨，是一本非常好的 Web 工程读物，认真阅读它会使你受益匪浅。自本书原版面世以来，我就是它的忠实读者，它是我从事“Web 工程”课程教学的重要参考书。在此，郑重推荐给读者。

本书的特点如下：

- 强调敏捷的过程和一些已经在行业应用中证实的简单而实用的方法。
- 各章后面提供了相应的参考文献信息。
- 全书贯穿了一个应用实例（基于 Web 的住宅安全系统——SafeHomeAssured. com），并以对话的形式逐步展开这一开发过程，使本书更具有可读性，引发读者的阅读兴趣。
- 以非正式的会话的风格，以回答问题的形式展现内容。

本书在前言中指出了其目的是提供一本可供实践者、大学生或者研究生使用的书。除此之外，教师可以把本书作为“Web 工程”课程的教材。在作为教材时，可能还需要自行补充一些材料：习题和课程实习。对于教师或者有一定 Web 应用开发经验的研究生和专业人员，本书提供了全面的、丰富的、代表当前发展水平的 Web 工程知识。

参加本书翻译工作的主要有：闫兵（第 1 ~ 10 章、第 13 ~ 14 章）、王蕾（第 11 ~ 12 章）、陈威（第 16 ~ 17 章）完成了翻译初稿。本人完成了第 15 章的翻译，并对全部译稿做了仔细的审核与大量的修改，闫兵和陈静玉对全部译稿也进行了仔细的审核。

尽管本书的翻译工作历时超过半年，但仍然感觉时间太紧，而且 Web 工程作为一门新的学科，很多概念和术语的翻译还很不成熟。限于译者水平，我们对内容的理解和中文表达难免有不当之处，在此敬请读者批评指正。

霍秋艳

2009 年 10 月

前　言

在开始计划写这本书的时候，我们曾担心它会淹没在数以百计（不！是数以千计）的相关书籍中，已经有大量关于“Web 设计”、HTML、Java、XML 的技术书，以及构建成功的基于 Web 的系统和应用（WebApp）必须要理解的无数的技术书籍。奇怪的是，我们发现一个至关重要的主题（其他的技术都被应用的过程）被覆盖得相对较少。我们称这个过程为 Web 工程（Web engineering），同时我们相信应用这个技术更可能构建出满足用户需要并给他们客户的业务或者组织带来真正收益的 WebApp。

强调 WebApp 会是现实中所有的业务和组织成功的关键已经是陈词滥调。然而，很多的 WebApp 仍然是这样构建的：基本不考虑问题分析、有效的设计、可靠的测试以及变更管理的基本原则。所带来的后果就是，很多 WebApp 无法满足最终用户的需要和委托给开发者的业务目标。

现在，为了应对下一代的基于 Web 的系统和应用所提出的挑战，我们要从守旧的方法转变到 Web 工程。这个行业正向更加实效的 Web 工程过程（展示出敏捷性和适应性）的方向发展。同时，这个过程必须要传递一个学科方法的完整性。

本书面世了，它全面展示了工程化的基于 Web 的系统和应用的实践过程。为了在这个全新的工程学科中引导读者，其内容是以一种非正式的、会话的风格，以问答的形式进行展示。

在全书中，我们强调一个敏捷的过程和一些已经在行业应用中证明过的简单而实用的方法。同时，我们有意较少涉及基于 Web 的特定的工具和技术。这不是因为我们认为它们不重要，而是因为已经有了数以千计的强调它们的书籍、文献和网上资源，而它们几乎都没有用联系的观点来看待 Web 工程。因此，我们自然把重点放在 Web 工程上。我们的目的就是提供一本可供实践者、大学生或者研究生使用的书。

Web 工程过程强调一种敏捷方法，并展示一些简单而有效的方法，以获取和分析问题需求，设计一个有效的解决方案，然后构建和测试一个高质量的 WebApp。但是，这个过程不仅仅是关于技术的。我们还将呈现一些已经被证实的用于项目管理、变更和内容管理以及质量保证的技术。在全书中，我们用一个精心设计的案例来展开叙述我们介绍的方法和技术。Web 站点 www. SafeHomeAssured. com 对这个案例补充了一些深入的细节，同时还提供了其他额外信息。

本书受益于许多讨论构建高质量的 WebApp 的准则和技术的出版物和 Web 资源。我们对本书中引用到的原始资料的作者以及在过去的几年中对我们的想法进行指导的数以百计的同行和作者表示衷心的感谢。同时，要特别感谢 Didar Zowghi、Norazlin Yusop、Xiaoying Kong 和 Rechatrin Tongrungrojana。

在全书中，一些引用的文本、图片以及 SafeHome 这个案例来自于 Roger Pressman 的《Software Engineering: A Practitioner's Approach》（sixth edition）[⊖]。我们经过允许直接使用或加以改造，以满足特定 Web 工程师的需求。

我们两个作者的家庭都有 4 个成员，在此我们想对他们在这件事情上做出的努力表示感谢。

⊖ 该书由机械工业出版社引进出版，中文书名为《软件工程：实践者的研究方法》（原书第 6 版），书号为 7-111-19400-4。——编辑注

我们杰出的妻子（Barbara 和 Catherine）忍受了为出版一本书所花费的大量用于写、修订和四处奔波的时间。Roger 的儿子（Mathew 和 Michael）已经长大了，有他们自己的事情要做，并且每天都要使用因特网和 Web。David 的儿子（Oliver 和 Dominic）很年轻，都有他们自己的未来，同时他们将来的职业生涯中会有相当长的一部分时间能见证未来网络的发展。我们希望本书所展现的思想可以对他们的职业成长有所帮助。

Roger Pressman
美国佛罗里达州贝卡拉顿市
David Lowe
澳大利亚悉尼

目　录

出版者的话
译者序
前言

第 1 章　基于 Web 的系统 …… 1
1.1　网络 …… 1
1.2　Web 应用 …… 2
1.2.1　一个学习案例 …… 2
1.2.2　WebApp 就是计算机软件吗 …… 3
1.2.3　WebApp 的属性有别于传统软件的属性吗 …… 3
1.2.4　在 WebApp 演变的过程中会涉及哪些范畴 …… 5
1.3　WebApp——一种哲学的观点 …… 7
参考文献 …… 7
第 2 章　Web 工程 …… 8
2.1　什么是 Web 工程 …… 8
2.1.1　敏捷意味着什么 …… 8
2.1.2　什么是 WebE 框架 …… 9
2.1.3　适应框架时应该遵循哪些原则 … 10
2.1.4　旧方法有什么价值吗 …… 11
2.2　Web 工程的组件 …… 11
2.2.1　软件工程如何发挥作用 …… 11
2.2.2　为什么 WebE 过程的敏捷性如此重要 …… 12
2.2.3　过程框架内有哪些 WebE 方法 …… 13
2.2.4　工具和技术不就是 Web 工程的一切吗 …… 13
2.3　Web 工程的最佳实践 …… 14
2.4　本章小结 …… 15
参考文献 …… 15
第 3 章　Web 工程过程 …… 17
3.1　定义框架 …… 17
3.2　增量过程流 …… 19
3.2.1　框架活动是如何进行的 …… 19
3.2.2　框架如何进行精化 …… 21
3.3　WebE 框架的一般动作和任务 …… 22
3.3.1　沟通活动应当如何进行精化 …… 22
3.3.2　制定一个增量计划需要哪些任务 …… 23
3.3.3　什么是建模 …… 24
3.3.4　可以应用哪些分析建模任务 …… 24
3.3.5　设计模型的要素有哪些 …… 25
3.3.6　可以应用哪些设计建模任务 …… 26
3.3.7　应用哪些构建任务 …… 27
3.3.8　WebApp 增量如何进行部署 …… 28
3.4　保护活动 …… 28
3.4.1　WebE 团队如何管理变更 …… 28
3.4.2　如何保证增量的质量 …… 29
3.4.3　如何管理风险 …… 29
3.4.4　如何管理工作 …… 29
3.5　本章小结 …… 30
参考文献 …… 30
第 4 章　沟通 …… 31
4.1　沟通活动 …… 31
4.2　规划 …… 32
4.2.1　应当和谁沟通 …… 32
4.2.2　可以使用哪些技术进行沟通 …… 32
4.2.3　不会有不同的视角吗 …… 33
4.2.4　应该问什么问题 …… 33
4.2.5　如何促进协作 …… 34
4.3　提取需求 …… 35
4.3.1　在提取需求的会话之前做什么 … 36
4.3.2　利益相关者如何进行准备 …… 36
4.3.3　提取会话中执行哪些任务 …… 37
4.3.4　WebApp 的用户种类 …… 38
4.3.5　如何识别内容和功能需求 …… 38
4.3.6　如何将约束和性能问题分离 …… 39
4.3.7　什么是使用场景 …… 40
4.3.8　什么是用例 …… 41
4.3.9　如何创建用例 …… 41
4.4　确认 WebApp 增量 …… 45
4.5　协商 …… 46
4.6　本章小结 …… 47
参考文献 …… 48

第5章 计划 …… 49
5.1 理解范围 …… 49
5.1.1 有哪些相关的沟通工作产品 …… 49
5.1.2 如需要更多的细节来理解增量会怎么样呢 …… 50
5.1.3 你的理解中仍然存在不足怎么办 …… 51
5.2 精化框架活动 …… 51
5.2.1 需要哪些动作和任务 …… 52
5.2.2 生产哪些工作产品 …… 53
5.2.3 评估质量的恰当方法 …… 54
5.2.4 如何管理变更 …… 55
5.3 组建一个 WebE 团队 …… 55
5.3.1 如何识别一个优秀的 WebE 团队 …… 56
5.3.2 团队为什么没有凝聚力，怎么改善 …… 56
5.3.3 WebE 团队可以管理自身吗 …… 57
5.3.4 如何组建一个成功的团队 …… 58
5.3.5 一个好的团队负责人有哪些特征 …… 58
5.4 管理风险 …… 59
5.4.1 如何识别风险 …… 59
5.4.2 如何评估风险 …… 60
5.4.3 如何开发应急计划 …… 61
5.5 开发进度表 …… 62
5.5.1 什么是宏观进度安排 …… 62
5.5.2 什么是增量进度安排 …… 63
5.5.3 如何估算工作量和时间 …… 64
5.5.4 如何描述任务之间的相互依赖关系 …… 65
5.6 管理质量 …… 66
5.6.1 团队可以使用哪些质量保证机制 …… 66
5.6.2 什么是结对走查方法 …… 67
5.6.3 什么是团队走查方法 …… 67
5.6.4 WebApp 存在质量标准吗 …… 68
5.7 管理变更 …… 68
5.7.1 如何评估变更的重要性和影响 …… 69
5.7.2 何时推迟做出变更 …… 70
5.7.3 所有的相关工作产品都应当做出变更吗 …… 71
5.8 跟踪项目 …… 72
5.8.1 进度问题有什么宏观指标吗 …… 72
5.8.2 可以使用哪些标准来跟踪进度 …… 72
5.9 外包 WebE 工作 …… 73
5.9.1 如何启动外包项目 …… 73
5.9.2 如何选择候选外包供应商 …… 73
5.9.3 如何评估报价的合理性和估算的可靠性 …… 74
5.9.4 需要什么程度的项目管理 …… 74
5.9.5 如何评估进度并管理范围 …… 74
5.10 本章小结 …… 74
参考文献 …… 75
第6章 建模活动 …… 76
6.1 建模的概念 …… 76
6.1.1 如何评判一个模型的有用性 …… 76
6.1.2 模型可以用于理解业务限制吗 …… 77
6.2 创建的模型 …… 78
6.2.1 过程告诉了我们关于建模的哪些信息 …… 78
6.2.2 WebApp 告诉了我们关于建模的哪些信息 …… 78
6.3 建模框架 …… 79
6.3.1 对 Web 来说，有建模框架吗 …… 80
6.3.2 建模如何与 WebE 过程关联 …… 80
6.4 建模语言 …… 82
6.4.1 应当具备哪些建模功能的能力 …… 83
6.4.2 应当具备哪些建模信息内容的能力 …… 83
6.4.3 在建模语言中，应当有哪些通用能力 …… 84
6.5 已有的建模方法 …… 85
6.6 本章小结 …… 87
参考文献 …… 87
第7章 WebApp 分析建模 …… 89
7.1 理解在 WebE 中的分析 …… 89
7.1.1 分析到什么程度才是足够的 …… 89
7.1.2 可以使用原型进行分析吗 …… 90
7.1.3 分析和设计不同吗 …… 91
7.2 对 WebApp 进行分析建模 …… 91
7.2.1 分析建模的输入是什么 …… 91
7.2.2 分析的输出是什么 …… 93
7.2.3 应当执行哪些分析任务 …… 93

7.2.4 可以使用哪些工具来建模…… 93
7.2.5 如何确定建模是否必要以及哪种方法最好呢 …… 94
7.3 理解用户 …… 95
7.3.1 为什么有必要重返用户等级…… 96
7.3.2 不加修改地应用使用场景吗…… 97
7.4 内容模型 …… 99
7.4.1 内容模型的结构元素有哪些…… 99
7.4.2 什么是信息交换，如何进行表示 …… 100
7.4.3 如何确定内容对象 …… 101
7.4.4 有描述内容之间的联系和层次的简单方法吗 …… 102
7.4.5 如何为 WebApp 选择和表示分析类 …… 103
7.5 交互模型 …… 104
7.5.1 用例从什么地方开始起作用 … 104
7.5.2 什么是序列图，何时进行开发 … 104
7.5.3 状态图如何描述 WebApp 的行为 …… 105
7.5.4 是否真的需要用例、序列图和状态图…… 106
7.5.5 为什么创建一个界面原型是很明智的…… 106
7.6 功能模型 …… 107
7.7 配置模型 …… 108
7.8 关系－导航分析…… 109
7.8.1 如何建立内容对象和功能之间的关系…… 110
7.8.2 如何分析导航需求 …… 110
7.9 本章小结 …… 111
参考文献…… 111
第 8 章 WebApp 的设计 …… 113
8.1 对 WebApp 进行设计 …… 113
8.1.1 WebApp 设计者需要知道什么 … 114
8.1.2 什么是逻辑设计 …… 114
8.1.3 什么是物理设计 …… 114
8.1.4 设计完成后生成哪些信息 …… 114
8.2 设计目标 …… 115
8.3 设计和 WebApp 的质量 …… 117
8.3.1 用户如何理解质量 …… 117
8.3.2 有以用户为中心的模型吗 …… 117
8.3.3 什么是质量框架 …… 120
8.3.4 有评估内容质量的方法吗 …… 121
8.3.5 有质量检查表吗 …… 121
8.4 设计过程 …… 123
8.4.1 WebApp 设计的元素有哪些 … 123
8.4.2 设计过程的特征有哪些 …… 124
8.4.3 对设计活动而言，一个增量的 WebE 过程意味着什么 …… 125
8.5 概念架构的初始设计 …… 126
8.6 技术架构的初始设计 …… 129
8.7 本章小结 …… 129
参考文献…… 130
第 9 章 交互设计 …… 131
9.1 界面设计原则和指导准则 …… 131
9.1.1 设计有效界面的原则 …… 132
9.1.2 有哪些实用的设计指导原则 … 135
9.2 界面设计工作流…… 135
9.3 界面设计的前期工作 …… 137
9.3.1 如何理解 WebApp 用户的特征 … 137
9.3.2 如何详细描述已经识别的内容对象…… 138
9.3.3 用户完成哪些任务 …… 138
9.3.4 如何详细描述已经确定的任务 … 140
9.3.5 如何为不同角色的不同用户进行设计…… 141
9.3.6 内容如何集成到界面描述之中 … 142
9.4 界面设计步骤 …… 143
9.4.1 如何布局界面对象和动作 …… 143
9.4.2 如何设计界面导航机制 …… 146
9.4.3 为什么界面的一致性如此重要 … 147
9.5 美学设计 …… 148
9.5.1 如何创建令人愉悦的布局 …… 148
9.5.2 什么是好的图形设计 …… 149
9.6 可用性 …… 150
9.7 设计问题 …… 151
9.7.1 哪些因素影响响应时间，怎么改进…… 151
9.7.2 如何设计“帮助”设施 …… 152
9.7.3 界面应当如何处理错误 …… 152
9.7.4 什么是“可访问性”？如何应用于界面设计中 …… 153
9.7.5 什么是“国际化”？如何应用到 WebApp 中 …… 154
9.8 本章小结 …… 154
9.9 参考文献 …… 154

第 10 章 信息设计 …… 156
10.1 信息架构 …… 156
10.1.1 什么是信息架构 …… 157
10.1.2 信息架构的元素有哪些 …… 157
10.1.3 好的信息架构有什么特征 …… 158
10.1.4 如何开发信息架构 …… 159
10.2 组织内容 …… 160
10.3 设计信息空间 …… 161
10.3.1 有哪些可能的信息结构 …… 161
10.3.2 什么可以产生好的结构 …… 164
10.4 蓝图：为结构添加细节 …… 165
10.5 访问信息 …… 167
10.5.1 如何确保用户可以理解上下文并且不会迷路 …… 167
10.5.2 如何帮助用户在信息结构内移动 …… 168
10.5.3 实现搜索机制的指导准则 …… 169
10.5.4 搜索机制会产生问题吗 …… 170
10.6 线框模型 …… 170
10.7 导航设计：创建详细的结构 …… 171
10.7.1 信息设计和导航设计模型是如何演化的 …… 171
10.7.2 RMM 模型如何用于导航设计 … 172
10.7.3 如何使用 WebML 创建导航设计 …… 175
10.7.4 可能创建链接了内容和功能的模型吗 …… 176
10.7.5 Web 自身的结构有影响吗 …… 176
10.8 总结设计过程 …… 177
10.9 本章小结 …… 178
参考文献 …… 179
第 11 章 功能设计 …… 181
11.1 WebApp 功能 …… 181
11.2 WebApp 功能的性质 …… 182
11.2.1 功能示例 …… 182
11.2.2 功能分类 …… 182
11.2.3 信息和功能总是可区分的吗 … 184
11.3 设计过程中的功能设计 …… 185
11.3.1 功能设计过程包含哪些要素 … 185
11.3.2 需要多少功能设计 …… 186
11.3.3 SafeHomeAssured.com 的初始功能设计应该如何进行 …… 186
11.4 功能架构 …… 188
11.4.1 功能架构是什么样的 …… 189
11.4.2 如何开发功能架构 …… 190
11.4.3 处理异常的功能如何开发 …… 191
11.4.4 在功能设计时可以使用架构模式吗 …… 192
11.5 详细功能设计 …… 193
11.5.1 WAE 建模如何用于详细设计中 …… 193
11.5.2 WebML 为什么适合于工作流建模 …… 194
11.6 状态建模 …… 196
11.7 本章小结 …… 198
参考文献 …… 198
第 12 章 构建和部署 …… 200
12.1 WebE 过程中的构建和部署 …… 200
12.1.1 构建和部署之间的相互作用是什么 …… 201
12.1.2 部署环境扮演什么角色 …… 201
12.2 构建 …… 204
12.2.1 有通用的构建任务吗 …… 204
12.2.2 什么是重构？如何应用它 …… 205
12.3 构建的原则和概念 …… 206
12.4 部署 …… 207
12.4.1 是否存在通用的部署任务 …… 207
12.4.2 指导 WebE 团队的部署原则 … 208
12.4.3 如何使用版本控制和 CMS … 209
12.5 构建和组件的使用 …… 210
12.5.1 什么是通用组件 …… 211
12.5.2 如何定义面向对象组件 …… 211
12.5.3 如何定义传统组件 …… 212
12.5.4 好组件有哪些特征 …… 212
12.6 组件级设计指导准则 …… 213
12.7 组件的设计步骤 …… 215
12.8 本章小结 …… 216
参考文献 …… 217
第 13 章 设计模式 …… 219
13.1 模式：理解概念 …… 219
13.1.1 什么是模式 …… 219
13.1.2 模式怎么描述 …… 220
13.2 WebApp 模式：设计焦点和粒度 … 221
13.2.1 如何使用设计焦点来确定模式 …… 221

13.2.2 粒度为什么是模式的一个重要特征 …… 221
13.3 模式仓库 …… 222
13.3.1 什么是模式库 …… 222
13.3.2 Web 工程师可参考哪些模式资源 …… 222
13.3.3 WebE 团队能够创建自己的模式集吗 …… 223
13.3.4 如何发现和使用模式 …… 224
13.4 模式示例 …… 225
13.4.1 有可能定义解决业务层问题的模式吗 …… 225
13.4.2 因为交互很普遍，就必然有很多交互模式吗 …… 226
13.4.3 有哪些可用的导航模式 …… 229
13.4.4 内容和表示模式适合使用的地方 …… 232
13.5 本章小结 …… 233
参考文献 …… 234
第 14 章 技术和工具 …… 235
14.1 常见问题 …… 235
14.1.1 分而治之如何影响工具和技术 …… 235
14.1.2 选用哪种技术——开源还是私有 …… 236
14.1.3 WebE 技术应用种类的影响 …… 237
14.2 实现工具和技术 …… 237
14.2.1 什么是应用框架 …… 238
14.2.2 如何应用内容管理系统和版本控制技术 …… 238
14.2.3 WebApp 怎样提供搜索能力 …… 239
14.3 开发工具和技术 …… 239
14.3.1 怎样获得有助于建模活动的工具 …… 239
14.3.2 专门针对 WebApp 的测试工具 …… 240
14.3.3 支持 WebE 过程管理的工具 …… 240
14.4 本章小结 …… 241
参考文献 …… 241
第 15 章 WebApp 测试 …… 242
15.1 测试概念 …… 242
15.1.1 什么是质量维度 …… 242
15.1.2 WebApp 环境中发生什么类型的错误 …… 243
15.1.3 应该应用什么测试策略 …… 243
15.1.4 多少测试计划是必需的 …… 244
15.2 测试过程概述 …… 245
15.3 内容测试 …… 247
15.3.1 内容测试的目标是什么 …… 247
15.3.2 数据库测试如何用于验证内容 …… 248
15.4 用户界面测试 …… 249
15.4.1 界面测试策略 …… 249
15.4.2 如何测试界面机制 …… 250
15.4.3 如何测试界面语义 …… 252
15.5 可用性测试 …… 252
15.6 兼容性测试 …… 255
15.7 组件级测试 …… 255
15.8 导航测试 …… 256
15.8.1 如何测试导航语法 …… 257
15.8.2 如何测试导航语义 …… 257
15.9 配置测试 …… 259
15.9.1 如何测试服务器端 …… 259
15.9.2 如何测试客户端 …… 259
15.10 安全性和性能测试 …… 260
15.10.1 如何确定 WebApp 是安全的 …… 260
15.10.2 如何测试 WebApp 的性能 …… 262
15.10.3 性能测试的目标是什么 …… 262
15.10.4 负载测试如何评估性能 …… 263
15.10.5 压力测试如何评估性能 …… 263
15.11 本章小结 …… 266
参考文献 …… 266
第 16 章 变更管理和内容管理 …… 268
16.1 变更 …… 268
16.1.1 变更的属性有哪些 …… 268
16.1.2 为什么需要变更 …… 268
16.1.3 WebApp 的哪些元素变更了 …… 269
16.2 Web 工程的变更管理 …… 269
16.2.1 为什么需要管理变更 …… 269
16.2.2 应该如何识别将要发生变更的对象 …… 271
16.2.3 如何控制将要发生的变更 …… 271
16.2.4 如何管理一个 WebApp 或其组件的不同版本 …… 273

16.2.5 WebE 团队如何确保一个变更恰当地实现了 …… 274
16.2.6 如何让利益相关者知道实施的变更 …… 274
16.3 内容管理 …… 274
16.3.1 如何使用内容管理系统 …… 275
16.3.2 CMS 的主要元素有哪些 …… 275
16.4 CMS 的实现标准 …… 277
16.4.1 数量如何影响内容管理 …… 277
16.4.2 内容创作者的人数会影响 CMS 吗 …… 278
16.4.3 变更量如何影响变更管理的正式性 …… 278
16.4.4 发布量如何影响内容管理的正式性 …… 279
16.5 本章小结 …… 280
参考文献 …… 281
第 17 章 未来趋势 …… 282
17.1 Web 和 WebApp 具有不断变化的特征 …… 282
17.1.1 基于 Web 的内容和功能的交付会如何变化 …… 282
17.1.2 WebApp 将如何变化 …… 283
17.1.3 为了适应这些变化，Web 工程师必须做些什么 …… 283
17.1.4 Web 可以作为应用软件的平台吗 …… 284
17.1.5 未来的 Web 会是一个操作系统吗 …… 284
17.1.6 语义 Web 将如何改变事物 … 285
17.2 演化中的 Web 技术和 Web 2.0 … 286
17.2.1 什么是 Web 2.0 …… 286
17.2.2 哪些技术支持 Web 2.0 …… 287
17.2.3 在技术演化中应该考虑哪些关键问题 …… 290
17.2.4 Web 2.0 将会如何发展 …… 290
17.3 对于未来的观点 …… 291
17.4 Web 工程的变化特性 …… 293
参考文献 …… 293

第1章　基于Web的系统

我们先回顾一下计算机软件开发的前几十年的情况。在20世纪五六十年代的时候，几乎没有人意识到基于计算机的系统的重要性，事实上也没有人预见到计算机的软硬件会在20世纪末和21世纪初期对社会的各个方面产生全球化的影响。在早期，与计算机打交道的绝大多数人会陷入业务的困惑之中，这是因为创建计算机程序的过程中，混杂了许多非正式、紧迫、直觉和艺术等因素。当事情完成得好时，这种方式会带来在计算方面的重要提升。然而，事情不可能总是一帆风顺的。基于计算机的系统常常无法完成其应有的功能，导致交付延迟甚至无法交付，并很难（有时甚至不可能）在任何合理的时间框架内进行纠正、适应和改进。很遗憾，过去使用的是一种碰运气的方式。

但是旧式的思想已经建立起了一种文化，并很快变得难以改变。非正式、紧急、直觉和艺术是大多数基于计算机的系统开发者活动的驱动力。总之，非正式（informality）带来的是宽松的工作环境——你可以在其中做自己的事情。紧迫（urgency）带来行动和快速的决策。直觉（intuition）是一种让你在复杂的情形下“感觉”自己的解决办法的无形的素质。同时，艺术（art）会带来美学的形式和功能——使接触它的人们感到快乐的东西。也许你会问，那有什么不对呢？

当我们把焦点从过去转移到现在的时候，你就会发现这个问题的答案和Web工程有莫大的关系，同时它也是本书所讨论的主题。

1.1　网络

今天，我们生活在因特网快速发展的年代。关于这个时代令人振奋的描述已经很多了，所以如果不涉及这些已经算是陈词滥调的对话，就无法去讨论因特网和万维网的影响。你已经知道Web“大”，很大。尽管Web的大小及其增长率令人吃惊，但是我们并不是指传统意义上的“大”（例如，网页和网站的数目、用户量和传输于网络的数据量），我们指的是社会和文化角度的“大”。

Web已经成为贸易、商业、通信、教育、工程、娱乐、金融、政府、工业、媒体、医药、政治、科学和交通等方面不可或缺的技术，而这些只是影响你生活的诸多方面中的一小部分而已。但是，“不可或缺的技术”只是揭示了Web对我们每个人的表面影响。它已经改变了人们购物（电子商务）、与人见面（在线约会）、理解世界（门户）、获得新闻（在线媒体）、发表自己的观点（博客）、自我娱乐（从音乐下载到在线娱乐）以及上学（在线学习）等方式。

所有的这些影响有一个共性——它们需要传输工具，它可以携带与兴趣域相关的原始信息；以一种有意义的方式加以结构化；构建组织好的、美学的、处理过的和交互的（如果需要）包装的表示；同时以启动会话（conversation）的方式将包装的表示传送到Web浏览器。

你和一个Web应用之间的会话可以是主动的或者被动的。在被动会话中，你选择将要呈现的信息，但是不直接控制它的大小、类型或者结构。在主动会话中，你提供输入，因此所呈现的信息是通过定制来满足你的特定需要。

这个获取信息、组织信息、构建包装的表示并对其进行传输的工具称为Web应用（WebApp）。当一个WebApp和客户端与服务器端的硬件、操作系统、网络软件以及浏览器结合起来的时候，一个基于Web的系统就出现了。

1.2 Web 应用

在万维网（World Wide Web）的早期（大约从 1990 年到 1995 年），"Web 站点"仅包含链接在一起的少量超文本文件，这些文件使用文本和有限的图标来表示信息。随着时间的推移，一些开发工具和技术（如扩展标记语言 XML、Java）使 HTML（Hypertext Markup Language，超文本标记语言）得到增强，它使 Web 工程师能够在客户端和服务器端的内容中提供计算能力。基于 Web 的系统和应用[2]应运而生。今天，WebApp 已经发展为先进的计算工具，不仅可以为最终用户提供独立的功能，而且也已经和公司以及政府的数据库和应用进行了集成。

1.2.1 一个学习案例

CPI 公司是一个虚构的公司，为家庭和小型企业来构建、买卖、销售、监控安全系统。CPI 公司还没有 Web 的展示，现在它希望推出一个"里程碑"的网站，来介绍它新的安全传感器生产线和一系列全新的基于 Web 的服务。他们希望能在名为 SafeHomeAssured.com 的 WebApp 的开发中得到你的帮助，同时希望你能够在他们创建增加市场份额的新 Web 服务时提供支持。

你被通知去参加一个讨论基本想法的会议。在会议中，你得知 CPI 已经设计出了一个简洁的无线传感器控制器，这个控制器将成为被称为 SafeHome 的商业和住宅安全系统的一个新生产线的核心部件。以下是会议谈话的一个片段。



一个项目开始了

场景：CPI 公司的会议室里。

参与者：一个高级业务经理，一个产品开发经理，一个营销经理，一个工程经理，还有你，Web 工程的专家。

会话：

业务经理（对产品经理说）：我听说你们正在开发一个产品？一个通用的无线盒？

产品经理：它是相当酷的产品，只有一个小火柴盒的大小。我们可以把它与各种只要使用了 IEEE 的任何无线协议的传感器（比如一个数码相机）相连接。它允许我们通过无线的方式访问设备。我们认为它将带来一代全新的产品。

业务经理（望着营销经理）：你同意吗？

营销经理：我同意。事实上，从今年迄今的销售业绩来看，我们需要一些新的产品。我们已经做了一些市场调查，同时已经有了一个可能很大的产品和服务线。

业务经理：多大？底线是多少？

营销经理：这是全新一代的"住宅管理系统"，我们称之为 SafeHome。产品采用新的无线接口，为家庭和小型企业使用者提供一个由电脑通过因特网控制的系统（家庭安全、监控、应用和设备控制）。当你在开车回家的路上的时候，可以打开家里的空调，或者诸如此类的事情。我们同时也考虑了住宅和企业内部的视频监视和控制。另外很重要的一点是，我们试图把产品与监控服务直接集成起来，允许客户通过 Web 去访问他们的账户，做出一些决定，比如什么时候对系统进行监控，什么时候撤销监控，以及在一个特定的时间段内发生了什么事情。我们还想通过 Web 来完成维护诊断。

[2] 在本书中，Web Application（WebApp）这个术语涵盖的范围是广泛的，小到可以驻留在用户计算机上实现简单 Web 功能，大到为成千上万用户提供全面的数据库服务的站点。这类应用包含在因特网上运行，企业内部和外部网络上的完整的网站，网站中的特定的功能以及信息处理应用。

产品经理：工程部已经对这些想法进行了技术可行性分析。结论是，这个项目可行，而且可以做到低成本。绝大多数的硬件都是现货供应的。基于 Web 的软件是个问题，但也并非是我们做不了的事情。我们已经注册了一个域名：SafeHomeAssured. com。[所有 CPI 经理们都微笑着看着你。]

业务经理：有意思。现在，我来问一下底线。

营销经理：在美国，有 PC 机的家庭占有很大的比例。如果我们定价合适，它将会是一个非常成功的应用。只有我们才拥有这一无线盒的专利（所有权）。我们将在这个方面拥有两年的竞争优势。至于收入？在第二年大概会有三千万到四千万美元。

业务经理（笑容满面）：让我们进行更进一步的讨论。我很有兴趣。

就这样，一个项目开始了。你会注意到现阶段很少涉及细节。有许多事情需要被定义、具体化，然后实现。对产品的内在理解和支持它的基于 Web 的系统会发生变化。但是在现在早期阶段这确实没有什么关系。SafeHome 有高级管理者（可以预见到巨大的潜在利益的人）的支持，而且你有机会成为完成这一工作的一员。

在本书中，我们会不停地回到 SafeHome 和 SafeHomeAssured. com 的 WebApp 上来，把这个项目作为一个案例，来描述 Web 工程中的许多方面。但是现在回到对 WebApp 的介绍性讨论上来，并研究一下它们和传统的计算机软件之间的相似性。

1.2.2 WebApp 就是计算机软件吗

如果把 WebApp 看作是一组指令和数据，可为最终用户提供信息和功能，那么毫无争议，WebApp 就是计算机软件。因此，也就是说，我们有理由设想，可以借鉴在过去几十年构建传统的基于计算机的系统时所学到的一些（如果不是全部）知识来构建 WebApp。同时也有理由认为，我们将会遇到很多（如果不是全部）在早期曾经经历过的问题（包括文化上的和技术上的）。本书的后面将对其进行讨论。

1.2.3 WebApp 的属性有别于传统软件的属性吗

人们在这个问题的回答上存在争议。有些人认为一个 WebApp 不过是一个高度重视美学表示（例如，布局、图像、音频和视频元素）和功能的客户端 - 服务器应用而已，认为 WebApp 和传统客户端 - 服务器应用有着相同的属性。但是另一些人（包括本书的作者）认为，当考虑它们的整体性时，一个 WebApp 的完整的特征集确实使基于 Web 的系统有别于更传统的基于计算机的系统。绝大多数 WebApp 都会具有如下属性。

网络密集性（network intensiveness）。每一个 WebApp 都驻留在网络上，并且必须为大量不同客户群体的需要提供服务。就 SafeHome[⊖]的产品而言，许多即将由 CPI 实现的新特征都将通过 Web 进行初始化、控制和监控。网络使 SafeHomeAssured. com 这个 WebApp 的基于客户的特征与由 CPI 建立的服务器之间建立了通信。

并发性（concurrency）。大量的用户可能会同时访问某个 WebApp。在很多情况下，最终用户的使用方式差异非常大。有时，一个或一组用户的行为可能会影响到其他用户的行为或者展示给其他用户的信息。就 SafeHomeAssured. com 而言，数以万计的住宅将被并发地监控；几百或几千个客户可能在任意时间访问 WebApp；几十个服务技术人员也可能同时在线。

⊖ SafeHome 是本章前面介绍过的一个基于 Web 的系统所支持的一个安全系统，将用作全书中连续的例子。

不可预测的负载（unpredictable load）。访问 WebApp 的用户数目每天都可能会有数量级的变化。至于 SafeHomeAssured. com，被监控的住宅和业务的数量变化将很缓慢。但是 WebApp 必须能够处理同时发生的数目不可预测的事件（例如，防窃报警器、火警装置和一氧化碳检测装置）。周一，也许每个小时会报告 10 个事件；周二，也许会记录 100 个事件；周三（在一个区域大规模断电之后）每分钟就可能报告几千个事件。

性能敏感性（performance sensitive）。如果一位 WebApp 用户（对于访问、服务器端的处理、客户端的格式化和显示）必须要等很长时间，该用户就可能会转到其他地方。具体到 SafeHomeAssured. com，它的性能是至关重要的，因为人们的生命可能受到威胁。如果 WebApp 对事件的响应太慢，诉讼就可能发生。

高可用性（high availability）。虽然说期待百分之百的可用性是不现实的，但是流行的 WebApp 的用户通常要求基本的“24/7/365”（全天候）的可访问性。就 SafeHomeAssured. com 而言，其目标是百分之百的可用性（已经说过，这一系统是关于住宅安全的），因此 WebApp 必须被设计成满足这个理想状况（或者非常接近它）。

数据驱动（data driven）。很多 WebApp 的主要功能是借助于超媒体来向最终用户展示文本、图像、音频和视频内容。另外，WebApp 通常用于访问那些存在于数据库中的信息，而这些数据库并非基于 Web 的环境的整体的一部分（例如，电子商务或金融应用）。就 SafeHomeAssured. com 而言，所有这些属性都将是很明显的。而且，这个 WebApp 必须访问存储了每个用户信息的数据库；用户具有的系统配置；以及给定系统的监控需求、一个事件日志和一个维护日志。

内容敏感性（content sensitive）。内容的质量和美学性依然是 WebApp 质量的重要决定因素。就 SafeHomeAssured. com 而言，WebApp 的一个重要的用户类将是“平民”，即那些要求简单而有意义的内容展示的非技术人群。

持续演化（continuous evolution）。传统应用软件随一系列计划好的时间间隔发布而演化，而 WebApp 是持续地演化。对某些 WebApp（尤其是 WebApp 的内容）而言，根据以分钟计的进度表进行更新，或者针对每个请求进行独立的内容计算是司空见惯的事。在本书的后面，我们将会看到，随着时间的推移，SafeHomeAssured. com 这个 WebApp 将随着对系统变化的理解而演化。WebApp 的演化将需要一种“增量”的开发方法。

即时性（immediacy）。虽然说即时性（也就是把软件尽快推向市场的强制性要求）是很多应用领域的特点，但是将 WebApp 投向市场常常是几天或者几周的事情[⊖]。Web 工程师们必须使用为满足 WebApp 开发所要求的紧迫时间进度而修改过的方法来进行计划、分析、设计、实现以及测试。就 SafeHomeAssured. com 而言，CPI 管理的焦点在于短期的收入增加和中期的显著收入。当这发生时，就体现出“昨天”需要 WebApp 了。

安全性（Security）。由于 WebApp 是通过网络进行访问的，所以即使可能也很难限制访问这个应用的最终用户的数量。为了保护敏感的内容，并提供安全的数据传输模式，在支持 WebApp 的整个基础架构以及应用本身内部都必须实现强有力的安全措施。就 SafeHomeAssured. com 而言，信息从人们的住宅和商务中流入流出，使这个 WebApp 成为那些有犯罪意图的人的绝好的目标。它必须是安全的。

美学性（aesthetics）。WebApp 吸引力的一个不可否认的部分是它的外观和感觉。当设计好一个应用要推向市场、出售产品或者想法、提供服务来产生收入时，美学与技术设计同样关系到该应用的成功。就 SafeHomeAssured. com 而言，WebApp 将提供的内容和功能的多样性（将在第 4

⊖ 有了现代的工具，复杂的 Web 页面在几个小时之内就可以完成。

章讨论）要求它们的表示既简单又优雅。美学就是这个系统能够被接受的一个关键因素。

1.2.4 在 WebApp 演变的过程中会涉及哪些范畴

继续进行你和 CPI 人员的会议，通过与产品经理和技术人员交流，可以更好地理解 SafeHomeAssured. com 的当前状态。很明显，SafeHomeAssured. com 这个 WebApp 将会相当重要。到目前为止，还没有什么确定的内容，但是看起来下面的这些特性（内容和功能）将被实现：

- 关于 CPI 及其产品和人员的信息。
- 所有安全类硬件组件的规格说明书，包括图片、技术性描述、安装指南、定价以及其他相关信息。
- 安全系统的设计支持，可以让客户指定生活或者业务空间（例如，房间、门、窗户），然后可以得到一个安全性系统的半自动化布局。
- 可以让客户订购安全类的硬件和监控服务的电子商务的功能，这个功能将会和支撑客户购买的后台系统进行结合。
- 客户通过因特网进行监控，使房主或者公司人员可以利用视频进行实时监控。
- 用户账号访问功能。
- 用户服务访问功能，包括特定的室内功能。
- 技术类的服务人员访问功能，包括特定的内部功能。

另外，CPI 想抛弃那种传统的销售策略（例如，售货员、店面）而转向一种 21 世纪的销售模式。这个公司计划专门通过 Web 进行销售。

但是 CPI 却没有一个针对性的 Web 展示，更不必说已经写出 SafeHomeAssured. com 这个 WebApp 的需求将是什么，甚至还没有对 Web 的真实能力的比较深入的理解。例如，当你提到最终在一两年的时间内完成通过一个包含对客户住宅三维重现（就像 SecondLife[⊖]一样）的虚拟世界到客户的安全系统的接口这件事的可能性时，CPI 的人员都一脸茫然地看着你。仅仅为了开始，你决定给他们提供一个速成的 Web 站点的例子。真实的 SafeHomeAssured. com WebApp 将会以阶段的方式进行演化，我们称之为 WebApp 增量。在 WebApp 演变的过程中，它将呈现出以下类别[⊜]：

信息型（informational）WebApp。你决定去构建一个主页面和一些支持页面来描述 CPI 以及它的产品和服务。你已经完成的是构建一个信息型 WebApp——包含一些只读的内容和简单的导航和链接。

下载型（download）WebApp。几周以后，你开始增加一些描述 SafeHome 的传感器和其他安全系统硬件的内容。CPI 提供了描述他们的 PDF（Portable Document Format，可移植文档格式）说明文件。你增加了一个允许访问者去 SafeHomeAssured. com 下载产品说明的功能。这个 WebApp 现在合并了信息和下载的功能。

可定制型（customizable）WebApp。当你从 CPI 利益相关者那里得到更多的信息之后，很明显你将会有以下四类潜在的最终用户：房主、小企业务拥有者、CPI 的客户服务人员和 CPI 的技术服务人员。你想通过使用术语和展示对在 Web 站点上展示的内容进行裁剪以满足每种顾客的特殊需要。你对自己原来的 WebApp 进行一项大的革新，即创建一个使每个用户可定制的新应用。

交互型（interaction）WebApp。通信量增加很快，没多久就有了几百个访问者（毕竟，人们会关注对住宅和企业安全的有效的解决方法）。你想在访问者中营造一种社区的氛围，在这里人

⊖ 参见 http：//secondlife. com。

⊜ 下面 WebApp 的分类改编自［Dar99］。

们可以聊天、互相问答、提供产品的证明等。你决定对 SafeHomeAssured.com 的实现进行扩展，使它支持聊天室的特性。你现在已经为自己的 WebApp 提供了一个交互组件。

用户输入型（user input）WebApp。CPI 的管理层想从通过电子邮件和电话请求产品报价单中解脱出来，你实现基于表单的输入，使每个报价请求都以一种可预知的方式进行组织。虽然你要使用其他自动操作来生成报价单，但是至少不必去转录形式各异的输入和信息源。

面向事务型（transaction-oriented）WebApp。基于表单输入的报价单可以很好地工作，但是 CPI 的管理层很快意识到整个报价过程可以自动化。他们为你提供了一系列计算软件和建立在基于表单输入之上的监控定价的算法。现在，基于用户通过表单提供的输入，立即提供给他们一个报价。用户和 WebApp 之间的事务就产生了。

面向服务型（service-oriented）WebApp。你现在准备好提供一种全面的设计辅助功能。用户以图形的方式输入对一个空间的描述，然后就能得到针对那个空间的安全系统在设计上的支持。这种服务可以直接带来销售收益。另外，它强调了 CPI 和 SafeHome 产品的总体质量。

门户型（portal）。随着时间的推移，你付出的很多工作收到了每天几千个访问者回报。CPI 的员工每天会收到几百个与安全相关的问题，他们没有时间去——回答。为了解决这个问题，你开始提供一些有这些问题的答案的站点的链接。不久以后，站点的一部分引导用户成为了很多种有用的信息源。SafeHomeAssured.com 现在已经有了门户的特征。

数据库访问型（database access）。你的生产线和客户的基数急剧地增加，因此，很有必要去建立三个新的数据库：1）所有 SafeHome 的产品及其技术规格说明书，定价（对于客户也略）、安装指南以及交付和可用性信息；2）所有与客户相关的信息；3）所有与监控相关的信息。这些数据库可以通过 WebApp 的用户输入要素的特征进行查询。

数据仓库型（data warehousing）。CPI 正快速地变成安全性产品的一个主要国际供货商。为了满足众多国家的需要，你必须使用当地的建筑规章、供货商、安装费等类似的信息，需要获得对多个数据库的访问并提取对客户有用的信息。你开始为 SafeHomeAssured.com WebApp 构建一个大规模的数据仓库组件。

SafeHomeAssured.com WebApp 将沿着上述的每种类型进行演化。在更进一步的讲解过程中，我们将会对引起 SafeHomeAssured.com 演化的需求有一个更细节化的认识。WebApp 的属性和种类总结在图 1-1 中。



图 1-1 WebApp 的属性和种类

1.3 WebApp——一种哲学的观点

在早些时候我们提到过，Web 很大，在物理上和文化上都很广大，但是它的影响无法用巨大来形容。有理由断言 Web 代表了一种全局的理解（数据、信息、知识甚至是智慧的汪洋大海），包含了不同实体（人、机构、文化、国家）的共同的“思想”。

相对来说，这个全新的全局概念中的一些元素是比较平凡的。我们可以应用一些简单的数据挖掘技术来合并一些表明上看起来无关的 Web 源信息，从而创造出一种看待世界的全新的方式（这种类型的应用通常被称为混搭（mashup））。例如，从 maps. google. com 中获得的信息可以和 SafeHomeAssured. com 数据库中的信息进行结合生成一幅城市地图，图中显示一个城市中 SafeHome 监控的每一个住宅的位置。

全局理解的其他方面也是很有诱惑力的，但是实现的难度大得多。例如，对于先进的搜索引擎来说，也许可能通过检测全部 Web 源（例如，博客、在线媒体、聊天室、商业数据源、在线的技术杂志、娱乐站点）来（为商业、娱乐、政治）获得发展趋势，而这些趋势通过一两个孤立的 Web 源是不可能得到的。CPI 公司可以使用基于 Web 的数据挖掘技术来收集某个国家的城市中附近地区的与犯罪相关的统计信息，然后使用基于 Web 的人口统计数据把附近地区的一些家庭作为目标。他们是 SafeHome 的潜在买家，最终，以这些家庭为重点目标开展市场计划。

今天，这个因特网海洋冲洗着我们每个人。我们在网上冲浪（使用这个术语非常有趣），去查找数据、信息和知识。但是我们几乎从未对海洋表面下涌动的意识有过一瞥。随着时间的推移，我们步入了 21 世纪，Web 工程师们将开始去创造一些可以让我们所有人以一种全新的方式获取数据、信息和知识的系统，而不仅仅是在表面获取。

这本书的大部分内容都是关于原理的，但是是一种更实用的原理。在后面的章节里，我们将讨论 Web 工程——一个可以帮助我们创建 WebApp 的框架，它将最终指导我们走向全局的概念。

参考文献

[Dar00] Dart, S., *Configuration Management: The Missing Link in Web Engineering*, Artech House, 2000.

第2章 Web工程

你想创建一个WebApp吗？当然，你可以使用在第1章的开始讨论过的旧方法——使用非正式、紧迫、直觉和艺术的混合来创建一个WebApp。如果进展顺利，你和同事将成为英雄，同时将产生一个有意义的WebApp。

然而，事情并非总是被很好地完成，尤其是当你的方法完全依赖于非正式、紧迫、直觉和艺术的时候。当这种情况真的发生的时候，“英雄”这个词汇就会分崩离析。WebApp也许并不会做它该做的事情，可能会延期交付甚至无法交付，或者难以（甚至不可能）在一个可以被急速的Web世界所接受的时间框架内进行纠错、适应和改进。

如果你使用旧式WebApp开发的基本原理，那么你将冒很大的风险。如果这仅仅涉及你个人，那么就继续向前，做一个冒险家——投掷骰子。我们对此没有异议。但是它很少是仅仅关于你个人的。客户想要一个可靠的、可扩展的、功能性的WebApp来满足他们的需要。管理层（当你为一个商务机构、一个教育机构或者政府工作时）很可能已经把一个WebApp的存在作为了一个更大的商业战略的一部分。你的同事依赖于WebApp的及时交付，以和他们正在开发的系统和过程相一致。人们需要的是可以工作的WebApp。他们并不想承担太大的风险。

有一种旧方法的替代方法——一种在创建高质量的WebApp时可以降低（不是消除）风险，并具有更高的成功的可能性的方法。这个替代品就是Web工程（Web engineering，WebE）。

2.1 什么是Web工程

我们简明扼要地回答这一节的标题所提出的问题：Web工程主张用一种敏捷（agile）而规范的框架来构建高质量的WebApp。这看起来足够简单，但是很重要的一点是，你要理解我们的答案中的两个关键词：敏捷和框架。

2.1.1 敏捷意味着什么

Web工程师必须明白，现代商业要求适应性、商业战略和规则的快速变化、管理需要接近实时的响应（即使这样的要求是完全不合理的），而且即使利益相关者（stakeholder）[⊖]要求快速交付，他们也会不断改变想法。顾客关注的是，当他们需要WebApp的时候，就交付给他们，而不是你创建一个可交付的WebApp而付出的努力。记住这些，一个WebE团队必须要强调敏捷性（agility）。Ivar Jacobson［Jac02］为这一概念提供了有益的讨论：

> 一个敏捷团队是一个灵活的团队，可以恰当地响应变化。变化就是软件开发自身很大的一部分。构建中的软件本身的变化、团队成员的变化、使用新技术带来的变化等都会对开发的软件产品以及创建软件产品的项目本身产生影响。我们拥抱“支持变化”，它应当存在于软件开发的任何过程中，因为它是软件的心脏与灵魂。敏捷团队能够认识到软件是由团队中的所有人所共同开发的，这些人的技能和合作能力是项目成功的关键所在。

⊖ 利益相关者（stakeholder）是在项目上分享利益的人，包括业务经理、最终用户、Web工程师、支持人员等类似的人员。Rob Thomsett曾开玩笑说：“利益相关者就是掌握巨额精明投资的人。如果你不顾忌你的利益相关者，那么投资将会终结”。

在 Jacobson 的观点中，无处不在的变化是敏捷性的基本驱动力。Web 工程师要适应 Jacobson 所描述的快速变化，就必须要加快步伐。

2.1.2 什么是 WebE 框架

框架（Framework）[⊖]定义了若干可以被应用到所有 WebApp 项目中的框架活动（activity），为完整的 Web 工程过程建立了基础，而不考虑项目的复杂性。此外，框架还包含了一组适用于整个 WebE 过程的保护性活动（umbrella activity）。

如图 2-1 所示，每一个框架活动都是由一组 Web 工程动作（Web engineering action）组成的，每个动作都是一系列相关的用来产生一件工作产品（例如，设计就是一种 WebE 动作）的任务的集合。每一个动作都是由多个单独的工作任务（work task）组成，每个工作任务用来实现一部分特定动作所定义的工作。

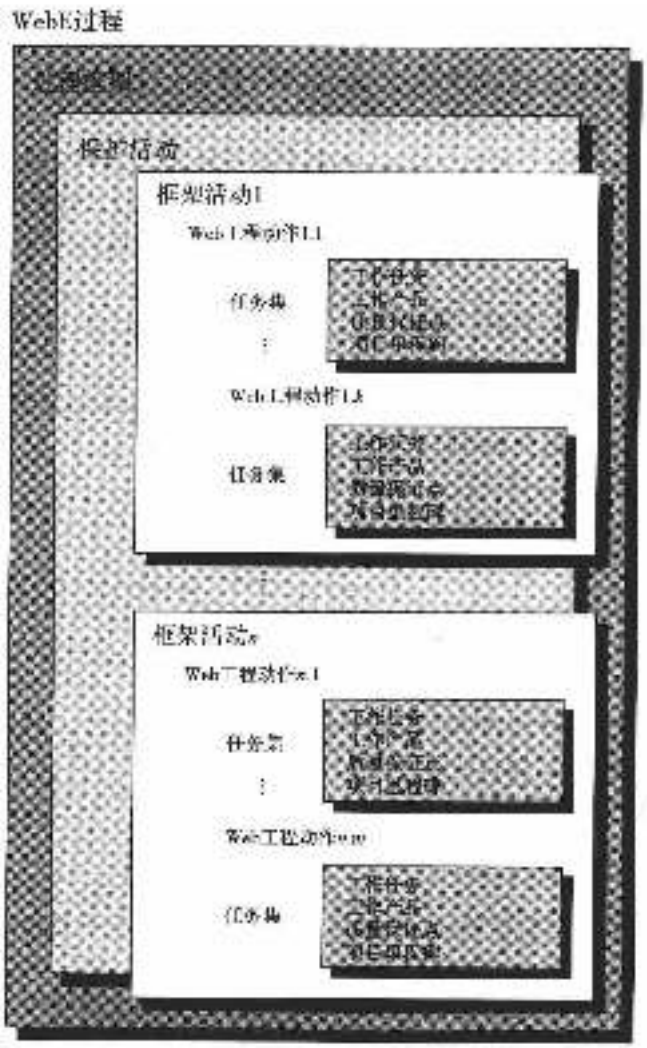


图 2-1 WebE 过程框架

⊖ 术语过程（process）、过程模型（process model）和过程框架（process framework）也会在这个上下文中使用。

下面的 WebE 活动是一个通用框架（generic framework）的一部分，可以应用于绝大多数的 WebApp 工程：

沟通（communication）——包含了与客户（和其他的利益相关者）之间大量的交流与协作，还包括需求获取和其他的相关活动。

计划（planning）——为 WebE 工作建立一个增量计划（incremental plan）[⊖]。它描述了将要发生的 WebE 动作、将要实施的技术任务、可能的风险、将需要的资源、将要生产的工作产品以及一个工作进度计划。

建模（modeling）——包含了有助于开发者和客户更好地理解 WebApp 需求的模型的创建，以及实现这些需求所进行的设计。

构建（construction）——把 HTML、XML、Java 以及类似代码的生成和用于发现代码中错误的测试结合起来。

部署（deployment）——把一个 WebApp 的增量[⊜]交付给客户进行评估，并基于评估提供反馈。

这五个通用的框架活动可以用于任何大小和复杂性的 WebApp 的开发过程。不同案例之间的框架细节可能有很大的不同，但是框架活动都一样。

在图 2-1 中，每一个 Web 工程动作都由一组任务进行表示，这些任务中的每一个都是由 Web 工程工作任务、相关工作产品、质量保证点和项目里程碑等组成的。选择最能适应项目需求和团队特征的任务集。这就意味着一个 Web 工程动作（例如，需求获取）可以适应于具体的 WebApp 工程需求和项目团队的特点。

保护性活动（例如，风险管理、质量保证、内容管理）应用于 WebE 全过程，在本书的后面将对其进行详细讨论。

对任何框架的明智的应用都必须认识到，（对问题、过程、团队和组织文化的）适应性是成功的关键。适应性影响到下面所有的框架特征：

- 活动、动作和任务的整体流程，以及它们之间的依赖关系。
- 每一个框架活动中的工作任务的确定程度。
- 工作产品的识别和需求程度。
- 质量保证活动的应用方式。
- 项目跟踪和控制活动的应用方式。
- 流程描述的全局的细化和严格程度。
- 客户和其他的利益相关者对项目的参与程度。
- 赋予软件项目团队的自主权的级别。
- 团队组织和角色的明确程度。

2.1.3 适应框架时应该遵循哪些原则

WebE 的适应性框架应当强调项目的敏捷性，同时要遵循敏捷开发联盟［Agi03］采用的如下 12 条敏捷性原则：

- 我们最优先考虑的事情是，通过尽早和持续地交付有价值的软件来使客户满意。
- 即使是在开发的后期，也欢迎需求的变更。敏捷过程利用变更为客户创造竞争优势。

⊖ 增量的计划假设 WebApp 将会以一系列的“增量”进行交付，每次的交付都会不断地提供更加完善的需求集合。

⊜ 一个 WebApp 增量把所选定的内容和功能交付给最终用户，后续的增量将对前面提交的内容和功能进行扩展，直到整个 WebApp 部署完毕。

- 频繁地交付可工作的软件，交付的时间间隔可以是几周到几个月，交付的时间间隔越短越好。
- 在项目的整个过程中，业务人员和开发人员必须天天都在一起工作。
- 围绕受激励的个人来构建项目，给他们提供所需的环境和支持，并信任他们可以把工作完成。
- 在项目开发团队内部及团队之间，表达信息的最有效果和效率的方法是面对面的交谈。
- 可工作的软件是进度的首要度量。
- 敏捷过程提倡可持续的开发。发起者、开发者和用户应当能够保持一种长期且稳定的速度。
- 持续关注杰出的技能和好的设计可以提高敏捷性。
- 简单（使未完成的工作最大化的艺术）是必要的。
- 最好的架构、需求和设计都来自于自组织（self-organizing）的团队。
- 每隔一定时间，团队反省如何提高效率，然后相应地调整和改变自己的行为。

2.1.4 旧方法有什么价值吗

有理由问一问，非正式、紧迫、直觉和艺术在 Web 工程中是否有用武之地。答案绝对是肯定的。但是每种影响力都应当被一种原理加以弱化，以减小风险，同时增加成功的可能性。

敏捷性鼓励非正式，同时认可每一个工业质量的 WebApp 的开发过程都是与紧迫感相关的。每一个 WebE 团队都应当把通用框架最好地适应于手边的问题，同时凭借直觉和过去的经验来指导产生适应的方式。作为适用框架一部分的 WebE 动作和任务采用定义好的技术方法（用于需求分析、设计、代码生成和测试）。然而，只有当（这些方法产生的）技术和（每一位高技能的 Web 工程师带给所完成的工作的）艺术进行紧密结合之后，这些方法才能成功。

2.2 Web 工程的组件

我们在本章前面的讨论表明，Web 工程涵盖了整个软件工程实践和过程流的范围，但是它是以把过程和每个实践适应和提取到 WebApp 的特定属性和特征的方式来完成的。

希望你已经对软件工程的实践和过程流有了一些了解，但是为了防止你不是足够了解，我们先花一点儿时间来对它们进行讨论。

2.2.1 软件工程如何发挥作用

在 IEEE Software［Pre98］出版的一个虚拟的圆桌会议中，Roger 指出了 Web 工程和软件工程的结合点：

> 在我看来，任何重要的产品或者系统都是值得工程化的。在你开始构建它之前，你最好先理解问题，设计一个可行的方案，以一种可靠的方式实现，并且彻底地进行测试。在你工作的时候，最好能够控制它的变更，并有一些机制来保证最终结果的质量。有很多 Web 开发者对此并无争论，只是认为他们的世界确实不同，并且认为传统的软件工程方法完全不能应用。

他的意思是，软件工程的原理、概念和方法可以应用到 Web 开发中，但是在应用时需要的是一种和在基于传统软件的系统开发中使用的稍微不同的方法。

软件工程[1]是将一套原理、过程、方法的集合和一个工具集应用于软件开发。然而，软件工程仍然至少和一些（常常是直言的）软件开发者之间存在公共关系方面的问题。一些开发者认为软件工程冗长而缓慢，认为它就是产生文档——很多文档。他们认为项目计划和项目管理是对团队的约束，增加了软件团队的负担。问题是：他们错了。如果软件工程应用不当，它可能就会成为一些人所担心的那样，但是如果它是以一种敏捷的方式加以应用的，对于应用了它的系统来说，它只会有助于改进系统的质量，并加快系统的交付速度。

软件工程是一种层次化的技术。如图2-2所示，它的基础是一个组织对质量的承诺[2]——一个促进持续过程改进文化的承诺。正是这种文化最终引导了人们开发更加有效的软件工程的方法。



图2-2 软件工程层次

过程（process）层是把各个技术层次结合在一起的粘合剂，使得合理而及时的计算机软件开发成为可能。过程构成了软件项目管理控制的基础，并且建立了一个环境。在这个环境中，技术方法被应用，工作产品（例如，模型和文档）被生产，里程碑被建立，质量被保证并且变更被合适地管理。

软件工程的方法（method）为构建软件提供技术解决途径。方法包含了很广的动作和任务的范围，包括沟通、需求分析、设计建模、程序构建、测试和支持。这些方法依赖于一系列的基本原则，这些原则涵盖了技术的各个领域，并包括了建模活动和其他的描述性技术。

软件工程的工具（tool）为过程和方法提供了自动化或半自动化的支持。当这些工具被集成，使得一个工具生成的信息可以被其他的工具使用时，一个支持软件工程的自动化环境就建立起来了。

2.2.2 为什么WebE过程的敏捷性如此重要

我们已经提到过一个WebE过程模型（在本章的前面称之为框架）应当是敏捷的。这就暗示了一种开发周期很短的高效工程方法。每一个周期都是以一个WebApp增量的部署而结束。Aoyama [Aoy98] 以如下的方式描述了敏捷方法的动机：

> 因特网把软件开发最优先考虑的事情从做什么变成了什么时候做。更短的上市时间成了领先企业所争取的竞争优势。因此，缩短开发周期就成为了软件工程目前最重要的任务之一。

虽然说较短的周期时间支配着开发思想，但是很重要的一点是，要认识到WebE框架必须定义在一个过程内，这个过程必须：1）接受变化；2）鼓励创造性、开发人员的独立性以及和WebApp的利益相关者之间的密切沟通；3）采用多个小的开发团队来构建系统；4）强制使用短的开发周期进行增量开发。

在第3章，我们将讨论可以为很多WebE项目很好地服务的过程框架的细节。现在，你应该

---

[1] 如果你想在这对软件工程的更全面的讨论，我们推荐你去看《软件工程：实践者的研究方法》[Pre09]。

[2] 全面质量管理（Total quality management）、六西格玛和ISO 9001是旨在促进质量文化的全面质量体系。

认识到敏捷性将是 Web 工程工作的基本原理。但是你同时应当理解，对敏捷开发的重视绝不会削弱对一个有原则的工程方法的需求。

2.2.3 过程框架内有哪些 WebE 方法

WebE 方法大体上包含了一组技术性任务，这些任务使 Web 工程师能够理解并描述 WebApp 的特征，从而构建出高质量的 WebApp。WebE 方法（将在第 6 到第 15 章进行详细描述）可以按如下的方式进行分类：

沟通方法（communication method）。定义了便于 Web 工程师和其他的 WebApp 利益相关者（例如，最终用户、业务客户、问题域专家、内容设计者、团队负责人、项目经理）之间进行沟通的方法。沟通技术在需求获取和评估一个 WebApp 增量时显得尤为重要。

需求分析方法（requirement analysis method）。为如下的几个方面提供了基础：理解一个 WebApp 要交付的内容，提供给最终用户的功能，以及通过 WebApp 内的导航时，各种类型的用户所需要的交互模式。

设计方法（design method）。包括了一系列的设计技术，用来描述 WebApp 的内容、应用和信息架构、界面设计和导航结构。

构建方法（construction method）。把一套各种各样的语言、工具和相关的技术应用于创建 WebApp 的内容和功能之中。

测试方法（testing method）。包括对内容和设计模型的技术评审和各种各样的测试技术。这些测试技术针对于组件级和架构级问题、导航测试、可用性测试、安全性测试和配置测试。

除了刚刚列举的技术方法，一系列的保护性活动（以及相关的方法）对一个成功的 Web 工程来说也是必不可少的。其中包括项目管理技术（例如，评估、计划进度、风险分析）、软件配置管理技术和评审技术。[⊖]

2.2.4 工具和技术不就是 Web 工程的一切吗

工具和技术可以增强一个技术专家构建基于计算机的系统的能力。如果使用得当，好的工具可以让我们更快地工作，创造出更高质量的最终产品。这就是为什么每一代的技术专家都会青睐于工具和技术。WebApp 的开发者也不例外。

但是工具和技术不能滥用。如果你没有真正理解问题所在，如果你没有办法来适应一定会发生的变化，如果你没有花一定的时间设计出一个可用的方案，如果你无意保证已经生成的“方案”（使用功能强大的工具）满足了你的利益相关者的要求，那么你就是在滥用。灾难就这样形成了。

我们想说的是：*工具和技术非常重要，但是，只有当它们用在 Web 过程的敏捷框架之内，并与那些已经证明过的用于理解问题、设计解决方案和进行全面测试的方法联合使用时，才能够很好地工作。*

在过去的十年里，随着 WebApp 变得更加复杂和普及，已经开发出了大量的工具和技术。这些技术包括广泛的内容描述和建模语言（例如，HTML、VRML、XML）、编程语言（例如，java）、基于组件的开发资源（例如，CORBA、COM、ActiveX、.NET）、浏览器、多媒体工具、网站编写工具、数据库连接工具、安全工具、服务器和服务器实用工具，以及网站管理和分析工具。

⊖ 这些技术，如提到的保护性活动，将在第 4 章和第 16 章中进行讨论。

对 Web 工程的一些更重要的工具和技术的概述将在第 14 章中予以描述。对于更全面的讨论，你可以访问如下的一些站点："Web Developer's Virtual Encyclopedia"（www.wdvl.com）、"WebDeveloper"（www.webdevelop.com）、"Developer Shed"（www.devshed.com）、"Webknowhow.net"（www.webknowhow.net）或者"WebReference"（www.webreference.com）。



对一个过程达成一致

场景：项目启动之前，CPI 公司的 Web 开发项目组的会议室。

参与者：技术经理、WebE 团队成员

会话：

技术经理（对全组说）：让我们简要总结一下。我们已经提到了构建 SafeHomeAssured.com。它是最大的，当然也是我们曾经尝试构建的最显著的 WebApp。毫无疑问，我们要做很多工作来简单地定义这件事情，但是我希望你们开始考虑将如何去着手这个项目的工程部分。

成员 1：看起来我们在过去的 Web 项目的方法中好像组织得相当混乱。

成员 2：我不同意，我们一直都在推出产品。

技术经理：对，但是也不是没有遇到麻烦，而且这个项目看起来比我们过去所做的都要大而且复杂。

成员 3：不要那么悲观，但是我同意……我们应用于过去的项目的特定方法在这里可能并不适用，尤其是当我们的时间非常紧的时候。

技术经理（微笑）：我想让我们的方法更专业一些。上周我去参加了一个短期课程，学到了很多关于 Web 工程的知识……好东西。在这里我们需要一个过程。

成员 2（皱了下眉头）：过程？我的工作是去创建 WebApp，而不是来撰写文档。

技术经理：在你否定我之前，请给它一个机会。我的意思是……［他继续描述本章所描述的过程框架，以及与此相关的敏捷 Web 工程原理。］所以不论如何，在我看来，用简单易行的活动来完成一个敏捷过程是可能的，而且它不会是"没有一大堆的文档"。

成员 1：是的，敏捷性听起来很好，还有那 5 个步骤，嗯，活动，你提到的只是一些常识。

成员 2：我同意，它们都是常识，所以我认为我们不需要去执行一个过程。

成员 3（笑）：是的，但是常识不是总是很普遍的。我们为什么不给它一次机会呢？

2.3 Web 工程的最佳实践

在进入本书的核心章节之前，一些很明显的讨论是非常合适的。我们认识到，你可能选择不去使用在本书的后续部分将要详细讨论的 WebE 过程框架和方法。我们知道，Web 工程团队有时顾着巨大的时间压力，并会设法走捷径（即使这些做法是不被提倡的，而且这样做会导致更多的开发工作量，而不是更少）。我们也接受这样的事实，一些 WebE 团队想去保持非正式性，同时仍然采用一个过程框架和已定义的方法。我们认为诸如此类的解释是错误的，但毕竟那是你的决定。

我们所希望的是，你能够在剩余的章节上花费足够的时间来评估我们所描述的每一个实践，然后接受那些你认为可应用的，拒绝那些你认为不可应用的。至少，我们希望当你和同事创建满足行业质量标准的 WebApp 时，将会采用如下的最佳实践：

1. 即使 WebApp 的细节很模糊，也要花一些时间去理解业务要求和产品目标。很多 WebApp 开发者错误地认为，（非常普通的）模糊的需求使他们可以不用去确认将要开发的系统只有一个

合理的商业目标。最终的结果（常常）是好的技术工作因为错误的原因为错误的用户构建了错误的系统。如果利益相关者不能阐明 WebApp 的业务需求，就要加倍谨慎。如果利益相关者不能清楚地确定产品（WebApp）的一系列明确的目标，就不要进行下去，直到他们确定之后再继续开发。详细信息参见第 4 章和第 5 章。

2. **使用基于场景的方法来描述用户如何与 WebApp 进行交互**。应该说服利益相关者去设计场景（第 4、5 和 7 章），通过场景来反映各种用户如何与 WebApp 进行交互。然后，这些场景可以被用于：1）项目计划和跟踪；2）指导分析和设计建模；3）作为测试设计时的重要输入。

3. **制定一个项目计划，即使这个计划很粗略**。基于所有的利益相关者都能接受的过程框架（第 3 章），建立项目计划（第 5 章）。因为项目的时间线很短，所以要使用一个细粒度的时间表。例如，在很多情况下，项目的进度计划和跟踪应当以天为单位。

4. **花费一些时间来对你将要构建的系统进行建模**。通常，综合性的分析和设计文档并不会作为 Web 工程工作的一部分进行开发。然而，有针对性的图形化模型（第 6 章 ~ 第 12 章）确实可以阐明重要的工程问题。

5. **对模型进行一致性和质量的评审**。结对走查和其他类型的评审（第 5 章）都应当贯穿于整个 WebE 项目。在评审上花费的时间会带来很重要的价值，因为它通常会减少返工，并产生高质量的 WebApp，从而提高客户的满意度。

6. **使用那些能够让你尽可能多地使用可重用组件来构建系统的工具和技术**。有各种各样的 WebApp 工具可用于 WebApp 构建（第 14 章）的几乎每一个方面。在这些工具中，有很多都能够使 Web 工程师可以使用可重用的组件来构建应用的重要部分。

7. **如果可以重用，就不要重新开发**。已经开发出了很多针对 WebApp 的设计模式。这些模式使 WebE 团队可以使用一些已经证明过的模板来快速开发架构的、导航和组件级的细节。

8. **不要依赖于早期用户来调试 WebApp——在发布系统之前，设计全面的测试并执行这些测试**。WebApp 的用户通常都只会给 WebApp 一次机会，一旦 WebApp 执行错误，用户就会到别的地方去——永远都不会再回来。正因为这样，即使必须要拖延最终期限，“先测试，后部署”也应该是首要的原则。详细信息参见第 15 章。

2.4 本章小结

随着我们开始考虑 Web 工程以及作为其基础的过程框架，敏捷性成了一个非常重要的概念。作为 Web 工程师，你必须要加快脚步。你的工作是去快速地构建高质量的 WebApp。但是在你这样做的时候，必须要适应伴随着工作的进行而持续演化的系统；必须把一个通用框架适应到每一个 WebApp 增量之上，并把一系列的 WebE 方法和工具集成到这个框架之内；必须遵循一系列敏捷性原则和一组最佳实践来指导团队走向成功。

当我们进入到第 3 章时，你将学到更多关于 WebE 过程框架的知识。更重要的是，我们提出了一系列的 WebE 动作和任务，作为把过程框架适应到问题、人员和项目的特定需求的一个基础。

参考文献

[Agi03] The Agile Alliance Home Page, www. agilealliance. org/home (accessed July 24, 2007)

[Aoy98] Aoyama, M., “Web-based Agile Software Development,” *IEEE Computer*, November/December 1998, pp. 56-65.

[Fow01] Fowler, M., and J. Highsmith, “The Agile Manifesto,” *Software Development Magazine*,

August, 2001, www. sdmagazine. com/documents/s = 844/sdm0108a/0108a. htm (accessed July 24, 2007)

[Jac02] Jacobson, I., "A Resounding 'Yes' to Agile Processes——But Also More," *Cutter IT Journal*, vol. 15, no. 1, January 2002, pp. 18-24.

[Pre98] Pressman, R. S. (moderator), "Can Internet-Based Applications Be Engineered?" *IEEE Software*, September 1998, pp. 104-110.

[Pre09] Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 7th ed., McGraw-Hill, 2009.

第3章　Web工程过程

一些人认为，一个组织可以事先为Web应用开发完整地定义开发过程；把它放在书架上（打比方）；然后在一个新的Web工程项目出现时把它拿下来，拍掉灰尘并应用它。不幸的是，事情并不是这样发展的。

的确，你和同事可以开发一个有效的WebE过程框架，并且把它应用到随后的每一个WebApp项目。但正如第2章提到的那样，这个框架必须要适应问题、项目以及明确说明需求和具体做工作的人员的特有特征。

一个通用的WebE过程框架为你提供了理解问题是什么的能力（从业务背景和技术两个方面）。有了这些基本的理解，你就可以精化或者修改基本过程，以满足手边问题的特定需要。你需要考虑开发团队的文化、客户的要求、问题需求中的稳定度以及管理者们（对监督、投放市场的时间和质量）的要求。

随着工作的开始，你要不断地对过程进行修正来确保：1）它并没有妨碍敏捷性；2）它确实适应WebE团队成员的需求；3）它生产的中间工作产品都是可以让你快速交付计划好的WebApp增量的；4）它允许你连续地对工作进行质量评估；5）它可以让你适应那些已经开始影响你的工作进度和方法的变化。

从本质上来说，过程框架保持不变，但是你应用它的方式将随着每一个项目和项目团队发生变化。随着每一个WebApp增量的创建，框架会被反复应用。

3.1　定义框架

在为WebE定义一个过程框架之前，我们必须重申在大多数WebApp项目中都会遇到的一些事实：

1. **需求随着时间演变**。当你开始一个WebApp项目的时候，业务策略的一些元素、交付的内容和功能、互操作性问题以及问题的其他很多方面都可能有一些不确定性。

2. **变化会经常发生**。因为不确定性是大多数WebApp项目的一个固有组成部分，所以需求的变化是很寻常的。另外，用户的反馈（基于对已交付增量的评估）和不断变化的业务环境都可能引发变化。

3. **时间线非常短**。这减少了对大量工程文档的创建，但是这并不排除一个简单的事实，那就是必须以某种方式对问题分析、设计和测试都进行文档化。

由于这些事实，WebApp常常都是增量式地进行交付的。也就是说，每当一个WebApp增量被工程化和交付（图3-1）的时候，框架活动就会重复进行。而且，第2章中描述的敏捷性原则应当加以应用。然而，我们也不能教条地应用这些原则，有时可以内在地采用它们，没有必要把每一个原则都应用到你所选择的过程框架中。

考虑到这些问题，参照图3-2，我们对第2章中描述的Web工程框架展开讨论。

沟通（communication）。在WebE过程中，沟通的特点在于如下3种WebE动作：规划（formulation）、提取（elicitation）和协商（negotiation）。规划定义了WebApp的业务和组织环境。除此之外，还包括确认利益相关者，预测业务环境或者业务需求方面的潜在变化，以及定义We-

bApp 和其他业务应用、数据库和功能之间的集成。提取是一个涉及所有利益相关者[1]的需求搜集活动。其目的是使用收集到的最有价值的信息描述 WebApp 要解决的问题（以及 WebApp 的基本需求）。另外，会试识别出不确定的领域和可能发生变更的地方。最后，通常都需要协商来消解项目的各个利益相关者之间的差别。

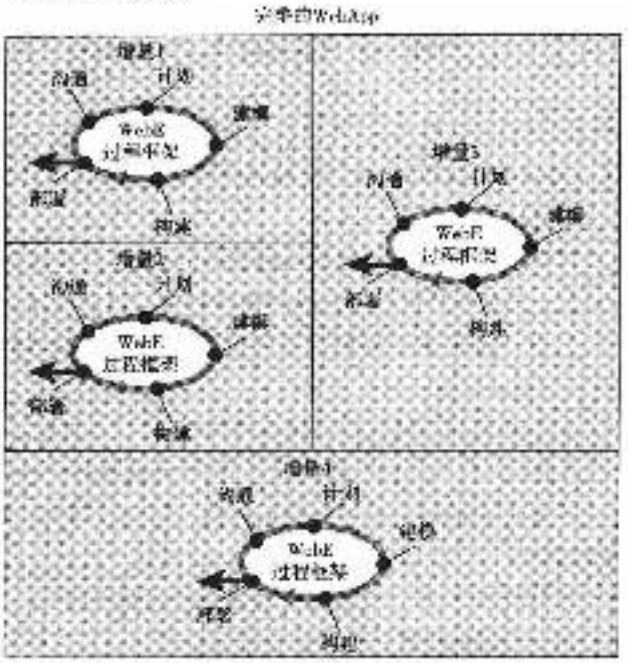


图 3-1 在 4 个增量中交付的一个 WebApp

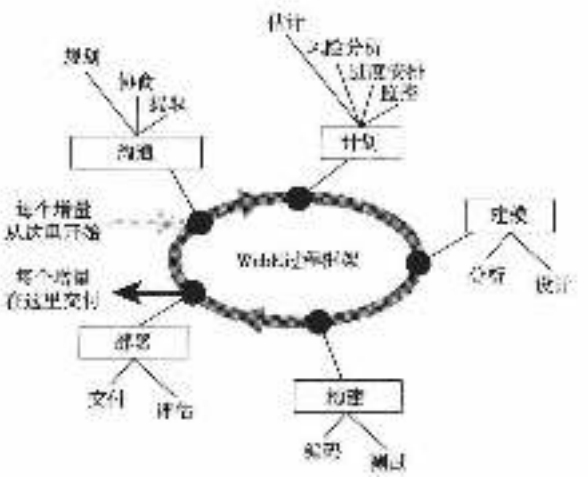


图 3-2 对 WebE 动作的过程流

计划（planning）。确定 WebApp 增量的总体数目[2]，同时为将要部署的下一个 WebApp 增量产生大体的项目计划（第 5 章）。估计增量所需的资源，考虑风险，选择任务并安排其进度。同

[1] 就像我们在第 2 章中提到的，利益相关者是任何一个在 WebApp 上有既得利益的人。利益相关者可参与定义需求（如市场团队人员）、特别项目的业务方面（如经济资源），使用 WebApp（如最终用户）或者开发 WebApp 本身（如 WebE 团队）。

[2] 随着从早期部署的增量中得到反馈，将要交付的增量为数目可能会发生变化。

时项目的跟踪和监控也将开始。在大多数情况下，计划的工作产品由任务定义和 WebApp 增量开发所需时间段（通常以周计）的时间线进度表组成。

建模（modeling）。传统软件工程的分析和设计任务被应用到 WebApp 开发中，合并并且最终融入到 WebE 建模活动之中（第 6 章～第 13 章）。其目的是开发用来定义需求的敏捷的分析和设计模型，同时展示一个可以满足需求的 WebApp。

构建（construction）。WebE 工具和技术（第 14 章）用于构建已经模型化的 WebApp。一旦构建出 WebApp 的增量，就会执行一系列快速测试，以确保发现设计中（如内容、架构、界面和导航）存在的错误，并对 WebApp 的其他特征进行附加测试。

部署（deployment）。WebApp 对它的操作环境进行配置，然后交付给最终用户，评估阶段就开始了。评估结果反馈给 WebE 团队，必要时对增量进行修改。

这 5 个 WebE 框架活动应用在图 3-2 所示的增量过程流中。

3.2 增量过程流

在第 1 章里，我们把你介绍给了 CPI 公司和它的 SafeHome 安全产品以及 SafeHomeAssured. com——一个复杂的 WebApp，它将作为产品市场和销售的中心，同时也是一个新一代安全监控服务的焦点。现在，除了一个大概的策略和一些好的想法，别无它物。

你已经被授权组建一支构建 WebApp 的 WebE 团队。需求是紧迫的，同时一个 Web 展示必须要尽快完成并开始运行。很明显，你必须要敏捷。还很明显的是，当你开始构造 WebApp 的时候，事情将发生（可能急剧）变化。

你决定使用那些为通用过程框架所定义的活动（沟通、计划、建模、构建和部署），而交付 WebApp 的唯一合理的方式就是增量。这些增量可能是什么呢？

在第 4 章中，我们将讨论用于定义可交付 WebApp 增量的一个特定技术。就目前而言，我们将只注意这些增量作为第一次迭代中的沟通活动的一部分而被定义。增量列表可能会随着后续的每一次迭代而得到进一步完善。

3.2.1 框架活动是如何进行的

再一次参阅图 3-1 和图 3-2，迭代过程流将被应用于每一个增量。最先开始的一次迭代主要关注定义总体的 WebApp 需求和确定在后续迭代中将要部署的增量。

第一次迭代。第一次沟通活动被初始化。目的是定义业务环境，建立总体需求，创建一系列使用场景，协商利益相关者之间有冲突的需求，然后从这些信息中得到将要交付的 WebApp 增量集。进行这些任务的详细指南将在第 4 章和第 5 章中展示。现在，我们假设第一次将要部署的增量是一个介绍 CPI 公司和它的产品的信息型 WebApp。利益相关者表示，这个信息型 WebApp 必须在一周内部署。

第二次迭代。即使时间非常短，你也要让自己受制于 Web 工程过程框架。第二天早上，你通过主持跟利益相关者的一个 1 个小时的会议，开始了第一个增量的沟通活动。你对这个增量的需求有了相当好的了解。你欣慰地认识到，增量所需的全部内容都在已经存在的文件中，同时一个内部的图形美工师已经开始设计 CPI 想要的美学外观。会议结束后，你检查你的笔记：

徽标和图形——需要美学设计。
一或两段的介绍。
CPI 公司的经营宗旨（文件已存在）
给访问者的话（某人将在明天写）

基本的导航条看起来就像……
关于公司
我们的产品
住宅安全产品（在下一层分级）
监控服务（一个列表）
我们的技术（新的传感器）
联系我们
其他事项：
信息内容将随着时间而变化。
这个“主页”将会是后续增量所需要的内容和功能的导航起点。

SafeHomeAssured. com 的第一个增量的需求是显而易见的。

因为你已经决定了遵循 WebE 过程框架，你的下一个活动就是做计划。由于所有的内容都已经有了，而且现阶段也没有要实现的功能，因此，你决定自己做这个工作（同时还想着为后续增量组织一个小的 WebE 团队）。在接下来的半个小时里，你制定了一个简单的时间线：

第一天：为 WebApp 创建一个原型化的布局（一个模型）。
收集并检查所有现存的 CPI 内容和图形。
如果可能，获取利益相关者对原型的反馈。
第二天：以原型为指导，开始增量的构建。
建立导航条。
布置内容的区域。
集成图形、链接等。
测试所有链接的有效性。
检查所有内容的完整性和正确性。
第三天：使用 FTP 把所有文件传输到（一个已经存在的）域。
执行导航测试。
部署：通知被选中的利益相关者，增量已经可用了。
第四天：调查利益相关者的反馈意见。
根据利益相关者的反馈意见作一些修改。

为 SafeHomeAssured. com 的第一个增量所做的计划已经完成了。在接下来的 3 ~ 4 天的时间里，我们将要做建模、构建和部署的工作。

下一次迭代。一旦（第一个增量）部署完成，你就要准备去启动 WebE 过程的下一次迭代。对于 SafeHomeAssured. com 而言，在这个第二次迭代中的沟通活动将确定第二个增量[⊖]的需求（包括内容和功能）。所以你要从头开始这个过程流，为这个增量执行沟通活动。你选择的填充于增量的框架活动中的任务可能和前面增量所执行的任务不同，但是总体过程流保持不变。

随着增量变得越来越复杂，每一个框架活动所需要的工作任务的数量以及这些任务的复杂性很可能会增长。含义很明确：用于一个特定增量的给定框架活动的方法可能和用于下一个增量的同一个活动的方法有所不同。在每一次迭代中，框架活动都必须加以适应和完善。

当通过每一次迭代的时候，敏捷性都应当贯彻始终。你和利益相关者是一个团队，同时团队成员之间互相交谈——沟通，作为每一个增量流的一部分。你和同事应当做计划，但是只要能够

⊖ 出于这个讨论的目的，我们将假设第二个增量交付选择和下载产品说明和相关信息的功能。

提供到达最终结果的路线和跟踪过程的方法就可以了。你应当建模，但是只有当将要创建的增量非常复杂、需求方法或设计方法没有你想要的那么清晰的时候才这样。在部署增量之前测试这个增量，同时还要关注并执行测试。你要听取最终用户的反馈，不仅将它们考虑进已经部署了的增量中的问题，而且在进行到下一个增量时还要用它指导你的工作。你同时将执行保护活动来保证质量、控制变更、管理风险和监督项目。

9.2.2 框架如何进行精化

精化每一个框架活动所需要的 Web 工程动作和任务取决于 Web 工程团队的判断。在有些情况下，框架活动是以非正式的形式执行的。而在其他情况下，团队成员将会定义和执行一个清晰的动作集合。当一个动作（如设计）很复杂的时候，它可能会被进一步精化为一系列的 Web 工程任务（例如，美学设计、内容设计、架构设计、导航设计和组件设计）。

但是如何决定应用哪一个 WebE 任务呢？要回答这个问题，就必须明白一个特定增量的框架活动的范围。比如，沟通是应用到所有 Web 工程项目的一个框架活动。考虑对作为 SafeHomeAssured. com 这个 WebApp 第一部分进行交付的两个不同增量的沟通进行精化。

对第一个可交付的增量（一个信息型 WebApp）你必须理解将要展示的内容，Web 站点所需要的总体美学要求（例如，颜色方案、布局、菜单格式、字体和风格），将要展示的不同信息种类的数量，以及相关的信息。因为你的基本原则强调敏捷性，你想尽可能快地确定信息（毕竟，你只有 5 天的时间去部署 WebApp 增量）。因此，沟通只涉及一个单一的动作：满足客户。

回想我们在本章前面所做的讨论，并没有产生正式的工作产品，同时提问和回答这些任务包含的问题所需要的时间相对较短。

现在考虑 SafeHomeAssured. com 的第二个可交付增量的沟通活动。为了这次讨论，让我们假设第二个增量增加了电子商务的功能。对这个增量而言，沟通活动更加复杂。三个动作——规划、引导和协商，都将被执行。每一个动作的工作任务都将被定义。

你必须理解业务背景（规则），识别电子商务（例如，基于表单处理需求、产品定价需求、运费、税收、特殊的订单）的一组特定的内容和功能需求（提取），并调解利益相关者之间的差别（协商）。你还必须要识别约束和性能需求。

你必须继续和利益相关者开会，但是会议的持续时间和必须要完成的 WebE 任务的数目都将增加。你很可能将要生产一些正式的工作产品（例如，为各类用户设计的使用场景、订单录入表单的一个草图），并且要花一些时间和你的利益相关者一起来检查它们。作为另一种选择，或许可以使用一个已经存在的电子商务模板（我们将推荐这一方法，参看第 13 章），使你能够给利益相关者展示一个原型，由此就加快了沟通任务的进程。



为每一个 WebApp 增量精化 WebE 过程

场景：当 SafeHome 的一个信息型 WebApp 的第一个增量结束时，Web 工程组在会议室

参与者：技术经理、WebE 团队成员

会话：

技术经理（对整个团队说）：看起来我们好像将会在周三的时候让第一个增量上线。

团队成员 2：那是小菜一碟。但是下一个增量看起来更加复杂一些，而且后面的那些增量……

() 保护活动将在本章的后面予以讨论。

团队成员1：有一些事情我不明白。得到第一个增量的需求很简单——简单的交谈，然后使用已有的内容。但是其他的东西就复杂多了，而且我们还没有需求呢。

团队成员3：没错。我们必须要在周四开工，而且还有很多不明确的事情。

技术经理：这就是为什么我们要再一次从沟通开始下一个增量。

团队成员1：但是沟通活动的过程是什么呢？对第一个增量而言，它不过是一个快速的交谈而已。

技术经理：就像我们已经决定的那样。我们把过程（就是沟通活动以及其他活动）调整到即将开始的增量上来。这一次我们必须要更正式一点儿，而且要定义一些特定的沟通任务。

团队成员2：并且我们将在剩下的每一个增量中都使用它们？

技术经理：不，每一次当我们开始一个新的增量的时候，我们都将考虑它的复杂性，然后决定一个看起来合适的任务集。

团队成员3（笑）：敏捷而且可适应。

技术经理：就是。

3.3 WebE框架的一般动作和任务

虽然说适应你自己的Web工程动作集和任务集是必要的，但是必须从某个地方开始。正因为这样，我们提供了一般动作和任务的概述，这些动作和任务可以作为你和团队在精化将要交付的WebApp增量的每一个框架活动时的起点。很重要的一点是，要注意这里展示的每一个活动都会在后续章节中进行更加深入的讨论。

3.3.1 沟通活动应当如何进行精化

如果你不知道目的地，那么就很难到达想要到达的地方和知道你可能何时到达那里。这是常识，但是仍然有很多Web开发者在开始WebApp的开发旅程的时候，没有或者基本没有考虑过他们将要到哪里。也许他们可能最终可以到达一个可接受的目的地，但是他们将走很多弯路，同时在他们的旅途中将遇到很多死胡同。结果就是，他们将会很迟。有时，他们根本就到不了。在其他的情况下，他们到达了，却不知道何时停止，因此就继续不必要地前进。

沟通就是为一个WebApp项目建立“目的地”的活动。对于一个简单的目的地，要确保你知道你的去向，需要的信息动作和任务的数目相对较少。如果目的地比较难以描述，你就需要更加用心地精化沟通活动。下面的任务和相关问题可以让你开始工作：

- **识别业务利益相关者**。准确地说，谁是WebApp的“客户”？什么样的业务人员将作为专家，什么样的业务人员可代表最终用户？谁将作为团队的有用成员？利益相关者之间的一致程度如何？当利益相关者之间出现争执的时候，谁将作为仲裁者？
- **识别用户种类**。有多少种不同类型的用户将和WebApp交互？每一类用户的背景和熟练程度如何？谁将为每一类用户识别特定需求，以及这些需求是什么？每一类用户需要的特殊内容和功能是什么？
- **阐述业务背景**。WebApp如何适应更大的业务策略？策略建立好了吗，已存在的业务规则得到了很好的理解吗？
- **为WebApp定义关键业务目标和目的**。如何从定性和定量两个方面对WebApp的成功进行度量？如果存在多个目标，它们的优先级是什么？不同的利益相关者有不同的目标和目的吗？所有的目标和目的之间相互一致吗？

- **识别问题**。WebApp 解决哪些特定的问题？将为最终用户生产什么信息？什么信息是由最终用户输入的？需要什么样的功能来处理数据？WebApp 使用哪些已经存储的信息？哪些已经存在的系统将和 WebApp 进行互操作？
- **定义信息目标和应用目标**。将为最终用户提供哪些种类的内容？这些内容的状态如何？内容的动态性如何，也就是说，它变化的频率怎样？当使用 WebApp 的时候，将完成哪些功能和用户任务？所需功能的稳定性如何？
- **收集需求**。WebApp 增量将支持哪些用户任务？将开发什么内容？将使用什么样的交互隐喻？WebApp 提供什么样的计算功能？为了提高网络的利用率，将如何对 WebApp 进行配置？需要什么样的导航模式？对增量而言，存在什么约束？必须考虑哪些特殊的性能需求？
- **开发使用场景**。是否考虑了将和增量交互的所有用户种类？使用场景完整且与增量的需求一致吗？使用场景是否需要进一步精化？

就像我们在前面提到的那样，如果目的相对简单，与合适的利益相关者进行非正式的会议以及做笔记就可以完成每一个沟通任务（以及提到的问题）。然而，如果目的更加复杂，可能就不得不为每一个任务实现更加结构化的方法。我们将在第 4 章对这种方法进行更加详细的讨论。

3.3.2 制定一个增量计划需要哪些任务

沟通活动已经为你提供了目的地，现在你准备好为旅行做计划了。你的路线也许很复杂。因此，你决定通过定义*航点*（way point）来逐步地计划行程路线，这将确保前进方向的正确性，并且让你逐步接近最终目标。

事实上，沟通活动的第一次迭代就建立了航点（WebApp 增量），但是，是由计划活动来定义到达每一个航点所需的资源和估计到达航点所需的时间的。在开发一个增量计划的过程中，下面的任务和相关问题应该对你有所帮助：

- **精化你对将要交付的 WebApp 增量的描述**。请求的变更（任何利益相关者提出的）需要对剩下将要交付的增量的数目和定义进行修改吗？如果需要修改，需要对内容和功能进行什么样的改变？对于剩下需要交付的每一个增量，大概需要花费多少工作？每一个增量的预计部署日期是什么？
- **选择现在将要交付的 WebApp 增量**。有足够的关于这个增量的信息来开始其他框架活动吗？对这个增量要交付的内容和功能有清晰的理解吗？对约束和性能问题有清楚的理解吗？所有必要的使用场景都是可用而且完整的吗？
- **估计部署增量所需要的工作量和时间**。建模、构建和部署这个增量将需要多少工作量（人-天）和时间（日历天）？做这项工作需要哪些资源（人、硬件和软件）？
- **评估增量交付的风险**。在这个增量的开发过程中，哪些风险应当加以注意？如何降低那些可能性大、影响大的风险？应当考虑哪些长期的风险？
- **为增量定义开发进度表**。应当如何沿着这个增量的时间线来分配任务？将建立哪些中间的里程碑？
- **确定将要生产的作为每一个框架活动结果的工作产品**。在增量继续进行的过程中，将开发哪些工作产品（例如，书面化的场景、草图、模型、文档）？
- **定义你控制变更的方法**。在其他开发活动的环境中，内容和功能上的变更如何被请求、评估和执行？

- **建立你的质量保证方法**。在对增量进行建模、构建和部署的过程中，团队如何进行质量评估？如果有的话，将进行什么样的复审？使用什么度量？

因为增量常常是在几周而不是几个月的时间里开发出来的，所以我们应当询问为 Web 工程调整计划是否合理。虽然建立到目的地的路线总是一个好的想法，但是花太多的时间来做计划，从而导致你没有时间来到达目的地却并不明智。我们主张为每个增量做*敏捷计划*（agile planning）——一个单一的、面向团队的会议，在会议中，WebE 团队的所有成员都参与到做计划当中。我们同时提倡*适应性计划*（adaptive planning）。随着 WebApp 增量的开发，事情总会发生，因此，计划可能不得不进行修改。

3.3.3 什么是建模

对于 Web 工程过程框架来说，*建模*（modeling）就是创建将要构建的 WebApp 某些方面的一个或多个概念性表示的活动。*概念性表示*包括如下的一种或多种形式：书面文档、草图、概要图形、图形化模型、书面场景、文件或可执行原型，以及可执行代码。[⊖]有两种 Web 工程动作在建模过程中发生：分析和设计。

3.3.4 可以应用哪些分析建模任务

分析的起点是使用在沟通活动中收集到的信息来检查利益相关者的需求。在很多实例中，沟通过程中收集的信息足以作为创建增量的基础，没有必要进行分析建模。然而，当一个增量的需求复杂的时候，有时创建一个模型来精化你对 WebApp 的理解反而是一个很好的主意。大体来说，分析模型的重点在于 WebApp 的内容、交互（包括导航）模式和功能，以及技术配置[⊜]。下面的任务和相关问题应该有助于你决定是否要开发一个分析模型：

- **判断是否需要一个需求模型**。已有的信息（在沟通活动中所收集的）是否提供了足够的关于 WebApp 的内容、需要的交互方式、需要的功能和技术的配置问题这几个方面的细节？使用场景的开发是否足够详细，以指导设计和构建活动？如果这些信息存在而且是完整的，那么就没有必要为增量构建分析建模。如果信息不完整或者暗含了需要进一步检查的复杂度，那么就去做后续的分析建模任务。
- **表示 WebApp 的内容**。将要表示哪些内容？它出自哪里？谁负责获取并开发它？把内容组织成种类的集合可以吗？内容种类之间的关系复杂吗？哪些内容种类是静态的（不会随着用户种类或者输入而变化），哪些是动态的（基于用户种类或者输入产生）？
- **识别内容之间的联系**。一种内容如何与其他的种类关联？每一个内容种类的表单和风格是什么？
- **精化并扩展用户场景**。哪些用户任务将作为这个增量的一部分完成？用户如何完成任务？用户需要哪些信息来完成任务？为了完成一个任务，用户需要提供哪些信息？用户与 WebApp 交互需要哪些步骤，更确切地说，是如何与 WebApp 交互？必须存在哪些功能来使用户完成任务？
- **复审使用场景**。在一个场景中，有不一致或者遗漏吗？每一个场景都足够详细吗？场景和 WebApp 增量中实现的内容和功能一致吗？
- **为复杂场景创建一个交互模型**。如果一个场景中指定的动作序列比较复杂，那么用户任

⊖ 这些概念性表示将在第 6 章～第 13 章中进行更深入的讨论。

⊜ 在这个上下文中，技术配置是指 WebApp 将要驻留的硬件和软件环境。

务和每一个任务需要的内容之间的关系是什么？可以识别哪些从外部可见的状态[⊖]？哪些用户动作可以引起从一个状态到另一个状态的转换？

- **精化界面需求**。WebApp 界面的外观和感觉与已经定义的用户场景一致吗？菜单、布局和导航需要修改吗？
- **识别功能**。WebApp 将为用户完成哪些功能？用户为了使用这些功能需要提供哪些数据？对每一项功能所包含的算法有很好的理解吗？
- **定义约束和性能需求**。约束和性能需求（作为沟通活动的一部分被定义）已经表示得足够详细吗？将实现哪些隐私策略？
- **识别数据库需求**。将访问哪些数据库？数据库接口协议的定义良好吗？将涉及哪些内容种类？

在进行分析建模任务的时候，可以使用一些有用的 WebE 方法和符号。这些将在第 7 章进行讨论。

3.3.5 设计模型的要素有哪些

设计是一个核心的 Web 工程活动。在 20 世纪 90 年代早期，Mitch Kapor[⊖]在《Dr. Dobbs Journal》[Kap91] 上发表了一篇“设计宣言”。我们已经获得了把他关于软件设计的讨论适应到 Web 工程世界的许可：

> 什么是设计？设计是你一身处于在两个世界（技术世界和人类目的的世界），而你试图把这两个世界连接起来……罗马建筑批评家 Vitruvius 提出，设计良好的建筑应该展现出坚固、适用和令人赏心悦目的特点。对于好的［WebApp］同样如此。坚固（firmness）：一个［WebApp］应该不含任何妨碍其功能的缺陷。适用（commodity）：一个［WebApp］应当适合它被期望的目标。赏心悦目（delight）：使用［WebApp］的体验应当是愉快的。这里我们开始介绍［Web 工程］设计理论。

Web 工程设计的目标是创建出一个坚固、适用和赏心悦目的模型或者表示。

当我们创建一个 WebApp 增量的时候，是否总要创建一个设计模型呢？答案是肯定的，但是模型的表现形式对每个增量来说可能是不同的。如果对增量的理解很透彻，而且增量很容易构建，那么唯一的设计模型可能就是一个简单的草图。另一方面，如果增量比较复杂，那么可能就要创建一个更详细的设计模型。这个模型可以考虑如下 WebApp 设计方面中的一些或者全部。

- **界面设计**（interface design）。描述用户界面的结构和组织形式。包括屏幕布局的表示、交互模式的定义以及导航机制的描述。
- **美学设计**（aesthetic design）。也称作图形设计，描述了 WebApp 的“外观和感觉”。包括颜色方案、几何布局、文本大小、字体和位置、图形的使用以及相关的美学决策。
- **内容设计**（content design）。针对作为 WebApp 的一部分进行展现的所有内容，定义其布局、结构和轮廓，建立内容对象之间的关系。
- **导航设计**（navigation design）。针对 WebApp 的全部功能，描述内容对象之间的导航

⊖ 状态是行为的一个外部可见的模式。在 WebApp 环境中，行为模式也许是在屏幕上展示的内容、一个计算功能的初始化里或工作流内的一个特定条件里发生的显著变更。参见第 7 章和第 9 章。

⊖ Mitch Kapor 是一位著名的软件开发者和企业家，他开发了第一个真正成熟的电子制表软件应用程序——Lotus1-2-3。作为因特网 WebApp 的一名创始人，他近期的工作是 Scott Rosenberg 所写的《Dreaming in Code》（Crown Publishers，2007）这本书的主题。

流程。

- 架构设计（architecture design）。确定 WebApp 的全局超媒体结构。
- 组件设计（component design）。开发实现功能组件所需要的详细处理逻辑，以实现一个完整的 WebApp 功能。



建模真的有必要吗？

场景：在 Web 工程开发者范围内的谈话

参与者：WebE 团队的两个成员

会话：

团队成员 2：Doug［项目经理］坚持要我们必须对第二个增量进行建模，可我认为这没有必要。

团队成员 1：嗯，他的真正意思是我们应当对增量中特别复杂或者市场需求比较模糊的那些方面进行建模。

团队成员 2：也许吧，但是这看起来似乎是在浪费时间……除此之外，我们还必须要学习一种新的建模符号，他们叫它什么来着，UML？如果你问我，我会说它只是额外的文书工作。

团队成员 1：但是，如果要是它有助于我们在设计中巩固需求或者发现一些漏洞呢？

团队成员 2：在我们编码实现 WebApp 以后，我们无论如何都会发现那些漏洞。

团队成员 1：我无法确定那是不是真的。要是我们没有建模，然后由此我们得到错误的需求呢？我们编码实现错误的方案，然后让客户尖叫。我们就得返回去重做一次。那是重做，需要时间。

团队成员 2：比建模用的时间还多吗？

团队成员 1（笑）：是的，比建模用的时间还多。

3.3.6 可以应用哪些设计建模任务

一个设计模型对在前面一节中提到的每一个方面的处理程度取决于将要创建的 WebApp 增量的复杂性。[1]在你考虑如何开发一个设计模型的时候，下面的任务和相关问题应该可以帮助你：

- **设计界面（第 9 章）。**交互任务和子任务将如何作为界面的一部分进行展示？需要什么样的界面控制机制（例如，链接、按钮和菜单）？如何在一个 Web 页面上放置控制机制？设计和每一个使用场景都一致吗？
- **为 WebApp 设计美学（第 9 章）。**[2]如何实现页面布局？颜色和表单会随着环境变化而变化吗？应当如何放置和展示导航机制？是不是所有的徽标、图形、图像和背景都已经实现了而且是可用的？美学设计在不同增量之间一致吗？
- **设计导航方案（第 10 章）。**需要什么样的导航链接和节点？将使用什么样的导航约定和辅助？全局导航流程定义了吗？导航机制与界面需求和设计是否对应？导航针对不同的用户种类进行过优化了吗？导航语义支持每一个使用场景吗？
- **设计 WebApp 架构（第 10 章）。**对内容和功能来讲，将使用什么样的架构风格？
- **设计内容和支持内容的结构（第 10 章）。**哪些内容必须作为 WebApp 增量的一部分被设计？为了实现功能或者展示内容，需要哪些大的数据结构和数据库？已存在的那些数据库的界面是否定义在设计层？

---

[1] 在大多数情况下，设计模型包括这里讨论的设计为图中的一部分，而不仅是全部内容。

[2] 值得注意的是，工作级质量 WebApp 的美学设计应当交给专业人士（如图形美工师）。

- **设计功能组件（第 11 章）**。必须要开发哪些组件？是不是所有的算法都已经定义好了？当处理需要的时候，合适的内容是否可用？
- **选择合适的设计模式（第 13 章）**。什么样的架构模式适合于信息型空间？导航设计问题是否可以使用已存在的模式？交互模式是否可以作为界面设计的一部分加以使用？工作流、行为、处理和沟通是否可以通过功能模式来完成？
- **设计合适的安全和隐私机制**。用户访问系统时需要的安全级别是什么？为了保护服务器端和客户端的功能和内容不被未授权的客户访问，需要什么级别的安全和隐私保护？
- **对设计进行审查**。设计和客户需求是否一致？设计是否可以按照增量部署进度表完成？

在很多例子中，采用一些和用于构建（第 14 章）的一样的工具进行设计。这些工具允许 Web 工程师：制定架构和导航的结构，对内容进行布置和详述；制定表单级的输入；同时对 WebApp的美学设计进行原型化。使用这种方法，从设计到构建之间几乎变成无缝转换。

3.3.7 应用哪些构建任务

如果所有的框架活动都进行得顺利，那么你已经确定了用户对 WebApp 增量的需求。同时你也已经开发了一个设计，它将作为构建活动的基础。随着构建活动的进行，你将执行两个 WebE 动作：*代码生成和测试*。下面的任务和相关问题应当有助于你为代码生成动作做计划。

- **构建和获取所有的内容，并把内容集成到 WebApp 架构之中**。哪些 WebE 技术和工具将被用来构建内容和功能组件？在构建过程中，可以使用哪些已经存在的表单、模板和模式？
- **选择适合于产生 HTML 代码的工具集**。工具集能专有地使用吗？专门的功能是否必须要手工编码？
- **实现每个页面的布局、功能、表单和导航能力**。所有的内容都可用于集成到增量的每一个 Web 页面中吗？到所有功能的链接都已经实现了吗？已经使用了哪些链接机制？
- **实现所有的计算功能**。必须要实现哪些表单、脚本和数据库接口？计算算法是否已经进行了充分的设计？功能是部署在客户端还是服务器端？
- **处理配置问题**。在客户端和服务器端将支持什么样的浏览器、插件和操作系统环境？

WebApp 一旦构建完毕，就必须要对其进行测试。测试开始于相对来说比较窄的关注点，然后持续地运用 WebApp 的更宽视角。下面的任务和相关问题应当可以帮助你去为测试动作（第 15 章）做计划：

- **测试所有的 WebApp 组件（内容和功能）**。在用户任务的环境下，将测试哪些组件？设计的测试能完全用于测试功能吗？
- **测试导航**。在用户任务的环境下，将测试哪些链接？哪些用户场景可以应用到 WebApp 增量来开发合适的导航测试？设计的测试能完全用于导航结构吗？
- **测试可用性**。为了让使用简单化，必须测试哪些交互机制？哪些用户场景可以应用到 WebApp 增量来开发合适的可用性测试？已设计的测试可以确保每个可用性场景都是被支持的吗？
- **测试（要求的）安全性和性能**。我们如何应用所有的安全性过滤器并测试增量的总体性能？已设计的测试可以保证客户端和服务器端的功能都是安全的吗？
- **测试不同配置的 WebApp 增量**。一系列所有的技术性配置都已经开发了吗？设计的测试可以在所有的操作配置下应用 WebApp 增量吗？

看起来这些构建任务的全部，至少是绝大多数，都将应用于你所开发的每一个增量。然而，

如果情况允许，一个 WebE 团队可以对这些任务进行组合或者简化（或者增加额外的任务）。

3.3.8 WebApp 增量如何进行部署

你已经完成了 WebApp 增量，[⊖]现在你已经准备好把它交付给最终用户。你的意图是：1）为一个或者多个用户提供功能；2）从最终用户那里得到反馈，以决定是否已经满足了增量的需求；3）为作为部署的结果而肯定会发生的修改建立基础。下面的任务和相关问题应当可以帮助你去部署 WebApp 增量：

- **把 WebApp 增量交付给预先定义的域中的服务器**。是否遵循了所有的文件和目录命名约定以及链接引用惯例？已经给用户提供访问信息了吗？合适的安全性元素（如密码检查）是否到位并可用？
- **为最终用户建立一个在线反馈机制**。在第一个 WebApp 增量的交付中，已经实现了一个在线的反馈表单吗？反馈是自由格式的还是有一系列指定多项选择题的？可以定量地评估反馈表单吗？
- **评估最终用户的交互**。用户如何与系统交互？用户将遇到哪些方面的问题？交互的哪些部分是不清楚的、模棱两可的或者遗漏的？哪些内容或者功能是不正确的或者遗漏的？
- **评估了解到的教训并考虑所有最终用户的反馈**。基于用户的反馈，需要做哪些改变？改变应当立即进行还是作为将要开发的下一个增量的一部分？
- **对 WebApp 增量做必要的修改**。必须要对这个增量做哪些修改？在后续的增量中，必须有哪些变化（针对需求和设计）？

有很多的 WebE 原则、指南和方法（包括计划、分析、设计、构建和测试）支持这几部分中讨论的框架活动、动作和任务。本书的剩余章节将对它们进行更加深入的讨论。

3.4 保护活动

随着前面几节中所讨论的框架活动应用到每一个 WebApp 增量，在这样的环境下，就出现了一系列的*保护活动*（umbrella activity）。因为这些活动对项目的成功同等重要，所以 WebE 团队应当明确地考虑它们。

虽然可以定义很多的保护活动，但是只有 4 个对成功的 Web 工程项目而言是关键性的：

- **变更管理**（change management）。在开发每一个增量的时候，管理变更的影响，集成所有 WebApp 内容管理工具。
- **质量保证**（quality assurance）。定义并执行有助于确保每个工作产品和部署的增量都有很高质量的一些任务。
- **风险管理**（risk management）。在开发增量的时候，考虑项目和技术风险。
- **项目管理**（project management）。在开发增量的时候，跟踪并监控其过程。

它们都将在本书的后面予以详述。现在，我们来考虑关于保护活动的几个重要问题。

3.4.1 WebE 团队如何管理变更

在任何技术项目中，变更都会造成混乱。它扰乱正常的过程流（例如，团队准备去部署一个 WebApp 增量，但是现在，由于一个变更，必须延迟）；它影响其他工作（人们响应变更，导致正常的工作任务被暂时搁置）；同时它使团队的关注点变得模糊（目的地现在不明确了）；然

⊖ 如果这个增量不是最后一个被提交的，那么你就还没有完成 WebApp。过程流将以一种迭代的方式持续下去。

而，变更是不可避免的，而且常常都是建设性的。那么该做些什么呢？

我们提倡 Web 工程使用的增量策略可以帮助一个团队去管理变更。变更请求（来自利益相关者和 Web 工程师）随时都可能发生，但是一旦一个 Web 增量开工，那么就先把变更排成队列，直到这个增量部署完毕，再对变更做评估和处理。这就减弱了变更请求的扰乱本性，使得一个团队可以进行工作，而且仍然允许变更相对较快地被评估（回想一下，在一个增量上花费的工作通常都是持续几周，而不是几个月）。

当然，还有相当多的关于这个重要保护活动的内容。我们将在第 16 章中详细讨论变更管理。

3.4.2 如何保证增量的质量

每一个 WebE 团队都力求开发出高质量的 WebApp。但是，在这个上下文中，“质量”是什么？而且为了达到这一目标，可以使用哪些指南和特定任务呢？

WebApp 的质量是通过稳固的设计来实现的，稳固设计的目标就是“坚固、适用和赏心悦目”。[⊖]当一个 WebApp 是可靠的，并且内容和功能都没有被错误地表示时质量的第一个属性（坚固）就达到了。当 WebApp 满足所有利益相关者的需要时，质量的第二个属性（适用）就达到了。当 WebApp 超出了用户和技术期望的时候，质量的第三个属性（赏心悦目）就发生了。

作为 Web 工程的一个保护活动，质量保证关注在创建一个增量的过程中生产的每一个工作产品。其重点是确保当一个增量被部署的时候，结果会是稳固性、有用性和赏心悦目。那些可以帮助你去评估并达到这些属性的技术将在本书的剩余部分进行讨论。

3.4.3 如何管理风险

什么会出错？这是一个多么明显的问题，以致于我们有时会忘了去问。对这个问题的回答就是与风险管理有关的全部内容。风险是可以引起一个 WebE 项目出错的事件或者条件。你需要识别风险，然后对它们进行管理。

风险的识别贯穿于整个 WebE 过程。一旦有利益相关者（Web 工程师们是利益相关者）识别出有东西可能要出错，就应当把它记录下来用于讨论并（如果需要）采取行动。需要描述潜在的风险，并注意其潜在的影响。

影响大的风险应当被管理起来，有两种方法可以做到这一点：被动管理和主动管理。被动风险策略被嘲笑为“印地安纳·琼斯学派的风险管理”[Tho92]。印地安纳·琼斯在以其名字命名的电影里，每当遇到巨大困难的时候，他总是一成不变地说，“不用担心，我会想出办法来的！”印地安纳·琼斯从不担心任何问题，直到问题发生，再对其做出一些英雄式的反应。

遗憾的是，一般的 Web 工程师并不是印地安纳·琼斯。然而，大部分的 WebE 团队都仅仅只是依赖于被动风险策略——寄希望于当问题发生的时候，会有人像英雄一样予以解决。被动策略至多不过是监测项目中的那些可能发生的风险。直到风险发生的时候，才会拨出一些资源来处理它们。更常见的是，团队对风险不闻不问，直到出现问题才会赶紧处理。

更加明智的风险管理策略是主动风险管理。识别潜在的风险，评估它们发生的概率和产生的影响，并根据它们的重要性进行排序。然后，WebE 团队建立一个计划来管理每一个风险。我们将在第 5 章中讨论主动风险管理。

3.4.4 如何管理工作

在绝大多数情况下，一个 WebE 团队自己管理自己。每一个团队成员都在分配的时间内使自

⊖ 回顾我们前面提到的 Mitch kapor 对设计的讨论。

已受制于将要执行的任务并勤奋地来完成他们的工作。从本质上说，团队是自组织和自监督的。

然而，随着增量复杂性的增长，工作任务之间的相互依赖性常常需要一个团队领导或者经理来予以跟踪和控制。毕竟，明确项目是否落后于进度并快速地采取措施补救局势是很重要的。

项目管理是一个保护活动，是计划的产物。它包括变更管理、质量保证以及风险管理。同时，在创建增量的过程中，通过检查每一个框架活动的结果来对过程进行评估。用于项目管理的技术将在第5章中讨论。

3.5 本章小结

WebE过程框架是用于创建Web应用的一个敏捷的、适应的和迭代的指导方针。这一框架包括五个活动（沟通、计划、建模、构建和部署）以及可用于任何工业级质量Web应用的每一个WebE团队的一个保护活动集。框架活动被加以修改来适应问题、项目以及细化需求并做工作的人员的具体特征。

每一个框架活动都集成WebE动作和任务。一个动作包括多个任务，同时作为更大的框架活动的一部分被加以应用。一个任务代表了一个工作单元，它可以作为WebApp计划的一部分被跟踪。选择并适应那些对将要部署的下一个增量而言合适的动作和任务是WebE团队的分内工作。

一个完整的WebApp是以增量的方式予以交付的。也就是说，内容和功能被分割成可交付的组件，称为增量。每当一个增量被计划和部署以后，对最终用户来说，一个增量的更加完整的WebApp版本就可用了。

到本书的这个地方，我们已经从大体上讨论了WebE活动。该是变得更加具体的时候了。在第4章中，你将学到如何以一种有效的方式与利益相关者进行沟通，并为后续的计划、建模和构建活动建立基础。

参考文献

[Kap91] Kapor, M., "A Software Design Manifesto," *Dr. Dobbs' Journal*, January, no. 172, 1991, pp. 62-68.

[Tho92] Thomsett, R., "The Indiana Jones School of Risk Management," *American Programmer*, vol. 5, no. 7, September 1992, pp. 10-18.

第4章 沟 通

在不平常的20世纪90年代，因特网的繁荣产生的狂妄自大比计算机历史上的其他任何事件带来的都要多。数以百计的新兴网站公司的WebApp开发者认为，系统开发的一个新的模式已经出现，旧的规则已经不再适用，同时上市时间已经压倒了一切。他们嘲笑那种认为在构建开始之前应当仔细进行问题表述的观念。[1]谁可以争辩呢？钱主宰了一切；24岁就成为了百万富翁（至少纸面上是这样）——事情也许真的已经变了。然后根基倒了。

惨痛的事实却是，那种"构建它，然后它们就会到来"的哲学并不生效，为了保证一个WebApp确实是需要的，问题规划是必需的，而且如果WebApp想要满足最终用户的需求，那么需求的启发是必需的。Constantine和Lockwood［Con02］在他们写作的时候，提到了这种情况：

> 尽管有让人厌恶的言论宣称Web代表了新规则所定义的一种新模式，但是专业的开发者正在认识到，在因特网时代之前学到的关于软件开发的知识依然是适用的。Web页面是用户界面，HTML编程是编程，部署了基于浏览器的应用就是软件系统，这些可以从基本的软件工程原则中设益。

软件工程最基本的原则是：在你开始解决问题之前，先理解问题，同时确保你所构想的解决方案是人们真正想要的。建议这个原则应当成为Web工程的基石也是合理的。

沟通活动为WebE团队提供了一个有组织的方法，用于从利益相关者那里引出需求。本质上来说，在创建每一个WebApp增量的时候，它建立了团队的目的地。

4.1 沟通活动

WebE过程从沟通活动开始。参见图4-1，沟通作为过程流的入口。Web工程师和利益相关者在这里开展如下的一系列WebE动作：1）询问并回答关于WebApp和其业务环境的一系列基本问题；2）提取需求将作为所有后续活动的基础；3）协商对实际时间、资源和技术需求。这些动作分别称为规划、提取（也叫需求获取）和协商。

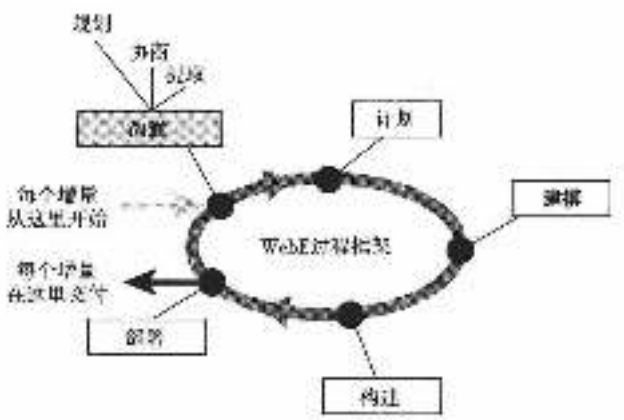


图4-1 WebE过程流中的沟通

[1] 几乎没有人想要做计划或者建模；毕竟，一直都有一个"沟企业"，不是吗？如果你停下来去做计划，那么你就将冒着过会了的风险！他们人约就是这样想的。

脑海中有了这些动作之后，我们为 SafeHomeAssured. com 过程框架的第一次迭代做准备。

第一次迭代中的沟通活动将从把 WebApp 作为一个整体识别其业务环境、目标和目的开始。[⊖] 这个规划动作对利益相关者（确保每个人都在同一个页面上）和 WebE 团队（确保将要创建的 WebApp 有一个业务需求）来说同等重要。在规划期间得到的信息和在提取期间收集的需求耦合在一起，将作为识别在后续迭代将要提交的增量的基础。

4.2 规划

规划是一个 WebE 动作，开始于识别业务需求，然后进入对 WebApp 目标的描述、定义主要的 WebApp 特征，以及为其后续的提取动作建立基础。规划允许利益相关者和 WebE 团队为每一个 WebApp 增量的创建建立一个通用的目标和目的集；同时识别开发工作的范围，并提供一种判定结果成功的方法。

在我们更深入地考虑规划之前，应当问一问规划在哪里停止，提取在哪里开始。对于这个问题，没有一个简单的答案。规划关注于全局视图——在于业务需求和目标以及相关信息。然而，利益相关者和 Web 工程师同时会列举特定的特征，定义需要的内容，讨论特定的功能，同时识别最终用户和 WebApp 交互的方式。当抽象层次从全局的业务问题（和 WebApp 相关的）转到更加具体的面向 WebApp 的需求时，从规划到提取的转变就发生了。

4.2.1 应当和谁沟通

在你为将要问的规划问题担忧之前，你必须识别问题的对象是谁。你需要确定利益相关者。一个利益相关者可以定义为“将从正在开发的系统中，以直接或间接的方式获得利益的任何人”[Som97]。

在最好的情况下，你的问题的焦点仅仅是一个人。这个人拥有合适的商务经验、良好的技术知识，并有从交付时间和资源方面协商需求的权力。然而，更可能的情况是，你将和很多不同的利益相关者打交道，他们中的每一个都有不同的视角、（有时）背离的需求，以及大量不同的业务和技术知识。总的来说，利益相关者包括：业务经理、产品经理、市场营销人员、内部和外部的客户、最终用户、顾问、产品工程师、Web 工程师以及支持和维护人员。

4.2.2 可以使用哪些技术进行沟通

除非 WebApp 相当简单同时最终用户资料是可预测的，否则使用只从一两个人那里收集的信息作为规划或者分析的基础是不可取的。应当考虑更多的人（和更多的观点以及视角）。

沟通可以用如下的一种或多种机制来实现 [Fuc98]：

- **传统焦点组**。一个经过训练的主持人和一小组（通常少于 10 个人）具有代表性的最终用户（或者是担任最终用户角色的内部利益相关者）会面。目的是讨论将要开发的 WebApp，通过讨论来更好地理解系统的需求。
- **电子焦点组**。同一组有代表性的最终用户和利益相关者进行适当的电子形式的讨论。参与的人员可以更多一些。因为所有的用户可以同时参与进来，所以可以在更短的时期内就能收集到更多的信息。由于所有的讨论都是基于文本的，所以可以自动产生同期讨论的记录。
- **迭代调查**。使用一系列简明的调查，通过一个 Web 站点或者电子邮件发送给具有代表性

⊖ 沟通活动中后面的迭代将关注于在第一次迭代中定义的单个增量。

的用户，要求他们回答一些关于 WebApp 的特定问题。对反馈加以分析，并用于调整后面的调查。

- **探索性调查**。一种基于 Web 的调查，是关于一个或多个 WebApp 的，这些 WebApp 拥有和将要开发的 WebApp 很相似的一些用户。用户链接到这个调查，并回答一系列的问题（参与的用户通常都会收到一些报酬）。
- **场景创建**。通过对被选择的最终用户（或者其他扮演最终用户角色的内部利益相关者）进行咨询，来创建用于描述与系统特定交互的使用场景。

4.2.3　不会有不同的视角吗

每一个利益相关者对于 WebApp 都有不同的视角，当 WebApp 被成功部署的时候获得不同的利益，同时在部署工作失败的时候都会遭受不同的风险。例如，业务经理们对那些将使公司销售增长和收入增加的特征集感兴趣；市场营销组将会对那些激发潜在市场，导致新用户产生并增加销量的那些特征感兴趣；产品经理希望 WebApp 不超过预算，同时将会满足已确定的市场需求；最终用户可能想要那些对他们而言已经很熟悉，而且易学易用的特征；Web 工程师可能关心对非技术的利益相关者不可见，但是能提供支持更多市场性特征的基础结构的功能。

当收集了多个视角的信息之后，出现的需求可能相互不一致或者可能彼此之间有冲突。在规划和提取阶段，你的工作就是以某种方式对所有的利益相关者信息（包括不一致和冲突的需求）进行分类，从而为最后的 WebE 动作（协商）做准备。

4.2.4　应该问什么问题

是时候去和 CPI 公司的利益相关者见面并开始沟通活动了。来自业务管理、市场营销、产品设计、顾客服务、销售以及产品支持的代表都已经到位。你的目的就是去简述 SafeHomeAssured. com 业务环境的蓝图。

礼仪之后，你把答案论述为 3 个基本问题［Pow98］：

1. WebApp 的主要动机（业务需求）是什么？
2. WebApp 必须完成的目标是什么？
3. 谁将使用这个 WebApp？

在一些讨论过后，大家同意了下面关于动机的简单陈述：

> SafeHomeAssured. com 将允许顾客来评估、配置和购买安全系统产品和监控服务。

值得注意的是，在这个陈述里面没有提供细节。这里的目标就是要为 WebApp 的总体目标划定边界，并把它置于一个合理的业务环境之中。

在与各种利益相关者进行了另外的讨论之后，得出了第二个问题的答案：

> SafeHomeAssured. com 将允许我们（把商品）直接卖给客户，因此就消除了中间商的成本，也增加了我们的利益空间。它同时可以让我们在现有年销售量的基础上增加 25%，并且可以让我们进入到我们当前没有销售经销店的地区。它可以提供允许客户直接访问监控信息的特性，来减小扩展我们的客户服务呼叫中心的需求。

最后，处理第三个问题（谁将使用这个 WebApp?），并为 WebApp 定义用户特征：

> SafeHomeAssured. com 所规划的最终用户是房主和小型商务的业务拥有者。然而，CPI 的销售和服务人员也会使用到这个应用。

假定主要人群对安全系统技术的知识有限，同时拥有非技术性的计算机技巧。

这些回答暗含了 SafeHomeAssured. com 这个 WebApp 的一系列特定目标。总的来说，定义了两类目标［Gna99］：信息型目标表示了为最终用户提供特定内容和信息的意图，应用型目标表示了在 WebApp 中完成某些任务的能力。

在 SafeHomeAssured. com 这个 WebApp 的背景下，开发了一系列信息型目标：

- 允许用户了解 CPI 公司及其人员和产品。
- 为用户提供详细的产品说明，包括技术性描述、安装指南和定价信息。
- 展示最终用户定义的设备（住宅或者业务）的安全配置（包括所有的硬件）。
- 使用户可以获得产品成本的报价。
- 创建一个账户数据库，为客户提供账户信息。

SafeHomeAssured. com 的应用型目标包括：

- 为用户提供使他们可以展示将要被保护的空间（例如，住宅、办公室/零售区）布局的工具。
- 制作可以在用户的空间中进行使用的安全和监控产品的定制建议。
- 将 SafeHomeAssured. com 基于表单的输入和 CPI 现存客户以及订单条目数据库进行集成。
- 允许用户下订单订购安全硬件。
- 允许用户控制其空间内的监控设备（例如，摄像机和麦克风）。
- 使用户可以签订监控服务的合同。
- 允许客户查询监控数据库中有关他们账户的活动。

一旦识别出所有的信息目标和应用目标，就标志着用户资料开发的完成。用户资料捕获“与潜在用户相关的特征，包括他们的背景、知识、爱好甚至更多信息”［Gna99］。

就 SafeHomeAssured. com 这个案例而言，一个用户资料可以识别安全系统（这些信息由营销部门提供）的一个典型买主的特点。

一旦 WebApp 的目标、目的以及用户资料开发完毕，规划活动将关注于 WebApp 的范围陈述。在很多情况下，信息和应用目标提供足够的信息来识别这个问题的范围。

4.2.5 如何促进协作

如果有 5 个利益相关者参与一个 WebApp 项目，那么对于适当的需求集合，你将会有 5 种（甚至更多）不同的观点。如果要产生一个成功的系统，那么利益相关者就应当在他们内部（避免为鸡毛蒜皮的事情争论）以及与 Web 工程师之间进行协作。但是如何实现这个协作呢？

你的工作就是在规划和需求提取的时候，充当顾问的角色——识别大家公认的方面（例如，所有利益相关者都同意的需求）和有冲突或者不一致的方面（例如，一个利益相关者期望的需求，但是与另一个利益相关者的需求相冲突）。当然，后面的一类代表了挑战。

有一种方法能够解决相互冲突的需求，同时更好地理解所有需求的相对重要性，那就是使用基于*优先级点*（priority point）的投票策略。给所有利益相关者提供一定数量的优先级点，这些优先级点可以用于一个或多个需求（在这个阶段，由信息目标和应用目标表示），数量可以不同。列出一系列的需求，然后每一个利益相关者通过在每个需求上面分配一个或多个优先级点来表示该需求的相对重要性（从他或她的视角）。优先级点不能重复使用。一旦某个利益相关者的优先级点用完，他就不能再对需求实施进一步的操作。所有的利益相关者在每项需求上的优先点总数就表明了这个需求的总体重要性。



确定 WebApp 目标（最高级的需求）的优先级

场景：CPI 公司的会议室

参与者：SafeHomeAssured.com 的所有利益相关者（一起来确定顶层项目需求的优先级）。一个主持人（你）主持着这个会议。

会话：

主持人：如你们所知，我们已经见到了在这里的每一个支持者为 SafeHome 定义的基本项目级需求。我刚发给你们的资料上面列出了这个 WebApp 的 5 个信息目标和 7 个应用或者功能目标。

团队组长：这些太含糊了，我认为我们需要给出一些细节，例如，这样的需求……

协调者（打断）：我们马上将对这些进行详尽说明，但是现在的焦点是建立一些最高级的优先级。

营销人员：就我看来，它们都很重要，所以……

主持人（又一次打断）：我理解，但是重要的是，我们应当建立优先级，这样我们就能开始考虑在线使用的内容和功能的交付顺序。

业务经理：你的意思是它将不会一次交付完毕？

团队组长：我们将按阶段交付这个 WebApp，我们称这些阶段为"增量"。它将使我们更快地开发出功能并运行。

业务经理（表示怀疑）：多快？

团队组长：我们将在一周内完成某些事情——尽管很简单。在大约 2 周的时间内会完成更多的内容。这是正确的方法。

营销人员（点头）：我们同意。

[业务经理以耸肩的方式表示同意。]

主持人：我们要明白我们定义的增量也许会和这些需求中的一些结合得很好［看着资料］，所以……

营销人员（打断）：但是我们还没有定义任何增量。

主持人：一旦我们建立了优先级，我们就将开始定义增量。这里是我们将如何来做。[解释优先级点。]

协作并非意味着个体人员来定义需求。在很多情况下，利益相关者通过各自对 WebApp 需求设定优先级来进行协作。可以由一个强势的"项目优胜者"（例如，一个业务经理或者一个资深的技术专家）来最终决定选择哪些需求。

4.3 提取需求

人们已经提出了很多不同的协同需求获取方法。每一种都利用一个稍微不同的场景，但是所有的方法都应用以下的基本原则作为基础原则：

- 会议由所有利益相关者共同举办和参与。[3]
- 制定准备和参与会议的规则。
- 建议拟定一个会议议程，这个议程既要足够正式，使其涵盖所有的重要点，但也不能太正式，以鼓励思想的自由交流。
- 由一个主持人（可以是一个客户、一个 Web 工程师或者一个局外人）对会议进行控制。
- 使用某种定义机制（可以是工作表、活动挂图、不干胶贴纸、电子公告牌、聊天室或者虚拟论坛）。

目的是识别一系列需求，这些需求处理 WebApp 的内容、功能、用户交互，以及和存在的业

[3] 这个"会议"可以是有形的，也可以是虚拟的。

务系统及数据库的互操作性。

为了更好地理解在会议召开时发生的事件流，我们展示了一个简略的场景，这个场景概述了准备需求收集会议的事件、在会议中发生的事件，以及会议之后的任务。

4.3.1 在提取需求的会话之前做什么

在规划期间，基本问题和问题的答案建立了 WebApp 的范围，提供了对 WebApp 特性的一些理解。如果范围显示开发工作相对较重，那么在需求收集（提取）之前进行准备是很明智的。如果时间允许，你可以考虑写一页的 WebApp 描述，作为需求收集的基础。WebApp 描述把从规划会议中已经得到的信息进行组合。就 SafeHomeAssured. com 而言，描述也许看似这样：

> SafeHomeAssured. com 将允许 CPI 公司直接把安全产品和监控服务卖给客户，因此也就消除了中间商的成本，同时增加了我们的利益空间。为了实现这个目标，此 WebApp 将合并实现了如下与产品相关特性的内容和功能：
>
> - 最终用户可以检查 SafeHome 的产品线和所需产品的规格说明。
> - 用户可以通过如下的方式对一个安全系统进行配置：展示一个“空间”（例如，房子、办公室/零售空间）的布局，然后初始化 SafeHomeAssured. com 中的功能来完成可以在这个空间中使用的安全和监控产品的定制建议。
> - 用户可以请求对产品和系统定价的及时报价。
> - 用户可以下订单订购传感器、控制器、音频和视频硬件以及相关的基础设施。
>
> WebApp 将合并实现了如下与监控相关特性的内容和功能：
>
> - 用户可以签署监控服务的合同，定购一个 SafeHome 系统的安装，以及协同其他将引导安全产品的购买的配置活动、它们的安装以及与 CPI 监控合约的执行。
> - 合约客户可以在他们的空间里控制安全和监控设备（例如，摄像机和手机），使用 SafeHomeAssured. com 这个 WebApp 来获得安全设备的输出，同时向他们自己的客户进行展示。
> - 客户可以查询关于他们账户活动的 CPI 监控数据库。
>
> SafeHomeAssured. com 的实现必须使得它的用法是凭直觉可获知的。它必须有很强的安全性，同时要 24/7/365 可用。
>
> SafeHomeAssured. com 也将为内部的 CPI 员工提供特殊的特性。这些特性将提供更强的客户支持和安全系统的技术性支持和维护。

有一种尝试写出一个“完美的”WebApp 描述的趋势，但是在现阶段实在是不可能的（即使曾经可能）。这样做将出现歧义，也很可能遗漏，同时也可能发生错误。你的目的是建立一个分离点。

选择一次会议的地点、时间和日期，以及一个会议主持人（很可能是你自己）。邀请所有的利益相关者出席。至少比会议时间提前 24 个小时把 WebApp 描述分发给所有的利益相关者。

4.3.2 利益相关者如何进行准备

每个利益相关者都被告知在需求收集会议之前检查 WebApp 描述，同时列出一个“内容对象”[⊖]列表，包括：1）系统周围的环境部分；2）由系统生成；3）被系统使用来完成系统的功

⊖ 一个内容对象是一个被命名的相关信息集合。有时是指一个“合成的数据条目”，一个内容对象包括那些在定义一个对象实例时，具有特定值的数据条目。

能。另外，每一个出席者都要给出一个功能列表，这些功能控制内容对象或者与这些内容对象进行交互。最后，也应当开发出约束（例如，影响所提供特性的业务规则）列表和性能标准（例如，速度、精确性、隐私性和安全性）。出席者被告知列表不需要很全面，但是应当反映每个人对系统的认识。

每一个利益相关者都开发前面描述的列表。用于描述 SafeHomeAssured. com WebApp 的内容对象可能包括：

- 公司概况
- 产品概述
- 产品的规格说明
- 安装指南
- 产品数据库（包括定价、存货清单、运输成本等）
- 报价模板
- 订单表单
- 订单录入数据库
- 用户空间的布局图
- 用户数据库（包括用户的确认、专门的安全系统配置，以及其他与用户相关的信息）
- 监控请求表单
- 一个合同用户的监控“仪表板”（包括账户信息、监控视频窗口、监控音频端口）
- 监控数据库
- 客户服务仪表板（在后面进行定义）
- 技术支持仪表板（在后面进行定义）

这些内容对象中的每一个都代表了一个初步的内容需求，同时也暗含了一个用于获取、操作或产生内容对象的功能集。

SafeHomeAssured. com WebApp 的功能列表可能包括：

- 提供产品报价
- 处理安全系统的订单
- 处理用户数据
- 创建用户资料
- 绘制用户空间布局
- 为布局推荐安全系统
- 处理监控订单
- 获取并显示账户信息
- 获取并显示监控信息
- 客户服务功能（将在后面进行定义）
- 技术支持功能（将在后面进行定义）

以类似的方式，每一个利益相关者开发约束和性能标准列表。

4.3.3　提取会话中执行哪些任务

当需求获取会议开始后，讨论的第一个主题就是需求和（推出）新产品的理由——每个人都应当赞同产品的合理性。一旦达成了共识，主持人指出大家必须完成如下 4 个任务：

1. 定义用户种类，同时给出每一类的描述。
2. 使用每个人准备的列表对内容和功能进行精化。
3. 考虑特殊的约束和性能问题。
4. 为每一类用户写场景。

并非所有的这些任务都要在一个简单需求获取会议期间内进行，但是它们都必须在 WebE 过程继续进行之前实现。

4.3.4 WebApp 的用户种类

可以证明，WebApp 的复杂性与该系统的用户种类数成正比。为了定义一个用户种类，必须要涉及如下一系列的基本问题：

- **用户在使用这个 WebApp 时，他的总体目标是什么**？比如，SafeHomeAssured. com 的一个用户可能对获取住宅管理产品的信息感兴趣。另一个用户可能想做一个价格比较。第三个用户可能想购买 SafeHome 产品。第四个用户已经拥有了 SafeHome，需要监控服务或者账户信息。每一个人都代表了一个不同的用户类，每一个人都将有不同的需要，不同地在这个 WebApp 内导航。
- **和这个 WebApp 内容和功能相关的用户的背景和熟练程度如何**？如果一个用户有技术背景和很好的熟练程度，那么基本的内容和功能提供的帮助就很少。另一方面，一个新手则需要基本的内容和功能，如果没有这些，他会很困惑。
- **用户如何到达这个 WebApp**？用户将从另一个站点或者搜索引擎的链接到达（这可能影响 WebApp 应用的内容和功能的设计），还是通过主页以一种更加受控的方式到达？
- **用户喜欢和不喜欢哪些一般的 WebApp 特征**？不同种类的用户也许有不同的、可预测的喜好和厌恶。有必要确定他们喜欢还是不喜欢。在很多情况下，这个问题的答案可以通过询问他们最喜欢的和最不喜欢的 WebApp 来确定。

使用这些问题的答案，应当可以定义最小的合理的用户类集合。随着需求获取的进行，在开发使用场景的时候，每一个已定义的用户种类都应当被考虑。

4.3.5 如何识别内容和功能需求

一个需求提取会议的核心活动就是识别内容和功能需求。每个利益相关者已经通过准备内容对象和 WebApp 功能开始了这项工作。一旦会议开始，这些列表可以展示在一个大的钉在房间墙上的纸质表里，显示在背面为胶的贴纸上贴在墙上，或者写在一个白板上。另外，这些列表也可以张贴在一个电子的公告板上，发布在一个内部网站上，或者发布在一个聊天室环境中，用于会议前的检查。理想情况下，每个列出的条目都应当可以独立地进行操作，因此，列表就可以被合并起来，可以删除条目，同时也可以增加。这个阶段严禁评论和争论。

来自每个列表的条目被整合为一个合并的列表。在这个列表中，消除冗余的条目，增加在讨论过程中出现的任何新的想法，但是不删除任何东西。当所有主题领域（对象、功能、约束和性能）的合并列表都创建之后，讨论（由主持人协调）随之产生。合并列表被加以缩减、加长或者编辑以恰当地反映将要开发的 WebApp。目标就是在每个主题领域开发一个一致的列表。然后这些列表将被放在一边，用于后面的动作。



一个需求获取会议

场景：会议室。在建立了基本的顶层需求之后，第一个需求收集会议就开始了。

参考者：主持人（你）和来自业务管理、营销、Web 设计、安全产品支持的利益相关者

会话：

主持人（指着白板说）：这就是 SafeHomeAssured.com 当前的对象和功能列表。

营销人员：那是从我们的视角来看待它的。

产品支持人员（向着列表点头）："获取并显示监控信息"的意思是你想允许客户通过网络访问远程的音频和视频？

营销人员：是的……那将使我们在竞争中占据领先地位。

主持人：那会不会同时增加一些约束呢？

业务经理：当然会。在技术上和法律上都会。

营销人员：什么意思？

团队组长：我们最好保证一个局外人不能侵入这个系统。解除监控，然后修改客户的空间或者做出更精彩的事情。

业务经理：我们担负的责任重大。

团队组长：非常正确。

营销人员：但是我仍然需要……只是确保阻止黑客侵入。

团队组长：说起来容易，做起来难，而且……

主持人（打断）：我现在不想争论这个问题。让我们注意这个约束并继续进行。［一个出席者作为会议的记录员，做了合适的记录。］

主持人：我感觉我们要考虑的东西还有很多。

［小组在接下来的 30 分钟时间里讨论列出的其他条目。］

4.3.6 如何将约束和性能问题分离

你不可能不考虑内部和外部约束就创建一个成功的 WebApp。理解内部约束（internal constraint）的最好方法就是考虑 WebApp 将驻留的技术环境，以及创建 WebApp 的项目环境。对技术环境的考虑也许会发现特定的数据库协议、不同 Web 浏览器的异常、操作系统的特性和客户端－服务器问题。所有的这些都会在 WebApp 的设计方法上有一些影响。项目环境包括可用的 WebE 工具、开发硬件、软件标准和员工对各种 WebE 技术的熟练程度。

外部约束（external constraint）可以通过考虑 WebApp 的业务和用法环境来列举。业务规则、最终用户的嗜好、安全性需求、隐私问题、运行时性能、互操作性的需求、法律上的限制以及政府规章都是一些可能附加在将要执行的 Web 工程工作之上的外部约束。

作为一个约束可能如何被识别的例子，考虑前面在 SafeHomeAssured.com 的需求获取会议上的对话。在对话过程中，发生了下面的沟通：

> 团队成员：我们最好保证一个局外人不能侵入这个系统，解除监控，然后修改客户的空间或者做出更精彩的事情。
>
> 业务经理：我们担负的责任重大。

这些主要的讨论应当为团队增加一个"约束标志"。很明显，如果监控功能或者与它相关的内容对象被破坏，那么将会发生非常严重的后果。因此，安全性就成为一个约束，这并非因为对每一个 WebApp 来说，安全性是不重要的，而是因为必须要为所有的 WebApp 实现特殊的注意，这些度量用于必须实现的那些特性。其含义是需要在安全性上进行重点设计，同时进行大量的

安全性测试。这对 WebApp 增量的定义、资料和进度安排的影响将在计划阶段确定。

4.3.7 什么是使用场景

一旦识别内容对象、功能和约束的一张列表完成，利益相关者就被要求扮演用户的角色（一个特定的用户种类），为每一个列表上的一个或多个条目开发使用场景[⊖]。一个典型的使用场景就是一到两个叙述性的段落，描述了一个最终用户如何应用或创建一个内容对象或与一个 WebApp 功能进行交互。然而，应当注意的是，更正式的模板已经提出了。它们被称为用例（use case）[Bit03] [Coc01]。



开发一个初步的用户场景

场景：会议室，接着进行第一个需求获取会议

参与者：WebE 团队的成员、营销组织的成员（扮演最终用户的角色）、一个产品设计代表以及一个主持人

会话：

主持人：我们一直在讨论访问 SafeHomeAssured.com 的安全性。我想试着去做一些事情。让我们为住宅安全性功能的初始访问开发一个用户场景。

营销人员：怎么做？

主持人：我们有很多种不同方式可以来完成它，但是现在，我想保持事情的非正式性。告诉我们，作为营销人员，你对访问系统是如何预想的？

营销人员：嗯……好吧，这类事情都是在我远离住所，同时我不得不让某人进入它的时候才去做的，例如，一个管家或者维修人员，他们没有安全密码。

主持人（笑）：这就是你要做它的原因……告诉我你实际是如何做到这一点的。

营销人员：我需要的第一件东西是一个 PC 机。我会登录到网站上，提供我的用户 ID 和……

团队成员（打断）：Web 页面必须是可靠的、加密的，以保证我们是安全的而且……

主持人（打断）：那是很好的信息，但是它是技术性的。让我们只关注于最终用户将如何使用这个功能，好吧？

团队成员：没问题。

营销人员：就像我刚刚说的，我将登录到一个 Web 站点，提供我的用户 ID 和至少两级密码。

团队成员：如果我忘记了我的密码呢？

主持人（打断）：很好的想法，但是我们现在不处理这个。我们将把它记录下来，并把它称为一个异常。我相信还会有其他的问题。

营销人员：当我输入密码之后，一个显示所有 SafeHome 功能的屏幕将会出现。我选择住宅安全功能。系统可能要求我验证自己的身份，索要我的地址或者电话号码或者其他信息。然后它将展示一个安全系统的控制面板图和一个我可以执行的功能列表——装备系统、解除系统、解除一个或多个传感器。我猜想它同时将允许重新配置安全性区域和其他类似的东西，但是我不确定。

[在营销人员继续谈论的过程中，主持人做了详细的笔记。这些笔记构成了第一个信息型使用场景的基础。作为选择，也可以要求营销人员写下场景，但是这将会在会议之后进行。]

⊖ 这可以在会议的背景下完成，或者会议之后在离线创建用户场景的时候保证一段时间，另一种方式是让一个实际的用户去完成它。

4.3.8 什么是用例

用例是一种广泛使用的用于创建用户场景的方法。用例描述了一个特定的用户种类（称为参与者（actor））为了完成一个特定的动作，将如何与 WebApp 进行交互。这个动作可以像获取定义好的内容一样简单，也可以像引导远程监控设备详细的指导用户控制设备那样复杂。用例从用户的角度来描述交互。

总的来说，用例可以迭代地进行开发。在增量的沟通活动中，仅仅开发那些对将要构建的增量来说必要的用例。在很多情况下，仅仅考虑 WebApp 主要功能（在需求获取阶段进行考虑）的用例。这些用例可能会在增量的分析建模活动（第 7 章）中进行精化。

虽然开发和分析用例要耗费时间，但是它可以提供很大的帮助：

- 用例为有效的计划和建模活动提供详细的必需品。
- 用例帮助开发者理解用户如何看待他们和 WebApp 的交互。
- 用例有助于划分 Web 工程工作，因为它们可以被组织到 WebApp 增量之中。
- 用例为那些必须对 WebApp 进行测试的人员提供了重要指导。

用例已经被赋予“行为的约定”［Coc01］特性。这个约定定义了一个参与者[⊖]使用一个 WebApp 来实现某个目标的方式。从本质上来说，一个用例捕获了发生在信息的生产者和消费者之间以及系统本身的交互。

理解用例的概念相对来说是比较容易的。它从一个定义好的参与者的角度，以一种简单易懂的语言，描述了一个特定的使用场景。但是我们如何知道：1）写什么东西；2）写多少东西；3）我们对细节的描述达到什么程度；4）如何组织这些描述？

4.3.9 如何创建用例

需求获取为我们提供了开始书写用例所需要的信息。要开始开发一个用例集合，需要列出一个特定的参与者执行的功能或活动。这些可以从利益相关者开发的列表或者与客户或最终用户的谈话中得到。



开发另一个基本的用户场景

场景：会议室，在第二个需求获取会议中

参与者：WebE 团队成员、营销成员、一个产品设计代表和一个主持人

会话：

主持人：到了继续讨论 SafeHomeAssured.com 的监视功能的时候了。让我们为它的访问开发一个用例。

团队成员：这次谁将扮演参与者的角色？

主持人：我认为 Meredith（一个营销人员）一直在做那个功能。你为什么不扮演这个角色呢？

Meredith（营销人员）：你想让我们像上次那样来做，是吗？

主持人：是的……同样的方式。

Meredith：好吧，监控的目的很明显是为了允许房主在他离开时，对住宅进行检查、记录和回放获取的视频等。

团队成员：视频应当是数字化的。它储存在磁盘上？

⊖ 一个参与者不是一个特定的人，而是在一个特定的环境下，一个人（或者设备）所扮演的角色。一个参与者“请求系统提交它的一个服务”［Coc01］。

主持人：很好的问题，但是现在让我们推迟实现的问题。Meredith?

Meredith：好的，所以在监控功能上基本上有两个部分：第一部分配置系统，包括设计一个建筑平面图——我们必须要有工具来帮助房主做这件事情；第二部分是真正的监控功能本身。因为布局是配置功能的一部分，我将只关注监控功能。

主持人（笑）：你讲出了我想要讲的话。

Meredith：嗯……我想通过因特网获得对监控功能的访问。我想在我的 PC 机上显示摄像机图像，并控制一个特定摄像机的缩放。我可以从住宅的建筑平面图上选择来指定一个摄像机。我想有选择地记录摄像机的输出，并对摄像机的输出进行回放。我还想通过一个特定的密码来阻止对一个或多个摄像机的访问。我还想有一个选项，可以看到一些小窗口显示来自所有摄像机的图像，然后我可以选择我想放大的那个。

团队成员：这些称为缩略图。

Meredith：对，然后我想要所有摄像机的缩略图显示。我还想要监控功能的界面，以获得和其他所有的 SafeHomeAssured. com 界面相同的外观。我希望它是直观的，就是说我不想因为要使用它而不得不阅读一些手册。

主持人：好。现在，我们进一步深入到这个功能的细节……

上面的讨论建议了一个功能，我们称之为由房主（最终用户种类）执行的通过因特网访问摄像机监控。另外，下面的子功能（一个缩略列表）都是隐含的，但是仍然都没有被定义：

- 选择摄像机观看。
- 请求来自所有摄像机的缩略图。
- 在一个 PC 机窗口中显示摄像机图像。
- 控制一个特定摄像机的缩放。
- 有选择地记录摄像机的输出。
- 重放摄像机的输出。

随着与利益相关者（扮演房主角色的人员）进行更进一步的会谈，需求获取团队将为每一个记录下来的子功能开发用例。作为选择，子功能用例的产生也可以推迟到 WebE 建模活动（第 7 章）。

一般来说，用例首先以一个用户场景的非正式叙述性方式进行书写。如果需要更正式的形式，同样的用例将使用本章后面讨论的结构化格式进行重写。

为了进行说明，考虑 SafeHomeAssured. com 的一个称为通过因特网访问摄像机监控的功能。扮演房主的利益相关者将进行下面的叙述：

用例：通过因特网访问摄像机监控

参与者：房主

叙述：如果我在很远的地方，我可以使用任何具有合适的浏览器软件的 PC 机登录到 SafeHomeAssured. com 网站。我输入用户 ID 和两级密码，并且一旦验证通过，就将获得对我所安装的 SafeHome 系统所有功能的访问。要访问一个特定的摄像机图像，我从显示的菜单项中选择“监控”。然后选择“选择一个摄像机”，就出现了住宅的建筑平面图。然后我选择感兴趣的摄像机，或者，我可以选择“所有的摄像机”作为我的观看选择，同时就可以看到所有摄像机的缩略图快照。一旦我选择了一个摄像机，然后选择“图像”，一个摄像机图像就会出现在由摄像机 ID 所识别的一个观看窗口内。如果

我想要转换摄像机，我选择“选择一个摄像机”，前面的观看窗口就消失了，同时再一次出现住宅的建筑平面图。然后我选择我感兴趣的摄像机。一个新的观看窗口就出现了。

另一种书写用例的方法就是以用户动作有序序列的方式对交互进行表示。每一个动作都表现为一个说明性的句子。

用例：通过因特网访问摄像机监控
参与者：房主
动作：
1. 房主登录到 SafeHomeAssured. com 这个站点。
2. 房主输入他的用户 ID。
3. 房主输入两级密码。
4. 系统显示所有的主功能按钮。
5. 房主从主功能按钮中选择“监控”。
6. 房主选择“选择一个摄像机”。
7. 系统显示住宅的建筑平面图。
8. 房主从建筑平面图中选择一个摄像机图像。
9. 房主选择“图像”按钮。
10. 系统显示由摄像机 ID 识别的一个观看窗口。
11. 系统在这个观看窗口内以每秒一帧的速度播放视频输出。

很重要的是，注意这个序列化的展示并没有考虑任何可选择的交互（叙述是一种更自由的流动，同时确实描述了一些选择动作）。

当然，对可选择交互的描述对于彻底地理解正在被描述的功能是必不可少的。因此，基本场景的每一步都通过询问下面的问题进行评估［Sch98］：

- 在这一点，参与者能否采取其他行动吗？
- 参与者在这一点是否有可能遇到一些其他的错误状况吗？如果有，它将可能是什么？
- 参与者在这一点是否有可能遇到一些其他行为呢（例如，参与者控制之外的事件所调用的行为）？如果有，它将可能是什么？

这些问题的答案将创建一系列的次要场景，它们是原始用例的一部分，但是描述了可选择的行为。

例如，考虑前面展示的主场景中的第 6 步和第 7 步：

6. 房主选择“选择一个摄像机”。
7. 系统显示住宅的建筑平面图。

在这一点，参与者可以采取其他行动吗？答案是肯定的。参考自由流动的叙述，参与者可能选择同时观看所有摄像机的缩略图快照。因此，一个次要的场景也许就是“观看所有摄像机的缩略图快照”。

在这一点，参与者有可能遇到一些其他的错误状况吗？在操作一个基于计算机的系统的过程中，任何数目的错误状况都可能发生。在这种环境下，我们仅仅考虑那些可能作为第 6 步和第 7 步中描述的动作的直接结果的错误状况。这个问题的答案也是肯定的。也许从来都不会配置一个有摄像机符号的建筑平面图。因此，选择“选择一个摄像机”将产生一个错误状况：“没有为

这个住宅配置建筑平面图"。[⊖]这个错误状况就变成了一个次要场景。

在这一点，参与者是否会遇到一些其他行为呢？这个问题的答案同样是肯定的。跳过第6步和第7步的进行，系统可能会遇到一个警告状况。这会导致系统发出一个特殊的警告通知（类型、位置和系统动作），并为参与者提供与这个警告本身相关的一些选项。因为这个次要场景会发生在实际所有的交互中，它将不会成为通过因特网访问摄像机监控这个用例的一部分。在一定程度上，另外一个用例（遇到警告状况）将被开发，同时需要对其他的用例进行参考。



监控功能的用例模板

用例：通过因特网访问摄像机监控

主要参与者：房主

环境下的目标：在任何很远的地方，通过因特网来观看住宅中所布置的摄像机的输出

前提条件：系统必须被完整地进行配置；必须获取了合适的用户ID和密码

触发器：不在家的时候，房主想看一看住宅内的情况

场景：

1. 房主登录到SafeHomeAssured.com这个站点。
2. 房主输入他的用户ID。
3. 房主输入两级密码（每级密码的长度至少都是8个字符）。
4. 系统显示所有的主功能按钮。
5. 房主从主功能按钮中选择"监控"。
6. 房主选择"选择一个摄像机"。
7. 系统显示住宅的建筑平面图。
8. 房主从建筑平面图上选择一个摄像机图标。
9. 房主选择"观看"按钮。
10. 系统显示由摄像机ID所识别的一个观看窗口。
11. 系统以每秒一帧的速度在观看窗口中显示视频输出。

扩展：

1. ID或密码错误，或不被识别。参见用例验证ID和密码。
2. 并没有为这个系统配置监控功能——系统显示合适的错误信息。参见用例"配置监控功能"。
3. 房主选择"观看所有摄像机的缩略图快照"。参见用例"观看所有摄像机的缩略图快照"。
4. 建筑平面图不可用或者没有进行配置——显示合适的错误信息。参见用例"配置建筑平面图"。
5. 遇到一个警告状况。参见用例"遇到的警告状况"。

优先级：中等优先级。将在基本的WebApp功能之后进行实现

何时可用：第六个增量

使用频率：中等

参与者的使用方式：通过基于PC或者移动设备浏览器和因特网连接

次要参与者：系统管理员，摄像机

次要参与者的使用方式：

1. 系统管理员：基于PC的系统
2. 摄像机：无线连接

开放问题：

1. 什么机制来保护CPI公司的员工对这个功能进行未授权的使用？
2. 安全性是否足够？对这个特性的破坏将表示对隐私的主要侵入。
3. 在摄像机图像所需的给定带宽下，通过因特网的系统响应是否将可以接受？
4. 当更高的带宽连接可用的时候，我们是否将开发那些提供更高的每秒帧数速率的功能。

⊖ 在这种情况下，另一个参与者，系统管理员，将不得不配置建筑平面图，实施和初始化（例如，分配一个设备ID）所有的摄像机并通过测试来保证每一个都可以通过系统和建筑平面图访问。

参见上面展示的正式用例模板，次要场景是作为描述通过因特网访问摄像机监控的基本序列的扩展。

我们有理由来问一问，对于每一个已经识别的功能，所有的这些是否真的都是必要的。一般来说，不是的。你应当为每一个将交付给一个 WebApp 增量的主功能开发一个叙述的使用场景。但是，如果一个功能特别复杂或者特别重要，那么也许就需要有一个更详细的描述（使用用例模板）。

4.4　确定 WebApp 增量

回顾 WebE 过程框架的迭代属性，在每次开发一个新的 WebApp 增量的时候，都将应用在本章前面所描述的规划和提取动作。但是如何去定义这些增量呢？

我们在第一次迭代时通过沟通活动得到了整体的 WebApp 需求。包括主要内容对象、WebApp 功能和使用场景（或者用例）的导出。在为第一个增量做计划可以开始之前，有必要开发一个将要部署的所有潜在增量的有序列表。⊖

一种方法［Fuc98］是创建一个“卡片”栈，栈中的每张卡片上都有一个使用场景或者用例。每张卡片包含有用例的名称、一个简短的描述，以及一个*工作量指标*（effort indicator）——通常是 1～4 的一个数字。工作量指标由 WebE 团队分配给每一个用例，并提供创建要实现用例所需的 WebApp 功能所需要的相对工作量程度的估计。工作量指标 1 也许就可以分配给一个相对简单的场景。工作量指标 4 可能就分配给一个需要大工作量的场景。

卡片被打乱成随机的顺序，然后分配给选定的利益相关者，要求他们把卡片分组，以反映他们所希望的内容和功能（使用场景隐含）的交付方式。卡片的分组方式是被一个工作量最大量 M 所限制的。没有任何卡片组的累计工作量指标值会比 M 大，在这里，M 是由 WebE 团队定义的，是可用资源和每一个增量的期望交付时间的函数。

作为一个例子，我们假设 SafeHomeAssured. com WebApp 的第一个沟通活动已经完成，同时利益相关者已经定义了一系列的使用场景，它们以一个卡片栈的方式进行表示：

卡片号	卡片名称	工作量指标
1	了解公司及其产品	1
2	下载产品规格说明	2
3	获取为我的用户种类定制的信息	2
4	查询一个特定的传感器	1
5	获取一个产品的报价	2
6	从将要被监控的空间开发一个布局	4
7	获取对我的空间所推荐的传感器布局	3
8	下订单订购产品	2
9	请求监控服务的信息	1
10	下订单订购监控服务	2
11	控制传感器	3
12	控制摄像机	3
13	通过因特网访问摄像机监控	4
14	获取账户信息	2
……		

⊖ 很重要的是，要注意在 WebE 工作继续进行之前，增量的有序列表很可能被修改很多次。

WebE 团队已经建立了一个工作量最大值，$M=4$，意思是，只有当一个增量所分组卡片的联合工作量指标值小于或者等于 4 的时候，它才能被定义。另外，卡片还必须以一种耦合的处理功能的方式进行分组。

对于在栈中进行定义的卡片而言，SafeHomeAssured.com 的基本增量集可能是：

增量 1：在卡片 1 中反映的用例

增量 2：在卡片 2、3 和 4 中反映的用例

增量 3：在卡片 5 和 8 中反映的用例

增量 4：在卡片 6 和 7[1] 中反映的用例

增量 5：在卡片 9 和 10 中反映的用例

增量 6：在卡片 11、12 和 13 中反映的用例

增量 7：在卡片 14 中反映的用例

利益相关者何时能预期这些增量的部署呢？在进行做计划（在第 5 章中[2]讨论的下一个框架活动）之前，我们确实不知道。

还有很重要的一点是，我们要认识到定义的第一个增量集可能是不完整的。随着工作的进行，一个重要的特性可能会被加入到 WebApp 之中，指示创建一个或多个新的用例。结果可能增加新的增量，也可能修改一个存在的增量，或者去掉一个已计划的增量。计划必须反映将要发生的这些变化。但是现在，我们与目前所拥有的相伴。

4.5 协商

信息



协商的艺术

学习如何有效地进行协商可以让你在人生和技术生涯中受益匪浅。下面的指导原则非常值得考虑：

1. 认识到这不是竞争。为了成功，双方都必须感受到自己正在获胜或者获得什么，双方将不得不妥协。
2. 制定策略。决定你希望获得什么，对方希望获得什么，以及你们如何让这两方面的希望都能实现。
3. 主动聆听。在对方说话的时候，不要陈述你的回应。聆听，你可能获得一些有助于你在协商中更好地说明你的立场的信息。
4. 关注对方的兴趣。如果你想避免冲突，那么请不要采取强硬的立场。
5. 不要进行人身攻击。集中于需要解决的问题。
6. 要有创新性。当你陷入僵局的时候，不要害怕而应该考虑如何摆脱困境。
7. 随时准备作出承诺。一旦达成了某项共识，不要闲聊胡扯；马上作出承诺然后继续进行。

理想情况下，沟通确定了足够详细的利益相关者的需求，可以进行到一个增量的后续框架活动。但现实情况是，Web 工程师和其他利益相关者往往进入一个协商的过程，此时，利益相关者也许需要考虑成本和交付时间，平衡功能、性能和其他的产品或者系统特性。这个协商的目的是建立增量需求，来满足客户的需求，同时反映 WebE 团队所处的现实世界的限制（例如，时间、人员和预算）。

---

[1] 在这里 $M=5$，但是增量依然是有意义的。工作量最大值是一个有用的指导原则，但是有时它又必须被违反。

[2] 而且即使到那时，在最好的情况下，我们将只是在第一次很少的增量上有比较好的进度可见度。

最好的协商是力求一个双赢的结果。[注]即客户“赢”在获得满足大多数需要的 WebApp，WebE 团队“赢”在按照实际情况在可实现的时间线内完成工作。

在任何协商开始之前，一个很好的主意就是确定每一个利益相关者的“赢的条件”，并制定一个策略，以把所有的利益相关者融合于对所有参与者（包括 WebE 团队）来说都是“双赢”的条件之中。



沟通中的需求协商

场景：会议室，在第二个需求获取会议中

参与者：WebE 团队成员、营销人员、一个产品工程代表以及一个主持人

会话：

主持人（对着营销人员说）：我想花一点儿时间来讨论卡片 6 和 7。你们知道，它们是“空间布局”和“监控布局的推荐”。基于我们基本的用例，看起来你们想要尽可能多的自动化。

营销人员：理想情况下，它应当是全自动的。只需要描述出住宅的布局，剩下的工作由 SafeHomeAssured. com 完成。

团队成员（做了个鬼脸）：我们将开始讨论它。

营销人员（看起来稍微有些恼怒）：这真的是一个很棒的特性。我们必须拥有它。

主持人（认识到一个协商已经开始了）：让我们后退一步，看一看这个问题。我同意你[看着营销人员]那是一个很重要的特性的说法，但是我们必须考虑一个全自动方案的复杂性[看着 WebE 团队成员]、其对交付进度安排的影响，以及全自动在收益上的总体重要性[看着产品工程代表]。

[每个人都在他们的座位上动来动去，但是没有人说话。]

主持人：我希望我们可以达成一致，就是你们的需求可以被满足[看着营销人员]，但同时我们保证交付时间线的合理性[看着 WebE 团队成员]。我们如何做到这一点呢？

团队成员：听着，我们知道那是一个很酷的特性，而且我们想去实现它。只是一个全自动的方案确实很复杂，更严重的是，它可能产生并非客户真正所需的推荐。我认为一个半自动化的方案，用户交互地参与进来，是一个更好的实施办法。这就是我认为我们应当做的……

主持人：等一等。你认为这一方案怎么样？[看着营销人员。]

营销人员：好吧，如果我们能获得我们需要的功能，那么它可能就没有问题，但是我需要知道更多细节。

主持人（看着一个 WebE 团队成员）：你的提议是什么？

[协商继续进行，直到双方达成一致，创建了新的用例以反映新版的特性。]

4.6　本章小结

沟通活动的目的就是为将要创建的 WebApp 增量建立一个需求集合。像所有框架活动一样，沟通也是迭代发生的。在第一次迭代中，勾画了 WebApp 大概的总体需求，然后从这些需求定义了一组可部署的增量。沟通包括三个 WebE 动作：规划、提取和协商。

规划通过检查业务动机、运作目标和用户特性识别将要创建增量的边界。目的是为将要完成的 WebE 工作提供一个边界描述。为了获取边界，利益相关者相互协作来建立优先级。

注　有很多关于协商技巧（例如，[Lew00]、[Fis97] 和 [Dun96]）方面的书籍。它是一个年轻（或者年长）的 Web 工程师或者经理可以学会的很重要的事情之一。读一本这样的书。

提取通过识别 WebApp 增量将要使用的内容对象和增量将要实现的功能来获取需求。使用会议的形式，需求收集任务包括定义用户种类，精化内容和功能，识别约束和性能问题，以及开发使用场景。在提取中识别的使用场景将用于识别 WebApp 增量，这些增量将在 WebE 过程框架的后续迭代中进行开发。

协商试图解决利益相关者之间的冲突，目的是建立一套增量需求，以满足所有利益相关者的需求，同时反映现实世界的限制（例如，时间、人员和预算限制）。最好的协商力求一个“双赢”的结果。

沟通为计划和建模活动设置场景——只有在识别了基本的 WebApp 需求之后才会发生的框架活动。在第 5 章中，我们将探究如何为一个 WebApp 增量的部署建立计划。

参考文献

[Bit03] Bitner, K., and I. Spence, *Use Case Modeling*, Addison-Wesley, 2003.

[Coc01] Cockburn, A., *Writing Effective Use Cases*, Addison-Wesley, 2001.

[Con02] Constantine, L., and L. Lockwood, “User-Centered Engineering for Web Applications,” *IEEE Software*, vol. 19, no. 2, March/April 2002, pp. 42-50.

[Don96] Donaldson, M. C., and M. Donaldson, *Negotiating for Dummies*, IDG Books Worldwide, 1996.

[Far97] Farber, D. C., *Common Sense Negotiation: The Art of Winning Gracefully*, Bay Press, 1997.

[Fuc98] Fuccella, J., and J. Pizzolato, “Creating Web Site Designs Based on User Expectations and Feedback,” *ITG Newsletter*, June 1998, www.internettg.org/ newsletter/june98/web_design.html (accessed July 25, 2007).

[Gna99] Gnado, C., and F. Larcher, “A User Centered Methodology for Complex and Customizable Web Applications Engineering,” *Proc. First ICSE Workshop in Web Engineering*, ACM, Los Angeles, May 1999.

[Lew00] Lewicki, R., Saunders, D., and J. Minton, *Essentials of Negotiation*, McGraw-Hill, 2000.

[Pow98] Powell, T. A., *Web Site Engineering*, Prentice-Hall, 1998.

[Sch98] Schneider, G., and J. Winters, *Applying Use Cases*, Addison-Wesley, 1998.

[Som97] Somerville, I., and P. Sawyer, *Requirements Engineering*, Wiley, 1997.

第5章 计 划

虽然我们当中的很多人（在我们更困难的时刻）赞同 Dilbert 对项目计划的观点，但是在创建一个 WebApp 时，它仍然是一个非常必要的活动。虽然每个人都在某种程度上做计划，但是计划活动的范围对于一个 WebE 项目所涉及的人员来说，却是各种各样的。一个 Web 工程师管理日常工作——计划、监控以及控制技术性任务。团队负责人计划、监控并协调一个 WebE 团队的联合工作。其他利益相关者通常不关心计划活动的细节，但是非常关心它的结果。我们做计划使得他们的期望得以实现。

因为绝大多数的 WebApp 增量都是在不超过 6 周的时间跨度内进行交付，所以我们有理由去问："我们真的需要花时间来计划 WebApp 的工作量吗？为什么我们不能让 WebApp 自然地演化，而没有计划或者没有明确的计划，并且只有间接的管理呢？"有很多的 Web 开发者会选择这种方式，但是这种方式并不会让他们成功。

把敏捷方法引入计划活动，你把花费在计划上的工作量和时间转移到将要部署的 WebApp 增量的复杂性上。如果 WebApp 增量很简单（例如，内容定义良好而且功能简单易懂），那么计划将花费很少的时间。另一方面，如果增量在内容、功能、限制和性能方面是复杂的，计划将需要更多的工作量，并且将包含本章所讨论的每一个动作和任务。不管增量的特征如何，你都必须进行计划。

5.1 理解范围

在沟通活动中花费的很多工作量都是关注于 WebApp 范围的定义。范围（scope）可以通过回答下面的问题进行定义：

背景。WebApp 如何适应到业务背景之中，同时背景将带来什么约束？

信息目标。WebApp 增量将使用和产生哪些用户可见的内容对象？

功能性。为了满足在使用场景中定义的需求，最终用户将启动哪些功能，或者 WebApp 将内部调用哪些功能？

约束和性能。哪些技术和环境的约束将影响到后面的框架活动？哪些特殊的性能问题（包括安全性和隐私性问题）将需要设计和构建工作？

在没有理解 WebApp 增量范围的情况下，开发一个有意义的计划，即使有可能也非常困难，所以在沟通活动中得到的信息必须要小心地进行评审。如果现在范围是模糊的（真的可能是），那么在计划可以开始之前，你需要做一些事情。

5.1.1 有哪些相关的沟通工作产品

所有的沟通工作产品都和计划活动相关。在某些情况下，可用信息也许只有你的书面笔记（总是做笔记！）和一系列使用场景，这些使用场景提供了对 WebApp 增量的内容和功能的描述。如果你采用一种更正式的沟通方法，那么下面的这些工作产品（第 4 章）对计划活动可能是有用的：

- 描述了 WebApp 总体业务动机的陈述
- WebApp 总体目标的陈述
- 用户种类列表

- 将要计划的 WebApp 增量的信息型目标列表
- 将要计划的 WebApp 增量的应用（功能）型目标列表
- 对增量的描述（范围陈述）
- 增量的内容对象列表
- 增量的功能列表
- 描述了每个用户种类如何与增量交互的使用场景集合

尽管当所有的这些信息都可用的时候的确很棒，但是它对于计划活动来说却并不充分。在沟通活动中得到的信息也许并不像 WebE 团队想要的那么完整。因此当这种情况发生时，重要的是知道去做什么。

5.1.2 如需要更多的细节来理解增量会怎么样呢

当人类面对一个复杂或者模糊的问题时，他们倾向于应用一种分治的策略。简单地说，就是把问题分解为更容易理解的更小问题。因为沟通活动并不试图完全地定义 WebApp 将要交付的内容和功能的每一个方面，所以当以一种更加完整的方式来理解 WebApp 增量的范围而需要更多的细节时，[⊖]可以使用一种*细化*策略。

细化开始于作为沟通活动的一部分而已经获得的信息。它被应用到两个主要的方面：随着 WebApp 增量而交付的内容和作为增量的所有用户场景的一部分而被调用的功能。

回想在第 4 章中执行的工作，SafeHomeAssured. com WebApp 的第四个增量实现了两个使用场景：

开发将要被监控的空间的布局。
获取我的空间的传感器布局的推荐。

使用场景隐含了一些主要内容对象以及与它们相关的功能的存在。然而，这些内容对象和功能还没有在沟通中进行明确的定义。你有三个选项：1）继续计划，然后在建模活动中担心内容和功能性的细化；2）现在就做一点儿细化，那么计划就能更加可靠；3）做完全的细化来确保你真正理解这个增量。

这里没有“最好的”选项。团队也许觉得技术性工作（在这里就是分析建模）应该开始了（时间总是非常重要的），同时可以选择使用现在可用的所有信息来开发他们所能开发的最好的计划。或者，团队可以选择去做少量的细化，这样基于将要得到的内容和功能，计划就可以恰当地分配技术性工作。最后，团队可以决定现在就做分析建模，并且在开发计划之前进行完全的细化。

作为一个例子，我们假设选择了第二个选项。团队现在将做一点儿细化，因此计划可以是完整的，但是将会把一个更完整的细化推迟到建模活动开始之前。[⊜]因为使用场景涉及“空间”，所以 WebE 团队就在这个概念之上进行细化：

空间——住宅或者小型商务的一个定义好的建筑平面图
内容对象：
空间标识符
空间名称

⊖ 对内容和功能更完整的定义，将作为建模活动中分析模型动作的一部分进行开发。
⊜ 值得注意的是，要注意现在进行的细化工作并不是“浪费”，因为它将在后续过程中节省时间和工作量。

客户名字
墙壁
墙壁名称（显示为一个数字）
起点和终点的坐标
门道
墙壁的 ID
门的尺寸
门的起点坐标
窗户
墙壁的 ID
窗户的尺寸
窗户的起点坐标

……和这个增量有关联的所有内容对象。

功能（与“空间”有关）：
明确说明并画出墙壁
明确说明并画出门道
明确说明并画出窗户
计算每一个房间的尺寸
保存/检索一个被命名的空间
需要安全性保护
更新/删除一个被命名的空间
需要安全性保护
打印一个被命名的空间

……与这个增量有关联的所有功能。

与这个精化相关的工作量相对来说是比较小的，但是它的收效却会很显著。例如，你现在知道团队将必须开发和一个用户空间相关的至少 7 个功能。与之相关的工作将作为计划活动的一部分进行分配。

5.1.3　你的理解中仍然存在不足怎么办

你不可能在计划阶段就期望得到一个全面的理解。你将必须接受这样的现实——即使在你细化了从沟通活动中得到的信息之后，事情仍然有一些不确定性，它是所有工程工作的固有风险中的一个。更重要的是，你将不得不使用有缺陷的信息来完成计划活动，并继续前进。

当你进一步深入到 WebE 过程时，随着建模活动的进行，你的理解将会加深。然而，我们有理由去假设，随着构建的进行，仍然会有一些惊奇出现（有些可能是严重的）。团队必须要适应，并继续前进。

5.2　精化框架活动

WebE 团队现在必须选择对于将被应用到增量上的剩余 WebE 工作来说正确的框架活动和任务。回顾在第 3 章中的讨论，下面的这些框架活动必须被加以精化，以使用到将要创建的增量之上：

- **建模**。导致分析和设计模型的创建的动作和任务，这些模型有助于 WebE 团队和其他利益相关者更好地理解 WebApp 以及它是如何被构建的。
- **构建**。代码生成（手工或者自动）和测试所需要的动作和任务。
- **部署**。把 WebApp 增量交付给最终用户的动作和任务，最终用户将对增量进行评估并提供基于评估的反馈。

这些框架活动可以应用于所有的 WebApp 增量。问题是要去精化对于这个增量而言合适的动作和任务集合。

5.2.1 需要哪些动作和任务

WebE 框架动作和任务是通过“融合”增量和过程的特点来进行精化的。一种实现方式就是去创建一个和图 5-1 类似的表格。表头的一行列出了将作为增量部署的一部分进行交付的关键内容对象和功能。第一列列出了用于建模、构建和部署的框架动作和任务集合。

框架动作和任务 \ 内容和功能	…	墙壁	门道	窗户	详细说明并绘制墙壁	详细说明并绘制门道	详细说明并绘制窗户	计算每个房间的大小	保存/检索一个命名空间	更新/删除一个命名空间	打印一个命名空间	…
建模分析												
审查用户场景												
显示内容关系												
创建交互模型												
详细说明内容细节												
定义数据库需求												
精化功能需求												
精化界面需求												
设计												
实施界面设计												
详细说明交互细节												
精化页面布局												
显示导航机制												
实施美学设计												
⋮												

图 5-1 合并问题和过程

WebE 团队选择那些对将要设计的 WebApp 增量来说最合适的动作和任务集合。虽然一些团队成员将会主张使用一个非常稀疏的列表，但是最好花一些时间制作一个动作和任务的集合，来得到一个高质量的结果。值得注意的是，一个任务只是被列在表格之中，并不意味着非得把它

应用于每一个内容对象和功能。最终的决策由将做工作的 Web 工程师来做出。然而，如果 Web 工程师决定提前开始 WebE 任务，那么他就必须能够证明决策的合理性，同时将最终为生产的工作产品的质量负责。

图 5-1 仅仅说明了将为 SafeHomeAssured. com 这个 WebApp 创建的表格的一部分[⊖]。在所展示的部分中，说明了使用场景的内容和功能：

开发将要被监控的空间的布局。
获取我的空间的传感器布局的推荐。

内容对象（墙壁、门道和窗户）列在第一行的第一部分。功能（例如，明确说明并绘制墙壁）列在后面。

在一个简单的讨论之后，WebE 团队选择将用于建模、构建和部署这些内容对象和功能的任务集合。参见图表，你将会发现团队对第 3 章中建议的一般框架进行了改编，对一些任务进行了重新安排，修改了其他的任务，同时增加了一些任务。在每一个实例中，过程的敏捷性和最终结果的质量将在做出决策的时候进行指导。表内部的矩阵用于表明哪些动作和任务对于哪个内容和功能来说是合适的。并非每一个动作和任务都将被应用到每一个内容对象和功能。

值得注意的是，为一个增量定义的动作与任务集合和为下一个增量所定义的集合不必是相同的。和每个增量相关工作的规模和复杂性将指导 WebE 团队选择将要应用的框架任务。

5.2.2　生产哪些工作产品

中间工作产品（例如，建模表示、界面草图、导航图）的数目应当保持对于为下一个框架动作或任务提供合适的指导来说是必要的最小值。应当有三个标准指导团队：

1. 对于更好地理解 WebApp 增量和获得一个高质量的结果而言，工作产品是否完全必要？
2. 对于将在后面的 WebE 任务中进行的工作来说，工作产品是否是作为它的一个有用的基础？
3. 工作产品是否提供了将在其他 WebApp 增量的设计中使用的信息？

如果这些问题的答案都是肯定的，那么你应当考虑创建这个工作产品。创建不必要的工作产品不仅会浪费精力，同时会僵化 WebApp，并且会鼓励开发者去抵制那些可能是非常合适的变化。

作为例子，考虑一种状况——团队正在讨论是否有必要去创建一个分析建模工作产品。在这里，在这个场景中，工作产品就是一组 UML 图（第 7 章），它将肯定会有助于理解增量，从而也就改进了质量。这些图形将作为后续设计工作的指导，而且也许会在为部署安排的接下来的两个增量中起作用。因为这三个标准都满足了，所以 WebE 团队决定去开发这个工作产品。

有一个团队成员反对这个决定。

“我有一种很强烈的感觉，利益相关者将在这个功能上改变主意”，他说道，“而且我们将不得不一次又一次地对这组图形进行更新。这将是浪费时间。”

每个人都愣了一会儿。Web 工程师已经做了非常好的工作。如果内容和功能非常有可能变化，那么讨论中的工作产品将不得不随着每个变化进行更新——一个消耗时间并且很可能有潜在错误的任务。怎么办呢？

团队负责人考虑了一下这种情况。

⊖ 这个表格常常会使用一个电子表格模型来创建，从而易于在团队成员之间进行修改和分发。

"要是我们假设不会发生变化来进行工作产品的创建会怎么样。它们对于指导设计来说确实是有用的，"她说，"如果你是正确的并且利益相关者确实改变主意，那么我们将放弃这些图形并对设计本身进行适当的改变。"

团队花了一点儿时间来讨论这个问题，然后再次回到其决策上来。

5.2.3 评估质量的恰当方法

你可以正确地完成它，或者你可以从头重做一次。如果一个 WebE 团队在所有框架活动中都强调质量，那么团队将减少必须重做的工作量，结果是只需要更低的成本，更重要的是，加快了上市时间。

但是仅仅在口头上说质量很重要是不够的。你必须明确定义当你说"WebApp 质量"的时候，你的意思是什么，而且定义一组任务，这组任务将有助于确保每一件工作产品都表现出很高的质量。

当我们在本书中讨论 WebApp 质量的时候，我们指的是问题定义的完整性和准确性、解决方案设计的有用产品、构建的稳定性，以及 WebApp 增量满足所有利益相关者需求的程度。[⊖]

为了在产生工作产品时保证它们的质量，WebE 团队可以进行结对走查（pair walkthrough）。这种方法改编自极限编程方法［Bec99］［Jel01］中的结对编程的概念，它提出所有的工作产品都由一对 Web 工程师来进行评审。工作产品的生产者把它交给另一个团队成员来查找错误、不一致和遗漏。结对的两个人通过在一起工作设法去改进工作产品的质量。

在工作产品复杂或者它的影响对项目的成功来说很关键的情况下，WebE 团队可以安排一个团队走查（team walkthrough）——一种涉及一些团队成员的审查形式，由一个很清晰的规则集合进行指导。关于结对走查机制更详细的讨论以及团队走查更多的细节将在本章的后面进行讲解。



结对走查

场景：建模活动工作开始之前，Web 工程区域的会议室

参与者：SafeHomeAssured.com 的 WebE 团队成员

会话：

团队负责人：明天我们将开始建模，我想转为使用"结对编程"的方法。

团队成员 1：结对编程？我以为我们将只是使用结对走查。我被弄糊涂了。

团队成员 2：它们实际上是同样的，至少我是这么理解的。你在做工作的时候进行审查。

团队成员 3：但是我们每个人都要做自己的工作，对吧？

团队负责人：你们作为一对进行工作。

团队成员 3：呃？

团队负责人（在文件夹中寻找一篇文章，发现了它）：听一下这个引用，它来自我发现的关于结对编程的一篇文章［Wil99］："在结对编程中，分配两个程序员共同生产一件产品（设计、算法、代码等）。这两个程序员就像一个连贯而聪明的有机体，使用同一种思想，负责这件产品的每一个方面。一个人打字或书写，另一个则不断地对工作进行审查。但是，两个人在这个过程中是平等的参与者。如下的事情是不可接受的，'你在设计中有一个错误，'或者'那个瑕疵来自你的部分。'取而代之的应该是'我们把设计弄糊了，'或者，'我们刚刚通过了没有瑕疵的测试！'两个合作者共同拥有所有的事情。"我认为那只是把它概括起来。

⊖ 在第3章中，我们用在用户指目的的适当性，也就是说，设计是否符合利益相关者的需求？这两种指错误被消除的程度。

团队成员 2：所以，我们中的两个人将一起工作来开发一个模型，同时在开发过程中寻找错误。
团队成员 1：呀！我不知道这个。对我来说，它看起来就是冗余。我认为它将使我们慢下来。
团队负责人：恰恰相反。事实上，有很多案例证明我们可以更快地交付 Web 工程工作产品，而且会有更高的质量，更少的故障。
团队成员 3：但是我习惯了单独工作。对于这，我一无所知。
团队成员 2：我认为我们还是给它一个机会。如果我们失败了，我们将重新评估，但是我有一个感觉，它将会成功。
团队成员 3（想了一下）：那么当我们使用结对编程和走查的时候，就没有分离的评审了？
团队负责人：在某些点上，我们也许仍然有一个团队走查，但是我们大多数的质量控制将作为这种方法的一部分进行。
[团队成员们点着头，热情不是很高，但也没有任何敌意。]
这就是我刚引用的文章。今天早上读一下，然后我们将在今天的晚些时候开会，对结对编程方法进行更详细的讨论。

5.2.4　如何管理变更

如果你不管理变更，那么它就会管理你。那绝不是好事。一连串未控制的变更可以很容易把一个运行良好的 WebE 项目搞得一团糟。正因为这样，对变更进行计划是计划活动必不可少的一部分。

变更有很多不同的来源：1）非技术的利益相关者可能会对总体 WebApp 或者当前正在开发的增量有一个事后想法[⊖]；2）最终用户可能需要不同的交互模式或者不同的功能或内容；3）Web 工程师也许认识到要完成 WebApp 需求，需要意料之外的修改。

在每一种情况下，请求的变化都必须进行明确的描述，进行评估来决定对增量和总体 WebApp 的影响，同时进行评估来估计做出变更需要的工作量粒度。问题是，所有的这些变更管理活动都要消耗资源和时间——在 WebE 团队工作于一个增量时，这些常常都是供不应求的。然而，利益相关者想要做出变更。

在本章的后面我们将讨论变更管理——一个保护活动，它被加以设计来处理那些强制的需求来做出请求的变更，同时减轻一个专注于建模、构建和部署的 WebE 团队在变更方面当前的负担。

5.3　组建一个 WebE 团队

Web 工程强调敏捷性，同时敏捷的哲学理念强调个人能力与团队协作联系在一起的重要性。这些对于一个 WebE 团队来说是关键的成功因素。Cockburn 和 Highsmith［Coc01］在写作的时候，提到了：

> 很有趣的是，在一起工作并有着良好沟通和交互的人们的工作水平明显高于他们独立使用个人才能时的水平。我们无数次地在头脑风暴和联合解决问题的会议中看到这一现象。因此敏捷的项目团队［WebE 团队］致力于增加个人能力和协作水平两个方面。

为了有效利用每一个团队成员的能力，同时促进一个项目内的有效合作，WebE 团队应当是

⊖ 事后想法可能由如下的事物促成：新的业务需求（例如，规划的市场人口统计规律和最初规划的不一致）、新的或者修改了的内容或功能（例如，最初展示的内容不是现在想要的）、管理上的变更（例如，一个新的执行经理“想走另一条路”）等。

自组织的（self-organizing）。

在 Web 工程的环境中，自组织包含了三件事情：1）WebE 团队对要完成的工作进行自组织；2）团队对过程框架进行组织来最好地适应它的局部环境；3）团队对工作进度表进行组织来最好地完成 WebApp 增量的交付。自组织有很多技术方面的好处，但是更重要的是，它可以改进协作并且提升团队的士气。从本质上来讲，团队充当了自己的管理部门。Ken Schwaber［Sch02］在写作的时候，提到了这些问题："团队选择它认为自己在迭代［增量］中可以完成的工作量，并且团队保证完成工作。没有什么比其他人为自己做保证更能让一个团队失去动力了。没有什么比接受完成自己给自己做的承诺的责任更能激发一个团队了。"

5.3.1 如何识别一个优秀的 WebE 团队

每一个组织的目标都是去创建一个展示出团结的 WebE 团队。DeMarco 和 Lister［Dem98］在其论著《Peopleware》（人件）中，讨论了这个问题：

> 在商业界，我们趋向于相当随便地使用团队这个词，把分配到一起工作的任何团体都称为"团队。"但是有很多这样的小组看起来并不像团队。他们没有一个统一的成功定义或者任何鲜明的团队精神。它们缺少的就是我们称之为凝聚力的珍贵东西。
>
> 一个有凝聚力的团队是一组团结紧密的人员，他们的整体力量大于他们所有个体的总和……
>
> 一旦一个团队开始具有凝聚力，成功的可能性就会大大提高。这个团队可以变得不可阻挡，成为成功的象征……他们不需要按照传统的方式进行管理，他们当然也不需要被激励。他们已经有了动力。

DeMarco 和 Lister 认为，有凝聚力的团队同一般的团队相比，具有更高的生产率和更大的动力。他们拥有一个共同的目标和共同的文化，而且在很多情况下，一种"精英意识"使他们很独特。

5.3.2 团队为什么没有凝聚力，怎么改善

不幸的是，一个 WebE 团队可能遭受团队毒性（team toxicity）——由"培养一个潜在有毒团队环境"的以下 5 个因素所定义的一个弊病［Jac98］：

1. 混乱的工作环境，团队成员在其中浪费精力，并失去对将要完成的工作目标的注意力。
2. 引起团队成员间产生摩擦的个人、业务或者技术因素所带来的重大挫败。
3. "零碎的或者不协调的步骤"或者一个定义拙劣或者选择不恰当的过程模型，它将成为实现之路上的障碍。
4. 对角色没有清晰的定义，导致缺少义务以及由此带来的相互指责。
5. "连续不断地重蹈覆辙"导致丧失自信和士气低落。

总的来说，如果 WebE 团队是自组织的，那么每个毒素都是可以避免的。一个自组织的团队可以访问完成工作所需要的所有信息，因此也就避免了一个混乱的工作环境，人们在其中争抢着去发现必不可少的信息。一个自组织的团队拥有对所使用的过程、生产的工作产品、定义的工作进度以及所实现的质量和变更管理活动的控制。因此，团队就避免了由于缺乏控制所引起的挫败。一个自组织的团队建立其自己的责任机制（例如，结对走查就是一种很好的完成方法），并且定义当团队的一个成员完成任务失败时的一系列纠正方法。

每个 WebE 团队都会经历小的失败。避免失败的关键是去建立一个基于团队的用于反馈和解决问题的技巧。而且，任何团队成员的失败必须视为团队自身的失败。这会带来一种面向团队的

方法——它可以纠正错误，而不是在有毒的团队中快速增长的相互指责和不信任。

5.3.3 WebE 团队可以管理自身吗

一些人可能认为自己动手的团队管理和自己动手的脑部手术相似，但是我们不同意（至少在某些时候[5]）。如果一个 WebE 团队的经验很丰富而且有能力，那么在没有一个正式“项目经理”的情况下可能产生出 WebE 增量。应当任命一个团队负责人来协调和沟通工作任务，但是团队成员可以通过进行日常会议来对进展和问题进行评估，这个会议用来协调和同步当天必须完成的工作。这些简短的会议（例如，15～20 分钟）解决 4 个问题：

1. 从上一次会议到现在，我们实现了什么？
2. 在下一次会议之前，需要实现什么？
3. 每个团队成员如何为需要完成的事情做出自己的贡献？
4. 存在那些必须要克服的障碍？



团队结构

场景：工程经理的办公室

参与者：SafeHome 产品线的经理和 WebE 团队的 3 个成员

时间：SafeHomeAssured. com 项目启动之前

会话：

经理：你们都仔细看过营销人员准备的关于 SafeHomeAssured. com 的基本信息了吗？

团队负责人（看看他的团队成员并点头）：是的，但是我们有很多问题。

经理：我们过一会儿再讨论这些问题。我想讨论我们将如何组织一个团队，谁将负责什么……

团队成员 1：我真的被敏捷方法所迷住了……我认为我们应当是一个自组织的团队。

团队负责人：我同意。给定严格的时间线和一些不确定性，而且事实是我们都是真正有能力的［笑］，这看起来是正确的方法。

经理：这对于我来说没有问题，但是你们知道如何操作这件事？

团队成员 2（边笑边讲，就好像在背诵）：我们做了一些战术决定，确定由谁做、做什么、什么时候做。按时交出产品是我们的责任。

团队成员 1：还要有质量。

经理：很正确。但是记住有一些约束，市场部决定要生产的增量——当然，是在与我们协商之后。

团队成员 2：还有呢？

经理：还有就是，我们将使用 WebE 过程框架来完成这个工作。

团队成员 2：但是保持无关的文档是最少的。

经理：那请和我联络？

团队负责人：我们决定由我来做技术负责人，因为我最有经验，所以我是你的联络人，但是欢迎和我们中的任何人进行交流。

经理（笑）：别担心，我会的。

随着这些问题答案的出现，团队以实现将要完成工作的方式对其方法进行了改进。随着时间的推移，持续的自组织和协作把团队推向了一个完整的 WebApp 增量。

---

[5] 如果 WebApp 项目非常复杂，可能需要一个大团队或者多个团队。在这种情况下（或者团队成员都没有经验的情况下），项目经理将扮演更重要的角色。

5.3.4 如何组建一个成功的团队

Tracy Kidder［Kid00］在他的关于很久以前计算机行业的畅销书中讲述了一个故事，一个计算机公司的壮举，它试图创建一台计算机来应对一个比其强大的竞争对手创建的一个新产品所带来的挑战。[⊖]这个故事是如下情况的一个隐喻：团队工作、领导力，以及当关键项目的进展不像所计划的那样顺利的时候，所有技术人员都会遇到的难以忍受的压力。

Kidder的书的总结几乎无法正确地使用，但是在你创建一个有效的WebE团队的时候，下面的这些关键点［Pic01］特别重要：

应当建立一套团队指导方针。这些指导方针包括：每个人的期望是什么，问题如何被处理，以及随着项目的进行存在哪些机制来改进团队的效率。

必须有坚强的领导力。团队负责人必须要以身作则，与团队成员进行沟通，同时必须展现出一定程度的热情来让其他的团队成员从心理上为他们面临的工作“签订合同”。

尊重个人才能很关键。没有人对所有的事情都精通。最好的团队利用个人的力量。最好的团队负责人允许个人在有好想法的时候自由地去做。

每个团队成员都应当做出承诺。Kidder的书中的主人公称之为“签订合同”。

开工很容易，但是保持势头却很难。最好的团队从不会让一个“不能克服的”问题阻止他们。团队成员开发一个“足够好的”方案并前进，希望前方的进步势头可以在后面的项目中带来一个更好的方案。

Scott Rosenberg［Ros07］在标题为令人好奇的《Dreaming in Code》（梦断代码）一书中讨论了软件开发者（和Web工程师）所面临的挑战。在书中，一个开源团队努力开发一个被称为Chandler（chandler. osafoundation. org）的产品，用于支持工作组的革新软件。在对技术团队所面临的挑战的讨论中，他写道：

> 随着时间的流逝，一组软件开发者在一系列项目中一起工作是非常罕见的。在这里他们不像运动队、部队或者音乐剧团，而更像一组组合起来的有相同意见的人制作一部电影，然后解散并在下一部电影开始时重新结合。所以，虽然单个程序员和经理可以具有在过去都很好地为他们服务的丰富的技术经验和知识，但是每当他们组建成新的团队并开始一个新的项目时，他们很可能会最终按下重置按钮，同时不得不从最初的原则开始设计一个工作过程。

我们认为Rosenberg有一半是对的。一般而言，Web工程师和软件开发者确实非常像一个电影制作公司——每一个人在他的特定领域都是很有创造性和博学的。但是我们认为每一次初始化一个新项目时都“从最初的原则开始设计一个工作过程”既没有必要，也是不可取的。成功的团队可以利用一个定义良好的过程框架而没有任何的重新发明。这将会为他们节省时间，而且更重要的是，它建立了一个基础，高质量的产品可以从中产生。

5.3.5 一个好的团队负责人有哪些特征

在一本关于技术领导力的优秀书籍中，Jerry Weinberg［Wei86］提出了领导力的MOI模型：

激励。鼓励技术人员去发挥其最好才能的一种能力。这可以通过为好的表现提供奖励并对

⊖ Kidder的《The Soul of a New Machine》（计算机之魂），最初出版于1981年，非常适合那些想以计算作为职业或者打算把计算作为职业的人。

差的表现强加惩罚来实现。

组织。塑造能够将初始概念转化为最终产品的现有过程（或发明新的过程）的能力。

想法或创新。即使必须工作在一个特定 WebApp 所建立的范围之内，也能鼓励人们去创造并让人感到自己有创造性的一种能力。

Weinberg 提出，成功的项目负责人应采用一种解决问题的管理风格。也就是说，WebE 团队的负责人应当关注于理解将要解决的问题，管理思想流，同时，让团队中的每个人知道（通过语言，更重要的是通过行动）质量是很重要的，不能妥协。

5.4 管理风险

风险管理包括一系列的任务，用来帮助一个 WebE 团队理解和管理那些将破坏一个 WebApp 项目的很多问题。风险是一个潜在的问题——它可能发生，也可能不发生。但是，不管发生还是不发生，我们都应该去识别它，评估它发生的可能性，估算它的影响，并制定一个应急计划来应对问题的真正发生。

一个 WebE 团队从两个不同的粒度层次上对风险进行考虑：1）风险对整个 WebApp 项目的影响；2）风险对当前正在设计的 WebApp 增量的成功部署的影响。

在项目层次上，必须要询问和回答很多和风险相关的问题：计划的 WebApp 增量是否可以在定义的时间框架内交付？在设计其他增量的同时，这些增量是否能为最终用户提供不断向前发展的价值？对变化的要求将如何影响交付进度？团队是否理解了需要的 Web 工程方法、技术和工具？可用的技术对于这个工作来说是否合适？可能发生的变化是否需要引入新技术？

在增量层次上，关注点更偏向基础。沟通活动是否为建模、构建和部署开发了足够的信息？精化了的过程框架是否适合将要开发的增量？团队是否有正确的技能组合来构建这个增量？内容和功能是否进行了充分的定义？增量是否将提出技术挑战？

风险管理在计划活动中启动，但是事实上它是一个在过程流中被重复访问的保护活动。一个 WebE 团队的挑战就是做足这个工作以便在风险中占据主动，但是不要多到把其他的开发工作减慢到缓慢的进行。我们来研究这些基本原理。

5.4.1 如何识别风险

在计划活动的某些点上，WebE 团队共同提到一个基本的问题："什么会出错？"每个团队成员都被要求做一个风险列表，这些风险可以归为如下几类：1）人员风险；2）产品风险；3）过程风险。

人员风险（people risk）是那些可以直接追溯到一些人的动作或者失败之上的潜在问题。例如，WebApp 增量很可能将需要一些使用 XML 开发的组件，但是团队中当前没有人有 XML 的经验。一个与技术相关的人员风险就被识别了。或者一个特定的利益相关者在过去做信息请求时可能不愿配合，而这些请求的信息对于一个成功的结果来说又是很关键的。一个与沟通相关的人员风险就被识别了。

产品风险（product risk）通常可以追溯到和 WebApp 的内容、功能、限制或性能相关的潜在问题。例如，当构建开始之后，才发现了一个风险，它可能影响 SafeHomeAssured. com 第一个增量（一个信息型 WebApp）很紧的交付时间。一个主要内容对象（SafeHome 产品的一个综合描述）过时了，并且可能需要进行大量的修改之后才能进行部署。与 SafeHomeAssured. com 下一个增量（室内视频监控）相关的性能风险可能是可疑的控制和用于视频与音频设备的监控界面。

过程风险（process risk）是那些与团队选择的框架动作和任务紧密相关的问题。有时候，过

多的过程可能就是一个潜在的风险。例如，团队决定去做一个彻底的分析模型（第7章）并详述实现它的动作和任务（以及相关的工作产品）。对于SafeHomeAssured. com WebApp的第四个增量（实现了将要被确保安全的空间布局并且为此空间推荐传感器）来说，一个过程风险也许是开发一个完整的分析模型所需要的相关工作可能过于消耗时间并将导致设计和构建活动的推迟。

潜在风险列表从团队成员那里收集而来，并按照类别进行合并。然后，WebE团队一起来评估这些潜在的风险。

5.4.2 如何评估风险

一旦形成了合并的风险列表，WebE团队就要执行一个快速的评估。对每个风险都进行简要讨论并且以两种方式进行评估：1）风险变为现实的可能性或概率；2）如果风险发生，与其相关的问题所带来的后果。目的就是以一种可以排定优先次序的方式来考虑风险。没有任何WebE团队有时间或者资源以同样的严格程度来处理每一个潜在风险。通过对风险排定优先次序，团队可以把资源分配到它可以发挥最大作用的地方。

一旦估计了可能性和影响，团队就可以创建一个风险表。[⊖]一个风险表样本如图5-2所示。表的3列反映风险、它们发生的可能性（通常以百分比表示），以及它们对进度和成本的影响（常常表示为1［低］到4［高］之间的序数范围）。

风险	可能性	影响
人员		
团队缺乏XML经验	80%	3
利益相关者缺乏合作性	60%	2
高级经理可能在中途变化	40%	1
产品		
信息内容可能已经过时	50%	2
算法可能没有被充分定义	80%	3
WebApp的安全性比预期的要难	80%	3
数据库集成比预期的要难	40%	3
空间定义功能比预期的要难	70%	3
过程		
对沟通不够重视	60%	2
分析太多（花费太多时间）	30%	1
对导航设计不够重视	40%	2
⋮	⋮	⋮

图5-2 分类之前的样例风险表

一旦完成了风险表的列之后，就可以先按照概率然后按照影响进行排序。也可以计算出可能性×影响得到的综合分数，然后在这个基础上对表进行排序。高概率、高影响的风险放到表的上方，而低概率风险则移到表的下方。这样就完成了一级的风险排序。团队对产生的有序风险表进行研究并定义一个中截线（cutoff line）。中截线（在表中某一点水平绘制）表示：只有那些在中截线之上的风险才会得到进一步的关注。

⊖ 风险表可以实现为一个电子表格模型。这能简化条目的处理和排序。

5.4.3　如何开发应急计划

因为每个 WebApp 增量的总体开发时间很短，所以书面化的应急计划将不会作为 WebE 过程框架的一部分进行开发。但是这并不是说 WebE 团队简单地忽视高优先级的风险。

团队成员考虑风险表中落在中截线之上的每一个风险，并回答 3 个问题：

1. 我们如何能完全避免这些风险？
2. 为了决定风险的可能性将变大还是变小，我们可以监控哪些因素？
3. 一旦这个风险变为现实，我们将为此做些什么？

这些问题的答案可以由团队负责人来进行记录（作为非正式说明）。



分析风险

场景：为 SafeHomeAssured.com WebApp 制定计划期间，Web 工程区域会议室

参与者：WebE 团队的所有成员

会话：

团队负责人（看着图 5-2 所展示的风险表）：我们已经在风险表上做了一次排序，有 3 个风险吸引住了我：我们团队中 XML 经验太少，WebApp 的安全性可能比我们想象的要更复杂，而且空间定义功能（我们计划在增量 4 中进行交付的布局设计）肯定比我们想象的更复杂。我认为……

团队成员 1（打断）：我认为这 3 个风险对于我们来说都是致命的。事实上，我真的已经开始怀疑为这个项目所安排的进度了。这太疯狂了。

团队成员 2：我不得不表示同意。我只会一个 XML 皮毛，我也许需要帮助而且……

团队成员 3（打断）：而且在这个团队中，谁对于我们必须要在后续增量中实现的安全功能真正在行呢？肯定不是我……

团队负责人（举起他的手来阻止大家的讨论）：大家不要抓狂！我也认为这些问题都代表了风险，而且极是那些给定的可能性，它们很可能成为现实。但是，我们的工作是决定对于这些风险我们应当做些什么。有什么想法吗？

团队成员 2：好的，Tyson McNeil［另一个软件开发者］对 XML 比较在行。我想如果我遇到困难……

团队负责人：让我们更正式一些吧。我将查明接下来的 3 个月内 Tyson 的任务是什么。也许他可以在我们需要 XML 技能时，为我们做准职。如果不行，我们必须制定一个可靠的替代策略。

团队成员 1（改变了话题）：我不得不相信我们低估了布局设计功能所需要的工作量。我认为我们应当现在就去找市场人员并告诉他们。

团队负责人：我不行不同意。我将和市场人员商量，看我们能不能获得更多的时间。如果没有，我希望你们去看看有没有这个功能的简约版本，这样我们就可以在分配的时间内交付。如果他们不同意推迟交付，我想把它作为一种选择。

团队成员 1：会议一结束我就开始进行。

团队成员 3：那安全功能怎么办呢？

团队负责人（想了一下）：我认为我们可能需要一点儿外部帮助。在项目预算中有一部分资金用于咨询帮助，我将马上尝试邀请一个安全专家。

团队成员 1：其他的风险呢？

团队负责人：除了算法定义的潜在问题，它们的可能性都比较小，所以现在，我们将把它们放在一边。但是我们应当讨论算法……

5.5 开发进度表

曾经有人请教《The Mythical Man-Month》（人月神话）［Bro95］的著名作者 Fred Brooks，问他软件项目进度是如何延误的？他的回答简单而富有哲理："某天某时"。

任何技术性项目（不论它涉及水电站的建设，还是 WebApp 的开发）的现实情况都是，必须要以数以百计的小任务来实现一个大目标。这些任务中有些处于（动作）主流之外，其进度不会对项目的完成日期造成影响。而有些任务则位于"关键路径"之上，如果这些关键任务进度拖后，则 WebApp 增量的完成日期就受到威胁。

WebE 团队的目标是列出一个增量的所有 WebE 动作和任务，并创建一个描述它们之间依赖关系的网络，识别网络中的关键任务，然后跟踪它们的进展以确保任何延误都会被识别为"某天某时。"为了实现这个目标，团队负责人必须有一个定义好的进度表，其定义的解析度应当允许对进展进行监控，对项目进行控制。

WebApp 项目进度安排是一种活动，它通过在计划的时间线（持续时间）跨度内，对为特定 WebE 任务分配所估计的工作量来创建一个增量。然而，重要的是，要注意总体的 WebApp 进度会随着时间而不断演化。在 WebE 过程框架的第一次迭代中，开发一个宏观进度表。该进度表标识出所有 WebApp 增量和项目将要被部署的日期。随着增量开发的进行，宏观进度表上增量的条目将精化成详细的进度表。这样就识别出了特定的 WebE 任务（完成一个活动所需要实现的），并做出进度安排。

可以从两种大不相同的角度来讨论 WebE 项目的进度安排。第一种情况，WebApp 的最终发布日期已经确定（而且不可更改），WebE 组织必须将工作量分配在预先规定的时间框架内。WebApp 进度安排的第二种情况，假设已知大致的时间界限，但是最终发布日期由 WebE 组织自行确定。工作量是以能够最好地利用资源的方式来进行分配，并且在对 WebApp 进行仔细分析之后才确定一个最终发布日期。但不幸的是，第一种情况的发生比第二种情况的发生频繁得多。

5.5.1 什么是宏观进度安排

作为宏观进度安排的一个例子，考虑 SafeHomeAssured. com WebApp。回顾在第 4 章中进行的讨论，为这个项目识别了 7 个增量：

增量 1：基本的公司和产品信息
增量 2：详细的产品信息和下载
增量 3：产品报价和处理产品订单
增量 4：空间布局和安全系统设计
增量 5：监控服务的信息和订购
增量 6：监控设备的在线控制
增量 7：访问账户信息

WebE 团队和利益相关者进行协商，为所有的 7 个增量开发一个初步部署进度表。这个进度表的时间线如图 5-3 所示。

重要的是，要注意部署日期（表示为时间线上的菱形）是初步的，而且可能会随着增量进度安排的进一步细化而变化。然而，这个宏观的进度表为管理提供了内容和功能何时可用以及整个项目何时完成的说明。作为一个初步的估计，团队将工作 12 周的时间来部署所有的增量。还应当注意其中一些增量将进行并行开发（例如，增量 3、4 和 7）。这就要假设团队将有足够的人员来做这项并行的工作。

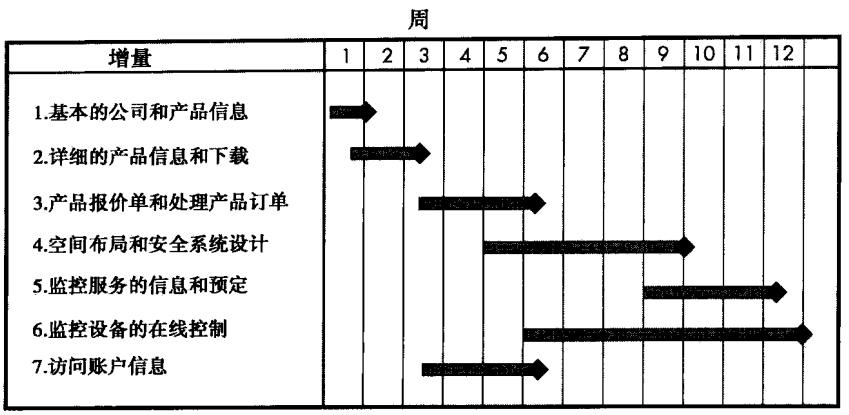


图 5-3　宏观项目进度时间线

5.5.2　什么是增量进度安排

一旦宏观进度表制定完毕，团队就要为一个特定的增量排定工作任务。为此，你可以使用一个对所有 WebApp 增量都适用的通用过程框架（第 3 章）。一个*任务列表*（task list）可以使用如下方式进行创建：以作为框架的一部分得到的通用任务为基础，考虑从特定的 WebApp 增量得到的内容和功能，对这些通用任务进行修订。

每个框架动作（及其相关任务）都可以通过如下 4 种方式之一进行修订：1）不修改而直接应用一个任务；2）去掉对于这个增量来说不必要的任务；3）增加一个新（定制）的任务；4）一个任务被分解（精化）为一些被命名的子任务，这些子任务都将成为进度表的一部分。

回顾在第 3 章中对建模活动的讨论，我们注意到，通用的“设计建模”动作可以通过应用下面的一部分或者全部子任务来完成：

- 设计界面。
- 对 WebApp 进行美学设计。
- 设计导航模式。
- 设计 WebApp 的架构。
- 设计内容以及支持它的结构。
- 设计功能组件。
- 选择合适的设计模式。
- 设计合适的安全和隐私机制。
- 对设计进行评审。

作为任务分解的例子，在通用任务“设计界面”被应用到 SafeHomeAssured. com 的第四个增量的时候，对其进行考虑。前面叙述过，第四个增量实现了描述 SafeHome 安全系统所保护的生活或者商务空间的内容和功能。参见图 5-3，第四个增量在第 5 周开始时开始，第 9 周结束时结束。

毫无疑问，“设计界面”任务必须要进行引导。团队认识到界面设计是这个增量成功的关键，决定对这个任务进行分解（精化）。对于第四个增量的“设计界面”任务来说，得到了如下子任务：

- 为空间设计页面开发一个页面布局的草图。
- 与利益相关者一起对布局进行评审。
- 设计空间布局的导航机制。

- 设计“制图板”布局。[⊖]
- 为图形化墙布局功能制定程序细节。
- 为墙的长度计算和显示功能制定程序细节。
- 为图形化窗户布局功能制定程序细节。
- 为图形化门布局功能制定程序细节。
- 设计选择安全系统组件（传感器、摄像机、手机等）的机制。
- 为安全系统组件的图示布局制定程序细节。
- 进行必要的结对走查。

这些任务成为了 WebApp 第四个增量的增量进度表的一部分，并且被分配到增量开发时间线之上，可以被输入到进度安排软件（例如，Microsoft Project）之中，并且用于跟踪和控制。

5.5.3 如何估算工作量和时间

对于绝大多数的 WebE 项目来说，估算的重点是宏观而不是微观的问题。WebE 团队评估一个计划好的 WebApp 增量是否可以在定义的进度表限制内，使用可用的资源开发完成。通过将每个增量的内容和功能作为一个整体进行考虑来完成这一评估。事实上，WebE 团队成员询问，“就我们当前对增量的理解，已经识别出的风险和为这个工作所定义的任务列表，3 个人工作 5 周是否就可以部署 SafeHomeAssured. com 的第四个增量?”

如果团队回答“是”，那么毫无争议并且不要有任何犹豫，不需要进一步的估算活动。相反，如果团队对于一个或更多利益相关者强加于他们的交付日期感到忧虑，那么可以做两件事：1）说出你的担忧，但是无论如何还要继续前进；2）做少量的详细估算来帮助你和利益相关者更好地理解需要的资源和时间。[⊖]

有两种切实可行（并且快速）的方法用于对 WebApp 进行详细估算。第一种，*基于使用场景的估算*，评审为将要创建的增量而定义的使用场景（如用例）。评估团队过去的历史，你确立一个值 E_{avg}，它表示部署一个使用场景所需要的平均工作量（以人·天为单位）。要对增量进行估算，统计出使用场景的个数并乘以 E_{avg}。可以根据对使用场景复杂性的理解对这个数字进行调整。工作量一旦确定，就可以随着时间线分配给 WebE 动作和任务。最后，这些估算可以用于评估增量部署日期的合理性。图 5-4 说明了把这种方法用于 SafeHomeAssured. com 的第四个增量，其中，团队过去的历史为 E_{avg} = 14 人·天。将要实现的两个使用场景的复杂性都比平均值大得多，它们的乘数分别是 2.5 和 2.0。因此，实现第四个增量所需要的总工作量被估算为 63 人·天，它们将在 5 周的交付期间内进行分配。

使用场景	平均工作量	复杂性	工作量
为将要被监控的空间开发布局	14	2.5	35
为我的空间的传感器布局获得推荐	14	2.0	28
总计			63

图 5-4 基于使用场景的估算

⊖ 在这个阶段，团队通过使用图形功能逐步绘制墙壁、窗户和门来预想空间的构建。墙壁线将“接在”控制点上。墙壁的尺寸将自动显示。窗户和门将进行图形化的布置。最终用户也可以选择特定的传感器、摄像机等，并且一旦空间定义完毕就对它们进行布置。

⊖ 当然，你希望得到的是对交付日期的延期。

现在到了需要小心的时候了。没有经验的团队（和团队负责人）会认为，在人员、工作量和进度之间存在线性关系，也就是说，如果建模、构建和部署一个增量需要 63 人·天，那么工作就可以按如下的方式完成：一个人工作 63 天，两个人工作 32 天，三个人工作 21 天，依此类推。遗憾的是，并不存在这样一种线性关系。

事实上，一个项目涉及的人员越多，花在沟通和协调（开会、E-mail 等）上面的工作量就越多。更多的时间将花在与项目无关的事情（例如，打电话，协调适度的等级）、管理问题（例如，申请一个新的安全保险计划）和其他的“非生产性的”工作上。其结果是，3 个人可能会需要花费 25 人·天或者更多的日历天来完成 63 人·天的工作量。

第二种估计方法使用一个*产品－过程表*。即，所有主要的 WebE 动作都列在表的第 1 列。增量所有主要的内容对象和功能都列在第 1 行。团队成员对每一个内容对象和功能估算进行 WebE 动作所需的工作量（以人·天为单位）。图 5-5 为这种估算方法用于 SafeHomeAssured. com 的第四个增量。这个工作量似乎比基于场景的估计要更乐观一些。必须要对这两种估算进行一致化来提供一个单一估计。增量 4 相对复杂的内容和功能需要比前面的增量多得多的 WebE 工作量。

内容和功能	分析	设计	编码	测试	交付	反馈	总计
墙壁、门道、窗户	1	2	2	2	0.5	0.25	7.75
传感器	0.5	1.5	1	1	0.25	0.25	4.5
详细说明并绘制墙壁、门道、窗户	1.25	3	3	3	1	0.25	11.5
计算房间大小	0.5	1	2	1	0.5	0.25	5.25
保持/检索命名空间	0	1	1	0.5	0.5	0.25	3.25
更新/删除命名空间	0	1	1	0.5	0.5	0.25	3.25
打印命名空间	0	1	1	0.5	0.5	0.25	3.25
推荐安全硬件	0.5	3	2	2	0.5	0.25	8.25
详细说明安全硬件	0.5	2	2	3	0.5	0.25	8.25
总计	4.25	15.5	15	13.5	4.75	2.25	55.25

图 5-5　用于估算的产品－过程表

5.5.4　如何描述任务之间的相互依赖关系

有些 WebE 任务或动作只有在另一个任务或动作所生产的工作产品可用的时候才可以开始。而其他的任务或动作可以独立发生。因此，当单个 WebE 任务之间存在顺序上的相互依赖时，我们不应当感到奇怪。

当有多个人参与一个 WebE 项目的时候，多个开发动作和任务很可能并行进行。当这种情况发生时，必须协调多个并发任务，以确保其能够在要完成的后续任务需要其工作产品之前完成。

任务网络（task network），也称为*活动网络*（activity network），是一个项目任务流程的图形表示。有时将任务网络作为在自动项目进度安排工具中输入任务顺序和依赖关系的机制。最简单的任务网络格式（用于创建一个宏观进度表时）只描述了主要的 Web 工程任务。图 5-6 所示为 SafeHomeAssured. com 第四个增量的任务网络示意图。

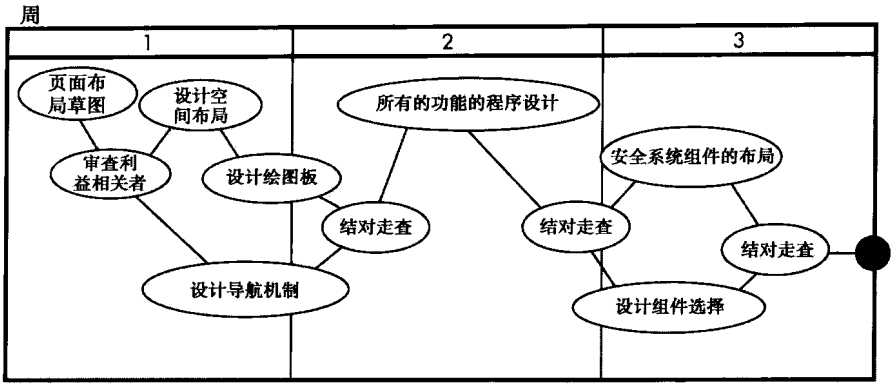


图5-6 第四个增量的一个任务网络

WebE动作和任务的并发性导致了在进度安排上的很多重要需求。因为并行任务是异步发生的，所以团队负责人必须确定任务间的依赖关系，以确保增量朝着完成的方向持续发展。另外，团队负责人必须注意那些处在关键路径上的任务。也就是说，要确保整个增量如期完成，就必须保证这些任务能够如期完成。

5.6 管理质量

建模、构建和部署WebApp的一个增量跨越相当短的时间。增量一旦发布给最终用户，它就被使用，同时WebE团队得到反馈。反馈提供了一个相当好的描述，指明哪些可以工作而哪些不能。

因此，很容易把对WebApp质量的任何考虑都推迟到一个增量的部署之后，在开始使用期间发现问题时进行解决。当然，这种方法有一些缺陷：

1. 对最终用户发现的质量问题进行改正的费用，几乎总是比这些问题在WebE过程流的早期被发现并进行改正花费得要多得多。在WebApp构建之前修正一个设计错误的花费比试图改正一个部署了的WebApp的花费（耗费的工作量）要少很多倍。

2. 最终用户发现的每一个问题都需要返工，而返工会占用那些应当用于下一个增量的资源（人员和工作量）。总体的项目就会落后于进度表。

3. 早期用户对增量的使用可能并不完全，有些内容和功能并未被触及到。在这些未被触及的地方可能存在质量问题。它们将在后面（如果你相信墨菲定律）最糟糕的时侯被发现。

因此，对于每个WebE团队来说，应该在构建增量时关注WebApp的质量。

5.6.1 团队可以使用哪些质量保证机制

虽然质量问题可以产生于各种各样的来源，但是它们的来源通常可以追溯到对最终用户需求的错误理解或者实现。因此，Web工程的第一个质量保证机制是一个考虑周全的、彻底的沟通活动。在第4章中，我们注意到沟通从WebApp的利益相关者那里导出需求。如果在进行需求获取的时候很审慎，那么出错的可能性就会小一些，不一致性或者遗漏就不会被传递到后面的框架活动中。结对走查（将在下面进行讨论）作为沟通的一部分可以用于评估开发的所有工作产品的质量。在这种情况下，走查的参与者包括一个Web工程师和一个或多个利益相关者。

一旦内容、功能、约束和性能的需求向前传递给了建模活动，那么分析和设计任务就开始了。在各个方面，分析和设计模型的质量都必须进行评估。在最简单的情况下，Web团队可以创建一个通用检查列表，可以用于评估模型。对于更复杂或者关键的模型，可以为创建的每个模型进行结对走查。

一旦 WebApp 增量的代码产生了，就可以启动系统的一连串测试来评估增量的质量。必须要注意，测试必须使用所有的用户需求，同时要评审增量的技术方面（例如，架构的完整性、功能组件或者内容对象的正确性，以及所有链接和导航机制的有效性）。

所有质量保证机制的结果都应当进行记录。这将在必须做出变更时为团队提供帮助，并有助于团队去改正可能导致一类错误的过程缺陷。

5.6.2　什么是结对走查方法

在本章前面，我们提到了结对走查是一种保证 WebE 活动所产生工作产品（例如，分析和设计模型、HTML 或者 XML 代码、各种脚本以及内容和功能）质量的有效而敏捷的机制。

作为结对走查方法的一个例子，我们考虑 SafeHomeAssured. com WebApp 第四个增量的分析模型的一部分。这个增量实现将要被监控的空间的布局以及为空间推荐传感器。

WebE 团队的一个成员，和营销部门的一个特定利益相关者一起工作，[⊖]产生出分析模型的一部分（本例中，是实现空间布局用户界面的图表化描述）。在开发界面模型的过程中，Web 工程师和一个博学的利益相关者（一个营销人员，可以充当最终用户的角色）一起工作，进行不断的结对走查。Web 工程师和利益相关者都会在界面模型演化的过程中对其进行审查，并提出类似下面的问题：

- 关于页面布局有没有什么问题？
- 菜单条目是否有意义、完整和直观？
- 空间布局方案易用吗？直观吗？
- 你能想象你是如何导航到这个页面并且从它导航到其他的功能和内容吗？
- 在这个页面中，我们遗漏了任何关键的功能或内容吗？
- 你对这个模型的假设是什么？

随着模型的设计，这些以及其他的一些问题会被重复提出。结果是对模型的不断修正和修改。

5.6.3　什么是团队走查方法

在有些情况下，整个 WebE 团队将对一件工作产品进行评审。这种情况发生在工作产品可能对整个 WebApp 产生广泛影响的时候，因此每个人都必须理解问题并对于评审有所贡献。

两个团队成员使用“结对编程”的方法来开发 WebApp 的某些方面。一旦他们相信工作产品的草稿阶段已经完成，他们（称为工作产品的*生产者*）就会要求团队的其他成员参与进行*团队走查*。生产者把已经产生的所有信息（如果可能，以打印稿的形式，或者是电子形式）提供给其他的评审者。评审者承诺至少花 30 ~ 45 分钟时间来评审工作产品，并列出任何的事项、问题或看法。

在 24 小时之内（最好更快）开始团队走查。工作产品生产者的工作开始于对工作产品进行的“走查”，解释它所代表的含义以及阅读者可能如何理解所显示的东西。这样，评审者提问（通常基于评审之前做的说明）并指出潜在的问题域。生产者记下所有这些，但不急着设法立即解决它们。在走查进行的过程中，参与者要遵循下面的这些准则：

1. **评审产品，而非生产者**。如果进行得当，那么团队走查将给所有的参与者留下温暖的成就感。如果进行不当，那么走查将呈现出审讯的气氛。应当和善地指出错误；走查的氛围应当是宽松而有建设性的；目的不是让人尴尬或轻视别人，而是去帮助。

2. **制定并遵守一个议事日程**。各种类型会议的一个主要弊端是容易跑题。走查应当保持在

⊖ 记住，非技术的利益相关者在敏捷 WebE 团队中也同样要看成是积极成员。

主题上，并且按进度表进行。

3. 限制争论和辩驳。当评审者提出问题的时候，对问题的影响未必达成共识。与其花时间来讨论这个问题，不如把这个问题记录下来，放在会后解决。

4. 阐明问题域，但是不要试图解决提出的每一个问题。走查并不是一个解决问题的会议。

5. 做书面笔记。可以把笔记直接录入笔记本电脑。

6. 花费足够的时间来发现质量问题，但是不要过多。一般而言，团队走查最多应当在 60 ~ 90 分钟完成。

作为需要团队走查的例子，整个 WebE 团队可能决定评审设计模型的各个方面，以确保在 SafeHomeAssured. com 的第四个增量的构建活动开始之前的代码生成开始之前发现错误、不一致和遗漏。

5.6.4 WebApp 存在质量标准吗

如果人们看着 WebE 工作产品而不知道去找什么，那么即使是最好的走查也不会有效果。下面列出了用于 WebApp 的提供了质量标准和准则的全面的在线资源集合［Qui01］。

信息



WebApp 的质量标准和准则

W3C：Style Guide for Online Hypertext（www. w3. org/Provider/Style）

The Soloid Guide to Web Design（www. sev. com. au/webzone/design/guide. asp）

Web Pages That Suck（www. webpagesthatsuck. com/index. html）

Resources on Web Style（www. westegg. com/unmaintained/badpages）

Gartner's Web Evaluation Tool（www. gartner. com/ebusiness/website-ings）

IBM Corp：Web Guidelines（www-3. ibm. com/ibm/easy/eou_ext. nsf/Publish/572）

World Wide Web Usability（http：//ijhcs. open. ac. uk）

Interface Hall of Shame（www. iarchitect. com/mshame. htm）

Art and the Zen of Web Sites（www. tlc-systems. com/webtips. shtml）

Designing for the Web：Empirical Studies（www. microsoft. com/usability/webconf. htm）

Nielsen's useit. com（www. useit. com）

Quality of Experience（www. qualityofexperience. org）

SAP Design Guide（www. sapdesignguide. org）

Creating Killer Web Sites（www. killersites. com/core. html）

All Things at Web（www. pantos. org/atw）

SUN's New Web Design（www. sun. com/980113/sunonnet）

Tognazzini，Bruce：Homepage（www. asktog. com）

Webmonkey（http：//hotwired. lycos. com/webmonkey/design/? tw = design）

World's Best Websites（www. worldbestwebsites. com）

Yale University：Yale Web Style Guide（http：//info. med. yale. edu/caim/manual）

5.7 管理变更

因为增量的过程流用于 Web 工程，所以如果你和其他的利益相关者有纪律和耐心，那么管理变更就相对简单。因为开发一个增量的时间很短，所以通常很可能把变更请求推迟到下一个

增量，从而，对必须匆忙实现的项目而言，降低了与变更相关的破坏性影响。

然而，这种策略暗含了在我们完成每个增量的方法上会有细微的变化。第 $n+1$ 个 WebApp 增量不仅仅是完成了和增量相关的内容和功能，同时它还可能包含对第 n 个增量请求的内容和增量的变更（被推迟到第 n 个增量部署之后的变更）。当然，只有当所有的这些额外变更都不是“项目终结者”，也就是说，交付的增量（不包含下一个增量包含的变更）仍然是可接受的解决方案时，它才是合理的。

5.7.1　如何评估变更的重要性和影响

为了评估任何请求的变化的重要性和影响，每一个变更都应当被归为下面 4 种类型中的一类：

第 1 类。修正较小的错误或改进局部内容或功能的内容或功能变更。

第 2 类。对增量中的其他内容对象或功能组件有影响的内容或功能变更。

第 3 类。对 WebApp 有广泛影响（例如，功能的主要扩展、内容的显著加强或减少、主要请求的导航变化）的内容或功能变更。

第 4 类。对一类或多类最终用户引人注目的主要设计变化（例如，界面设计或导航方法方面的变化）。

一旦请求的变更分类完毕，就可以按照图 5-7 所示的算法来进行评估。

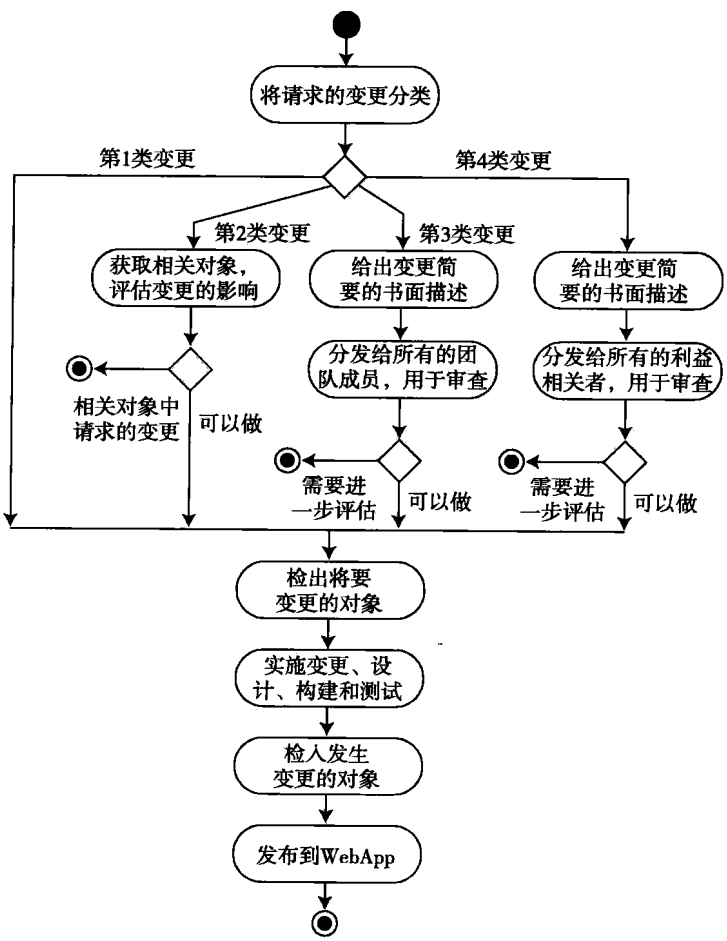


图 5-7　管理 WebApp 的变更

在图5-7中，第1类和第2类变更的处理是非正式的，并且按敏捷的方式进行处理。对第1类变更，由一个Web工程师评估变更的影响，但不需要外部的评审或者记录。随着变更的产生，由配置中心存储库工具（详见第16章）执行标准的检入（check-in）和检出（check-out）过程。对第2类变更，评审变更对相关对象的影响是Web工程师（或者要求其他负责这些对象的开发者来做）义不容辞的职责。如果变更不会引起对其他对象的明显变更，那么就不需要另外的评审或文档修改。如果需要做较大的变更，那么就需要进一步的评估和计划。

第3类和第4类变更也以敏捷的方式进行处理，但是需要一些描述文档和更正式的评审步骤。为第3类变化开发一个变更描述（change description）——描述变更并提供变更所产生影响的简洁评估。变更描述被分发给评审变更的Web工程团队的所有成员（包括其他有兴趣的利益相关者），由这些成员对其进行评审以更好地评估其影响。也为第4类变更进行变更描述，但是在这种情况下，应当由所有的利益相关者来进行评审。

5.7.2 何时推迟做出变更

就像在本节的介绍中提到的，变更常常扰乱工作并延误WebApp增量的部署。原因很明显——工作于一个增量相关内容和功能的人员必须停下他们手头的工作来解决变更，这要消耗时间。

为了避免增量工作进度表受到扰乱，和增量n相关的变更通常应当在增量n开始被部署以后再进行解决。也就是说，变更被正在开发中的增量“封条”。当然，请求的变更必须被加以评估并合并到增量中，只不过必须要在增量第一次交付之后。

这时，你也许会想：“我的利益相关者不会忍受这种方法。他们请求变更并希望立即实现。”

我们已经遇到了这样的情况，但是那并不意味着“现在就做出变更！”的文化不能被改变。为了改变这种文化，利益相关者必须明白变更并不是免费的——它们要耗费资源和时间。因此，利益相关者总是面临一个折中——可以打断工作并做出变更，但是交付日期将推迟，同时成本肯定会增加。在大多数情况下，最好把变更推迟到破坏性最小（由此，消耗的成本和时间最少）的时候才实现。

然而，在某些情况下，必须立即做出变更。如果：

1. 推迟变更比立刻做出变更将导致更多的工作。
2. 没有这个变更，对于最终用户来说，增量的可使用性将大大降低。
3. 如果没有立即做出变更，将会有相当大的资金损失。
4. 规章制度方面的需求需要部署包含这个变更。

如果满足这些标准中一个或多个，那么你就必须在当前增量中做出变更。



变更请求

场景：当工作进行到增量2（详细的产品信息和下载）时的Web工程区域

参与者：WebE团队的两个成员和团队负责人

会话：

团队负责人：我刚刚接到了营销部门打来的电话。他们想让我们做出一个变更，但是我认为这不是一个什么大问题。

团队成员1：但是我认为，按照规则，一旦我们开始一个增量，就不应当有变更。

团队负责人：没错，但是我们仍然需要快速地进行评估，以决定我们将如何处理它。

团队成员 2（做了个鬼脸）：现在？我们正在……半中间。

团队负责人：我知道，我知道，但是这不会超过 15 分钟。

［两个团队成员都耸了耸肩，但是他们看起来并不高兴。］

团队负责人：他们想让我们在产品信息内容中增加一类新的内容对象。他们有一个描述如何安装各种监控设备的视频库，希望用户可以使用它。

团队成员 1：所以除了说明性的文本、图片、产品技术说明等之外，他们还想要一个视频？

团队负责人：嗯哼。

团队成员 2：每一个监控设备都有一个视频吗？

团队负责人：我不确定，不过我会打电话问他们。

团队成员 2：我们需要知道这些文件的大小、视频的格式，并且当我考虑得更多时很可能出现的更多问题。

团队成员 1：我猜他们现在就想要。

团队负责人：是的。

团队成员 2：在我看来像是第 1 类或第 2 类的变更请求。我们将必须对我们的导航设计进行一点儿修改来找到视频文件，开发合适的视频显示功能，并且修改其他产品信息对象的内部链接。它将带来一些工作，但不会太多。

团队负责人："不会太多"是多少？

团队成员 1：给我们剩下的"15 分钟"，我们将告诉你。然后？

团队负责人：好吧，我们可以把它推迟到下一个增量，但是我倾向于现在就做这个修改。我怀疑如果我们推迟它，那么实际上将带来更多的工作。我们还没有深入到增量 2 中，所以……

团队成员 1：我们估计一下需要多少工作量，然后再做决定。

团队负责人：很好！当你对工作量有一个更清楚的估计的时候，顺便走访一下我的工作台。

5.7.3　所有的相关工作产品都应当做出变更吗

在 WebE 过程中生产的工作产品，有下面两个目的之一它们可以是，1）在一个 WebE 活动中获取的信息的非正式记录（例如，在沟通活动中获取的内容和功能需求列表）；2）WebApp 的表示（模型），用于指导后续的 WebE 活动（例如，一个界面设计模型将对构建活动中界面的编码进行指导）。

作为一个普遍的经验法则，归为第 1 类的工作产品在请求并做出变更时，不需要进行更新。而第 2 类工作产品应当被加以修改来反映任何的变更，这样它们就可以在后续的 WebE 活动中保持可用。

然而，很重要的是，要注意对一个 WebE 团队来说，保持对所有第 2 类工作产品（例如，设计模型）的更新可能是一个累赘。那么 WebE 团队在做出变更时，如何选择那些应当被更新的工作产品呢？

一个实用的方法就是回答问题："设计的哪些方面可以更好地表现为一个可执行的实现？"在某些情况下，WebApp 某方面的唯一永久性描述就是 WebApp 本身。例如，一旦用 HTML 实现了界面设计模型，就没有理由去维护界面的实际设计模型。界面的变更将由实现而不是模型进行指导。因此，模型可以被存档（出于历史的原因），但是不需要更新。另一方面，一个复杂功能的设计模型可能在功能实现并部署之后仍然有用。原因在于模型展示较高层次的抽象，它有助于起初没有参与实现的人员理解功能。这种情况下，当对功能做出变更时，功能模型应当被

更新。

5.8 跟踪项目

对于一个 WebE 项目而言，一个 WebApp 增量的按时部署常常是全局进度的主要度量。但是在增量可用之前，Web 工程师必然会遇到这样的问题，“我们现在在哪儿?”

为了给出这个问题的答案，你需要在一个 WebApp 增量的开发过程中跟踪进度。但是如何进行呢？对于小型和中型的 Web 工程项目而言，一个增量可能只需要 2 或 3 周的时间进行开发。至多，非正式地定义一些中间的里程碑，而且项目进度表也许并没有定义成有助于跟踪的粒度。

5.8.1 进度问题有什么宏观指标吗

在宏观层次上，有一些指标标示一个增量（或者整个项目）正处于麻烦之中。John Reel［Ree99］提出了 10 个指标，标示了一个信息系统项目正处于危险之中。我们把它们适应到 WebApp 之上：

1. WebE 团队没有理解其客户的需求。
2. WebApp 的范围定义不当。
3. 变更的管理不当。
4. 所选择的技术发生了变化。
5. 业务需求似乎改变（或定义不当）。
6. 时间表不现实。
7. 用户对这个 WebApp 并不真正感兴趣。
8. 失去了赞助（或者从来就没有真正得到过）。
9. WebE 团队缺乏有合适技术的人员。
10. 实践者避免最佳实践和教训。

在本章和前面几章中，我们已经讨论了可以帮助消除这些问题的一个 WebE 过程。认真采用这个过程，你将避免一些疲惫不堪的行业专家所谓的“90-90 规则”。即系统前面的 90% 消耗分配的工作量和时间的 90%。剩下的 10% 消耗分配的工作量和时间的另外 90% ［Zah94］。导致 90-90 规则的根源就包含在前面列表提到的宏观指标之中。

5.8.2 可以使用哪些标准来跟踪进度

在 WebApp 增量的开发过程中，追踪进度的一种方法就是调查 WebE 团队，以确定哪些框架内的活动已经完成。然而，这种方法可能是不可靠的，因为只有当适当的[⊖]工作产品开发完毕时，一个框架活动的完成才能算是进度指标。

另一种方法是确定已经实现了多少用户场景，还剩下多少用户场景需要实现。这提供了项目增量相对完成程度的粗略指示。

如果 WebE 团队已经花时间为增量构建了详细的工作进度表，那么对进度的跟踪就可以按照确定完成了多少工作任务、生产并评审了多少工作产品和单个的团队成员对增量完成日期有多少信心来进行。

⊖ 在这个上下文中，一个适当的工作产品就是一个正确且为后续 WebE 工作建立了坚实基础的产品。

5.9 外包 WebE 工作

绝大多数的 WebApp 都会外包[⊖]给那些专门从事基于 Web 的系统和应用开发的供应商。在这种情况下，企业（客户）从两个或更多的供应商那里询问 WebApp 开发的固定报价，评估这些竞争的报价，然后选择一个供应商来完成这项工作。但是签约组织要寻求什么？如何确定一个 WebApp 供应商的能力呢？如何知道一个报价是否合理呢？当组织（及其外包签约方）开始着手主要的 Web 工程开发工作时，期望在何种程度上进行计划、进度安排和风险评估呢？

为了回答这些问题，下面几部分将介绍一组指导原则。这些并非试图成为生产低成本、按时交付 WebApp 的万无一失的秘笈。然而，它们将有助于签约组织和供应商平稳地启动工作，并且使误解最少。

5.9.1 如何启动外包项目

如果 WebApp 开发选择了外包策略，那么一个组织在寻求外包供应商来完成此任务之前，应当完成下面的一系列任务：

1. **沟通应当在内部进行**。识别 WebApp 的用户，列出可能对该 WebApp 感兴趣的内部利益相关者，定义并评估 WebApp 的总体目标，刻画该 WebApp 将要提供的信息和服务，记录与其竞争的 Web 站点，定义对一个成功的 WebApp 进行定性和定量度量。这些信息应当记录在提供给外包供应商的产品规格说明中。

2. **WebApp 概要设计应当在内部进行**。很明显，专业的 Web 开发者会创建完整的设计，但是如果外包供应商已经确定了该 WebApp 的一个总体观感（在项目的初始阶段总是要被修改），则会节省时间和成本。设计应当包括 WebApp 将要展示的内容的类型和容量，以及将要完成的交互处理的类型（例如，表单、订单条目）。这些信息应当被添加到产品规格说明中。

3. **应当开发一个粗略的项目进度表，不仅包括最终的交付日期，还应该包括里程碑的日期**。在 WebApp 的演化过程中，里程碑应当附属于其可交付的增量。

4. **应当为内部组织和外包供应商创建一个职责列表**。实际上，这个任务就是指明了外包供应商将从内部组织那里所需要的东西（信息、联系和其他资源）。

5. **应当明确签约组织和供应商之间的监督和交互程度**。这应当包括明确供应商联络人的任命和联络人职责和权利，定义开发过程中质量评审点，以及明确供应商在组织之间的沟通方面的职责。

在这些步骤中取得的所有信息都应当组织在报价要求书中，并传递给候选的供应商。[⊜]

5.9.2 如何选择候选外包供应商

近几年，出现了数千家 Web 设计公司来帮助商业机构建立 Web 站点和从事电子商务。其中，很多已经在 WebE 过程方面非常熟练，但是，也有很多公司和新手没有多大区别。为了选择候选的 Web 开发者，你必须做一些该做的事情。你应当：1）拜访过去的客户以确定 Web 供应商的专业精神，满足保证进度和成本的能力，以及有效沟通的能力；2）确定在过去成功的项目中供应商的首席 Web 工程师的名字（然后，确保这个人通过合同的约束而有义务参与项目中）；3）仔细评审供应商完成的工作样品，该样品和将要签约的 WebApp 在观感（和业务领域）方面应

⊖ 虽然很难找到可靠的行业数据，但是我们可以肯定地说，WebApp 外包的百分比要比传统软件外包高很多。当然，随着 Web 开发技能逐渐成为主流，它将会稍微减少，但是它也很可能一直很高。

⊜ 如果 WebApp 的开发工作由内部组织完成，那么就不会有任何改动！项目基本上是按原来的方式进行启动的。

该具有相似性。即使在提供报价申请之前，面对面的会见也能使签约方和供应商更好地了解彼此之间的“符合”程度。

5.9.3 如何评估报价的合理性和估算的可靠性

因为几乎不存在相关的历史数据，并且，众所周知，WebApp 的范围是易变的，所以估算本质上就有风险。为此，某些供应商把大量保险额度加入到项目的成本报价中。这是可以理解的，也是合适的。问题不是“我们的付出是否得到了最好的回报?”而应该是：

- WebApp 的报价成本是否能提供直接或间接的投资回报，从而证明项目是否合算?
- 已经提供报价的供应商是否具有我们所需的专业精神和经验?

如果对这些问题的回答都是肯定的，那么报价就是值得考虑的。

5.9.4 需要什么程度的项目管理

与项目管理任务（由供应商和签约组织共同完成）相关的手续直接与 WebApp 的规模、成本和复杂度成正比。对于大型而复杂的项目，应当制定详细的项目进度表，包括定义工作任务、软件质量保证检查点、工程工作产品、客户评审点和主要的里程碑。供应商和签约方应该共同评估风险，并为那些被视为重要的风险制定减轻、监控和管理计划。应该明确定义质量保证和变更控制的机制。应该建立签约方和供应商之间高效沟通的方法。

5.9.5 如何评估进度并管理范围

WebApp 开发进度跨越的时间段较短，因此，开发进度表应该是细粒度的。也就是说，工作任务和较小的里程碑应该按日来安排。这种细粒度的安排使得签约组织和供应商能够在最终完成日期可以延后之前认识到进度的延误。

因为在 WebApp 项目进行过程中，其范围很可能会发生变更，所以 WebE 过程模型应该是可适应的和增量的。这种情况允许供应商的开发团队“冻结”某个增量的范围，使其可以创建一个可运行的 WebApp 发布版本。下一个增量可能会执行前面增量的评审中所建议的范围变化，但是，一旦第二个增量开始，范围就将再一次被暂时冻结。这种方法使得 WebApp 团队可以在不必顺应一个持续变更流的情况下正常地进行工作，但是，我们仍然要认识到大多数 WebApp 的持续演化特征。

5.10 本章小结

Web 工程的计划活动开始于对项目范围的考虑，带来对业务背景、信息对象、WebApp 功能、系统约束和性能问题的理解。这些信息源于沟通活动产生的工作产品。

项目的总体计划为所有 WebApp 增量的交付提供了一个路线图，但是计划活动本身关注于为一个特定的增量开发的任务和工作产品，同时制定在增量的开发过程中应用的风险、质量和变更管理机制。

计划由 WebE 团队的所有成员执行，由团队负责人进行协调。好的 WebE 团队将明确和避免当一组人员一起工作的时候将会带来问题的有毒特性。

一旦计划完成，技术工作就从建模活动开始。在后面几章中，我们将研究用到 WebApp 的建模的很多不同方面。

参考文献

[Bec99] Beck, K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999.

[Bro95] Brooks, M., *The Mythical Man-Month*, Anniversary Edition, Addison-Wesley, 1995.

[Coc01] Cockburn, A., and J. Highsmith, "Agile Software Development: The People Factor," *IEEE Computer*, vol. 34, no. 11, November 2001, pp. 131-133.

[Dem98] DeMarco, T., and T. Lister, *Peopleware*, 2nd ed., Dorset House, 1998.

[Jac98] Jackman, M., "Homeopathic Remedies for Team Toxicity," *IEEE Software*, July 1998, pp. 43-45.

[Jef01] Jeffries, R., et al., *Extreme Programming Installed*, Addison-Wesley, 2001.

[Kid00] Kidder, T., *The Soul of a New Machine*, Back Bay Books (reprint edition), 2000.

[Pic01] Pickering, C., "Building an Effective E-Project Team," *E-Project Management Advisory Service*, Cutter Consortium, vol. 2, no. 1, 2001, www. cutter. com/content/project/fulltext/summaries/2001/01/index. html (accessed July 24, 2007).

[Qui01] Quibeldey-Cirkel, K., "Checklist for Web Site Quality Assurance," Quality Week Europe, 2001.

[Ree99] Reel, J. S., "Critical Success Factors in Software Projects," *IEEE Software*, May, 1999, pp. 18-23.

[Ros07] Rosenberg, S., *Dreaming in Code*, Crown Publishers, 2007.

[Sch02] Schwaber, K., "Agile Processes and Self- Organization," Agile Alliance, 2002, www. agilealliance. org/system/article/file/784/file. pdf (accessed July 24, 2007).

[Wei86] Weinberg, G., *On Becoming a Technical Leader*, Dorset House, 1986.

[Wil99] Williams, L., and R. Kessler, "All I Really Need to Know about Pair Programming I Learned in Kindergarten," University of Utah, 1999, http: //collaboration. csc. ncsu. edu/laurie/Papers/Kindergarten. PDF (accessed July 24, 2007).

[Zah94] Zahniser, R., "Timeboxing for Top Team Performance," *Software Development*, March 1994, pp. 35-38.

第6章 建模活动

书面或口头语言是很出色和有表达力的沟通手段，但是它对于一个 WebApp 将要交付的内容和功能来说，却并不是一种准确的表达方式。自然语言可能有歧义、有矛盾或者不清晰。考虑如下需求："全部的产品信息应当只对注册用户是可用的。"除了"全部的产品信息"的含义缺乏准确描述之外，用户何时以何种方式被系统识别为"注册用户"也不清楚，甚至"可用的"的含义也是模糊的。未注册用户应当知道有其他信息存在，只是他们无法看到？还是他们应当不知道它的存在？

建模通过综合使用文本、图像和图表格式来描述内容和功能、架构和组件细节、接口、导航和美学方式解决上述问题。使用这些方法相对来说是易于理解的，更重要的是，对正确性、完整性和一致性的审查是直接的。

第3章简要地讨论了建模，指出了建模是一个活动，创建将要构建的 WebApp 的某些方面的一个或更多的概念表示。如图 6-1 所示，在 WebE 过程流中，建模包括分析和设计两个主要动作。在我们聚焦于这些动作并考虑和它们相关的"概念表示"之前，我们先进一步研究建模的概念。

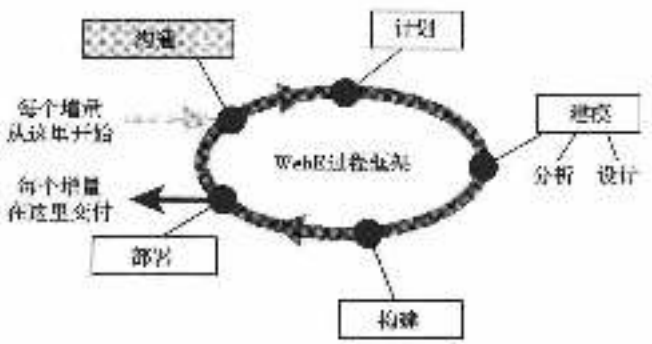


图 6-1 WebE 过程流中的建模

6.1 建模的概念

我们把对现实的认知进行建模以理解和改变它，但是我们对现实所构建的模型并不完美。著名的工业统计学家 George Box 曾经提到："所有的模型都是错误的，但是有一些模型是有用的。"意思是说，没有任何模型可以捕获被模型化的现实中几乎无限的所有细节。（否则的话它就是现实本身！）然而，当你创建一个模型的时候，你的目的是捕获现实中对你来说是有用的和有趣的那些部分。因此，一个好的模型只在与你不相干的那些方面是"错误的"（或者缺乏缺乏现实的细节）。模型的错误和你的目的无关。[1]其实，你可以通过把现实的一部分从你的模型中去掉，对其进行简化并关注那些对你来说重要的事情和你将要执行的 WebE 工作。

6.1.1 如何评判一个模型的有用性

我们评判一个模型有用性的唯一方法是理解模型的目的。模型试图帮助你理解什么？然而，

---

[1] 然而，重要的是，要注意一个 WebApp 模型可能是彻底失败的，如果我们遗漏重要的细节，展示有一致的表示，不听从利益相关者，或者犯很多其他错误。我们就会创建一个这样的模型：在最好的情况下，它对任何人来说是无用的；在最坏的情况下，它具有危险的。

值得注意的是，需要理解的重要事情在不同的时间以及对于不同的人员来说是不同的。

我们考虑 SafeHomeAssured.com 的一些利益相关者。对 CPI 的营销经理来说，重要的是理解用户对一个特殊产品可能的响应方式。产品开发的高级经理可能更关心理解所需功能和不同的安全或传感器技术的潜力和能力之间的关联。CPI 的工程经理想理解 WebApp 内容和功能所预测的复杂性。

在每一种情况下，感兴趣的现实部分都是不同的。根据 George Box 的意见，将需要使用不同的模型。如果建模可以以线性的方式进行（与迭代的方式相反），我们可以以一种良好的顺序方式很干净地从一个现实视图（一个模型）转移到下一个视图。

如同在前面所讨论的一样，问题是，Web 工程本身并不适合使用线性方法。你需要敏捷性来处理不断变化的需求、不断变化的技术、不断变化的人员——变化的现实。在这种环境下，这些模型之间的关系就变得重要得多。当我们共享模型内包含的信息时，利益相关者就能更好地理解其他利益相关者的观点。



关于建模的讨论

场景：建模开始之前，Web 工程区域的会议室

参与者：SafeHomeAssured.com 的 WebE 团队成员

会话：

团队负责人：我听到了关于建模的一些抱怨。我们中间至少有一个人想直接进行编码和 Web 页面的创建。

团队成员 2：那应该就是我。

团队成员 1（稍稍有些愤怒）：为什么我不感觉惊呀呢？你从一开始就反对 WebE 过程。

团队负责人（看着 WebE 团队成员 2）：我对你的态度所不理解的是，这个过程看起来是适合这个 WebApp 的。我们进行了相当好的需求收集工作，我们的计划似乎也合理。到目前为止，这个过程是适用的。

团队成员 2：但是建模肯定会使我们显著放慢。我们不是来创建不必要的模型的；我们是要创建 SafeHomeAssured.com 的。

团队成员 3：难道过有关创建不必要的模型的事情吗？

团队成员 1：我们不是为了创建模型而创建模型，而是在事物不清楚或者很复杂的时候来创建模型。

团队成员 2：但是我们如何确保模型是正确的呢？

团队负责人：这就是为什么我们要为创建的每个模型使用结对编程和结对走查的原因。

团队成员 2：那仍然不能确保它们是正确的。

团队成员 1：你没有抓住重点。建模的目的不是去创建一个完美的模型，而是为我们提供对 WebApp 某些方面更好的理解，将使我们在更短的时间内构建出错误更少的 WebApp。

团队成员 2：理论上。

团队成员 3（笑）：看起来我们并不能说服你，但是因为除了你我们都同意，所以你也要到这边来。如果你是正确的，我将第一个表示赞同。

团队负责人：看吧，只有当我们真正需要的时候才去建模，但是当它是必要的时候，我们也进行建模。同意吗？

[大家都点点头，但是热情度不同。]

6.1.2 模型可以用于理解业务限制吗

业务活动及其支持技术已经变得密不可分了。这种便利为业务活动提供了强有力的支持，但是它也意味着组织越来越多地受到技术的固有局限性所带来的限制。

正如在前面所讨论的，一个构建合适的模型可以让 Web 工程师理解业务和用户需求以及试

图实现它们的设计。但是在快速演化的业务环境中，有效的模型同等重要，它可以让开发者和业务分析员看到一个 WebApp 的架构或支持技术如何对业务组织进行限制。

作为这种“逆向建模”影响的一个例子，考虑 SafeHomeAssured. com WebApp。第四个增量的界面设计（已在第4、5章中介绍）模型产生了一个设计，它可以让最终用户开发 SafeHome 安全硬件的布局，以展示将要被保护或者监控的物理空间。这种技术方法对最终用户提出要求（必须愿意花时间来描述空间），因此也限制 CPI 把基于 Web 的监控仅仅出售给那些愿意做空间布局的人。这很有可能损失很大比例的潜在市场。CPI 的营销部门在对界面设计模型进行审查之后，立即认识到了这个隐式限制，并且开始讨论如何让那些不想做图形化布局描绘的人仍然可以达到目标。这导致了对两个新的业务规则的考虑：

1. 潜在顾客可以选择把一组住宅或者营业所的建筑平面图发送给 CPI 的客服。CPI 员工将把这些平面图转化为一个布局。

2. 潜在顾客可以打电话给 CPI，然后 CPI 将派一个销售代表去和他们见面。

提出的第一个业务规则可以让 CPI 适应更大范围的客户。提出的第二个规则，虽然很有意思，但是因为公司不想建立一支销售队伍（人员）而最终被否决。这就说明了 SafeHomeAssured. com 的界面设计模型（以及其他模型）可以用来逆向驱动新的业务规则的创建。

6.2 创建的模型

到现在为止，我们希望你会同意模型可以帮助你更清晰地理解项目的各个方面。但是你应当创建什么模型呢？显然，这取决于你想要理解什么。[⊖]识别哪里需要更好的理解（在建模中）的最佳方法是考虑如下两个方面的结合：1）使用的过程（以及因此与被执行的活动、动作和任务相关的观点）；2）正在开发的产品（和因此需要被理解和建模的事物）。我们首先要做的是对这些方面依次逐个进行审查，然后看它们如何联合为我们提供关于建模对象的更清晰视角。

6.2.1 过程告诉了我们关于建模的哪些信息

再次回顾最初在第3章（特别是图3-2）中概述的 WebE 过程。其中的一个关键 WebE 活动就是建模，而且它包含分析和设计两个主要动作。虽然这两个动作的方法大不相同，但是都旨在产生模型。分析检查利益相关者的需求以理解 WebApp 应当干什么。

从根本上来说，分析建模有助于你理解正在处理的问题的本质，以及 WebApp 的“形状”，以便让你解决问题。对于这一点来说，有很多和不同的观点相关的不同要素。CPI 业务开发的执行副总裁将关注业务环境和起作用的 WebApp 可能创造的业务机会。CPI 的营销经理将对用户如何与 WebApp 交互以及如何对其响应感兴趣。CPI 负责产品开发的高级经理将对 WebApp 的特定需求感兴趣。每一个兴趣集都和其他兴趣集相关，但是每一个都代表了一种不同的观点，而这可能会决定一个不同的模型。

相反地，设计建模是关于对正在开发的 WebApp 内部结构的理解，以及由此如何创建分析模型所确定的 WebApp 形状。同样，这里会有不同的视角。CPI 的系统架构师将关注 WebApp 的总体结构、将用到的主要技术以及 WebApp 如何与其他系统连接；信息设计师将强调 WebApp 中内容的性质、结构和管理；WebE 团队中的程序员将最关心用于实际实现的底层功能细节。

6.2.2 WebApp 告诉了我们关于建模的哪些信息

在第1章中，我们考虑了不同类型的 WebApp（例如，信息型、下载型、定制型、交互型）。

⊖ 它同时还将依赖于你所拥有的用于建模工作的时间。

每一类都以不同的方式将它的关注点融合到信息、功能和行为之中。类似地，Web 工程师将在 WebE 过程的不同阶段关注 WebApp 的不同方面——界面、内容、导航、功能。一个功能复杂的应用（例如，一个集成了对工作流详细支持的拍卖站点）更可能需要全面的功能建模。一个涉及丰富的、各种各样的信息的应用（例如，新闻网站）可能需要关注内容建模和描述用户如何与内容进行交互的交互模型。

除了应用的类型，应用中的不确定性领域也会影响构建的模型类型。例如，考虑一个富功能应用，其中，功能被良好地定义和理解，但是内容的性质不清晰。对于 WebE 团队来说，更明智的举措是关注（至少开始）于那些有助于澄清内容不确定性的模型。

6.3 建模框架

很多研究者已经提出了很多框架，它们阐明了可用于 Web 工程的模型集合。除了提出具体的模型类型，框架常常旨在建立相关的术语和概念。大多数框架都关注于应用的架构，所以常常被称为架构框架（architecture framework）。

出现最早、综合性最强、应用最广的一个框架是 Zachman 框架［Zac87］［Sow92］，如图 6-2 所示。Zachman 框架包括两维。第一维描述应用的利益相关者的视角，同时包括计划者、拥有者、设计者、构建者和分包者的视角。第二维列出 6 个问题：什么（数据）、如何（功能）、哪里（网络）、谁（人）、何时（时间），以及为什么（动机）。这两个维度建立了一个 5×6 矩阵。每个单元格描述一个独一无二的模型，与一个特定利益相关者询问的一个特定问题相关的架构或者描述。很多其他流行的框架，包括架构的 4+1 视图模型［Kru95］和模型驱动架构（MDA）［OMG01］，提供了类似的见解。[3]对这些框架的详细讨论以及它们与 Web 工程的关联参见［Kon05］。

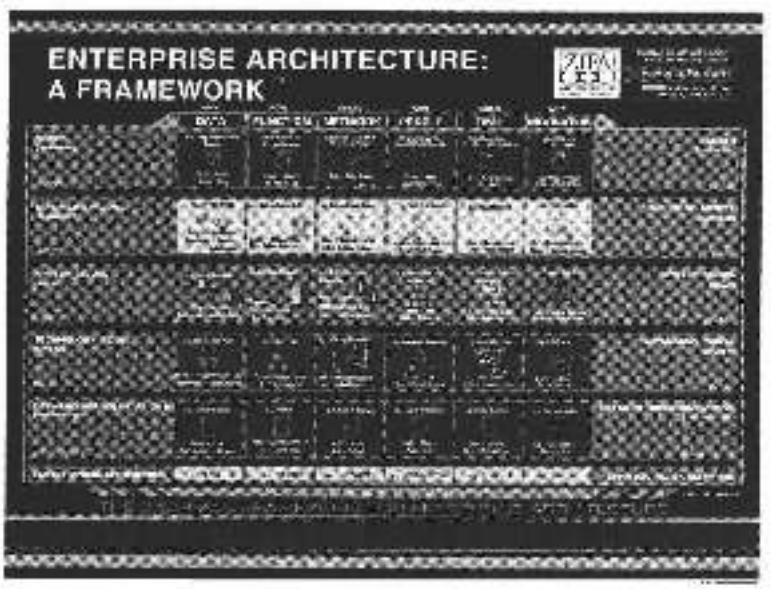


图 6-2 Zachman 框架

资料来源：转载自 www.zifa.com/framework.html，获得了 John Zachman 的许可。

[3] 除了已经讨论的这些框架之外，还有很多其他框架，其中很多是为特定的应用域而开发的，如联邦企业架构框架（FEAF——为美国联邦过程描述的共享的开发过程）；命令、控制、计算机、通信、智能、侦察和侦查（C4ISR）架构框架（为美国国防部和关键供货商架构上的指导）；美国财政部企业架构框架（TEAF——以工作产品的形式支持美国财政部的业务过程）。

6.3.1 对 Web 来说，有建模框架吗

过去的建模框架（例如，Zachman 框架）是为传统信息技术（IT）应用所开发的。它们直接应用于 WebApp 开发是有问题的。WebApp 的可用建模框架应当能够适应 Web 开放和模块化架构的开发。类似地，可重用信息和组件以及对用户界面质量增加的关注（Web 架构的所有特性）都必须要适应。

本书的作者（David）和几位同事（Xiaoying Kong 和 Li Liu［Kon05］）联合开发了一个更适合于 Web 的建模框架。这个改进的框架，称为 Web 应用架构框架（Web Application Architecture Framework，WAAF），把与 WebApp 开发相关的焦点归为两个维度。如表 6-1 所示，每一行（水平维度）显示了 WebE 过程不同参与者的视角：业务拥有者、WebApp 用户、信息设计师、应用架构师、开发者和测试人员。竖直维度（列）把架构归为 4 类，它们代表了主要的建模领域：结构（是什么）、行为（怎么做）、位置（在哪里）和模式。前 3 类（是什么、怎么做、在哪里）和 Zachman 框架中的相对应。而第 4 类（模式）则是基于模式[⊖]对一般软件应用的重要性的逐渐认识（尤其是对 WebApp［Pla02］［Mon03］）而添加的。框架中的每个单元格视情况而定，要么是一个模型，要么是一个描述，要么是一个架构[⊜]。

表 6-1 Web 应用架构框架（WAAF）矩阵

	结构（是什么）	行为（怎么做）	位置（在哪里）	模式
计划架构（计划者的角度）	对业务重要的事情列表	业务执行的过程列表	业务操作的位置列表	可能的业务模型和模式
业务架构（业务拥有者的角度）	例如，业务实体关系模型	例如，业务过程模型	例如，业务实体位置模型	例如，业务模型模式
用户界面架构（用户的角度）	例如，用户界面结构模型	例如，用户界面流模型	例如，用户站点地图模型	例如，界面模板、导航模式
信息架构（信息架构师的角度）	例如，信息字典	例如，信息流模型	例如，信息节点位置模型	例如，信息策略模式
系统架构（系统架构师的角度）	例如，系统功能模块/子模块/服务器页面结构	例如，模块/子模块/服务器页面的工作流模型	例如，模块/子模块/服务器页面的站点映射模型	例如，设计模式/表示风格
Web 对象架构（开发者的角度）	例如，物理对象关系	例如，源码中的算法	例如，网络部署模型	例如，商业现货软件、组件、代码库
测试架构（测试人员的角度）	例如，测试配置	例如，测试过程	例如，测试部署	例如，模板、测试文档的标准

6.3.2 建模如何与 WebE 过程关联

WAAF 框架指明了哪种模型（即 WebE 过程中的术语工作产品）适合于哪个目的和受众。然而，我们需要更进一步，考虑这些模型应该如何与 WebE 框架中建模活动的特定动作和任务进行关联。建模活动包括两个主要动作——分析和设计。这些动作的每一个都需要一系列的 WebE 任务和相关的工作产品，它们将在第 7 章～第 13 章中进行详细讨论。然而，现在审查它们是如何与 WAAF 框架融为一体的可能很有用。分析和设计动作由如下的 WebE 任务组成：

⊖ 模式是遇到的一般问题的通用解决方案。它们通常表示为一个描述或者模板，描述如何解决问题，可用于不同的环境下。WebApp 的设计模式将在第 13 章进行讨论。

⊜ 这些中的一部分将在第 7 章～第 11 章中进行讨论。

分析建模（第7章）

A-1. 决定是否需要一个需求模型。

A-2. 描述 WebApp 内容。

A-3. 识别内容之间的关系。

A-4. 精化和扩展用户场景。

A-5. 审查用户场景。

A-6. 为复杂场景创建交互模型。

A-7. 精化界面需求。

A-8. 识别功能。

A-9. 定义约束和性能需求。

A-10. 确定数据库需求。

设计建模（第8章～第13章）

D-1. 对 WebApp 进行美学设计。

D-2. 设计界面。

D-3. 设计导航策略。

D-4. 设计 WebApp 架构。

D-5. 设计内容以及支持内容的结构。

D-6. 设计功能组件。

D-7. 设计适当的安全和隐私机制。

D-8. 设计审查。

每个任务都会使用在前面的框架活动或者早期的模型（或者其他工作产品）中获得的信息，并增加新信息来对已存在的模型进行修改或者构建新的模型。我们把如何实现这一点的细节放在后面的章节里，但是一些例子可以帮助阐明我们的意思。

考虑任务 A-6，为复杂场景创建交互模型。如图 6-3 所示，这个任务的执行涉及对用户任务和每个任务所需内容之间关系的理解，以及相关的应用状态。[⊖]它也许还需要业务过程和信息结构方面的信息。任务 A-6 的输入一般包括结构和行为方面广泛的业务层信息（例如，业务拥有者的视角）。在这个上下文中，结构可以通过审查业务实体及其之间的关系（如顾客之间交换的信息）来进行定义；行为可以通过审查业务工作流来进行定义。产出是从用户的视角描述用户如何与应用（例如，应用行为）进行交互的一个模型。

作为任务 A-6 的一个简单的例子，考虑 SafeHomeAssured. com 的增量 5，“监控服务的信息和订购”。业务层结构处理对于监控服务建立一个新用户必需的具体信息。这可能包括具体的数据条目和数据库结构，这个数据库作为使用 SafeHome 监控服务的客户的基础。业务层行为处理已经建立了一个用户注册监控服务的工作流，包括一组合适的用户信息、用户的验证和审查（为了安全的目的）、支付选项和选择的说明，以及与注册用户所需要的其他细节的混合。另外，用于注册的用户信息必须以某种方式进行建模。应当注意，增量 5 中界面的实际设计直到任务 D-2 才开始。

图 6-4 所示为设计任务 D-3“设计导航策略”所需要的信息和所创建的模型。在这里，任务涉及从用户角度来看的结构（例如，和用户交互的内容以及内容之间的关系）和行为（例如，用户的交互模型）方面的输入。输出是一个应用信息架构模型。

⊖ 应用状态是应用所处条件的一个外部可观察的描述。例如，一个机票预订应用的状态可能有一个登录的用户和一个完全选好的航班，但是没有确认或者付款。

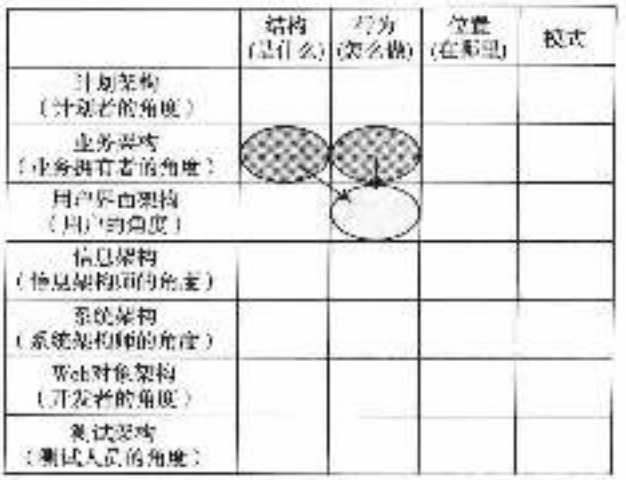


图 6-3 与分析任务"为复杂场景创建交互模型"有关的 WAAF 模型

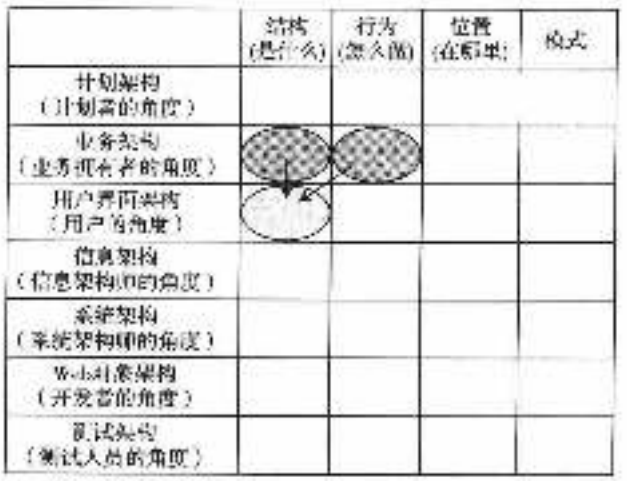


图 6-4 与设计任务"设计导航策略"有关的 WAAF 模型

作为任务 D-3 的一个简单的例子，考虑 SafeHomeAssured. com 的增量 4，"空间布局和安全系统设计"。用户层结构处理用户创建空间布局必须指定的具体信息，以及在增量 4 的范围内，一个数据条目如何与其他条目发生联系。它指明创建空间布局所需要的用户动作流，并与美化布局和推荐安全监控设备的系统响应进行耦合。这些信息被传递给"信息架构师"（WAAF 的术语），由他来建立总体的界面结构。

WAAF 提供了一个总体框架，有助于 WebE 团队选择可以适应不同 WebE 动作和任务的模型类型。对分析和设计可用的实际模型将在第 7 章到第 13 章中进行讨论。

6.4 建模语言

在 Web 工程上下文中，一种建模语言（Modeling Language）包含一系列的符号、术语和符号，以及建立它们之间联系的规则。一种建模语言常常都有一个正式的结构化表示和一系列图形元素。因为可以对一个 WebApp 的很多不同方面进行建模，所以 WebE 团队可能遇到两种不同类型的语言：

- 一种可进行多样化建模的丰富的语言

- 大量不同的语言，其中每一种专用于一个特定目的

UML（统一建模语言）[⊖] [Rum98] 属于第一类，它包括丰富的建模符号，包含了大量不同类型的模型。WebML [Cer00]（第二类语言）则具体得多，它关注于设计中和 WebApp 相关的一个特定方面。

Gu、Henderson-Sellers 和 Lowe 对 Web 建模语言进行了一些深入考虑，包括对不能满足 WebApp 建模需求的建模语言进行扩展 [Gu02]。这些需求很多都和我们想要对 WebApps 架构进行建模的方式有关。建模需求被分解为功能的、信息的和一般的需求。接下来的几节总结了这个早期工作的关键要素。

6.4.1 应当具备哪些建模功能的能力

为了建模 WebApp 交付的功能，建模语言应当提供如下的能力：

- **建模集成和连接的能力**。建模语言必须能够表示（子）系统和资源之间的集成。在很多组织中，新开发的 WebApp 需要和已存在的核心业务应用密切合作。这个挑战需要建模语言支持 WebApp 和这些遗留应用的无缝集成。类似地，因为组件的集成很大程度上依赖于接口说明，所以建模语言需要提供准确而清晰地对组件接口进行建模和文档化的能力，并能够表示连接机制。
- **支持模式建模的能力**。研究和实际经验都证明，只要有可能就使用模式。这包括对不同问题和方案之间的共同点进行建模的支持。建模语言应当可以确定模型和模式（例如，即使当模式是大体可应用的时候，从一个应用到另一个应用有很大不同的那些方面）中的变化点以及在特定场景中何时以何种方式对这些变化点进行应用的规则 [Fon00]。
- **以技术无关的方式表示概念的能力**。支持 WebApp 的技术变化很快。因此，创建依赖于特定技术的 WebApp 规范或设计架构是不明智的。的确，在 WebApp 开发中，技术无关的设计是非常必要的。
- **建模复杂系统功能的能力**。有很多 WebApp 支持作为组织业务模型重要部分的关键业务过程和工作流。[⊜]这些过程常常会涉及复杂的功能，而这些功能是应当能够被表示的。同时应当可以定义业务规则和用于确保整个应用完整性的约束。

6.4.2 应当具备哪些建模信息内容的能力

建模内容的能力是很多复杂的 WebApp 成功实现的关键。Web 信息说明、架构和设计解决内容及其如何被管理、信息结构化和访问、用户的语境化、导航设计和对导航的支持，以及信息视角和表示问题。建模语言应当提供如下的信息建模能力：

- **建模表示层概念的能力**。和传统的软件应用相比，WebApp 的表示设计有自己独有的特性，包括表示层更复杂的功能、表示层的各种媒体类型，以及非 IT 专注的设计者（例如，图形设计人员、多媒体工作人员、作者和市场分析人员）对建模语言的使用。
- **建模导航结构和行为的能力**。表现复杂导航行为和结构的能力是 WebApp 的一个关键特征。很重要的是，不仅能表示导航，还能表示导航上下文。
- **建模用户和信息交互的能力**。WebApp 通常支持非常复杂的用户交互，而这些交互受很多因素的影响。两个关键的因素是信息来源丰富的多样性和复杂的访问模式（例如，反应性的、主动的、指导的、自由格式的、探索的）。

⊖ 参见 www. omg. org/uml/ 中对 UML 的详细讨论。

⊜ 业务模型定义了可以使业务（或业务的一部分）吸引顾客、生产和出售产品并产生收入的战略和策略机制（例如，对业务工作流的建模、信息或订单跟踪以及事务处理）。

- **建模用户角色和用户组的能力**。WebApp 的成功很大程度上取决于用户的满意。这就意味着你需要理解用户的角色和目标。但同时也意味着建模某些方面（如个性化）是很关键的。
- **建模内容的能力**。对内容的有效管理是 WebApp 开发成功的一个关键因素。内容管理（第 16 章）通常涉及（但不限于）对内容结构的设计，绝大多数是以数据库模式的格式进行表达。

6.4.3 在建模语言中，应当有哪些通用能力

建模语言除了支持 WebApp 功能和信息架构的建模之外，提供一组通用能力也是很关键的，例如，以一致和内聚的方式对 WebApp 的功能和信息方面的模型进行连接。建模语言应当提供如下的通用能力：

- **建模业务领域概念的能力**。经验表明，大多数 WebApp 的一个关键特性是业务模型和技术架构之间很强的链接。[⊖]结果就是，建模质量很大程度上依赖于开发者在 Web 技术影响的背景下，对业务模型及其所需变化的理解。为了易于理解并对理解进行文档化，建模语言需要提供建模业务领域概念（例如，业务过程、业务实体、工作流、业务规则）以及用户的角色和责任的能力。与业务相关的开发和建模产品通常由既具有 IT 背景又具有业务背景的开发者进行创建和使用。因此，对业务领域概念建模的设计需要考虑目标用户类型，使这些模型产品可以在开发团队和业务部门之内或之间易于理解、沟通和修改。
- **连接业务模型和技术架构的能力**。本章前面曾提到“逆向建模”的重要性。建模语言需要提供表示业务模型和技术架构之间以及业务模型中的模型元素和技术架构中的模型元素之间链接的能力。这种互联需要在不同的抽象层次进行表示。
- **连接信息和功能的能力**。WebApp 的完整性和内聚性很大程度上取决于其信息架构和功能架构（这将会在第 8 章中进行更详细的讨论）之间紧密而灵活的互联。将信息和功能架构进行连接非常关键，因为这样 WebApp 就可以成功地以连贯的方式处理业务需求[Low01]。
- **保持系统完整性的能力**。WebApp 的完整性可能受到来自如下多方面的挑战：WebApp 的复杂性、利益相关者的不确定性所导致的需求变更、技术的快速变化、应用功能和结构的持续演化，以及多个学科和技术领域的结合。建模语言可以通过基于支持对模型的自动完善和参照检查的表示来帮助确保应用的一致性。
- **支持理解和沟通的能力**。为了在 WebApp 开发时便于沟通，建模语言应该提供从不同的视角和抽象层次[⊜]对 WebApp 的不同方面进行表示的能力。提供这些抽象层次之间协调的互联也很重要。也就是说，当 WebApp 模型的某个方面变化时，可以在相关模型的相关方面识别和突出变化所带来的潜在影响。
- **支持 Web 系统生命周期管理的能力**。因为 WebApp 在其生命周期中表现出细粒度的演化，所以维护（例如，对 WebApp 的纠错、修改和加强）扮演着越来越重要的角色。因此，建模语言需要支持易于演化的模型，并且便于对模型中那些可能是难以变化的方面进行确定和理解。

⊖ 换句话说，WebApp 和其他应用相比，其类型对业务和它的运作方式有更直接和更大的影响。它的极端例子就像 eBay 这样的组织，如果没有 WebApp 的支持，那么其业务模型根本就不存在。即使是更传统的组织，同样的事情也适用。例如，考虑在线银行业应用是如何成为现代金融机构的支柱的。

⊜ 抽象层次指表示 WebApp 的产品或技术细节的层次。



为建模做准备

场景：Web 工程区域，一个 2 天的关于 UML（将在后面章节进行讨论的一种建模语言）的短期课程完成之后

参与者：SafeHomeAssured.com 的 WebE 团队成员

会话：

团队成员 1：我认为我理解了 UML 的机制及其如何应用到传统的软件，但是我不完全清楚在 WebApp 分析和设计中我们应当使用多少。

团队成员 2：我认为我们还是谨慎使用它。有这么多可能的图形，而且我们的时间很紧。

团队负责人：我还记得教员说的两个意见。我们需要“把业务模型和技术架构连接起来”，同时“把信息和功能‘连接’起来。”

团队成员 3：它的确切意思究竟是什么？

团队成员 1：依我看，我们使用 UML 对 SafeHomeAssured.com 的营销部门所希望的特定业务需求进行建模，并用它来展示 WebApp 内容如何与操作内容的内置功能发生联系。

团队成员 2：那么要是需求是显而易见的呢？

团队负责人：那我们很可能就不需要 UML 模型。

团队成员 3：教员还说了一些关于如何考虑 WebApp 整个“生命周期”的事情。

团队成员 1：是的，我记得。如果我们做得好，那么我们创建的模型就能够支持我们事后做出的变更。

团队成员 2（笑）：但是我认为我们应当做一个好得不会有任何变更的建模工作。

团队成员 1：现实一点儿吧，变更总会有的！不过，UML 模型可以让变更容易处理，尤其是在一两年内其他人必须做的时候。

团队负责人：但是这只有在模型是精确而最新的情况下才是正确的。

团队成员 2：这也就是我们不想创建太多模型的原因。我们无法保持其是精确和最新的。

团队负责人：这是毫无疑问的！

6.5 已有的建模方法

目前，已经有了很多种不同的建模语言，对这些语言的各种优缺点的讨论信息也非常丰富[Gu02]。尝试考虑已有建模语言的优缺点。其中，通过考虑对 Web 专用建模语言（OOHDM[Sch98]、WebML[Ceri00]、Koch[Koc00]、WAE[Con02]、W2000[Bar01]和 HDM-Lite/Autoweb[Fra00]）进行扩展来满足前面章节中概括的需求。评估的结果如表 6-2 所示。

表 6-2 建模语言差异分析

建模语言的要求	OOHDM	WebML	Koch	WAE (Conallen)	W2000	HDM-lite/ Autoweb
建模非软件系统的能力	N	N	N	P	P	P
支持模式建模的能力	Y	N	N	P	P	N
以技术无关的方式表示概念的能力	P	Y	Y	P	Y	Y
建模复杂系统功能的能力	N	N	N	P	N	N
建模表示层概念的能力	Y	Y	Y	N	N	Y
建模导航结构和行为的能力	Y	Y	Y	P	P	Y

（续）

建模语言的要求	OOHDM	WebML	Koch	WAE (Conallen)	W2000	HDM-lite/ Autoweb
适应用户和产品交互的能力	P	Y	Y	P	Y	Y
适应用户角色和组的能力	N	Y	N	P	Y	Y
建模内容的能力	Y	Y	Y	P	Y	Y
建模业务领域概念的能力	P	P	P	P	P	P
连接业务模型和技术架构的能力	P	P	P	P	P	P
连接信息架构和功能架构的能力	N	N	N	N	P	N
保持系统完整性的能力	N	P	N	N	P	P
支持迭代和演进的能力	Y	Y	P	P	Y	Y
独立于平台的能力	N	Y	Y	Y	N	N
支持 Web 系统生命周期管理的能力	P	P	P	P	P	P

注释：Y = 是，N = 不是，P = 部分。

来源：[Gn00]

如表 6-2 所示，大多数建模方法都对基本建模需求进行了充分考虑。然而，其他方面却根本没有涉及。例如，没有一种建模语言很好地支持如下这些方面：

- 建模集成和连接性的能力。
- 建模复杂系统功能的能力。
- 建模业务领域概念的能力。
- 连接信息架构和功能架构的能力。

在后面的几章中，我们将更多地讨论如何处理这些方面的建模问题。

各种建模语言相对的关注点如图 6-5 所示。图中，用例和基于 UML 的建模记号可以很好地覆盖一般的软件应用，特别是 WebApp 的功能元素。但是 UML 并没有提供好的获取 WebApp 信息元素的代表性工具。已经发展的专门 Web 建模记号可以处理信息型元素（至少一部分），但是并没有提供和功能建模之间的良好连接。大多数建模语言都没有提供和在大多数 WebApp 中都非常重要的业务模型和过程之间好的建模连接。

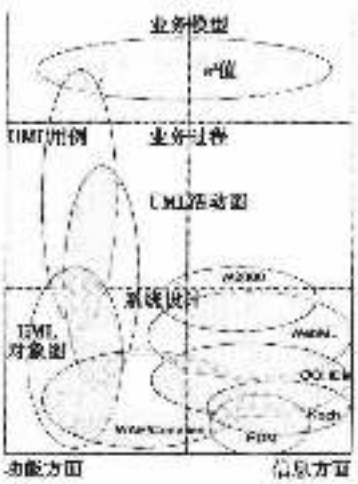


图 6-5 已有建模方法的差异分析

来源：[Cu02]。

6.6 本章小结

在本章中，我们把建模作为一个概念进行了讨论。Web 工程的建模活动创建了同一现实的不同表示。这些表示的有效性由使用它们的人和为建模创建 Web 应用架构框架（WAAF）的一系列标准来定义。

有很多可用于描述 WebApp 的建模语言。理想情况下，这些语言展示了一系列功能的、信息的和通用的需求，将产生有效的分析和设计模型。在本章中，我们描述了这些基本需求，并对其中一些语言进行了简洁概述。在后续几章中，我们将展示如何为 WebApp 开发有效的模型。在第 7 章，我们将详细讨论分析建模，在第 8 章我们将关注在 WebApp 的设计中用到的模型。自始至终，我们都将考虑如何为不同的任务选择正确的模型，以及如何应用和解释这些模型。

参考文献

[Bar01] Baresi, L., G. Garzotto, and P. Paolini, "Extending UML for Modelling Web Applications," *Proc. 34th Hawaii International Conference on System Sciences*, 2001, Hawaii, p. 3055.

[Cer00] Ceri, S., P. Fraternai, and A. Bongio, "Web Modelling Language (WebML): A Modelling Language for Designing Web Sites," *Proc. WWW9/Computer Networks*, vol. 33, 2000, pp. 137-157. See also www. webml. org/ (accessed July 25, 2007)。

[Con02] Conallen, J., *Building Web Applications with UML*, 2nd ed., Addison Wesley Professional, 2002.

[Fon00] Fontoura, M., W. Pree, and B. Rumpe, "UML-F: A Modelling Language for Object-Oriented Frameworks," *Proc. ECOOP* 2000 – *Object-Oriented Programming*: 14*th European Conference*, Sophia Antipolis and Cannes, France, June 2000, Springer, pp. 63-82.

[Fra00] Fraternali, P., and P. Paolini, "Model-Driven Development of Web Applications: the AutoWeb System," *ACM Transactions on Office Information Systems*, vol. 18, no. 4, 2000, pp. 323-382.

[Gu02] Gu, A., B. Henderson-Sellers, and D. Lowe, "Web Modelling Languages: the Gap Between Requirements and Current Exemplars," Paper presented at the AusWeb 02: *The Eighth Australian World Wide Web ConferenceAusWeb*, Sunshine Coast, Australia, July 2002, pp. 5-10. See also http: //ausweb. scu. edu. au/ (accessed July 25, 2007).

[Koc00] Koch, N., H. Baumeister, R. Hennicker, and L. Mandel, "Extending UML to Model Navigation and Presentation in Web Applications," *Modelling Web Applications in the UML Workshop*, UML2000, York, England, October 2000.

[Kon05] Kong, X., L. Liu, and D. Lowe, "Separation of Concerns: A Web Application Architecture Framework," *Journal of Digital Information*, vol. 6, no. 2, 2005, http: //jodi. tamu. edu/Articles/v06/i02/Kong/ (accessed July 25, 2007).

[Kru95] Kruchten, P., "The 4 + 1 View Model of Architecture," *IEEE Software*, vol. 12, no. 6, 1995, pp. 42-50.

[Low01] Lowe, D., and B. Henderson-Sellers, "Web Development: Addressing Process Differences," *Cutter IT Journal*, vol. 14, no. 7, 2001, pp. 11-17.

[Mon03] Montero, S., P. Díaz, and I. Aedo, "Formalization of Web Design Patterns Using Ontologies," *Atlantic Web Intelligence Conference2003*, Madrid, Spain, May 5-6, pp. 179-188.

[OMG01] Object Management Group, "Model Driven Architecture," 2001, www. omg. org/mda/

(accessed July 25, 2007).

[Pla02] Platt, M., "Microsoft Architecture Overview," 2002, http://msdn2.microsoft.com/en-us/library/ms978007.aspx (accessed July 25, 2007).

[Rum98] Rumbaugh, J., I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1998.

[Sch98] Schwabe, D., and G. Rossi, "Developing Hypermedia Applications Using OOHDM," *Workshop on Hypermedia Development Processes*, Methods and Models (Hypertext'98), Pittsburgh, 1998.

[Sow92] Sowa, J. F., and J. A. Zachman, "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal* vol. 31, no. 3, 1992, pp. 590-616.

[Zac87] Zachman, J. A., "A Framework for Information Systems Architecture," *IBM Systems Journal* vol. 26, no. 3, 1987, pp. 454-471.

第7章　WebApp分析建模

如果你没有理解需求是什么，那么就不可能开始对一个WebApp增量进行设计和构建。这就是Web工程过程建议一个沟通活动（第4章）作为从利益相关者那里引出需求的有组织的方法的原因。结果就是生成将要创建的每个Web增量的需求集合[⊖]。

但是我们如何知道这些需求是否充分？或者完整？或者是清晰并且没有矛盾的？这些就是对WebApp进行分析建模的目的。*分析建模*（analysis modeling）有助于确保你清晰地理解将要创建的事物。

7.1　理解在WebE中的分析

当对WebApps进行需求分析的思想被提出的时候，Web开发者常常都是持怀疑态度。“毕竟，”他们争辩道，“WebE过程必须是敏捷的，而分析是要花费时间的。它会在我们需要设计和创建WebApp的时候拖慢我们。”

分析问题确实要花时间，但是解决错误的问题会花费更多的时间。对每个WebApp开发者的问题是很简单的：你是否确定你已经理解了问题的需求？如果答案是肯定的，那么或许可以跳过分析建模；但是如果答案是否定的，那么就应当进行需求分析。Steve Franclin［Fra03］针对这一问题发表了自己的看法：[⊜]

> *开发者往往在没有真正理解他们想创建什么或者如何去创建的情况下，就直接进入编码阶段。服务器端的编码通常都是以特定的方式完成的，数据库表在需要的时候就进行增加，架构有时也以一种随意的方式演化。但是某种形式的建模和严格的软件工程可以让软件开发过程更加顺利，并且能确保Web系统在将来更加易于维护。*

在WebApp开发中，严格的分析建模和敏捷方法可以很好地共存吗？我们认为答案是肯定的，但前提是WebE团队要认识到分析建模必须根据问题的大小和后续设计任务的属性进行调整。

7.1.1　分析到什么程度才是足够的

Web工程师应当对分析的角色进行足够深入的理解，这样才能知道需要实施多少分析。强调分析建模的程度取决于下面的这些和规模相关的因素：

- WebApp增量的大小和复杂性
- 利益相关者的人数（分析有助于确定来自不同来源的有冲突的需求）
- WebE团队的大小
- WebE团队成员以前在一起工作的程度（分析有助于对项目形成一个共同理解）
- 组织的成功程度直接依赖WebApp的成功

这些观点反过来说就是，随着项目变得越来越小、利益相关者的人数越来越少、开发团队更加具有凝聚力、以及应用的关键性变弱，那么使用更加轻量级的分析方法就是合理的。

虽然我们提出应当在开始设计之前对问题进行分析，但这并不是说所有的分析都必须在所有的设计之前。事实上，对WebApp的特定部分的设计仅仅需要分析那些对这部分有影响的需求。在SafeHomeAssured.com中，我们可以在没有分析与产品销售相关的电子商务功能的功能性需求的情况下，正确地进行网站总体美学的设计（布局、颜色搭配策略等）。WebE团队仅仅需

⊖ 这些需求几乎肯定是不完整的，但是它们为后续的建模动作提供了一个起点。

⊜ 额外的重点。

要分析与将要交付的增量的设计工作相关的部分问题。

7.1.2 可以使用原型进行分析吗

对很多 WebApp 而言，最好的分析方法并不是建模，而是当 WebE 团队必须理解页面布局或者美学需求的时候，最好的选项常常是创建一个快速原型[1]。在这种背景下，一个原型（prototype）是 WebApp 增量（常常使用简单的 HTML 来实现）的一个或多个方面的一个精简版本。原型提供了对页面布局和美学、导航机制的感受，在一些情况下，提供对功能和信息内容的感受，但是并没有以一种完整的方式实现它们。

WebApp 的原型可以用作对它所基于的需求进行评估，同时对于 WebApp 的最终构建来说，它常常是有用的。利益相关者可以对原型进行试用，并且马上就能发现他们喜欢的和不满足要求的事物。他们可以指出遗漏的事物，同时调整他们对于真正所需事物的看法。所有的这些都是用于理解的很卓越的手段。

虽然原型化方法很奏效，但同时也会带来问题。很常见的是，利益相关者使用原型，然后询问它是否能很快地用于生产（例如，明天?）。WebE 团队告诉利益相关者，原型仅仅用于展示，还有很多缺陷，对于生产和使用来说是不合适的。然而，利益相关者仍然坚持要快速地使用。

WebE 团队常常都会屈从于这种请求。"我们或许可以忍受这些缺陷，"团队的一员冒险地说。"毕竟，我们将会有部分已经创建的 WebApp，而且时间确实很紧。我们将实现一些补丁，然后继续做下面的工作。"结果就会是一个设计不良的 WebApp，它也许很难去适应和改进。

原型法可以使用吗？"当然！但前提是你的团队既明白其危险，又明白其优点。



原型代替建模

场景：CPI 公司的 Web 工程区域

参与者：作为一对儿在一起工作的 SafeHome-Assured.com 团队的两个成员，开始增量 3（产品报价单和产品订单处理）的建模活动。

会话：

团队成员 1：我和营销人员谈过了，他们已经把报价单和订单处理的工作流给我了。没有什么很难的东西，但是有很多 CPI 特有的需求。我们将不得不从 WebApp 访问订单录入数据库，而且……

团队成员 2（打断）：因此，我们将使用 UML 来对它进行建模？

团队成员 1：虽然我确实认为我们需要对用例进行精化，而且 UML 在那方面也许是合适的，但是我认为 UML 在这里是不合适或者不必要的。我想做的是创建一个订单录入原型，把它早点给营销部门，获得他们的满意，然后使用它来驱动这个增量的完整设计和构建。

团队成员 2：但是其他的技术需求怎么样呢，比如和已有数据库的集成以及……

团队成员 1：这些部分的需求是相对简单易懂的，所以它们都可以等到设计阶段。

团队成员 2：所以取代 UML 模型（除了可能要对用例进行完善），我们使用 WebCarpenter［WebCarpenter 是团队选择来产生 Web 页面的一个虚构的现货供应的工具］来创建一个原型？

团队成员 1：嗯。而且那意味着我们将同时做一些设计工作，但为什么又不呢？它会让我们更进一步，让那些营销部门的人员好好看看他们将得到什么，同时要比 UML 快不少。

团队成员 2：那我们开始做吧。但是必须要确保营销部门明白我们将展示给他们的是一个原型，而不是一个可使用的 WebApp。

---

(1) 在很多情况下，创建一个有效的分析模型所需的时间比创建一个快速的设计原型所需的时间要多。

7.1.3 分析和设计不同吗

很多人认为技术（尤其是软件技术）是无限延展的，利益相关者的任何需求，不论它多么古怪，都可以在 WebApp 内部简单地通过对技术进行塑造以满足需要来实现。不幸的是，这可能导致灾难。

我们所使用的作为 Web 工程一部分的技术有一个和它们相关的确定的“纹理”。任何一个木匠都知道，如果你试图顺着木材的纹理对它进行切割和定形，那么它就会很容易。但是一旦你逆着纹理进行工作，那么事情就会变得困难得多。[⊖]

如果你构建了一个 WebApp，在它的内部，利益相关者的需求和技术能力非常适合，那么项目和最终结果都很可能成功。但是如果你试图与技术相抵触来做事，那么设计工作和它生产的产品将变得更加复杂、耗时和昂贵。

例如，Web 的“纹理”部分是其分布式的客户端 - 服务器架构。那么确保快速 WebApp 响应时间的需求就与这个“纹理”相抵触（主要是因为客户端和服务器之间的通信所带来的延时）。如果 WebE 团队必须满足这个需求，就必须要开发复杂的设计方案（例如，使用 Ajax）。相反地，从任何地点可以简单访问 WebApp 的需求就和“纹理”一致，同时使用 Web 技术的分布性，因此设计和实现就很简单了。

那么我们在这里得到了什么？简单地说，有一些需求本质上实现起来就比其他需求困难得多，代价也高得多。虽然分析可能帮助你理解需求本身，但是只有设计才能帮助你理解构建的成本[⊜]。WebE 团队和其他利益相关者必须要考虑实现需求的成本，以决定需要（需求）是否值得消耗这些成本，以及成本如何影响排列需求优先级的方法。

当然，这并不是说你应当忽视分析而直接跳到设计。WebE 团队是否选择非正式地进行分析，创建一个原型，或者完成更正式的分析模型（本章自始至终都在讨论），理解问题（和定义问题的利益相关者需求）仍然是很重要的——但是它确实需要和设计交织在一起。

7.2 对 WebApp 进行分析建模

我们希望你现在同意这样的观点：理解问题（捕获需求并进行分析）是开发一个 WebApp 很重要的部分。但是我们实际上如何分析需求呢？回答这个问题的一种有用的方法是考虑如下四个关键问题的答案：

1. 分析建模的输入是什么？
2. 分析的输出是什么？
3. 我们可以并且应当执行哪些分析任务？
4. 我们可以使用哪些工具来帮助我们建模？

7.2.1 分析建模的输入是什么

WebE 过程包含沟通活动（第 4 章），它识别将作为分析建模输入的信息。这个输入包括利益相关者和用户种类、业务背景、定义的信息型和应用型目标、总体 WebApp 需求和使用场景。这些信息以自然语言描述、粗略的提纲、草图和其他非正式表示方式进行表示。

⊖ 或者，如果这个类别对你不适用，那么考虑一下在一个繁忙的购物中心，与顺着人流相比，逆着人流或者从人流中横穿要困难多少。

⊜ 在这种情况下，构建成本包括这些方面，例如，完成工作需要的人员和技巧、完成需求需要的时间、需求集成的困难程度以及为了适应需求而对其他的内容或功能所做的必要的修改。

分析首先接收这些信息，使用一种正式定义的表示策略（在适当的时候）来对它进行组织，然后生产出更严格的模型作为输出。分析模型提供了对问题真实结构的详细表示和对解决方案轮廓的洞察。

如果我们考虑一个例子，那么这也许更容易理解一些。在第4章中，我们讨论了SafeHomeAssured.com可能的功能——其中一个称为“通过因特网访问摄像机监控”。当时，这个功能看起来是相对清楚的，而且作为一个用例的一部分进行了一些详细描述。我们将这个用例的一个版本进行复制：

> **用例**：通过因特网访问摄像机监控
>
> **参与者**：房主
>
> **叙述**：如果我在很远的地方，我可以使用任何PC机上合适的浏览器软件登录到SafeHomeAssured.com站点。我输入用户ID和两级密码，并且一旦我通过验证，将获得对我所安装的SafeHome系统的所有功能的访问。为了访问一个特定的摄像机视图，我从显示的主功能按钮中选择“监控”。然后我选择“选择一个摄像机”，住宅的建筑平面图就出现了。然后我选择感兴趣的摄像机。或者，我可以选择“所有的摄像机”作为我的观看选择，就可以看到所有摄像机的缩略图快照。一旦选择了一个摄像机，然后选择“视图”，一个摄像机视图就会出现在由摄像机ID所识别的一个观看窗口内。如果我想要切换摄像机，我选择“选择一个摄像机”，前面的观看窗口就消失了，同时再一次出现住宅的建筑平面图。然后我选择感兴趣的摄像机。一个新的观看窗口就出现了。

在我们继续下面的讨论之前，你应当再次检查这个用例，看一看你是否能发现任何遗漏、歧义或者不清楚的事情。

好的——你发现什么了吗？毫无疑问，你确定的一些问题将成为在设计过程中自然出现的解决方案的某些方面。例如，可能是功能按钮的特定布局、它们的美学外观和感觉、缩略图的大小、摄像机视图和住宅建筑平面图的布置，甚至是一些诸如密码的最大和最小长度这样的细节。这些方面中有一些确实是设计决定（例如，按钮的布局），而其他方面则是不会从根本上影响架构的设计决定需求（例如，密码的长度）。

但是你识别的一些问题是那些将影响到总体设计本身的事情，这些事情更多地涉及对需求的全面理解。我们在本章将对这些问题中的一些进行研究，但是为了给你一个对于一些遗漏的信息的初始认识，考虑下面的问题：

- 提供什么样的视频分辨率？
- 当对摄像机进行监控的时候，如果遇到一个报警状况会怎么样？
- 系统如何处理视野可以进行缩放的摄像机？
- 和摄像机视图一起，应当提供哪些信息（例如，位置、时间和日期、过去的最后一次访问）？

在用例的初始开发中，不会确定或者考虑这些问题。可是，这些问题的答案却可能对设计的不同方面产生重大的影响。

我们真正想说的是，虽然沟通活动为问题的理解提供了一个很好的基础，但是分析将通过提供更多的解释说明来对这个理解进行完善。随着问题的结构作为分析模型的一部分被勾画出来，问题肯定会出现。这些问题填补了鸿沟——或者在某些情况下，确实帮助我们从一开始就找到了鸿沟。

总之，分析模型的输入将是在沟通活动中收集的信息——从一个电子邮件中的非正式记录到有着综合用户场景和产品规格说明书的详细项目摘要。

7.2.2 分析的输出是什么

在本章的前面，我们提到了分析的目标就是理解利益相关者的需要。换句话说，分析提供了一种有准则的机制，用于表示和评估 WebApp 的内容和功能、用户将遇到的交互模式、WebApp 驻留的环境和基础结构。

这些特征中的每一个都可以表示为一系列的模型，这些模型可以使 WebApp 的需求以一种结构化的方式进行分析。虽然具体的模型很大程度上取决于 WebApp 的性质，但是我们仍然可以确定四类主要的模型以及这些模型所支持的相关分析。

1. **内容模型**。确定将由 WebApp 提供的内容的全部范围。内容包括文本、图形图像、照片、视频和音频数据。
2. **交互模型**。描述用户和 WebApp 的交互方式。
3. **功能模型**。确定将应用到 WebApp 内容之上的操作，以及描述与内容相互独立，但是对于最终用户来说却是必要的其他处理功能。
4. **配置模型**。描述 WebApp 驻留的环境和基础结构。

这些模型中的每一个都可以使用一种表示策略（常常称为一种“语言”）来进行开发，这种策略使它的目的和结构可以在 WebE 团队和其他利益相关者之间很容易地进行沟通和评估。结果就确定了一系列的关键问题（例如，错误、遗漏、不一致性、改进或修改的建议、疑惑），并对其采取行动。

7.2.3 应当执行哪些分析任务

很明显，分析任务的焦点将是对 7.2.2 节中讨论的模型的构建和诠释。典型的分析任务包括：

- 确定是否需要一个需求模型。
- 对 WebApp 的内容进行表示。
- 识别内容之间的关系。
- 精化和扩展使用场景。
- 审查使用场景。
- 为复杂的场景创建交互模型。
- 精化界面需求。
- 识别系统功能。
- 识别约束和性能需求。
- 识别数据库需求。
- 对功能需求进行表示。
- 对导航需求进行表示。

在本章的后面，我们将研究这些任务如何被用于创建作为分析活动输出的模型。

7.2.4 可以使用哪些工具来建模

尽管有很多构建 WebApp 的软件工具，但是专门为分析开发的却相对很少。然而，这并不意味着 WebE 团队就没有工具支持。在进行分析时，可以使用如下四类工具。

UML 工具。使用统一建模语言（Unified Modeling Language）来创建分析模型，统一建模语言是一种广泛应用于软件工程领域的建模记号。

原型化工具。实际上任何 WebApp 构建工具（例如，Adobe GoLive）都可以用来创建一个可运行的原型。这些工具允许快速的布局、内容的集成、框架的美学开发，所有的这些对一个快速的原型来说都是合适的。

问题跟踪工具。在对不断生成的分析模型进行集成时，可以用于对出现的问题的解决进行记录和跟踪。

内容管理工具。可以用于对作为 WebApp 的一部分进行使用的内容对象的性质和结构进行建模。

为了列出那些已有的可以帮助你的团队进行有效分析的工具，你可以访问那些对本书进行补充的网站。

7.2.5 如何确定建模是否必要以及哪种方法最好呢

分析还是不分析？这是一个 WebE 团队在进入设计阶段之前应当进行回答的合理问题。在有些情况下，WebApp 是复杂的，并且是关于关键业务的。这时可能就需要原型法和其他建模方法相结合。在其他情况下，需要进行严格分析的仅仅是 WebApp 特性很小的子集，其他的特性则在作为沟通活动的一部分所能获得信息的基础上有充分的理解。最后，WebApp 可能相当简单（即使它很大）。WebE 团队可以决定直接进入设计阶段。

在本章的前面，我们确定了当决定是否需要严格分析时必须要考虑的一些标准。下面的一系列关于你的项目的问题提供了一个很好的起点：

- WebApp 增量的功能或者内容是否复杂到足以需要不仅仅是一些页面或者一些简单的草图？
- 在不同的内容类之间是否存在复杂的关系，并且这些关系对如何实现 WebApp 是否有影响？
- WebApp 增量是否生产对用户很关键的信息？
- WebApp 增量是否交付那些现在还没有很好理解的功能？这些功能在算法上是否复杂？
- WebApp 增量是否对已经部署的增量产生显著的影响？对后面开发增量的影响呢？
- 这个 WebApp 是否是关于关键业务的？
- WebApp 增量是否可能对业务或者业务组织与其利益相关者之间的关系的其他操作产生影响？
- 利益相关者是否可能对 WebApp 增量的需求是不确知的？
- 是否有多于两种的用户种类将以完全不同的方式使用 WebApp？
- 是否很有可能在沟通活动中引入了错误、遗漏或者不一致？

如果对任何问题的答案是“是”，那么你就应当考虑创建一个分析模型。值得记住的一个关键因素是，你应当把建模的焦点放在不能清楚或者有可能在混淆的那些方面。



分析建模

场景：增量 3（获取产品的报价单并订购）开始之前，SafeHomeAssured.com WebE 团队的会议室

参与者：WebE 团队负责人和团队的其他两个成员

会话：

团队负责人：不错，我们已经完成了增量2，而且每个人都对结果很满意。

团队成员1：没有更多的变更吗？

团队负责人：没有，只有我们已经完成的那些。

团队成员2：酷！那么接下来呢？

团队负责人：我们可以开始下一个增量——创建 WebApp 的产品报价单和订购部分。营销部门已经对用例进行了修改，但是都不显著。[分发卡片5，“获取产品的报价单”，和卡片8，“订购产品”。] 我认为你们两个都已经看过这些了。我已经标出了那些变更。

团队成员1（边读边点头）：我认为这两个用例都是相当简单的。我们应当可以直接投入设计，并且很快做完这个增量。

团队成员2（皱眉头）：是的，也许，但是……

团队负责人（看着团队成员2）：听起来你好像并不相信所有的那些。

团队成员2：是的，很可能没有什么，我只是有点儿担心可能我们忽视了一些东西。

团队成员1（有些发怒）：噢，拜托。这只是一个简单的报价单请求和订单系统。Web 上的任何其他网站都有这些，而且它们都是很简单的。我同意监视方面的一些后续材料更复杂一些，但不是这个。

团队负责人（对团队成员1）：你可能是正确的，但是它不能阻止我们在投入之前花几分钟时间更加仔细地看一看它。

团队成员2（看起来如释重负）：我所担心的事情就是我们是否已经真正理解了 SafeHome 提供的报价单的性质。我们在早些时候和他们一起开发的这个用例显示了一个简单的过程：定位一件产品，然后请求报价单。但是我确信我听到了客户谈到的一些关于为不同种类的顾客提供不同定价的事情。想必那会让事情更复杂吧？

团队负责人：嗯。继续……[看起来很感兴趣。]

团队成员2：一旦一个客户为一件产品下了订单，我们甚至不知道他愿意接受的支付选项以及他希望的交付地点的限制，同时……

团队负责人（打断）：好的，好的。我明白了。我同意我们需要更仔细地看一看需求。对我来说核心功能看起来是相当清楚的，但是支持它的内容却是模糊的——像定价模型、支付方式等。我们构建一个内容模型，然后看一看它将把我们引领到哪里来进行下面的工作，怎么样？

团队组员1和2：[都点头表示同意。]

7.3 理解用户

到了一天结束的时候，使用 WebApp 的人们将对它的成功进行评判。因此，很重要的一点是，WebE 团队要理解谁将是用户，用户有什么样的背景和技能，每一类用户如何感知 WebApp 以及提供一个成功的用户体验所需要的特征和性能。

很多的这些信息作为沟通活动（第4章）的一部分得到。在捕获需求的时候确定每一个用户种类，以及通过回答如下问题来完成对每一类用户的描述：

- 当使用 WebApp 的时候，用户的总体目标是什么？
- 用户拥有与 WebApp 内容和功能相关的背景是什么，熟悉程度如何？
- 用户如何到达这个 WebApp？
- 用户喜欢和不喜欢哪些一般的 WebApp 特性？

在开始分析时，你应当再一次更加仔细地检查这些种类和描述，确保你已经完全理解了。这可能包括定义用户的等级，然后对已经开发的使用场景进行精化。

7.3.1 为什么有必要重返用户等级

用户种类作为沟通活动（第4章）的一部分进行识别，在很多情况下相对来简单易懂，而且在分析进行的时候保持不变。然而，在分析建模中随着对问题域有更多的了解，可能有必要把两个或者三个用户种类集合精化为等级。这有助于WebE团队更好地理解不同种类用户之间的关系以及WebApp本身。这种等级在定义连贯的、对不同用户可见的内容和功能集合的时候，也是很重要的。

SafeHomeAssured.com的用户已经进行了充分的定义，不需要进一步精化为更详细的等级。不过，为了说明这种方法，我们在图7-1[⊖]中描述了一个用户等级。

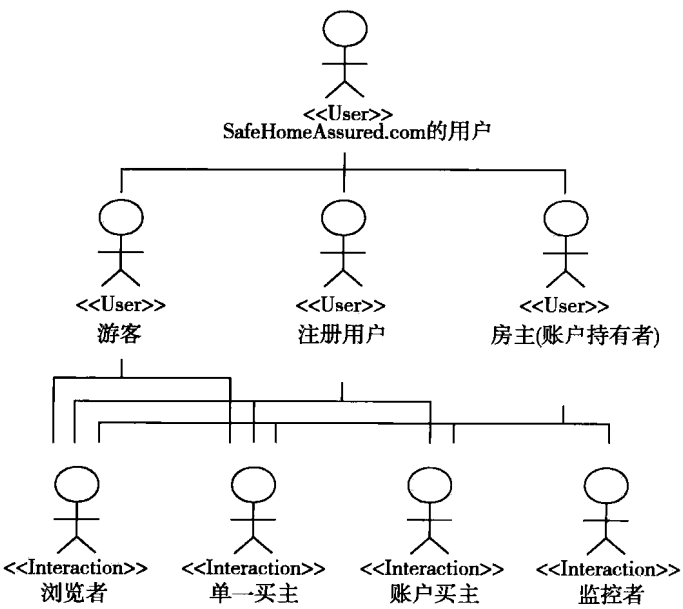


图7-1 SafeHomeAssured.com的用户层次

熟悉UML的读者可能已经注意到了在这个图中直接改变了UML参与者（actor）的符号。UML允许对参与者进行“建模”，但是在UML中，一个参与者严格代表了与系统的一个特定的理想化交互——而不是一个用户。同一个用户可以参与到不同的交互之中（因此也就表现为不同的参与者），但是这个用户会在这些角色之间带有他或她的背景。在我们看来，重要的是对用户背景进行建模，而不仅仅是对单个交互进行建模。为了达到这些目的，我们既展示了不同的用户（使用<<User>>参与者），又展示了这些用户参与的不同类型的交互（使用<<Interaction>>参与者）。

对SafeHomeAssured.com而言，有三种主要的用户类型，它们可以执行四种不同类型的交互：1）游客（guest）可以浏览产品方面的公共信息，下载规格说明和购买产品（但是系统不会保存他们的兴趣选项）；2）注册用户（registered user）可以和游客一样访问相同的功能，但是系统会保存他们的参数选项，并允许他们使用账户进行购买；3）注册房主（registered homeowner）

⊖ 重要说明：本书中很多地方，包括图7-1，我们都使用UML的记号。如果你对UML不熟悉，你可能要在下面提供了基本的UML入门知识的Web资源上花费一些时间：
- Object Management Group（对象管理组织）——UML：www.uml.org/
- UML Tutorial（UML教程），Ian Grahame：http://uml.tutorials.trireme.com/
- UML Primer（UML入门）：http://vinci.org/uml/

可以浏览信息并购买产品，但同时可以使他们的系统在线监控。

对于每一个 < < User > > 参与者，我们需要理解用户种类的性质——它的背景、需求和行为。第 4 章讨论了用户资料以及可以在哪里找到这些信息（常常是从营销部门）。然而，在那个初始阶段，资料很可能是相当不正式的。当我们进行分析的时候，或许有必要更详细地对用户资料进行检查，看看我们是否忽视了什么。

一个典型的用户资料要考虑年龄、性别、背景、专长和知识，以及应用型和信息型目标。在有些情况下，这些信息是相对简单的。在其他情况下，你可能就需要主动对用户进行调查——使用诸如用户会谈、调查或者讨论组这样的技术。例如，SafeHomeAssured. com 的 WebE 团队需要弄清楚 WebApp 用户有哪些技巧（根据对如何最有效地利用 Web 来对他们的住宅进行监控的了解）。可以在澳大利亚的政府网站 www. agimo. gov. au/publications/2004/06/toolkit 上找到一个获取用户资料的方法的很好的例子。

7.3.2 不加修改地应用使用场景吗

正如沟通活动帮助我们理解用户特征一样，它同时定义了使用场景（用例）来描述不同的用户种类在不同的情况下如何与 WebApp 进行交互。对很多项目而言，使用场景将是基本的，也就是说，它们对于初始计划和评估[⊖]来说是足够的，但对于分析和设计来说不够详细。很可能至少有一些用例需要大量的精化。另外，随着分析的进行，在一个用例中定义的动作必须在其他用例定义的用户动作的条件下进行检查。

在第 4 章中，我们为 SafeHomeAssured. com 确定了一套 14 个使用场景，其中每一个都可以使用一个用例进行表示。作为用例精化的例子，考虑框注中关于“获取产品报价单”用例的讨论。

细化用例的另一个关键是识别可能的重叠部分，以及由此产生的在系统实现的时候可能有用的变化。例如，增量 3 的开发涉及两个用例的实现：请求产品报价单和订购。增量 5 涉及的两个用例：请求监控服务的信息和订购监控服务。注意虽然这两个用例一个是关于产品的，一个是关于服务的，但都涉及订购。如果你在实现增量 3 的时候，没有考虑这个共性，那么你很可能最终创建一个产品订购系统，但在后面很难修改以用于订购服务——至少很难重用。在图 7-2 中，我们展示它是如何以一个用例图的形式进行表示的。

同时要注意，就简化应用设计和由此所带来的实现和改进应用的可用性所需要的工作量的减少而言，发现用例之间的共性也很重要。例如，考虑一个事件推广型 WebApp，第一类用户（赛场赞助人）可以查看所选择的赛场内将要举办的事件的描述，例如，体育、文化或者音乐事件。第二类用户（赛场拥有者）可以对他们赛场的事件描述进行编辑。这最初听起来好像是两个不同的用例（查看事件信息和编辑事件信息），但是如果我们认识到在两个用例中信息都是可用的，那么设计将大大简化，只是在一个用例中信息是可以改变的；也就是说，编辑事件信息可以看作是查看事件信息的扩展版本。同样的设计（和大多数同样的代码）用于这两个用例是相当可行的，只不过在查看用例中，编辑功能是禁用的（或隐藏的）。

即使 WebE 团队决定不加修改地应用使用场景，分析仍然可以识别用例之间的共性领域，并建立起用例之间的互连，这种互连可以被设计者用于创建 WebApp。

创建一个类似于图 7-2 所示的图花费相对较少的时间，而且它可能提供大量的分析。

⊖ 计划和估计都用于建立初步的 WebApp 增量集合。

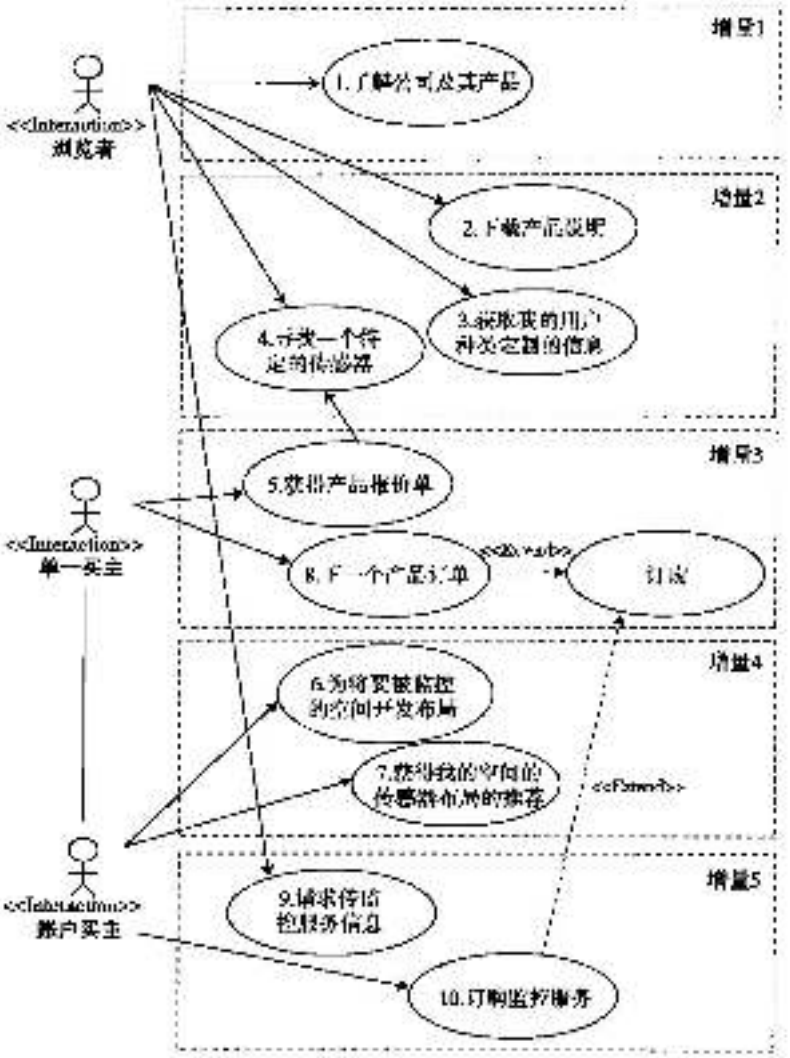


图 7-2 SafeHomeAssured.com 的几例图



精化用例

场景：增量 3（获取产品报价单和订购）开始之前，WebE 团队的会议室

参与者：WebE 团队负责人和 WebE 团队的另外两个成员

会话：

"……继续上一次关于在用例描述中需要更多细节的会话……"

团队负责人：很明显，我们都同意这两个用例需要更多的工作。[他指着卡片 5——获取产品报价单，和卡片 8——订购产品。]问题是我们如何着手？

团队成员 2：嗯，如果我们看一看用例图，就会发现订购研究是获取报价单的扩展。我们在上周已经讨论过这个问题，对获取报价单的人来说，那里更有意义，并且他们一旦有了报价单就可以订购——而不像更一般的处理过程那样，简单地把物品放进购物篮，然后付账离开。

团队成员 1（看起来有一点儿迷惑）：我还是不明白我们为什么要讨论这个方法。

团队成员 2：是的，我开始也很不解，但是在营销人员解释之后就明白了。我们想使订购和获取报价单尽可能紧耦合，作为让用户最大可能完成订购的方法。如果我们让获得报价单很简单而订购却很难，那么顾客就可能在提交之前再次对订购进行考虑。

团队成员 1：哦，好的——我明白了。他们一旦有了报价单，那么订购的步骤就应当是快速而简单的！我们确实还没有在用例中清晰地捕获这些。我猜这意味着我们应当对用例和用例图进行修改，来更清晰地体现这一点。

[然后他们开始继续工作，对用例进行修改。]

7.4 内容模型

使用在沟通活动中捕获的信息和在分析中进行的精化，WebE 团队现在应当对用户以及描述用户与 WebApp 交互的用例有了清晰的理解。下一步就是开发内容模型（content model）——清晰地表示支持一个使用场景所需要的内容。

7.4.1 内容模型的结构元素有哪些

内容模型包含提供对一个 WebApp 内容需求重要视角的结构元素。这些结构元素包括作为 WebApp 的一部分进行展示的内容对象（例如，文本、图形图像、照片、视频图像、音频）。另外，内容模型包括所有的分析类（analysis class）——用户与 WebApp 交互时创建或者操作的用户可见的实体。分析类以如下方式之一出现：

- 外部实体（external entity）：提供或者使用 WebApp 所使用的信息（例如，其他系统、数据库、人员）。
- 事物（thing）：作为问题信息域的一部分（例如，报告、展览、视频图像）。
- 事件或活动（occurrence or event）：在用户与 WebApp 交互的情况下发生的（例如，一次报价或一次订购）。
- 角色（role）：与 WebApp 交互的人员所扮演（例如，零售买主、客户支持、售货员）。
- 组织单元（organizational unit）：与一个应用相关（例如，部门、组、团队）。
- 位置（place）：建立了问题背景和 WebApp 的总体功能（例如，制造现场、装货处）。
- 结构（structure）：定义对象类或相关对象类（例如，传感器、监控设备）。

一个分析类包括了对其进行描述的属性、影响它所需行为的操作，以及允许它和其他类进行通信的协作。

属性（attribute）定义一个分析类，阐明在问题空间的情况下，这个类表示什么。为了给一个分析类开发有意义的属性集，你应当对一个用例进行研究，选择那些合理地属于这个类的“事物”。而且应当为每个类回答这样的问题：在当前问题背景下，哪些数据项（组合的或基本的）对这个类进行了完全的定义？

操作（operation）定义了一个类的行为。虽然存在很多不同种类的操作，但是它们通常可以被划分为四大类：1）以某种方式（例如，添加、删除、格式修改、选择）控制内容的操作；2）执行计算的操作；3）查询对象状态的操作；4）对用于一个控制事件发生的对象进行监控的操作。操作是通过对属性进行控制来实现的，同时它必须对类的属性的性质有所“了解”。

一个类可以使用它自己的操作来控制其属性，由此完成一个特殊的职责。此外，一个类也可以和其他类进行协作来完成一个职责。*协作*表示类之间的关系。当类进行协作来满足某些需求时，它们可以在设计过程中组织成 WebApp 的一个架构元素（例如，一个子系统）。

和分析模型的其他元素一样，内容模型也是源于对为 WebApp 所开发的用例的仔细审查。对用例进行审查来提取内容对象和分析类。为了实现这一目的，Web 工程师可以对为将要创建的 WebApp 开发的用例执行“语法分析”［Pre05］。可以通过标出每个名词和名词短语并把它们放在一个简单的表格中来识别类。同义词应当记录。如果某个类对于实现一个方案来说是需要的，那么它就是方案空间的一部分；否则，如果一个类只是对于描述一个方案来说是必要的，那么它就是问题空间的一部分。

对于简单的 WebApp，你可以直接从用例跳到内容对象，然后继续到分析模型的其他方面。但是复杂些的 WebApp 就需要一个更结构化的方法来定义分析类。一种有用的方法是利用 Web 信息交换图（Web information exchange diagram）［Ton04］。

7.4.2 什么是信息交换，如何进行表示

用例（以及其他类似的记号，比如使用场景）描述用户和 WebApp 交互的方式，因此它们隐式地捕获由 WebApp 所赋予的不同用户之间的价值交换（exchange of value）。[⊖]例如，对 SafeHomeAssured. com 来说，第一个用例是了解公司及其产品。这表示用户和 SafeHomeAssured. com 之间的一次交互。这个交互会发生，是因为用户得到价值——关于公司的信息，这些信息可以帮助用户确定 SafeHomeAssured. com 是否值得信赖。SafeHomeAssured. com 反过来也得到有价值东西——来自于那个用户的未来的业务潜力。

理解在一个 WebApp 中所需内容的一种方法是去审视这些价值交换（当被用例捕获的时候），确定隐含的、支持价值交换所需要的信息交换。这是 Web 信息交换图（WIED）的基础。图 7-3 展示了一个 WIED 符号的子集。

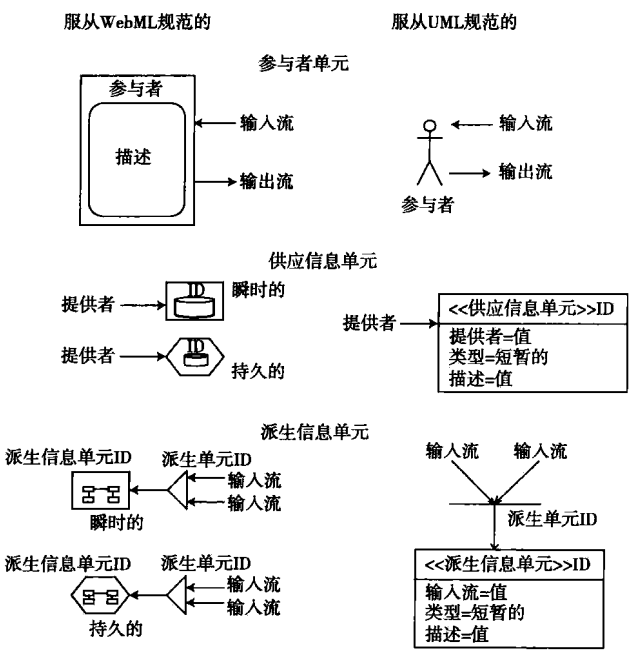


图 7-3 Web 信息交换图符号

来源：源自［Ton04］。

⊖ 很多业务建模方法中的一个关键的概念就是价值交换。对任何要发生的业务交互，所有参与者都必须有（可感知的）价值。通过对这些价值交换进行理解和建模，我们可以对业务进行改进。

应用如图 7-3 所示的符号，关于在 SafeHomeAssured. com 中使用的传感器位置信息的典型 WIED 如图 7-4 所示。在图中你会发现，对用例的分析不仅识别出了在每个交互中交换的信息，而且识别出了这些信息条目之间内在的相互依赖关系。这对于帮助识别应用中可能的约束来说是至关重要的。例如，获得对一个传感器布局的推荐需要一些信息，而这些信息只能通过对一个布局先前的开发得到。这就说明了这两个特殊用例之间的紧耦合。类似地，基于这个图表的讨论可能识别出了用户可以指定将要被监视的传感器，即使建筑平面图还没有录入，但是第一个版本仅仅允许显示作为建筑平面图布局一部分的注册传感器。对这个图表所做的后续修补，可能是手工绘制添加的，建议提供一种机制，即使他们没有提供一个建筑平面图，也为用户提供一系列他们的注册传感器。

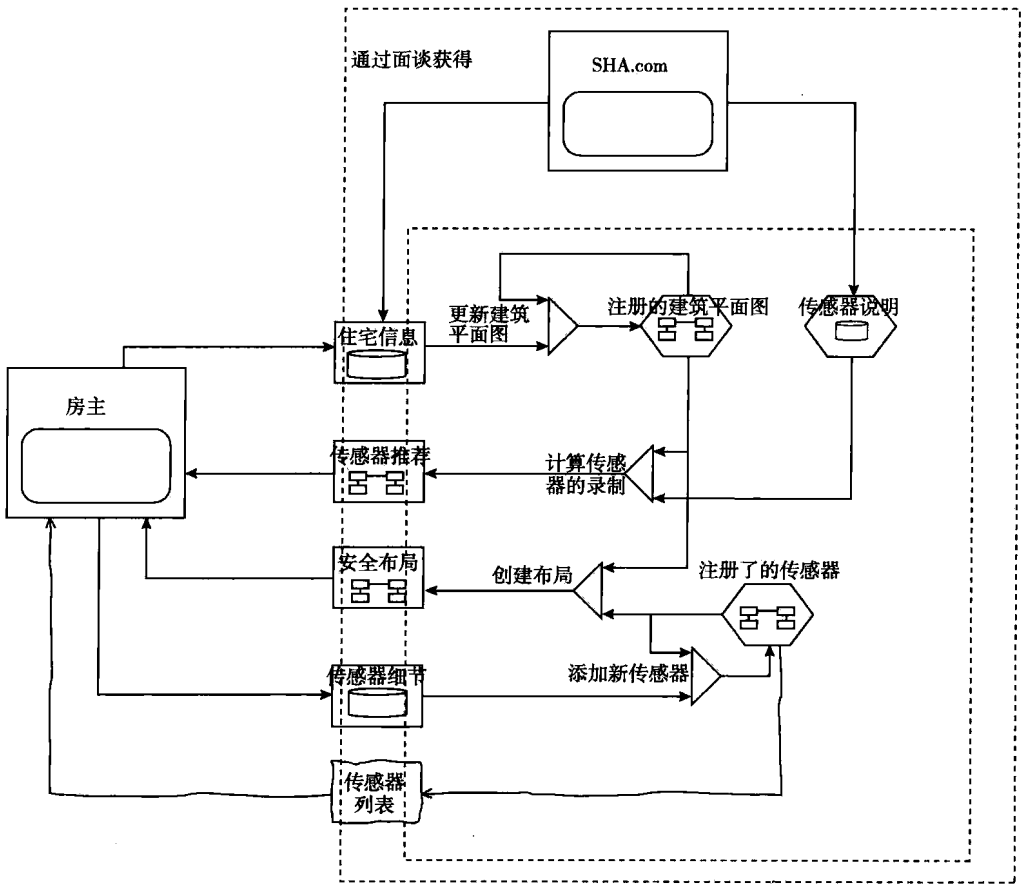


图 7-4 SafeHomeAssured. com 的部分信息交换图

7.4.3 如何确定内容对象

内容可以在 WebApp 实现之前、WebApp 创建时或者在 WebApp 投入运行很久之后进行开发。在每种情况下，内容都通过导航链接合并到 WebApp 的总体结构之中。内容对象可能是对产品的文本描述、描述了新闻事件的一篇文章、在运动中拍摄的一张动作照片、用户在一个论坛上的答复、公司徽标的一个动画展示、一个演说的简短视频，或者演示文稿幻灯片集合的音频插曲。内容对象可能的储存方式为单独的文件，直接嵌入到 Web 页面中，或者从数据库信息中动态生成。换言之，一个内容对象是将要展示给最终用户的内聚信息的任何项。

内容对象或者出来自于Web信息交换图的信息单元确定，或者（在简单的应用中，不需要WIED模型）通过检查内容直接或间接引用的场景描述，直接从用例中抽取得到。例如，在选择SafeHome组件用例中，我们碰到这样的句子：

我将能够获得每个产品组件的描述信息和定价信息

同样的信息也展示在WIED模型中。虽然没有到内容的直接引用，但是它是隐含的。你可以会见用例的作者，然后获得对“描述信息和定价信息”含义的更详细的理解。在这种情况下，用例的作者可能指出“描述性信息”包括：1）对组件的一段概要描述；2）组件的一张照片；3）对组件的多段技术性描述；4）组件的一个示意图，展示了它如何适应一个典型SafeHome系统；5）一个缩略视频，展示了如何在一个典型的家庭环境中安装组件。

值得注意的是，必须开发（常常由内容开发者，而非Web工程师来完成）或获取每一个内容对象，并把它们集成到WebApp架构中（在第9章讨论）。



定义内容对象

场景：增量2（详细的产品信息和下载）的分析建模开始之前，WebE团队区域

参与者：WebE团队作为结对编程中一对的两个成员

会话：

团队成员1：那么下一步就是定义内容，对吗？

团队成员2：是的，我们将检查和这个增量相关的所有用例，选择合适的内容对象，如果必要的话，也许会创建一个模型，同时……

团队成员1（打断）：有两个问题。第一，我不清楚如何选择内容对象；第二，我认为我们已经同意把建模保持在最低限度。

团队成员2：选择对象是需要一点儿技巧的，但是一种真正有效的方法就是做一个语法分析。例如，看一看来自于这个增量的一个用例摘录：“我希望能够得到任何SafeHome组件的产品信息。对SafeHome一个组件的描述包括所有的产品识别数据和描述性信息。”

团队成员1：为什么要用下划线呢？

团队成员2：它给出了内容对象，但是这些还不够。我们需要知道“产品识别数据”和“描述性信息”是什么——组件元素是什么。所以“画一幅图”，我已经和营销部门谈过了。这是我们提出的想法［和图7-5类似的一个示意图］。

团队成员1：我明白了。有关SafeHome一个组件的信息包括一个描述，以及你画出的其他条目……类似某种层次。

团队成员2：那将使我们开始工作。然后我们需要弄明白为了控制内容对象，需要哪些操作。例如，我们将要让所有的视频对象动起来并进行显示，同时……

团队成员1（打断）：我认为你是说我们将不得不对这东西进行建模。

团队成员2（笑）：我们刚刚做了。

团队成员1：嗯？

团队成员2（笑）：这张图［指着图7-5］称为一棵数据树，而且它是一种对内容对象层次进行建模的合理方法。

团队成员1：那很简单。

团队成员2：建模不一定非得是什么大不了的。

7.4.4 有描述内容之间的联系和层次的简单方法吗

在很多情况下，内容对象的一个简单列表加上每个对象的简短描述，对于定义那些必须被设计

和实现的内容需求来说，已经足够了。但是，在有些情况下，内容模型可能会从更丰富的分析中获益，比如实体关系图（entity relationship diagram）[Bag04]或数据树（data tree）[Sri01]。这些图形表示描述了内容对象之间的关系或一个 WebApp 维护的内容层次。

考虑图 7-5 中为 SafeHomeAssured. com 的一个组件所创建的数据树。该树体现了用于描述一个组件（在后面我们将会看到，一个 SafeHomeAssured. com 的组件实际上是这个应用的一个分析类）的信息层次。不带阴影的矩形表示简单或复合数据项（一个或多个数据值）。带有阴影的矩形表示内容对象。在图中，描述（description）由五个内容对象（有阴影的矩形）进行定义。在有些情况下，随着数据树的扩展，这些对象中的一个或多个将会被进一步精化。

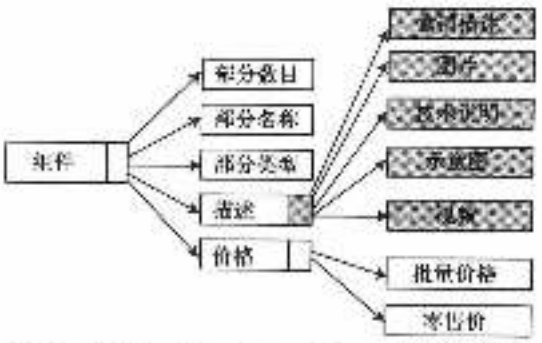


图 7-5 SafeHomeAssured. com 的一个组件的数据树

你可以为任何由多个内容对象和数据项组成的内容创建一棵数据树。开发数据树的目的是为了定义内容对象之间的层次关系，同时提供一种对内容进行复审的方法，通过复审，可以在设计开始之前发现疏漏和不一致。此外，数据树可以作为内容设计的基础。

7.4.5 如何为 WebApp 选择和表示分析类

正如我们已经注意到的，分析类是在用户和 WebApp 交互的时候，创建或者控制的用户可见的实体，而内容对象则表示了在 WebApp 内部所使用的原始信息或者媒体。

同时，我们也已经注意到，一个分析类包括描述它自身的属性、影响这个类需要的行为的操作，以及允许这个类和其他类通信的协作。分析类可以通过对每个用例进行检查来得到。为了加以说明，考虑在第 4 章中引入的用例“通过因特网访问摄像机监控”。

对用例进行快速语法分析，可以识别出 3 个候选类：HomeOwner（房主）、Camera（摄像机）和 House（住宅）。对每个类的初步描述如图 7-6 所示。

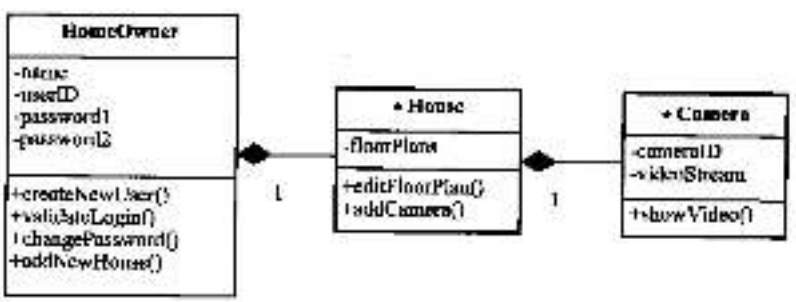


图 7-6 用例“通过因特网访问摄像机监控”的分析类示例

HomeOwner 类包含系统中有关 HomeOwner 用户所持有的信息，这些信息包括姓名、用户 ID

和密码。一个或者多个 HomeOwner 对象还可以连接到零个或多个包含有建筑平面图的 House 对象。一个建筑平面图反过来也可以与 Camera 对象关联。这些类的一些属性是简单或复合的数据项，其他的则是内容对象。图中同时还展示了与这个类相关的操作。

为 SafeHomeAssured.com 所识别的每个用例都要进行语法分析以识别分析类。为每个用例开发类似本节所描述的类模板。



监控用例

用例：通过因特网访问摄像机监控

主要参与者：房主

相关目标：通过因特网从任何远程地点观看布置在住宅中的摄像机的输出

前置条件：系统必须进行了完全的配置；必须获得了合适的用户 ID 和密码。

触发器：不在住宅的时候，房主决定看一看住宅的内部情况。

场景：

1. 房主登录到 SafeHome 的产品网站。
2. 房主输入他或她的用户 ID。
3. 房主输入两级密码（每个的长度都至少是 8 个字符）。
4. 系统显示所有主要功能的按钮。
5. 房主从主要功能按钮中选择“监控”。
6. 房主选择“选择一个摄像机”。
7. 系统显示住宅的建筑平面图。
8. 房主从建筑平面图中选择一个摄像机图标。
9. 房主选择“观看”按钮。
10. 系统显示一个与摄像机 ID 所识别的观看窗口。
11. 系统以每秒一帧的速度在观看窗口中显示视频输出。

7.5 交互模型

绝大多数 WebApp 都能够使最终用户与应用的功能、内容和行为“会话”。这种会话可以用交互模型来进行描述，这种交互模型由如下一种或多种元素组成：1）用例；2）序列图；3）状态图[1]；4）用户界面原型。除了这些表示之外，交互还可以用导航模型（将在第 7.8 节进行描述）进行表示。

7.5.1 用例从什么地方开始起作用

用例是 WebApp 交互模型中处于支配地位的元素。当分析、设计和构建大型复杂的 WebApp 时，描述 100 个甚至更多的用例是很平常的。但是，这些用例中只有少数一部分描述了最终用户种类（参与者）和系统之间的主要交互。而其他用例则对交互进行精化，提供指导设计和构建所需的必要的分析细节。

在很多情况下，一组用例就足以在分析级别描述交互（更多的精化和细节将在设计时引入）。然而，当交互序列很复杂，涉及多个分析类或很多任务时，有时值得用一种更严谨的图表形式来进行描述。

7.5.2 什么是序列图，何时进行开发

UML 序列图（sequence diagram）提供了用户动作（由用例定义的系统动态元素）与分析类

[1] 序列图和状态图使用 UML 的记号来进行描述。

（系统的结构元素）之间的协作行为的速记表示。因为分析类是从用例的描述中抽取出来的，所以有必要确保已经定义的类和描述系统交互的用例之间的可溯源性。Conallen［Con00］写道："在模型的可溯源性中，［分析］模型的动态元素和结构元素的合并是关键链接，应当高度重视。"

"通过因特网访问摄像机监控"用例的序列图如图 7-7 所示。图的垂直轴描述了用例内定义的动作。水平轴确定了在用例前进时所使用的分析类。例如，一个用户登录到 WebApp，定位到相关选项，并且选择了一个摄像机。然后就会显示那个摄像机的视频。序列图的横向和纵向移动把每一个分析类都绑定到用例动作上。如果一个用例动作从图中丢失了，那么你应当对分析类的描述进行重新评估，判断是否有遗漏的类。可以为每个用例以及为这个用例所定义的分析类创建序列图。

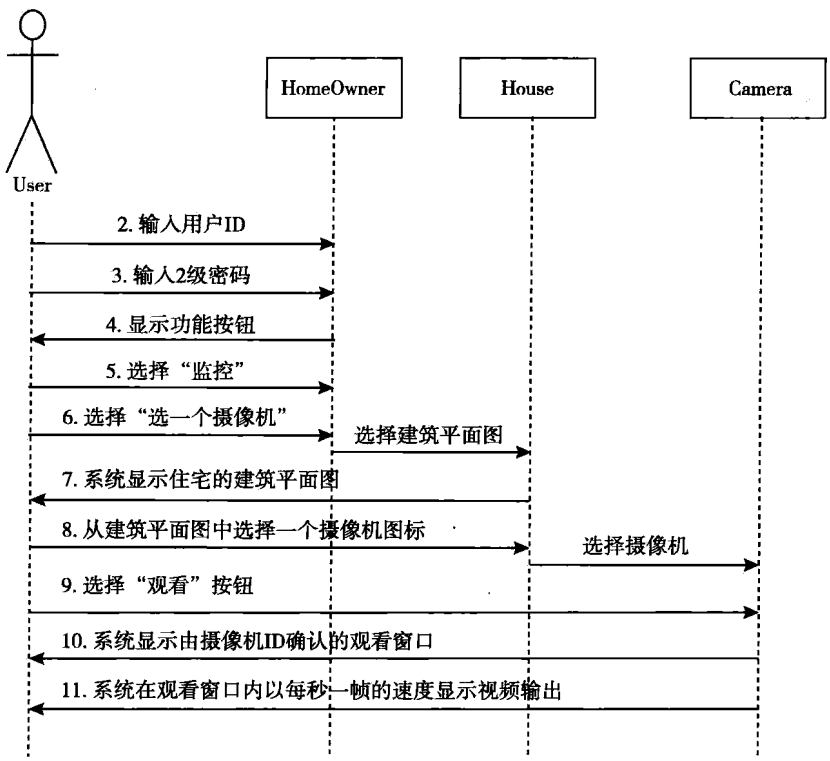


图 7-7 用例"通过因特网访问摄像机监控"的序列图

7.5.3 状态图如何描述 WebApp 的行为

UML 状态图（state diagram）提供了当一个交互发生时，WebApp 动态行为的另一种表示法。和在 Web 工程中使用的绝大多数建模表示法一样，状态图也可以表示不同的抽象层次上。状态图在一个用户交互触发 WebApp 的状态变化——由此改变它对用户的响应方式的时候是最有用的。

当用户观察到 WebApp 行为的一个新模式时，一个状态变化就发生了。例如，当一个正在浏览产品描述的用户请求产品报价单的时候，这个请求就触发了从产品综览状态到产品报价状态的状态转移。用户观察到的 WebApp 观感也有可能因此而发生变化。

这些状态转移常常会包含不同用例之间的互连。例如，"通过因特网访问摄像机监控"用例包括从一个建筑平面图选择一个摄像机——但是这只有在用户已经配置了一个建筑平面图（或

者配置 SafeHomeAssured. com）时才有可能。换句话说，HomeOwner 类（参见图 7-6）可能有也可能没有一个合法的 House，同时当一个 House 确实存在的时候，它可能有，也可能没有一个合法的 FloorPlan（建筑平面图）。

如图 7-8 所示为顶层（抽象层次高）状态图的一部分，描述了顾客和 SafeHomeAssured. com WebApp 之间的交互。这个状态图表明，WebApp 对“通过因特网访问摄像机监控”用例内各种事件的响应依赖于应用的状态，尤其是一所住宅对一个顾客是否可用以及是否有一个合法的建筑平面图。状态图指明了使新顾客从一种状态转移到另一种状态所需要的事件、当进入状态时所显示的信息、在一个状态内发生的处理，以及导致一种状态到另一种状态转移的结束条件。

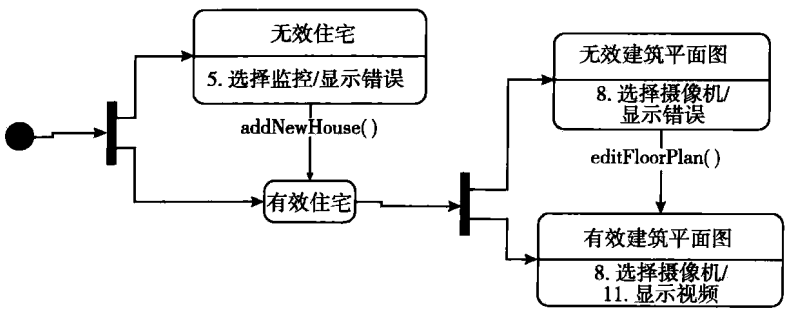


图 7-8 构建住宅平面图的部分状态图

7.5.4 是否真的需要用例、序列图和状态图

因为用例、序列图和状态图都表示相关的信息，所以我们有理由问一问为什么它们都是必要的。在某些情况下，它们并不是都必要的。在某些情况下，有用例就可能足够了，特别是当你已经彻底记录下了异常情况的时候。然而，用例只提供了交互的一维视图。序列图表示了在本质上更过程化（动态）的第二维视图。状态图提供了更侧重于行为的第三维视图，同时它包含了用例或序列图所不能提供的潜在的导航路径信息。

当三种维度的视图同时使用的时候，在一个维度上发现不了的遗漏或者不一致将在检查第二维（或第三维）时变得很明显。正是这个原因，大型复杂的 WebApp 可以从包含了所有这三种表示的交互模型中受益。

决定用例、序列图和状态图是否都必要的关键因素包括总体增量的复杂性、对用户来说一个错误带来的后果（想象一下一个未授权用户获得了对你住宅内视频摄像机的访问的后果），以及在后面的 WebE 过程中改正一个错误时可能遇到的困难。

7.5.5 为什么创建一个界面原型是很明智的

用户界面的布局、展示的内容、实现的交互机制以及用户与 WebApp 连接上的总体美学与用户的满意度和 WebApp 的总体可接受性密切相关。虽然可以认为用户界面原型的创建是一种设计活动，但是在分析模型的创建期间就创建界面原型是一个好方法。用户界面的物理表示检查得越早，越有可能满足最终用户的需求。用户界面的设计将在第 9 章进行详细讨论。

因为 WebApp 的构建工具种类繁多，而且相对便宜、功能强大，所以最好使用这些工具来创建界面原型。原型应该实现主要的导航链接，并采用构建应用的方式来表现总体的屏幕布局。例如，将提供给最终用户 5 个主要的系统功能，原型应当展现它们，如同用户第一次进入 WebApp 时所看到的那样。它将提供图示链接吗？将显示一个左侧（或者右侧）菜单吗？用户将看到哪

些其他的信息？类似这样的问题都应当由原型进行回答。图 7-9 显示了 SafeHomeAssured.com 主页的一个早期原型的截图。这个原型将 SafeHome 的布局和视频监控能力作为销售工具进行使用，同时为访问者提供了到其他重要的 WebApp 专题的链接。原型可能不会激活这些链接，但是提供它们可以给检查者一种对导航将被启动的方式的感觉。值得注意的是，在最终界面设计开发之前，这个原型会（而且常常是）不断地发生变化。

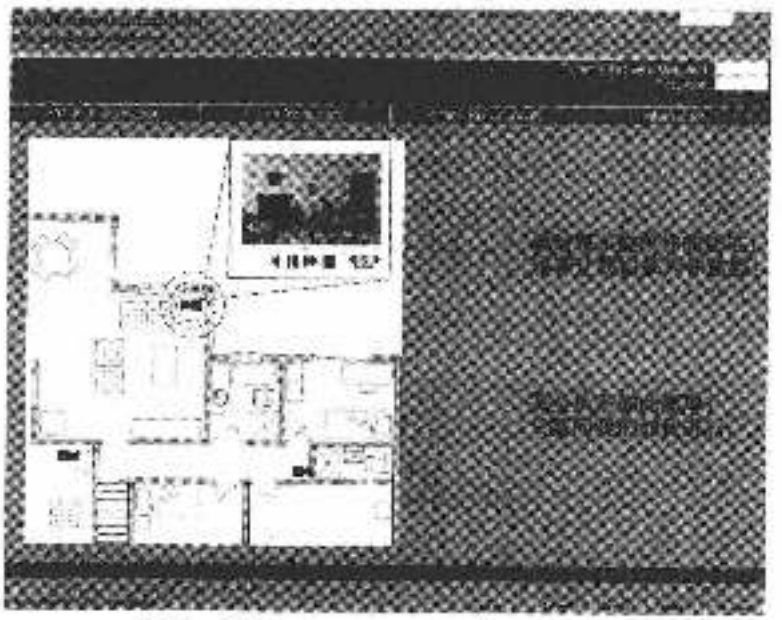


图 7-9 SafeHomeAssured.com 的主页面截图示例

7.6 功能模型

很多 WebApp 交付一系列计算和控制功能，这些功能可以直接和内容发生关联（使用或者生成内容），而这常常是用户与 WebApp 交互的主要目的。因此，除了内容和交互，WebE 团队还应该对功能进行分析。

功能模型（functional model）描述 WebApp 的两个处理元素，每一个处理元素代表过程上不同的抽象层次：1）WebApp 交付给最终用户的用户可观察到的功能；2）分析类内部包含的操作，这些操作实现与类相关的行为。

用户可观察到的功能包括任何直接由用户启动的处理功能。例如，一个财务 WebApp 可以实现多种功能（例如，一个大学学费计算器或者一个退休金计算器）。这些功能实际上可能要使用分析类中的操作来完成，但是从最终用户的角度来看，这些功能（更准确地说，是这些功能提供的数据）是可见的结果。

在过程抽象的更低层次，分析模型描述了由分析类的操作执行的处理。这些操作操纵类的属性，并参与类之间的协作来完成一些所要的行为。

不管过程的抽象层次如何，UML 的活动图可以用于表示处理细节。在分析层次，具有适当功能相对复杂的时候才使用活动图。WebApp 的很多复杂性并非在提供的功能之中，而是在可访问信息的本质和操纵信息的方式。

SafeHomeAssured.com 有两个很好的复杂功能的例子，它们得益于 WebE 团队的认真分析。第一个是用例“获得我的空间的传感器布局的推荐”。用户已经为将要监控的空间开发了布局，然后在

这个用例中选择此布局并请求那个布局内传感器的推荐位置。SafeHomeAssured.com 的响应是带有额外的传感器推荐位置信息布局的图形表示。这个交互非常简单；内容相当复杂一些，而潜在的功能非常复杂。系统必须对楼层平面布置图做一个相当复杂的分析，来决定最佳的传感器集合，必须检查房间的尺寸、门和窗的位置，并把这些和传感器的功能和说明书相协调。重要的任务！

第二个例子是用例"控制摄像机"。在这个用例中，交互相对简单，但是考虑到这个"简单"操作需要和远程设备进行复杂的通信，并且可以通过因特网进行访问，那么就有潜在的复杂功能。当多个授权人员试图同时监控或控制同一个传感器时，进一步可能的复杂性和控制的协商相关。

图 7-10 描述了操作 takeControlOfCamera()（控制摄像机）的活动图。这个操作是"控制摄像机"用例中分析类 Camera 的一部分。活动图类似于流程图，说明处理流及其内部的逻辑判定。应当注意到，在过程流内部调用了另外两个操作：requestCameraLock()（请求摄像机加锁）试图为这个用户锁住摄像机，和 getCurrentCameraUser()（获取当前摄像机的用户）获取当前正在控制摄像机的用户的名字。传递细节指出这些操作是如何被调用的，每个操作的接口细节直到 WebApp 设计开始时才考虑。

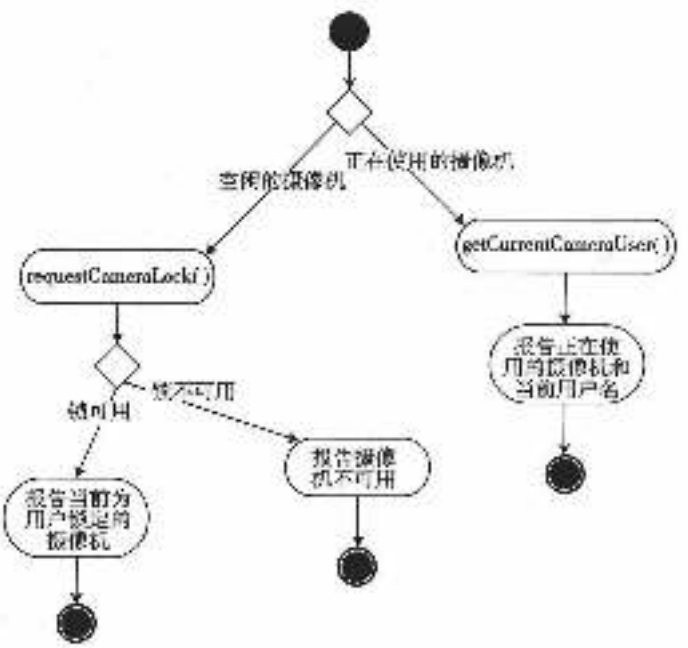


图 7-10 takeControlOfCamera()操作的活动图

7.7 配置模型

WebApp 的设计和实现方式都必须能够适应服务器端和客户端的多种环境[1]。WebApp 可以驻留在一个提供通过因特网、内联网或者外联网进行访问的服务器上。必须详细说明服务器的硬件和操作系统环境。另外，还必须考虑服务器端的互操作性问题。如果 WebApp 必须访问一个大的数据库，或者与公司的 IT 应用进行互操作，那么必须详细说明合适的接口、通信协议和相关的协作信息。

[1] 服务器则是 WebApp 和所有相关的系统特性的宿主机，这些特性被多个用户可以通过网络访问 WebApp。客户端提供了一个软件环境（例如，浏览器），此环境被终端用户能够在他们的桌面上与 WebApp 交互。

客户端软件提供了基础结构，它使得用户从所在的位置访问 WebApp。一般来说，浏览器软件用于显示从服务器下载的 WebApp 的内容和功能。尽管有标准，但是每个浏览器都有它自己的特性。因此，WebApp 必须针对每一种浏览器配置进行彻底的测试，这些配置作为配置模型的一部分进行详细说明。例如，一些内联网 WebApp 可能会假设使用一个特定的客户端，但是大多数公开访问的 WebApp 必须适应大范围的客户端。

在有些情况下，配置模型只不过是服务器端和客户端的属性列表。但是，对于更复杂的 WebApp，多种配置的复杂性（例如，在多个服务器之间分配负载、高速缓存的架构、远程数据库、同一个 Web 页面服务于不同对象的多个服务器）可能会对分析和设计产生影响。在必须考虑复杂配置架构的情况下，可以使用 UML 部署图（deployment diagram）。

对 SafeHomeAssured. com 而言，应当详细说明公共的内容和功能，使得可以通过所有主要的 Web 客户端（例如，那些市场份额大于等于 0.1 个百分点的[⊖]）来访问。相反，把更复杂的控制和监控功能（只有 HomeOwner 用户可以访问）限制在较小的客户端集合也是可以接受的。SafeHomeAssured. com 的配置模型同时将详细说明和已有的产品数据库和监控应用之间的互操作。

7.8 关系 - 导航分析

在 Web 应用中，每个架构元素都有可能被链接到所有其他架构元素。实际上，链接机制和因此能够进行的导航提供了 WebApp 内一种关键的访问工具。但是随着内容数量和用户交互复杂性的增加，链接的数目以及由此带来的 WebApp 内的导航复杂性也都增加了。因而，问题就是如何建立内容对象与提供用户所需能力的功能之间合适的链接。

关系 - 导航分析（relationship-navigation analysis，RNA）作为分析模型创建的一部分，提供了一系列用于确定已经发现的元素之间的关系的分析步骤[⊖]。Yoo 和 Bieber［Yoo01］以如下的方式对 RNA 进行了描述：

> *RNA 为系统分析员提供了一种系统技术，可以用于确定一个应用系统的关系结构，帮助他们发现应用领域内所有潜在的有用关系。这些关系在后续可以用链接来实现。RNA 也在这些链接上帮助确定合适的导航结构。RNA 通过扩大和深化领域的概念模型，来增强系统开发者对应用领域的理解。因此，开发者可以通过加入额外的链接、元信息和导航来改进实现。*

RNA 方法被组织为五个步骤：

1. **利益相关者分析**。确定多个用户种类，并建立适当的利益相关者层次。
2. **元素分析**。确定内容对象和最终用户感兴趣的功能元素。
3. **关系分析**。描述存在于 WebApp 元素之间的关系。
4. **导航分析**。检查用户如何访问单个元素或者元素组。
5. **评估分析**。考虑与实现前面确定的关系相关的实际问题（例如，成本效益）。

RNA 方法中的前两个步骤已经在本章前面进行了讨论。在下面的部分，我们将考虑内容对象和功能之间关系的建立方法。

⊖ 确定浏览器的市场份额是众所周知的难题，而且不同的调查，其结果也有很大的差异。虽然如此，在写作本书的时候，据报道，Internet Explorer 和 Firefox 是仅有的两个市场份额超过 10% 的浏览器，而 Netscape、Mozilla、Opera 和 Safari 是其他仅有的市场份额一直超过 0.1% 的浏览器。

⊖ 应当注意，RNA 可以应用到任何信息系统，它起初是为超媒体系统开发的，然而，它也可以很好地适用于 Web 工程。

7.8.1 如何建立内容对象和功能之间的关系

Yoo 和 Bieber［Yoo01］提出了一个问题列表来帮助你评估在分析模型中确定的每个元素（内容对象或者功能）之间的关系。下面是一个适用于 WebApp 的有代表性的列表［Yoo01］：

- 这个元素属于更广泛的元素种类吗？
- 已经为这个元素确定了哪些属性或参数？
- 有关这个元素的描述性信息是否已经存在？如果存在，这些信息在哪里？
- 在 WebApp 中，这个元素在不同位置出现吗？如果是，在哪些位置？
- 这个元素是由其他更小的元素组成吗？如果是，这些更小的元素是什么？
- 这个元素是一个更大的元素集合中的成员吗？如果是，这个更大的元素集合是什么，它的结构是什么样？
- 这个元素由一个分析类进行描述了吗？
- 其他元素与正在被考虑的元素相似吗？如果相似，可能将它们合并为一个元素吗？
- 这个元素在其他元素的特定顺序上进行使用吗？它的出现依赖于其他元素吗？
- 有另外一个元素总是在被考虑的元素出现之后就出现吗？
- 将要使用的元素必须满足的前置条件和后置条件是什么？
- 有特殊的用户种类使用这个元素吗？不同的用户种类对元素的使用方式也不同吗？如果是，如何不同？
- 这个元素能和一个具体的、明确表达的目标或目的联系在一起吗？还能和一个特定的 WebApp 需求联系在一起吗？
- 这个元素会和其他元素同时出现吗？如果会，是哪些其他元素？
- 这个元素总是和其他元素出现在同一个地方（例如，屏幕或页的相同位置）吗？如果是，是哪些其他元素？

对这些问题和其他问题的回答有助于你定位 WebApp 内不确定的元素，并建立这些元素之间的关系。针对提到的每一个显著问题，有可能建议一种关系分类体系，把所识别的每个关系进行归类。如果你感兴趣，可以参考［Yoo01］以了解更多的细节。

7.8.2 如何分析导航需求

一旦建立了分析模型内定义的元素之间的关系，你就应当考虑那些规定了一个用户种类如何从一个元素（例如，内容对象）导航到另一个元素的需求。导航机制定义为设计的一部分。在这个阶段，你必须考虑总体的导航需求。应当解决如下的问题：

- 某些元素比其他元素更容易到达（需要更少的导航步骤）吗？表示的优先级是什么？
- 为了迫使用户在自己的方向上进行导航，应该强调某些元素吗？
- 应当如何处理导航错误？
- 导航到相关的元素组的优先级应该比导航到某个特定元素的优先级高吗？
- 应该通过链接、基于搜索的访问还是其他方式来实现导航？
- 根据前面的导航动作，某些确定的元素应该展示给用户吗？
- 应该为用户维护导航日志吗？
- 在一次用户交互中，一个完整的导航地图或菜单（与单个“返回”链接或者定向指针相比）到处可用吗？
- 导航设计应该由大多数普遍期望的用户行为来驱动，还是由所定义的 WebApp 元素可感

知的重要性来驱动？

- 为了加快将来的使用，用户是否能够“存储”他们以前对 WebApp 的导航？
- 应该为哪类用户设计最佳导航？
- 应该如何处理 WebApp 的外部链接？应该覆盖现有的浏览器窗口，作为一个新的浏览器窗口，还是作为一个单独的框架？

提出和回答这些问题以及其他一些问题，是导航分析的一部分。

Web 工程团队及其利益相关者必须还要决定导航的总体需求。例如，提供的“站点地图”会让用户全面综览整个 WebApp 的结构吗？用户会使用将突出显示可用的最重要的元素（内容对象和功能）的“导游”吗？用户能访问内容对象或由这些元素已定义的属性所决定的功能（例如，一个用户可能想要访问一个特定建筑的所有图片，或者允许计算重量的所有功能）吗？

关系 - 导航分析带来了导航的一系列需求。从这些需求中，有可能开发一个信息架构——高层导航设计的第一步（第 10 章）。

为了阐明这一点，考虑下面的问题如何应用于 SafeHomeAssured. com：应当为哪类用户设计最佳导航？如图 7-1 所示，有三类主要的用户——游客、注册用户和房主。考虑原型主页面（参见图 7-9），可以看出导航选项隐含地对所有这些用户都是可用的。SafeHomeAssured. com 的系统架构师和 CPI 的营销团队之间的一个快速讨论指出应当优化导航的支持来吸引新顾客（例如，用户种类游客的导航需求应当优先考虑）。考虑这个导航需求，目前主页面没有为游客提供一个清晰的途径来判断如何在 SafeHomeAssured. com 上注册。

7.9 本章小结

分析建模的目的是创建沟通活动中获得的信息，精化和建模 WebApp 的需求，从而建立起对将要构建的 WebApp 本质的更深入的理解。此外，分析建模把需求转变为能够很自然地通向设计的形式。

分析建模精化了 WebE 团队对用户及其执行的任务的理解，而使用场景（用例）为捕获此理解提供了关键的机制，同时也是后续所有需求分析和建模活动的催化剂。

创建一个完整的分析模型包含四个分析任务：1）内容分析识别 WebApp 将要提供的全系列内容；2）交互分析描述用户与 WebApp 交互的方式；3）功能分析定义将应用于 WebApp 内容的操作，并描述其他独立于内容但对于最终用户来说是必要的处理功能；4）配置分析描述 WebApp 所驻留的环境和基础结构。最后，关系 - 导航分析确定在分析模型中识别的内容和功能元素之间的关系，同时为定义系统中合适的导航链接建立需求。一系列的问题有助于建立关系和识别那些将对导航设计产生影响的特征。

分析建模是 Web 工程过程中的第一个技术性活动。通过精化、改进和结构化 WebE 团队对需求的理解，为设计建立了一个坚实的基础。

在后面几章中，我们将深入研究设计——Web 工程的技术核心。我们将从第 8 章总体系统和架构的设计问题开始。

参考文献

[Baq04] Baqui, S., *Database Design Using Entity Relationship Diagrams*, Auerbach, 2004.

[Con00] Conallen, J., *Building Web Applications with UML*, Addison-Wesley, 2000.

[Fra01] Franklin, S., “Planning Your Web Site with UML,” webReview, 2001, www. ddj. com/documents/s = 2659/nam1012432061/index. html (accessed July 30, 2007).

[Fra03] Franklin, S., "Applying Rational tools to a simple J2EE-based project Part 9: Wrapping up development and testing," December 4, 2003, http://www.ibm.com/developerworks/rational/library/337.html (accessed October 11, 07).

[Pre05] Pressman, R. S., *Software Engineering: A practitioner's Approach*, 6th ed., McGraw-Hill, 2005.

[Sri01] Sridhar, M., and N. Mandyam, "Effective Use of Data Models in Building Web Applications," 2001, www.2002.org/CDROM/alternate/698/ (accessed July 30, 2007).

[Ton04] Tongrungrojana, R., and D. Lowe, "WIED: A Web Modelling Language for Modelling Architectural-Level Information Flows," *Journal of Digital Information*, vol. 5, no. 2, 2004, http://jodi.tamu.edu/Articles/v05/i02/Tongrungrojana/ (accessed July 30, 2007).

[Yoo01] Yoo, J., and M. Bieber, "A Systematic Relationship Analysis for Modeling Information Domains," in M. Rossi and K. Siau (eds.), *Information Modeling in the New Millennium*, Idea Group, 2001, http://web.njit.edu/~bieber/pub/millenium-v3.pdf (accessed July 30, 2007).

第8章　WebApp的设计

到了Web工程过程的这个阶段，你已经通过努力确保了自己对将要创建的WebApp有了一些清晰的理解。但是你却花了很短的时间考虑如何构建它。在本章和第9章～第11章，我们将考虑设计活动，这些活动将为WebApp本身和将要交付给最终用户的每个增量的构建创造条件。

作为分析建模（第7章）的结果，你已经确定了谁将使用WebApp，用户将完成哪些任务，需要哪些功能来支持这些任务，必须交换哪些信息以及支持这些信息的内容。最后，你确定了把这些因素联系在一起的交互特性。

在本章和后续几章，我们将考虑把对WebApp应当做什么的理解转化为WebApp应当是什么的表示的一些概念、原则和方法。换句话说，我们准备从做什么（what）转移到如何做（how），从WebApp分析（WebApp analysis）转移到WebApp设计（WebApp design）。

8.1　对WebApp进行设计

Jakob Nielsen［Nie00］在他的关于Web设计的权威著作中写道："本质上设计有两种基本途径：表达自己的艺术设想和为客户解决问题的工程设想。"在Web发展的前10年，艺术设想是很多开发者选择的方式。它以一种特别的方式进行设计，并且经常是在生成HTML时才进行设计。设计从艺术想象中发展而来，而艺术想象本身又随着WebApp构建而发展。

即使在今天，敏捷软件开发（第2章）的一些"极端"拥护者仍然使用WebApp宣传基于"有限设计"的应用开发。他们认为WebApp的直接性和易变性削弱了形式化设计，即设计随着对应用的构建（编码）而进化，并且应当花费相对少的时间创建详细设计模型。这种观点有其优点，但是这仅仅适用于许多相对简单的WebApp——当一个单独的开发者可以设计架构并在头脑中保持一个详细设计图的同时创建WebApp的时候。当功能和内容都很复杂的时候；当WebApp的规模包含好几百个的内容对象、功能和分析类的时候；当多名人员参与到设计中的时候；当WebApp的成功对于业务的成功具有直接影响的时候，设计就不应该被轻视了。我们必须要小心，不要把有限设计（limited design）（除了过分简单的应用外都不适用）误认为是轻量级设计（lightweight design）。后一种进行了很认真的设计，不过可能与正在逐渐开发的设计的某些方面和逐渐发展成实际WebApp的原型中记录下的结果一起进行。

这个事实使我们考虑Nielsen提出的第二种途径——"为客户解决实际问题的工程设想"。Web工程采用这种思想，并且一种更严格的WebApp设计方法使得开发人员能够实现这种设想。在每个实例中，应当在开始构建之前创建设计模型，但是优秀的Web工程师会认识到，在WebApp创建过程中，随着对利益相关者需求的了解增多，设计将会进行演化。因此，设计需要的详细程度很大程度上取决于需求可能的易变性。

创建有效设计通常需要多种技巧。有时，对于简单的项目，一个单独的开发者可能需要具备多种技能。对于大型项目来说，利用多个专家的技能是明智而可行的。这些专家包括Web工程师、图形设计人员、内容开发人员、程序员、数据库专家、信息架构师、网络工程师、安全专家和测试人员。利用这些不同技巧能够创建可以在内容和代码产生之前进行质量的评估和改进、对测试进行引导、有大量最终用户参与的模型。如果分析是建立WebApp质量的地方，那么设计就是质量真正嵌入的地方。

8.1.1 WebApp设计者需要知道什么

一个WebApp设计者应当很好地理解产生一个好的设计的那些特征——不管是通用的还是特定的WebApp。一般来说，一个好的设计应当指导WebApp的构建，并确保开发方案的简单性、一致性、相符性、健壮性和可用性的最大化。为了使设计成为构建的有效指导，它还应该是可理解的，并包含合适的精化程度。但是什么程度的精化才合适呢？就WebApp设计而言，我们需要足够的信息来使设计有用，但是不要多到成为设计过程的累赘。

就具体的WebApp而言，设计需要覆盖和此WebApp自身相关的那些方面。对某些应用来说，安全性根本就不是问题，设计可以有效地忽视这个方面（或者至少在构建期间对它进行非正式的处理）。对其他应用来说，安全性可能至关重要。这就意味着设计人员应当很好地理解设计所必须解决的那些安全性问题。

8.1.2 什么是逻辑设计

WebApp的设计包括逻辑设计和物理设计。*逻辑设计*（logical design）包括识别那些必须在WebApp中发生以满足利益相关者的需求的抽象事物，并符合分析模型（第7章）。例如，就SafeHomeAssured.com提供所需信息和功能而言，它可能需要包括一个安全监控子系统、一个订单管理子系统和一个顾客管理子系统等。在这个层次上，我们并不考虑实际的WebApp软件组件或者这些组件运行的硬件，而是考虑提供所需功能和内容的逻辑构件。

8.1.3 什么是物理设计

物理设计（physical design）把我们对逻辑设计的理解映射到将作为WebApp的一部分进行实现的实际物理元素——例如，Web服务器、数据库服务器、内容管理系统、处理脚本、Web页面和其他WebApp组件。在这个层次上，你要考虑总体的系统物理架构，然后深入详细的系统组件。

例如，SafeHomeAssured.com的第2个增量包括对可用产品的规格说明书的下载支持。在逻辑设计中，你开发了一个概念性架构，架构包括一个存储产品信息的产品信息仓库和一个支持访问这一内容的产品管理子系统[⊖]。在物理设计中，你决定如何实现这些组件。例如，你可以选择使用现有的内容管理软件[⊜]实现这些子系统。实际上，由于应用相对简单，由单个人员进行维护，同时内容是稳定的，没有持续不断的演变，而且已经存储在一个数据库中，所以SafeHomeAssured.com的开发团队决定使用现有的独立数据库作为产品信息的仓库，并使用手工编码ASP（Active Server Pages）脚本来访问这个产品管理的内容。

8.1.4 设计完成后生成哪些信息

所有设计任务的主要输出是一个*设计模型*，设计模型包含内容、美观、架构、界面、导航和组件级设计问题。设计模型为Web团队创建最终的WebApp（或者至少是正在设计中的那些WebApp的增量）提供了足够的信息。在设计过程中，应当考虑一些替代方案，同时要评估当前设计模型会带来一个有效实现的程度。

在本章中，我们将审查总体的设计过程以及如何确保逻辑设计和物理设计以及功能设计和信息设计之间的一致性。我们将研究会影响你设计方法的关键问题以及你应该如何解决这些问题。

⊖ 这些方面将在图8-4中所示的概念架构中进行更详细的展示。

⊜ 参考CMSMatrix网站（www.cmsmatrix.org/）中的典型软件，它们在实现产品管理子系统的过程中可能是合适的。

在第 9 章～第 13 章，我们将考虑具体的设计任务。第 9 章着眼于交互设计及其如何形成一个用于确保用户需求和 WebApp 设计之间连接的基础。第 10 章将研究支持交互的信息方面的设计。第 11 章考虑表示 WebApp 的功能设计所需要的系统功能和技术。在第 12 章中，我们从逻辑设计转到物理设计和构建，同时展示如何把逻辑元素映射到可以被实现的组件。最后，在第 13 章中，我们将考虑设计模式及其如何帮助设计的改进。

8.2 设计目标

在 Web 设计的定期专栏中，Jean Kaiser［Kai02］提出了如下一组设计目标，无论应用的领域、规模和复杂性如何，这些目标几乎可以适用于任何 WebApp。

简单性。虽然这看起来有些过时，但是“凡事适度”这句格言依然适用于 WebApp。一些设计者倾向于给最终用户提供太多的事物：详尽的内容、完美的视觉效果、插入的动画、大量的 Web 页面等。与功能膨胀相比，最好还是尽量做到适度和简单。为了感受这种说法的正确性，考虑使用最广泛的网站之一（www. google. com）的界面的简单性（虽然界面很简单，但是潜在的功能却复杂得难以置信）。

内容应当能够提供大量的信息，但同时应该简明扼要，并且应当使用一个对于正在交付的信息来说合适的交付模式（例如，文本、图形、视频、音频）。外观应该令人愉快，但是不要变得太夸张（例如，使用过多的颜色易于使用户分神，而不是改进交互），而且架构应当以可能的最简单的方式完成 WebApp 的目标。导航应当简单易懂，同时导航机制对于最终用户来说应当很直观。功能应当易于使用和理解。

一致性。这一设计目标几乎适用于设计模型的每个元素。内容的构建应当一致（例如，在所有文本文档中，文本格式和字体风格应该保持一致；而且图形应该有统一的外观、颜色配置和风格）。图形设计（美学方面）应该在 WebApp 的各个部分显示统一的外观[⊖]。架构设计应该建立起一个可以产生一致的超媒体导航结构的模板。界面设计应该定义一致的交互、导航和内容显示模式。应该在所有的 WebApp 元素中一致地使用导航机制。正如 Kaiser［Kai02］提到的，“记住，对一个访问者来说，一个 Web 网站就是一个物理空间，如果网站内的页面在设计上不一致，那么就会让人产生困惑。”

相符性。一个 WebApp 的美学、界面和导航设计必须要与将要创建的应用所处的领域保持一致。毫无疑问，一个嘻哈乐师（hip-hop musicians）的 Web 网站肯定和一个金融服务公司的 WebApp 有着不同的观感。WebApp 的架构会完全不同，界面会被构建成适合不同的用户种类，而且导航会被组织成完成不同的目标。Web 工程师（和其他的设计参与者）应该在设计过程中建立 WebApp 的相符性。

健壮性。在已经建立的相符性的基础上，WebApp 通常会给用户做出一个隐含的承诺。用户期望与他们的要求相关的内容和功能是健壮的。如果这些元素遗漏或者不足，那么这个 WebApp 很可能会失败。

导航性。我们已经在前面提到了导航应该简单而一致，也应该以一种对于用户来说是直观和可预测的方式进行设计。也就是说，用户不必搜索导航链接或说明就应该可以理解如何使用 WebApp。例如，如果一个页面包含一个图标或图像字段，其中一些可以作为导航机制来使用，这些必须以某种很明显的方式进行表示。没有什么比试图在很多图形图像中寻找合适的可用链接更令人沮丧了。

每个 Web 页面中，在一个可预测的地方实现到 WebApp 的主要内容和功能的链接也很重要。如果

⊖ Web 技术（如层叠式样式表（CSS））被加以设计，部分原因是用于支持这个目标——通过提供一个可以在整个网站内进行使用的样式定义，由此也就促进了一致性。

需要页面滚动（而且经常是这种情况），在页面顶端和底端的链接可以使用户的导航任务变得更简单。

视觉吸引。对所有的软件种类来说，WebApp 应用毫无疑问是最具有视觉效果的、最生动的、最具有审美感觉的。美丽的外观（视觉吸引）无疑是吸引观看者的眼球的，但是很多设计特性（例如，内容的观感、界面布局、颜色搭配、文本的平衡、图形和其他媒体、导航机制）确实对视觉吸引产生影响。

兼容性。绝大多数 WebApp 会在多种不同环境（例如，不同的硬件、Internet 连接类型、操作系统和浏览器）中进行使用，设计必须互相兼容。

实现了所有这些目标的设计将使最终用户满意，并为 WebE 团队带来声望。



建立设计准则

场景：就在开始第一个增量的设计之前，SafeHomeAssured. com 的 WebE 团队的会议室

参与者：WebE 团队负责人和所有的团队成员

会话：

团队成员 1（在大家到齐之前）：我不明白对于这个增量我们为什么需要标准。它仅仅是一个信息型网站。

团队成员 2：没错，但是其他的增量更复杂，而且除此之外，在开始设计和构建之前，我们最好都基于相同的页面。[停了下来，看着团队负责人走进来。]

团队负责人：我认为我们应当为将要在 SafeHomeAssured. com 上做的工作讨论总体的设计准则。[他分发给每个团队成员一份 6 页的文档。] 我已经对一位朋友给我的一组设计准则进行了修改——我想对它进行走查，获得你们的评论和建议，然后把它作为我们工作的约束准则。

团队成员 3（环顾桌子）：我们以前都做过 WebApp。我们真的需要用“标准”来限制我们吗？难道我们不是要成为自组织的团队吗？

团队负责人（微笑但很坚决）：第一，它们是指导准则，而不是严格的标准。第二，即使是自组织的团队也必须保持一致。这些准则将带给我们一致的设计方法。没有人会告诉你如何设计低层的东西，但是 SafeHomeAssured. com 的总体设计感觉应当是牢固的……而且是一致的。

[团队负责人对文档进行走查。于是产生大量的讨论，除了一些异议和反对方案，大家一致同意需要设计准则。然后继续进行导航设计准则的走查。]

团队负责人：好吧，下一个部分处理导航设计。[每个人都花了一些时间来阅读那个部分。]

团队成员 1：这就是说所有的主要功能都将显示在左手边，当选择一个主要功能的时候，有下拉的子功能。那几乎就是我们在分析中预想的界面方式。

团队成员 4：是的，但是它同时意味着我们要限制在一个页面内到文本链接的导航机制。我发现了一些我们可能想要使用图标的地方，例如……

团队负责人（打断并做记录）：我同意，那有很强的限制性。我已经做了记录要对它进行修改。

团队成员 2：这还表明我们“把导航深度限制在最多四级”。我不明白那意味着什么。

团队成员 1：它意味着任何内容或功能的访问对导航链接的鼠标点击数应该不超过 4 次。这是一个好主意……保持简单，但是我不确定它一直都是可行的。

团队成员 3：如果这些真的被保证，那么我们将一直会有例外。

团队负责人：我同意。

团队成员 4：它同时意味着……

[继续进行每个设计准则的讨论。]

8.3 设计和 WebApp 的质量

设计目标为 WebE 团队提供了一组总体目标，但是团队成员如何确定这些目标是否已经达到了呢？可以对哪些特性进行评估来确定 WebApp 设计的质量呢？

从一般的观点来看，设计将会产生一个用于对 WebApp 的构建进行指导的“模型”。不论模型是什么形式，它都应该包含足够的信息来反映利益相关者的需求（在沟通活动和分析模型中进行了定义）如何转化为内容和可执行代码。但同时设计必须是特殊的。它必须以一种能够使你有效地进行创建和测试的方式来确定 WebApp 的关键质量特性。

这意味着为了保证有效的 WebApp 设计，你需要对 WebApp 的质量有很好的理解。一个很好的起点是检查 WebApp 用户可能的期望是什么。

8.3.1 用户如何理解质量

每个进行过网上冲浪或者使用过公司内联网的人都对“好的” WebApp 有自己的看法。Web 上有大量提供好的和不好的 WebApp 设计的信息和相关例子的网站（至少有这些信息的作者认为的说明好的和不好的质量的例子）[⊖]。大家看待这一问题的角度也相差甚远。有些用户喜欢闪烁的图片，另一些人则喜欢简单的文本；一些人想要丰富的信息，另一些人则希望简略的陈述；有些人喜欢高级的分析工具或数据库访问，而另一些人只是喜欢保持简单。

事实上，用户对于“好的 WebApp”的理解（基于这种理解而接受或者拒绝 WebApp）可能比对 WebApp 质量的任何技术性讨论都要重要。例如，托马斯·爱迪生，留声机的发明者，尽管发明很卓越，但是他建立的营销留声机的公司都失败了——主要是因为他懂技术，但不懂最终用户。结果，他把营销焦点放在了技术而不是设备提供给顾客的娱乐收益。他的竞争对手营销的留声机可能在技术上并不领先，但是更适用于用户，他们在爱迪生失败的时候获得了成功（至少在经济上）。

但是如何认识 WebApp 的质量呢？在最终用户看来，必须要展示哪些特性才能达到优良呢？同时，展示哪些质量的技术性特性才能使你可以长期对应用进行修正性维护、适应性维护、增强性维护和支持呢？

实际上，软件质量的所有一般特征都适用于 WebApp。然而，其中一些关联性最强的特征（可用性、功能性、可靠性、有效性和可维护性）为评估基于 Web 的系统的质量提供了有用基础。

8.3.2 有以用户为中心的模型吗

不论一个 WebApp 满足了多少标准，如果用户不喜欢或者不满意，那么它都将失败。因此，为 WebApp 的质量建立一个以用户为中心的模型是很重要的。为评估（从而进行指导）WebApp 设计提供有价值的基础的一种有用方法是技术接受模型（Technology Acceptance Model，TAM）[Dav89]。TAM 广泛应用于信息系统研究和对用户对新技术（在这里，一个新的 WebApp 就是考虑中的技术）的接受和使用进行建模。这个模型包括两个关键的度量：感知有用性（perceived usefulness，PU），用户认为技术将帮助实现可使用目标的程度；和感知易用性（perceived easy of

---

⊖ 比较好的和差的设计的例子，可以参见下面的网站：
Vincent Flinder's Web Pages That Suck，www. webpagesthatsuck. com/
Bad Human Factors Designs，www. baddesigns. com/
Kerlins. Net Web Design，http：//kerlins. net/bobbi/technology/webdesign/
CoolHomePages Web Site Design Learning and Tutorials Academy，www. coolhomepages. com/cda/10commandments/
Poor Web Design Examples，www. 4webmarketing. biz/webdesign/poorwebdesign. htm
Sunday Software's Good and Bad Church Websites，www. sundaysoftware. com/good-badsites. htm

use，EOU），用户认为系统的易用（花费最少的精力）程度。

TAM 后来已经演化为 TAM2［Ven00］和 UTAUT（Unified Theory of Acceptance and Use of Technology，技术接受和使用统一理论）［Ven03］[⊖]。虽然这些模型主要是作为研究工具，但是它们确实提供了对影响用户对 WebApp 的反应方式的因素的一些有益见解[⊜]。例如，表 8-1 列出了在 UTAUT 中使用的条目，特别针对 WebApp 而不是一般的技术系统做了修改。

表 8-1 在评估用户对一个 WebApp 的响应时用到的项 （来源：摘自［Ven03］）

效用期望（Performance expectancy）
U6[⊜]：我会发现 WebApp 在工作中有用。
RA1：使用 WebApp 使我可以更快地完成任务。
RA5：使用 WebApp 提高生产率。
OE7：如果使用 WebApp，我将增加自己获得提升的机会。
努力期望（Effort expectancy）
EOU3：我和 WebApp 的交互应该是清晰易懂的。
EOU5：在使用 WebApp 时，我很容易变得熟练。
EOU6：我将发现 WebApp 易于使用。
EU4：对我来说，学习操作 WebApp 很容易。
使用技术的态度（Attitude toward using technology）
A1：使用 WebApp 是一个坏/好主意。
AF1：WebApp 使工作更加有趣。
AF2：使用 WebApp 工作很有意思。
Affect1：我喜欢使用 WebApp 工作。
社会影响（Social influence）
SN1：那些影响我行为的人认为我应该使用 WebApp。
SN2：那些对我重要的人认为我应该使用 WebApp。
SF2：对这个业务的高级管理在使用 WebApp 的过程中有用。
SF4：一般来说，组织支持使用 WebApp。
便利条件（facilitating condition）
PBC2：我有 WebApp 必要的资源。
PBC3：我有使用 WebApp 必要的知识。
PBC5：这个 WebApp 和我使用的其他 WebApp 相容（不相容）。
FC3：对于帮助 WebApp 的困难来说，有一个特定的人（或组）可用。
自身效率（Self-efficacy）
我可以使用 WebApp 完成一项工作或任务……
SE1：在我使用过程中，周围没有人告诉我做什么。
SE4：如果我在进行不下去时叫人帮忙。
如果我有很多时间来完成 WebApp 提供的工作。
如果我仅有内置的帮助功能用于辅助。
担心（Anxiety）
ANX1：我对使用 WebApp 感到害怕。
ANX2：当我想到使用 WebApp 点击错误链接而丢失很多信息时，就会感到害怕。
ANX3：我因为害怕做出自己无法纠正的错误而对 WebApp 的使用感到犹豫。
ANX4：WebApp 对我来说有些恐惧。
使用 WebApp 的行为目的（Behavioral intention to use the WebApp）
BI1：我想在接下来的 < n > 个月里使用 WebApp。
BI2：我预测自己将在接下来的 < n > 个月里使用 WebApp。
BI3：我计划在接下来的 < n > 个月里使用 WebApp。

⊖ 对 UTAUT 的一个好的综述可以在约克大学（York University）的关于对技术的接受和使用的统一理论网站（www. istheory. yorku. ca/UTAUT. htm）上面找到。

⊜ 阅读 UTAUT 可以为你提供从用户角度对设计质量的有用理解。

⊜ 在表 8-1 中使用的条目符号是 UTAUT 中所指定的，来自于对 UTAUT 的开发有贡献的那些原始资源。例如，条目 OE7 指的是在结果预期（Outcome Expectations，出现在 Social Cognitive Theory——社会认知理论内部）中的条目。PBC 条目指的是知觉行为控制（Perceived Behavioral Control，来自 Theory of Planned Behavior——计划行为理论）。所有的细节都包含在参考文献［Ven03］之中。

随着每个 WebApp 增量的引入，可以给一组最终用户的代表提供一个包含表 8-1 中列出的问题的调查表。在合适的地方，可以对问题进行定制来适应正在开发的特定 WebApp。一旦对这些问题的答案进行了统计分析，结果可以为 WebE 团队提供以用户为中心的对设计（和构建）活动的认识指标。

作为使用 UTAUT 的一个例子，考虑如下描述 SafeHomeAssured.com 的团队成员内的讨论：



评估设计质量

场景：增量 5（监控服务）的一个设计审查会议，SafeHomeAssured.com 的 WebE 团队的会议室

参与者：WebE 团队负责人和 WebE 界面设计人员

会话：

界面设计人员：我已经对用户对这个增量最初界面设计原型的反馈进行了评估。有一方面让我很担心，我想和你一起对它进行审查。

团队负责人：问题是什么？

界面设计人员：我使用你展示给我的 UTAUT 方法，并且使用和这个增量特定问题相关的一些附加问题对它进行修订。大多数的结果看起来都不错——除了关于用户优虑问题的答案。

团队负责人：请提示一下，问题是什么？

界面设计人员：好的，这是问题列表。[界面设计人员拿出了一张纸，上面写有下面的这些内容（一部分）]：

忧虑：

忧虑 1：我对监控服务的使用感到不安。

忧虑 2a：当我想到我会因为使用 SafeHomeAssured.com 监控服务而丢失大量信息的时候，就感到很恐惧。

忧虑 2b：我担心一个未授权的人员可以访问我的住宅安全信息或功能。

忧虑 3：我因为害怕犯出自己无法纠正的错误而对使用监控服务感到犹豫。

忧虑 4：监控服务对我来说有点儿陌生的感觉。

界面设计人员：我们把这个给了 20 个抽样的可能用户，让他们基于一个标准的五分利凯特式量表（Likert scale），从完全不同意到完全同意对这些问题进行评估。在关于忧虑的问题中，忧虑 3 和忧虑 4 的答案还行，这很可能意味着用户对界面一般是比较满意的。问题出在前三个忧虑问题上。这二个问题比界面中的其他问题严重得多。然后我返回到用户中，和他们对此进行了更详细的讨论。事实证明，他们确实担心其他人员闯入系统并访问他们的安全设备。基本上来说，诸如使用摄像机来监视他们或者甚至使安全设备失效而进行闯入。

团队负责人：但是我们在所有这些功能周围都已经有了很卓越的安全性了。

界面设计人员：是啊，但是那种安全性对生们来说实际上是不可见的。我认为我们需要对界面做出一点儿改变，让安全性方面对用户更可见一些。

团队负责人：好的。明天带着我们将如何完成它的一些建议来找我怎么样？

对很多 Web 工程项目来说，诸如 UTAUT 这样的正式方法可能是大材小用了。然而，当 WebApp 是关于关键业务时，当用户的接受是业务成功的关键时，以及当使用更加非正式的方法很难辨别用户对有用性和易用性的认识时，诸如 UTAUT 这样的基于问题的格式，并辅以后续的统计进行的分析，可以为 WebE 团队提供重要的指导。

8.3.3 什么是质量框架

如果不同类型的最终用户回答了表 8-1 包含的问题，那么 WebE 团队就将获得对所创建的 WebApp 的功效的深入理解。但是从用户的角度（虽然很重要）对“好的 WebApp”进行评估不是对质量的唯一判断。例如，一个电子商务网站对有用信息提供了很容易的访问方式，所以从用户的观点来看，它的设计非常好，但是实际上却不能售出任何产品。除了以用户为中心的模型之外，还有很多质量框架识别在设计中应当考虑的更多质量特性。

Olsina 和他的同事［Ols99］设计了一个“质量需求树”，它定义了一组可产生高质量 WebApp 的技术性属性——可用性、功能性、可靠性、有效性和可维护性[⊖]。图 8-1 总结了这些的工作。在图中提到的标准对于那些必须长期从事设计、创建和维护 WebApp 的 Web 工程师非常有帮助。

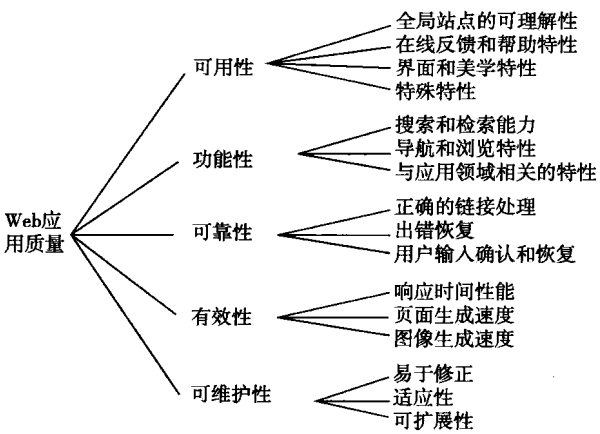


图 8-1 质量需求树

来源：来自［Ols99］

Offutt［Off02］对图 8-1 中描述的五个主要的质量属性进行了扩展，增加了下面的属性：

安全性。WebApp 已经和重要的公司及政府数据库高度集成。电子商务应用提取并存储敏感的客户信息。基于这些及许多其他原因，WebApp 的安全性在很多情况下都极为重要。例如，对 SafeHomeAssured. com 而言，未授权用户不能访问其他用户住宅的安全性信息，这一点非常关键。对安全性的关键度量标准是 WebApp 和服务器环境拒绝未授权访问和（或）阻挡恶意攻击的能力。对 WebApp 安全的详细讨论已经超出了本书的范围。如果你还有兴趣，可以参考［Cro07］、［Das07］或［Kal03］。

可用性。如果不可用的话，那么即使是最好的 WebApp 也不能满足用户的需要。从技术角度来说，可用性就是对 WebApp 可以使用的时间占总时间的比例的一种度量。一般最终用户都会期望 WebApp 是 24/7/365（一天 24 小时、一周 7 天、一年 365 天）都可用。对可用性的任何减少都被认为是不可接受的[⊜]。不过正常运行时间并非可用性的唯一指标。Offutt［Off02］认为，“使用那些仅限于一种浏览器或一个平台上可用的特性”会使 WebApp 在那些具有不同的浏览器或平台的配置环境中变得不可用。用户毫无疑问地会转到其他地方。

⊖ 这些质量属性和软件应用经常引用的那些质量属性非常相似。这就暗示所有的软件质量特性都是通用的。

⊜ 当然，这种期望除了最极端的例子外是不现实的。大多数 WebApp 都必须安排用于安装和升级的停机时间。但是，我们可以保证（通过对服务器群的使用，对维护和部署服务器的精心设计等）到 WebApp 新版本的过渡发生的时间很短，使得对用户来说是感觉不到的。

可扩展性。WebApp 及其服务器环境是否可以伸缩来处理 100、1000、10 000 或者 1 000 000 个用户？WebApp 和为其提供接口的系统能够处理容量和负载的显著变动吗？或者响应能力会显著下降（或者终止）吗？创建一个通常成功的 WebApp 是远远不够的。开发一个可以不断调节成功的负载（最终用户的显著增加）甚至和变得更加成功的 WebApp 同等重要。

面市时间。虽然说从技术角度来看，面市时间并不是一个真正的质量属性，但是从商业角度来看，它是质量的一种度量。第一个上市的 WebApp 常常获得非常多的最终用户。

更一般地讲，Miller［Mil00］提出了下面的“质量度量因素。”Web 工程师应当考虑这些因素，但这些因素其实表示了最终用户更可见的质量视角。

时间。WebApp 的变化有多快？从最后一次升级到现在，WebApp 变化了多少？你如何强调变化了的那些部分？

结构。WebApp 各个部分如何结合在一起？WebApp 的所有内部和外部连接都能工作吗？所有的图像都工作吗？是否有未连接的 WebApp 部分？

内容。关键页面的内容是否在其该在的地方？关键词组存在于不断变化的页面中吗？关键页面从一个版本到另一个版本保持了高质量的内容吗？动态生成的 HTML 页面怎么样呢？

准确性和一致性。今天下载的页面的副本和昨天的一样吗？足够相似吗？显示的数据足够准确吗？你是如何知道的？

响应时间和延时。Web 服务器可以在某些参数的范围内响应一个浏览器请求吗？在电子商务或工作流环境中，信息提交以后端到端的响应时间如何？网站上是否有一些部分太慢而导致用户感到沮丧？

性能。浏览器 - 服务器连接够快吗？在一天内的不同时间、不同负载和不同用法的情况下，性能如何变化？性能满足电子商务的应用吗？

作为一个 WebApp 设计者，在你设计 WebApp 的界面、导航方法、布局和美学的时候，你可以把这些问题作为参考。

8.3.4 有评估内容质量的方法吗

对于在万维网上搜索信息的人来说，有数以亿计的 Web 页面可用。即使是目标很明确的 Web 搜索，其结果也是超大量的内容。有这么多可供选择的信息来源，用户如何评估一个 WebApp 内显示的内容的质量（例如，真实性、准确性、完整性、时效性）呢？Tillman［Til00］提出了如下一组评估内容质量的有用标准：

- 能否很容易地确定内容的范围和深度，保证满足用户的需要？
- 能否很容易地确定内容作者的背景和权威性？
- 能否确定内容的通用性？最后一次更新的时间和更新的内容是什么？
- 内容和位置是否稳定（例如，它是否一直保持在被引用的 URL 处）？

除了这些与内容相关的问题，还可以增加下面的这些需要考虑的问题：

- 内容是否可信？
- 内容是否独特？也就是说，WebApp 能否给使用它的用户提供一些特有的好处？
- 内容对于目标用户群体是否有价值？
- 内容的组织是否合理？是否有索引？是否易于访问？

8.3.5 有质量检查表吗

如果我们能为 WebApp 建立一种单一的、明确的质量检查表，那确实很好。但问题是，不同

的 WebApp 利益相关者看重不同的质量特征。每个利益相关者都扮演着不同的角色，有着不同的观点。此外，不同的 WebApp 有着不同的需求和质量要求。

为了对你有所帮助，我们把前面讨论的各种资源整合成一个检查表。这个检查表提供了对某要素问题点的情况，这个情况可用作告知设计和进行设计审查的基础。

评估设计

WebApp 设计 IQ 检查表

下列 WebApp 设计的 IQ（重要性/质量）检查表（来源于本书所讨论的资源）提供了一系列在 WebApp 设计期间需要考虑的关键 WebApp 特性。通过考虑这些特性，Web 工程师和最终用户可以快速评估 WebApp 的整体质量，然后把他们的注意力集中在需要进一步考虑的特性上。

对于每个列出的特性，给定 I 和 Q 的分数，然后计算这两个值的乘积。那些特性具有整体的等级（或绩）或更可能需要进一步考虑。那些具有 10 或更高等级的条目是需要主要关注的。

I（重要性）等级：评价 WebApp 特性的重要。

0 = 对 WebApp 的重要性最小

5 = 对 WebApp 的重要性高

Q（质量）等级：评价 WebApp 设计或实现的当前质量级别。

0 = WebApp 的最小质量问题

5 = WebApp 的重大质量问题

IQ 设计特性

I Q 可用性

□×□=____全局可理解性（适当的隐喻等）

□×□=____适当的反馈和帮助功能

□×□=____界面美观（可学习性）

□×□=____结构性导航

□×□=____导航完整性（链接正确性、完整性等）

□×□=____站点总体视图（地图、层级和层级等）

□×□=____用户适应和个性化

□×□=____其他_____

内容

□×□=____内容完整性（广度和深度）

□×□=____内容正确性

□×□=____内容实时准确性

□×□=____内容相关性

□×□=____内容的准确性和权威性

□×□=____用户对内容的认识

□×□=____其他_____

功能的

□×□=____功能的可访问性

□×□=____功能的一致性

□×□=____功能的正确性

□×□=____搜索和索引功能

□×□=____异常处理和错误恢复

□×□=____其他_____

非功能的

□×□=____数据安全性

□×□=____用户对安全的认识

□×□=____WebApp 的可用性

□×□=____WebApp 的扩展性

□×□=____WebApp 的响应速度和性能

□×□=____其他_____

配置的

□×□=____易于改正

□×□=____扩展性和适应性

□×□=____其他_____

WebApp 设计应当努力解决所有这些质量需求，但是事实上，这些质量需求之间有时是互相冲突的。例如，一个金融服务公司（竞争性很强的环境）的营销部门希望最终用户可以尽可能快、尽可能容易地进入公司的网站。而对隐私和安全负责的利益相关者则希望确保网站不会被

有攻击意图的人员破坏，即使这会阻碍用户进入网站。WebApp 设计必须要响应这些有冲突（但是仍然有效）的需求。

前面一页中的检查表仅仅展示了在 WebApp 演化过程中应该解决的问题的很少的抽样，但是它已经被开发为解决关键问题然后把设计者的注意力聚焦于那些需要进一步注意的问题。Web 工程的一个重要的目标是开发与质量相关的问题都有肯定答案的系统。

8.4 设计过程

是时候开始更明确地讨论 WebApp 设计了。为了以一种可以实现高质量的方式进行设计，你应当理解和应用那些关键的设计任务——“如何”进行 WebApp 设计。

8.4.1 WebApp 设计的元素有哪些

Powell［Pow00］在他关于 Web 设计的早期作品中，试图在 Web 工程的环境下定义设计：

> 定义 Web 设计是很困难的。对有些人来说，Web 设计注重视觉效果。而对另一些人来说，Web 设计是通过文档空间构造信息和导航，还有一些人可能会把设计看成是一种构建交互式 Web 应用的技术。事实上，Web 设计应该包括所有的这些内容，并且可能包括得比这还多。

Web 设计关注基于 Web 的系统的很多不同方面。因为美学、内容和技术适当的结合会随着 WebApp 性质的不同而不同，所以强调的设计活动对每个项目都是不同的。一个关键的 Web 工程技巧是对这些不同元素的有效集成。

设计开始于关注对最终用户（每天都会使用 WebApp 的一类利益相关者）来说很重要的那些问题。几乎所有的 WebApp 都有一个交互性很强的组件。因此，交互设计是用户接受的枢纽，也是 WebApp 本身成功的关键。由此，交互设计就是 WebApp 开始的地方。

设计又在哪儿结束呢？这个问题的答案看起来似乎很简单——设计结束于构建（例如，HTML 编码）的开始。但是这个过分简单的答案有两个问题。第一，设计和构建之间的界限常常很模糊，在原型中确定了设计的很多方面，而原型成为了构建的基础。第二，即使是最好的设计者也无法预见将在构建活动中遇到的每一个意外事件。有些构建问题需要重新设计，因此，设计延续到了构建活动。

所以设计开始于界面，终止于构建的完成。但是中间会发生什么呢？

图 8-2 描绘了一个设计金字塔，金字塔包含了将在创建一个完整设计模型的过程中发生的设计动作（层次）。金字塔的最高两层（界面设计和美学设计）构成了用户与系统交互的基础，因此它们可以被归为一类——交互设计[⊖]。我们逐一考虑交互设计的这些方面：

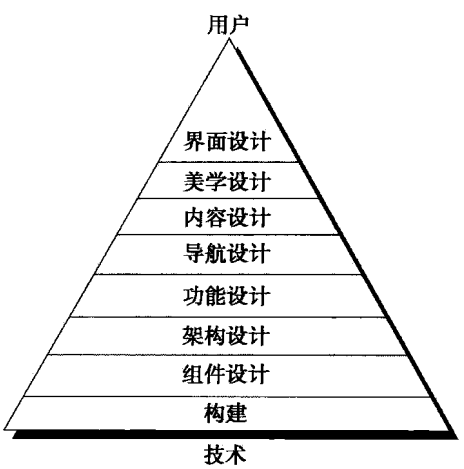


图 8-2 WebApp 设计金字塔

⊖ 交互设计将在第 9 章中进行详细考虑。

- *界面设计*描述了用户界面的结构和组织形式，包括屏幕布局的表示、交互模式的定义和导航机制的描述。对于不只是简单功能的网站来说，可能还包括对工作流顺序和相关问题的描述。
- *美学设计*，也称为图形设计，描述了 WebApp 的外观和感觉，包括颜色配置、几何布局、文字大小、字体和位置、图形的使用和相关的美学决策。

金字塔接下来的两层关注信息：支撑 WebApp 的内容、内容的组织方式和用户在 WebApp 内访问内容或功能的方式。我们分别考虑这些设计任务：

- *内容设计*定义所有内容的布局、结构和作为 WebApp 的一部分进行展示的内容大纲，并建立内容对象之间的关系。
- *导航设计*展示内容对象之间和所有 WebApp 功能的导航流程，描述了用户对内容进行定位和与内容交互的方式。

第 10 章将详细讨论内容和导航设计。

WebApp 已经逐渐发展为除了提供各类内容之外，还提供大量的功能，所有这些都依附于考虑的应用领域。例如，SafeHomeAssured. com WebApp 除了提供大量的内容（例如，产品描述、规格说明书、照片）之外，还提供了很多功能性能力（例如，允许用户定义空间布局、订购产品、通过因特网监控住宅）。在某些情况下，功能将采用既定模式。例如，business-to-consumer（B2C）WebApp 使用目录、购物车和付款台。在另外一些情况下，功能可能相当独特。当设计明确地关注功能时，我们采用如下的方式进行描述：

- *功能设计*识别 WebApp 所支持的总体行为和功能——包括如工作流支持、内容和界面适应和（或）定制、订单录入、数据库处理、计算性功能和其他等方面。

功能设计将在第 11 章中进行讨论。

设计金字塔的底层（归类为技术设计）考虑界面、导航、内容和功能将如何被集成和实现。技术设计为后面的构建活动提供基础。考虑 WebApp 的架构和组件设计：

- *架构设计*确定了 WebApp 的总体结构。*概念架构设计*确定了 WebApp 的概念组件及其之间的互联方式，确保应用的各个方面被合适地集成[⊖]。*技术架构设计*确定技术组件和构建 WebApp 所需的组件之间的交互。
- *组件设计*开发实现支持一组完整的 WebApp 功能的功能组件所需的详细处理逻辑。

应该澄清概念架构和技术架构之间的区别。概念架构勾画出系统需要的主要功能组件和信息组件。例如，一个简单的电子商务应用可能需要一个产品目录（有内容和搜索-浏览的功能）、一个购买系统（有购物车管理和支付功能）、一个客户管理系统（用于跟踪用户购买的东西，并执行网站的适用性和推荐），而且还可能需要一个管理系统（用于注册、购买请求的管理等）。

相比较而言，技术架构展示构建架构的具体设计组件如何映射到实现 WebApp 的具体技术。例如，一个电子商务应用可能包括一个 Web 服务器、一个支付通道、一个内容管理系统和一些 HTML 页面和 ASP 脚本。

8.4.2 设计过程的特征有哪些

WebE 的设计金字塔阐明了创建一个高质量 WebApp 需要执行的设计任务。但是执行设计的实际过程如何呢？应当何时考虑这些设计任务以及考虑的深度如何呢？

⊖ 例如，我们可以有一个独特的导航设计，它可以控制用户与 WebApp 的交互方式，但是为了使 WebApp 适应不同的用户，将对这个导航的功能进行修改。在集成导航和功能元素时，保持总体完整性非常重要。

要回答这些问题，我们先看总体的 Web 工程过程，然后关注它能告诉我们关于设计的那些内容。在前面几章中，我们已经提到 Web 工程增量地产生 WebApp。意思是你可以在一系列增量中为最终用户交付健壮性和复杂级别递增的内容和功能。Web 的特有性质以两种方法来推动这种策略。

第一，从一个 WebApp 增量到另一个更健壮的增量的转变，对用户来说通常是透明的。虽然说工作肯定是 WebE 团队完成，但是在最终用户级别无需任何动作[3]。因此，采用一个增量的设计和版本策略就合理得多。

第二个推动增量过程的特征就更微妙了。WebApp 的界面被分割成通过导航通道进行访问的页面序列。这意味着在对已有内容和功能干扰最小的情况下，额外的内容和（或）功能可以被添加到一个应用中。新的增量需要对已有 WebApp 做的改变可能像在一个页面上添加一个新链接那么简单。

8.4.3 对设计活动而言，一个增量的 WebE 过程意味着什么

在构造和计划活动（第 4 章和第 5 章）期间，对 WebApp 增量进行了描述。在分析建模时，WebApp 增量被进一步精化。在第 8.4.1 节中讨论的设计任务必须应用于每个增量。然而，有一个与总体架构完整性相关的重要的初步或初始设计步骤。



初始设计

场景：SafeHomeAssured.com WebE 团队的会议室，正在进行增量 1（公司和产品的基本信息）的初始设计会议

参与者：WebE 团队负责人和 WebE 团队的另外两个成员

会话：

团队成员 1：好了，我们终于可以开始一些实际设计了。看到 WebApp 的开始部分开始成形并得到一些可用的东西，感觉真的很好。

团队成员 2：是啊，当然了。我已经看过了对这个增量的描述。非常简单，只是一些简单的信息。诸如“你好，我们在这里”的一组页面。我认为我们可以在下周很快地把这些拼凑起来，运行一个 Linux 机器上 Apache Web服务器上的一些页面真的很简单，同时也意味着我们可以开始使用我最近找到的用于 Linux 的一些很好的版本控制软件。我说我们马上进入信息设计吧。

团队负责人：稍微等一下。确实，第一个增量是很简单——特别是它实际上不需要任何复杂的硬件或服务器软件。但是如果我们现在仅仅因为特定的平台没有多大关系就很随意地决定一个，那么当它在后面的增量中有关系的时候，我们就会陷入麻烦之中。你们是否想过其他增量通过什么可以添加一些高层架构上的约束？

团队成员 2（皱眉）：没有，真的没想过。我认为它没有关系。

团队负责人：那么，考虑实现空间布局的那个增量。我们在内部使用的软件都运行在 PC 机上，还使用一些我认为只能用在 PC 机上的专门程序库。如果我们现在就让自己受困于一个 Linux 平台，那么当我们到了那个特殊增量时，所有的问题都会出现。

团队成员 1：有道理。我认为我们应该开始思考一个更广的架构。一旦我们完成了，我们就可以考虑任何增量的细节。

---

[3] 这和传统软件相反。传统软件的一个新版本需要进行重新安装（以及随之带来的复杂性），而且常常由最终用户来执行。

由 SafeHomeAssured. com WebE 团队的成员进行的如上框中的讨论提出一个重要问题。在整本书中，我们都在强调敏捷性的必要性，并建议它作为创建 WebApp 的关键原则。加上一些敏捷过程模型的拥护者认为开发者应当避免把任何与未来增量相关的事物变成系统的一部分，其目的是能够专门集中于将被交付的增量，不会被“外来的”细节转移视线。

然而，在很多情况下，对 WebApp 架构问题更广的考虑不仅是合理的，而且是必需的。有一些敏捷方法确实把这个问题定位到有限范围内，但其他方法却没有强调架构设计的重要性，结果，会出现主要的架构问题。例如，在极限编程（XP）［Bec99］［Bec05］中，一个关键的步骤是开发系统隐喻，但是与其说它确保应用的完整性，倒不如说它支持开发者在设计过程中的沟通与理解。

可以通过考虑全部增量（或者至少是一个尽可能有代表性的样例）以及随之发生的总体概念架构和技术架构的开发而开始进行总体设计来解决这个问题。一旦这些架构的基本元素开发完成，那么每个增量的设计和构建就可以开始了。

然后就可以产生如图 8-3 中所列出的设计过程。从图中可以看出，概念架构和技术架构的总体结构都进行了初始开发，但是这些架构相关方面的设计会延迟到具体需要 WebApp 增量的时候。然后，随着每个 WebApp 增量的设计，交互层也进行了设计，然后告知核心的架构组件和信息及功能层的详细设计。这些设计一旦完成，设计模型相关的概念组件就会被映射到技术设计。

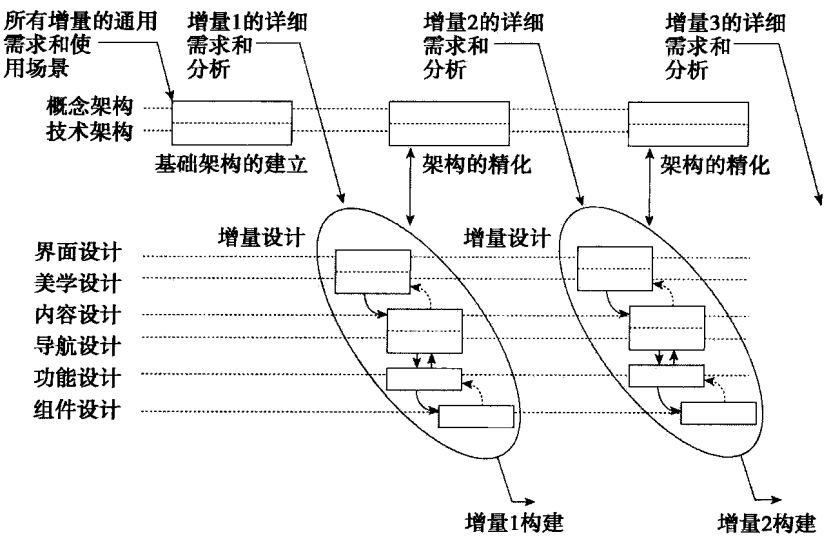


图 8-3 WebE 设计过程

8.5 概念架构的初始设计

概念架构为 WebApp 的设计提供总体结构。它表示 WebApp 主要的概念组件和信息组件，同时描述这些组件如何组成一个整体。具体架构将依赖于 WebApp 的性质，但是在每种情况下，都应该确保 WebApp 的信息和功能之间的牢固集成。

传统软件开发的高层设计活动常常关注一个应用的功能方面[⊖]。本质上，一个系统架构师试图识别[⊜]系统可能需要的功能模块或子系统，结果通常被记录为一个模块结构图，以描述这些子

⊖ 面向对象的方法是个例外——问题类从一个分析模型流出，并与控制它们的方法进行耦合。

⊜ 考虑到大多数作者都把架构设计看作一个有创造性的过程，直觉（Intuit）也许是描述这个实际过程更合适的词。

系统及其之间的接口。

当这个过程被应用到 WebApp 时，可能会有问题。它能正确地识别功能子系统，但却常常忽视信息模块或仅仅考虑其最表面。那么在概念架构中，你如何达到信息和功能的有效平衡呢?

工作流或功能性场景（它是系统功能的表达）和信息流（在第 7 章中进行了描述）是一个好的开头。如果你在工作流或关键功能内寻找共性，将会发现主要的功能组件；如果你在信息流内寻找关系，那么将会发现主要的信息组件。

作为简单的例子，考虑下面的 SafeHomeAssured. com（在第 4 章中确定的）关键功能：

- 提供产品报价单
- 处理安全系统订单
- 处理用户数据
- 创建用户资料
- 绘制用户空间布局
- 为布局推荐安全系统
- 处理监控订单
- 获取并显示账户信息
- 获取并显示监控信息
- 客户服务功能（将在后面进行定义）
- 技术支持功能（将在后面进行定义）

从这些关键的功能中，我们可以识别下面不完整的功能子系统列表：

- UserManagement（用户管理）。管理所有用户功能，包括用户注册、授权和配置、用户特定的内容以及界面的适应和定制。
- ProductManagement（产品管理）。处理所有的产品信息，包括定价模型和内容管理。
- OrderHandling（订单处理）。支持客户订单的管理。
- AccountAdministration（账户管理）。管理客户的账户，包括消费清单和支付。
- SecuritySystemSupport（安全系统支持）。管理用户的空间布局模型，并推荐安全布局。
- SeuritySystemMonitoring（安全系统监控）。监控顾客的安全系统，并处理安全事件。

当然，同时还有总体的管理子系统：

- ClientInterface（客户端界面）。为了满足用户的需要，提供用户和其他子系统之间的界面。
- SystemMaintenance（系统维护）。提供维护功能，如数据库清理。

现在，考虑下面的这些已经在第 4 章中引入的内容对象：

- 公司概况
- 产品概述
- 产品规格说明
- 安装指南
- 产品数据库（包括定价、存货清单、运费成本等）
- 报价模板
- 订单表单
- 订单录入数据库
- 用户空间的布局图
- 用户数据库（包括用户的确认、专门的安全系统配置，以及其他与用户相关的信息）
- 监控请求表单

- 一个合同用户的监控“仪表板”（包括账户信息、监控视频窗口、监控音频端口）
- 监控数据库
- 客户服务仪表板（将在后面进行定义）
- 技术支持仪表板（将在后面进行定义）

从这个列表中可以识别一组关键的信息组件：

- 用户
- 公司信息
- 产品信息
- 报价单和订单
- 安全系统的配置（空间模型和系统布局）
- 安全系统的数据（动态监控信息）

架构设计的挑战是以一种有效的方式将功能子系统和信息组件连接起来。为了实现这个目标，WebApp 的设计者开发了一个把功能组件连接到每个组件将要控制的信息的混合模型。这个混合模型同时把用户连接到他们需要交互的信息和功能。

图 8-4 展示了 SafeHomeAssured. com 的一个初始概念架构。图中显示了各种功能子系统和信息组件之间的链接。架构的一些方面值得注意。第一，设计中仅有的静态页面是公司的信息页面，其他所有的内容都从数据源动态生成，以适应不同的用户。由各种功能子系统执行这种动态生成。第二个有趣的方面是，主要的客户端界面（它可能在技术架构层包括实际 Web 服务器）使用各种其他子系统来生成需要的页面，然后交付给用户。最后，SafeHomeAssured. com WebApp 包含了支配网站总体观感的全局配置文件和风格模板。

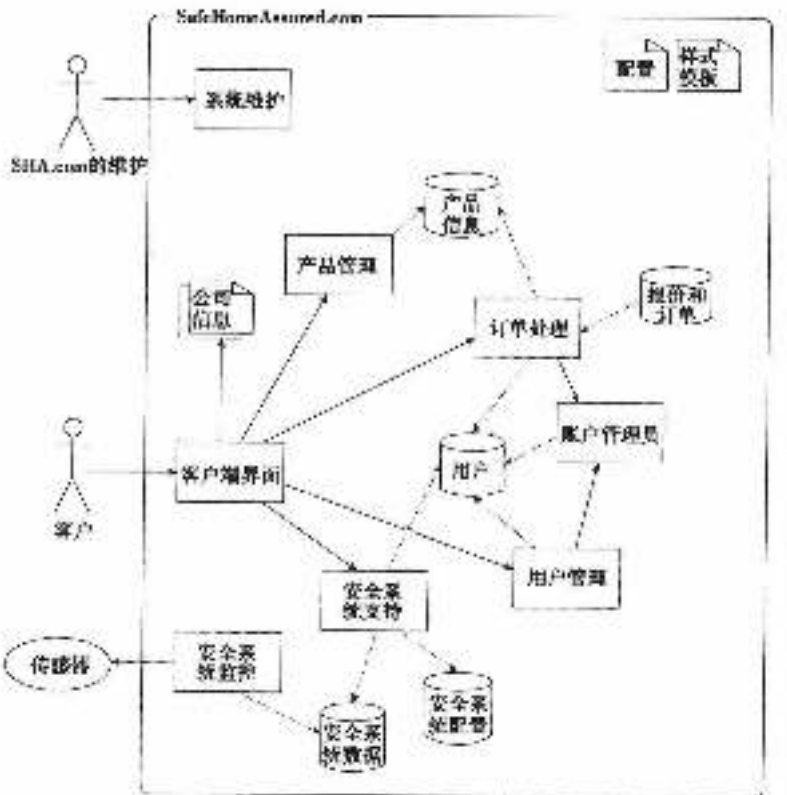


图 8-4 SafeHomeAssured. com 的概念架构示例（部分）

概念架构设计可以用于初始化设计过程，并建立每个增量的详细设计起点。随着在后续设计任务中获得知识，架构可能还要被精化和修改，但是它大体上形成了一个可以在每个增量的开发中确保合理完整性的基线。

8.6 技术架构的初始设计

概念架构为 WebApp 设计提供了总体结构，而技术架构显示全局架构如何映射到具体技术组件。我们将在第 12 章和第 14 章对技术组件进行更多的讨论。现在，应该注意对技术组件的决策有影响的各种关键问题。

任何关于一个组件如何映射到技术架构的决策都将影响到其他组件的决策。原因有很多，但通常是因为一个技术组件提供了广泛基础的支持或者在其他组件上加了约束。例如，WebE 团队可能选择的 SafeHomeAssured. com 设计是把产品信息存储为 XML 文件。后来，团队发现内容管理系统难以支持对 XML 内容的访问，而是假设把内容存储在传统的关系型数据库中。技术架构的一个组件就与另一个组件强加的约束相互冲突了。

很多（可能是大多数）技术组件并非只狭窄地考虑一个焦点而开发。相反，它们被设计为能够尽可能广泛地被适用。因此，这些组件常常适应性很好，但同时复杂性也很显著。一个充分展示商业力量的内容管理系统很明显会提供给简单的静态网站它们所需的功能，但也会提供它们不需要的那些额外功能。这些不必要的功能将使开发变得复杂——分散开发者的注意力，还需要额外的管理工作量。在有些情况下，一个简单得多的技术方案将更加合适。

最后，组成技术架构的技术组件的具体选择不仅取决于概念架构所强加的特殊需求，而且还取决于组件的运行环境（例如，WebApp 的主机环境可能基于一个 Linux 平台）和开发团队的技能。

8.7 本章小结

WebApp 的设计是艺术与工程的结合。为了有效，一个好的设计必须适应利益相关者在沟通活动中建立的每个需求。但是它同时也要适应发生在设计、构建和部署中的不可避免的需求变更。设计开始于对用户与一个应用交互方式的关注，然后考虑满足利益相关者的要求所需要的功能和信息内容。然后进行物理设计，其中，逻辑元素被映射为可以作为 WebApp 的一部分进行实现的表示。设计活动的主要输出是一个设计模型，模型包括界面描述、美学、内容、导航、架构和组件级设计问题。

每个 WebApp 设计者都必须认识到一组通用设计目标：简单性、一致性、相符性、健壮性、导航性、视觉吸引和兼容性。另外，很多质量框架都可以应用于 WebApp 设计。这些质量标准提供了实现 WebApp 设计质量的目标。此外，以用户为中心的质量模型提供了一个可靠的指标，指明了最终用户对 WebApp 交付技术的喜欢程度。WebApp 的设计者可以应用一个“设计 IQ（重要性/质量）检查表”来评估各种质量标准的重要性和 WebE 团队满足每种标准的程度。

设计过程是一个敏捷的、迭代的设计动作集合，这些设计动作应用于每个 WebApp 增量的创建。可以使用一个设计金字塔来描述将为每个 WebApp 增量执行的设计动作集。这些设计动作包括界面设计、美学设计、内容设计、导航设计、功能设计、架构设计和组件设计。

对 WebE 团队来说，每个设计动作都代表了一个挑战。在后面的几章中，我们将更详细地讨论每个设计动作的机制。

参考文献

[Bec99] Beck, K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999.

[Bec05] Beck, K., and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed., Addison-Wesley, 2005.

[Cro07] Cross, M., *Developer's Guide to Web Application Security*, Syngress, 2007.

[Dav89] Davis, F. D., "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly*, vol. 13, no. 3, 1989, pp. 319-342.

[Das07] Daswani, N., et al., *Foundations of Security*, Apress, 2007.

[Kai02] Kaiser, J., "Elements of Effective Web Design," About, Inc., 2002, http://webdesign.about.com/library/weekly/aa091998.htm (accessed September 15, 2004).

[Kal03] Kalman, S., *Web Security Field Guide*, Cisco Press, 2003.

[Mil00] Miller, E., "The Website Quality Challenge," *Software Research*, Inc., 2000, www.soft.com/eValid/Technology/White.Papers/website.quality.challenge.html (accessed July 31, 2007).

[Nie00] Nielsen, J., *Designing Web Usability*, New Riders Publishing, 2000.

[Off02] Offutt, J., "Quality Attributes of Web Software Applications," *IEEE Software*, March/April, 2002, pp. 25-32.

[Ols99] Olsina, L. et al., "Specifying Quality Characteristics and Attributes for Web Sites," *Proc. 1 st ICSE Workshop on Web Engineering*, ACM, Los Angeles, May 1999.

[Pow00] Powell, T., *Web Design*, McGraw-Hill/Osborne, 2000.

[Til00] Tillman, H. N., "Evaluating Quality on the Net," Babson College, May 30, 2000, www.hopetillman.com/findqual.html#2.

[Ven00] Venkatesh, V., and F. D. Davis, "Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," *Management Science* vol. 46, no. 2, February 2000, pp. 186-204.

[Ven03] Venkatesh, V., M. G. Morris, G. B. Davis, and F. D. Davis, "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly*, vol. 27, no. 3, 2003, pp. 425-478.

第9章 交互设计

不论是为 WebApp、传统的软件和消费产品设计的用户界面，还是为工业设备设计的用户界面，每一个用户界面都应该表现出这样的特性：易于使用、易于学习、易于导航、直观、一致、高效、无错并且有用。它还应该为最终用户提供令人满意的和有益的经验。界面设计的概念、原则和方法为 Web 工程师提供了实现这些特性所需要的工具。

WebApp 的交互设计并非首先考虑技术或工具，而是首先对最终用户进行仔细考察。在分析建模（第 7 章）过程中开发出了一个用户层次。每一类用户可能会有稍微不同的要求，可能会以不同的方式要求与 WebApp 进行交互，或者可能要求独特的功能和内容。这些信息在沟通活动中产生，可能在分析建模中进行精化，并且将作为交互设计的第一步被再次考虑。

Dix［Dix99］认为你应该设计这样的界面，以致能够为最终用户回答三个主要问题：

我在哪里？界面应该提供对已经访问过的 WebApp 的指示[⊖]，或者告诉用户他们当前在内容层次中所处的位置。

我现在能做什么？界面应该总是能够帮助用户理解他们当前的选项——哪些功能是可用的？哪些链接是可用的？哪些内容是相关的？

我去过哪里？又将去哪里？界面必须能够辅助导航，因此，必须提供一张“地图”（以一种易于理解的方式来实现）。这张地图显示了用户到过哪里，还能通过哪些路径去 WebApp 内的其他地方。

在最终用户导航内容和功能时，一个有效的 WebApp 界面必须回答所有这些问题。

9.1 界面设计原则和指导准则

一个 WebApp 的用户界面是它的“第一印象”。无论它的内容价值、处理能力和服务的复杂性以及 WebApp 自身的总体收益如何，一个设计拙劣的界面将令潜在用户失望，而且，事实上很可能使得用户转到其他地方。由于在实际中的每个领域都有海量的有竞争力的 WebApp，所以界面必须瞬时吸引住潜在用户。

Bruce Tognozzi［Tog01］定义了所有界面都应该具有的一组基本特性，这样就建立了所有 WebApp 界面设计人员都应该遵循的指导原则：

> 有效的界面在视觉效果上是明显而宽容的，并且慢慢地给用户灌输一种控制感。用户能够很快地看到他们的选择范围，领会如何达到他们的目标，然后做他们的工作。
>
> 有效的界面使用户不必关心系统的内部操作。工作被谨慎而连续地进行保存，从而使用户有充分的选择余地，可以在任何时刻取消任何活动。
>
> 有效的应用和服务从用户那里请求最少的信息，而完成最多的工作。

你也许会认为，每个人都知道这些特性，而且每个界面设计者都想去达到它们。可是，我们每个人都已经看到了很多确实差的界面，而且我们肯定将会看到更多这样的界面。

⊖ 我们都曾把网页加入书签，仅仅是为了稍后对它进行再次访问，书签并没有对站点或网页上下文的指示（也无法连接到网站的其他位置）。

9.1.1 设计有效界面的原则

为了设计有效的界面，Tognozzi［Tog01］确定了一组非常重要的设计原则[⊖]：

预测。一个 WebApp 应当被设计为它能够预测出用户的下一个步骤。例如，考虑一个由计算机的打印机制造商开发的顾客支持 WebApp。一个用户已经请求了一个内容对象，此对象表示出一个最新发布的操作系统的打印机驱动程序的相关信息。这个 WebApp 的设计者应该预测到用户可能会请求下载驱动程序，并且应该提供下载的导航辅助，而不需要用户查找这一功能。

传达。界面应该能够传达由用户启动的任何活动的状态。传达可以是明显的（例如，一条文本信息），也可以是隐含的（例如，一张纸在一个打印机内的移动表明打印机正在工作）。界面也应该在 WebApp 的内容层次中传达用户的状态（例如，用户的身份）。

一致性。在整个 WebApp 内，对导航控制、菜单、图标和美学（例如，颜色、形状和布局）的使用应该是一致的。例如，如果带有下划线的蓝色文本表示导航链接，那么内容就不应该包含不表示链接的蓝色有下划线的文本。另外，一个对象（如一个黄色的三角形）用于表示用户调用一个特定功能或动作之前的一条警告信息，那么它就不应该在此 WebApp 内的其他地方用于其他目的。最后，界面的每个特征都应该以和用户期望一致的方式来响应[⊜]。

自治。界面应该辅助用户在整个 WebApp 中移动，但是辅助的方式应该坚持使用那些已经为应用建立起来的导航习惯。例如，对 WebApp 安全部分的导航应该受到用户 ID 和密码的控制，而不应该提供能使用户避开这些控制的导航机制。

效率。WebApp 的设计和界面应该优化用户的工作效率，而不是优化设计和构建 WebApp 的 Web 工程师的效率，也不是优化运行 WebApp 的客户端 - 服务器环境的效率。Tognozzi［Ton01］在讨论这个问题的时候写道：

> 这个简单的事实就是，为什么对于参与一个软件项目的每一个人来说，意识到提高用户的生产率目标的重要性以及理解开发有效的系统和提高用户效率的根本区别非常重要。如果要达到这个目标，这个事实也是实现工程师和人机界面设计者之间密切不断的合作、沟通和共谋的需求的关键。

灵活性。界面应该足够灵活，既能使一些用户可以直接完成任务，也能使另一些用户以比较随意的方式浏览 WebApp。在每一种情况下，界面都应该能够使用户认识到他们在哪里，并且给用户提供撤销错误及从选错的导航路径返回的功能。

焦点。WebApp 的界面（和界面表示的内容）应该保持以手边的用户任务为焦点。在所有的超媒体中，都倾向于将用户引导到并不紧密相关的内容。为什么会这样呢？因为这很容易做到！问题是，在支持信息的很多层次中，用户会很快迷失，并且会错过他们最希望看到的最初内容所在的站点。

费茨法则（Fitt's law）。“到达目标所用的时间是到这个目标的距离和目标规模的函数”［Tog01］。在 20 世纪 50 年代进行的研究的基础上［Fit54］，费茨法则“是对快速和有目的的活动进行建模的有效方法，在这一活动中，一个附属肢体（像一只手）在某个特定的起始位置开始，并移动到一个目标区域内停下来。”如果一个用户任务定义了选项或标准化输入的顺序（选项有很多不同的排列顺序），那么第一个选择（例如，鼠标的选择）物理上应该与下一个选择靠

⊖ Tognozzi 的初始原则在本书中进行了改编和扩展。对这些原则的进一步讨论参见［Tog01］。

⊜ Tognozzi［Tog01］提到，确保用户的期望被正确地理解的唯一方式是通过一个全面的用户测试（第 16 章）。

得很近。例如，一个销售消费类电器的电子商务网站的 WebApp 主页的界面，提供了如下的菜单选项：

- 获得礼物建议
- 购买商品
- 比较采购
- 询问问题
- 获得技术支持

在这些选项中，每一个都暗示了一组顺序执行的用户选项或动作。例如，“购买商品”这个选项需要用户首先输入产品类别，然后是产品名称。一旦选择了“购买商品”，产品类别（例如，音频设备、电视机、DVD 播放器）就会以下拉菜单的形式显示，然后，下一个选项就会立即出现（在附近），而且获得此选项的时间是可以忽略不计的。另一方面，如果该选项出现在屏幕另一边的菜单上，那么用户获得选项（然后做出选择）的时间就会变得很长了。

用户界面对象。大量可重用的人机界面对象（和类型）库已经为 WebApp 开发出来。使用这些对象库。一些被最终用户所能“看到的、听到的、接触到的或者以别的方式感知到的”[Tog01] 任何界面对象都可以从大量对象库中获得。

缩短延时。WebApp 不应该让用户等待一些内部操作的完成（例如，下载一个复杂的图形图像），而应该使用多任务处理方式，从而使用户继续进行处理工作，看起来就好像前面的操作已经完成了一样。除了减少等待时间，如果有延迟事件发生，则必须通知用户，从而使用户了解正在发生的事情，包括：1）在选中选项后，如果 WebApp 没有立即做出响应，则应该提供声音反馈（例如，点击声或者铃声）；2）显示一个动态时钟或进度条来表示处理工作正在进行中；3）当产生漫长的处理过程时，提供一些娱乐活动（例如，一个动画或者文本演示）。

易学性。一个 WebApp 的界面应该设计为学习时间是最少的，并且一旦已经学习过了，当再次访问此 WebApp 时，所需要的再次学习的时间也是最少的。一般来说，界面应该强调简单、直观的设计，把内容和功能分类组织，这样对于用户来说很直观。

隐喻。只要隐喻适合于应用和用户，使用交互隐喻的界面就更加易于学习和使用。隐喻应该采用用户熟悉的图片和概念，但是不必是现实世界的精确再现。例如，一个为金融机构实现自动账单支付的电子商务网站，使用支票簿（不会令人感到惊讶）隐喻来帮助用户对账单支付进行详细说明和安排账单支付活动。然而，当用户“填写”支票时，用户不需要输入完整的收款人的姓名，而是可以从收款人的名单列表中进行选择，或者通过输入名字的前几个字母就能得到系统的选择提示。虽然隐喻的方式依然是没什么变化，但是用户从 WebApp 中得到了帮助。

保持工作产品的完整性。工作产品（例如，用户填写的一张表单、用户专用的数据清单）必须被自动保存，使得在有错误发生时这些信息不会丢失。我们每个人都有过这样不愉快的经历：在填写完一张冗长的 WebApp 表单的时候，最终却由于某个错误（我们自己的错误、WebApp 的错误或者客户端到服务器端的传输错误）导致内容的丢失。为了避免这种情况，一个 WebApp 应当设计成自动保存用户的所有专用数据。界面应该支持这个功能，并且为用户提供一个简单的用于恢复“丢失的”信息的机制。

易读性。不论对于老人还是年轻人，界面展示的所有信息都应该是易读的。界面设计者应该着重选择易读的字体、字号以及可以增强对比效果的背景颜色。

跟踪状态。在合适的时候，应该跟踪和保存用户交互的状态，使得用户能够退出系统，并且在稍后返回系统时又能回到他们退出的地方。一般来说，可以设计 Cookies 来存储状态信息。然

而，Cookies 是一种备受争议的技术，其他的设计方案也许对于某些用户来说更合适。

可视的导航。一个设计合理的 WebApp 界面提供了“一种假象，即用户在同一个地方没有动，工作被带到了他们的面前”［Tog01］。当使用这种方法时，导航就不再是用户关心的事情了。相反地，用户检索内容对象，并选择功能，这些功能都是通过界面显示并执行的。

这些设计原则都应该应用于创建初步界面设计的时候和审查设计的时候（参见框注）。



界面团队走查

场景：SafeHomeAssured.com 团队负责人的办公室

参与者：团队负责人和 WebE 团队的其他两名成员

会话：

团队成员 1（对团队负责人）：你有机会看过我们为 SafeHomeAssured.com 做的初步界面设计［图 7-9］了吗？

团队负责人：是的……我从技术角度对它进行了审查，做了一些笔记。我今天早上通过电子邮件把它们发给你了。

团队成员 1：我还没有看我的电子邮件……跟我说一下重要问题的概要吧。

团队负责人：总的来说，你干得不错，没有什么极富创造力的东西，不过它具有典型的界面、体面的美学，合理的布局。你关注到了所有的重要功能……

团队成员 2（可怜地笑着）：但是？

团队负责人：好吧，还有一些内容……

团队成员 1：例如……

团队负责人：你们还记得主页面上的第一次迭代吧，菜单上的主要功能是

关于公司
描述你的住宅
获取 SafeHome 组件的推荐
购买产品
获得技术支持

问题是我们忽略了一些内容，而这些功能又不在同一个抽象层次上。所以我们要对整个内容进行重新组织，而且你们有了这个［指着来自图 7-9 的初始原型］。

团队成员 2（笑）：我喜欢它。

团队负责人：我也喜欢，但是营销部门希望能更突出公司，并且希望把产品的和服务的照片区分开。所以……我认为我们的主菜单应该是：

关于公司
住宅安全
监控服务
我们的技术
联系我们

这些选项的每一个的下面都会有很多子功能。可以认为当选择了一个高层选项时，将为我们弹出一个子菜单。子菜单的每个选项都将把用户带到一个独立的 Web 页面，这些页面有自己的布局、内容对象和功能。

团队成员 1：但是我们不想注册那些想直接进入满足他们的页面的那些用吗？这里没有直接的机制来支持。

团队负责人：你说得对。所以提供一个像我们在前面的界面原型迭代中有的登录是一个不错的想法。我会把它告诉营销部门，但是我认为我们现在就可以开始做了。

在初始界面设计检查中提出的修改之后，WebE 团队开发了如图 9-1 所示的一个新的主页界面设计[⊖]。

⊖ 回忆我们提到过的当 WebE 团队进一步深入设计的时候，初步界面设计（图 7-9）很可能会发生变化。

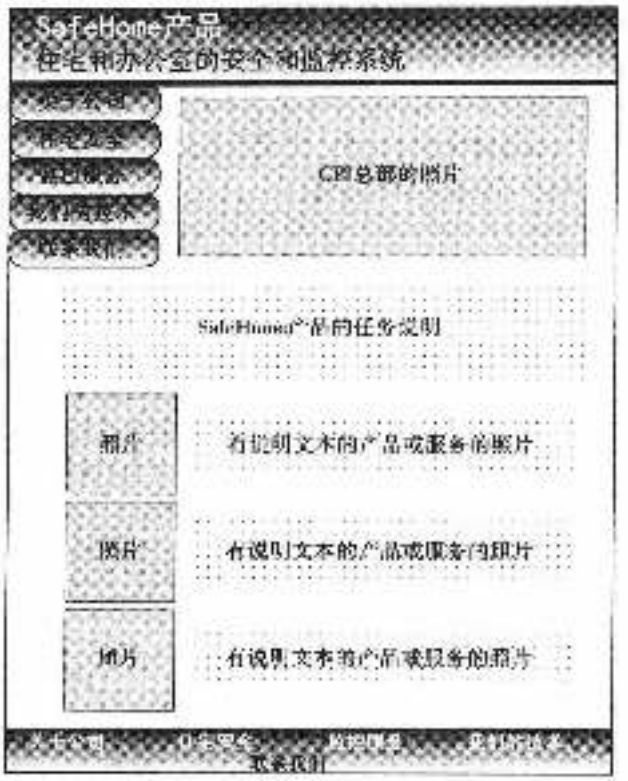


图9-1 主页的初步设计布局

9.1.2 有哪些实用的设计指导原则

Nielsen 和 Wagner [Nie96] 提出了一些实用的界面设计指导原则（基于他们对一个重要 WebApp 的重新设计），对本节前面提到的原则提供了一个很好的补充：

- 对电脑屏幕的阅读速度比对书本的阅读速度大概要慢25%。因此，不要强迫用户阅读大量的文本信息，尤其是当文本的内容是解释 WebApp 的操作或者帮助导航的时候。
- 避免“正在建设中”的标记。用户满怀期望而来，但是却看到这么一个不必要的链接，肯定会很失望或者很沮丧。
- 用户不喜欢使用滚动操作。重要的信息应该布置在一般浏览器窗口都可以显示的范围内。
- 导航菜单和标题条的设计必须是一致的，并且必须出现在用户可用的所有页面上。设计不应该依赖于浏览器功能来辅助导航（例如，返回箭头）。
- 美学效果永远都不应该取代功能性。例如，比起一个源头的但是内容不明确的图像或图标，一个简单的按钮可能是更好的导航选择。
- 即使对于一些偶尔的用户，导航选项也应该是明显的。不应该让用户花不满屏幕搜索才能决定如何链接到其他的内容或服务。

一个好的界面设计能够提高用户对网站提供的内容或服务的理解程度，它并不一定要有闪烁的动画，但是应该是结构合理及功效健全的。

9.2 界面设计工作流

WebApp 的界面设计首先确定用户、任务和环境需求，这些活动是作为在本书前面讨论的需

求收集和建模的一部分来完成的。一旦确定了用户任务，就应该创建和分析用户场景（用例），并定义一组界面对象和动作。

分析模型（第7章）中包含的信息构成了创建描述性设计和符号布置的屏幕布局、定义描述性屏幕文本、标注窗口的初始说明和标题，以及明确说明主要和次要菜单条目的基础。然后使用工具来进行原型化，最终完成界面设计模型。

下面的工作代表了WebApp界面设计的一个基本工作流程：

1. **审查用户特征和种类、用户任务、用例和包含在分析模型中的相关信息，并根据需要进行精化。**

2. **开发一个粗略的WebApp界面布局设计原型。**界面原型（包括布局）可能已经作为分析建模活动中的一部分进行了开发。如果布局已经存在，那么应该根据需要对其进行检查和精化；如果还没有开发界面布局，此时WebE团队就应该和利益相关者合作完成开发。如图9-1所示是SafeHomeAssured.com的一张初步界面设计（作为分析建模的一部分而创建的原型的一个修改版本）。

3. **把用户目标映射到具体的界面行为。**对于大多数WebApp来说，用户主要目标相对较少（一般来说，有4～7个主要目标）。如图9-2所示，应该将这些目标映射到具体的界面行为。本质上来说，设计必须回答下面的问题："界面如何能够使用户完成每个目标？或者如何为相关任务提供启动点？"

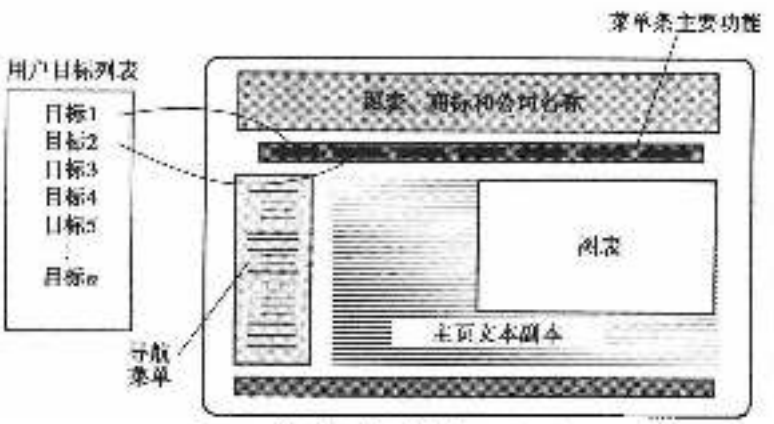


图9-2 把用户目标映射到界面行为

4. **定义和每个行为有关的一组用户任务。**每一个界面行为（例如，"购买一件商品"）和一组用户任务相联系。在分析建模的过程中已经确定了这些任务。在设计期间，这些任务必须与明确的交互对象建立对应关系，这些交互对象包括导航问题、内容对象和WebApp的功能。

5. **为每一个界面行为开发屏幕图像。**当考虑每一个行为时，应该创建序列屏幕图像来描述界面是如何响应用户的交互行为的。应该明确内容对象（即使它们还没有被设计和开发），展示WebApp的功能，并指明导航链接。

6. **利用从美学设计中的输入来精化界面布局和屏幕图像。**Web工程师完成粗略布局，但是重要商业网站的美学感通常是由艺术家而不是技术专家，专业人员（如图像设计者）来开发的。美学设计（在本章后面进行讨论）会和界面设计者完成的工作集成。

7. **确定实现用户界面所需的界面对象。**这个任务可能会需要在现有类库中进行搜索，找出那些适合于WebApp界面的可重用对象（类）。另外，在此时定义任何需要的自定义类。

8. **开发用户与界面交互的过程表示。**这个可选的任务使用UML序列图和活动图来表示用户

与 WebApp 交互时的活动（和决策）流程。

9. **开发界面的行为表示法**。这个可选的任务利用 UML 状态图来表示状态转换和引起转换的事件，并定义控制机制（例如，通过用户可用的对象和行为改变 WebApp 的状态）。

10. **描述每一种状态的界面布局**。使用在任务 2 和任务 5 中开发的设计信息，把确定的布局或屏幕图像与任务 9 中描述的每一个 WebApp 状态联系起来。

在所有这些设计任务进行的过程中应该进行结对走查（第 5 章），而且应该注重可用性。另外，值得注意的是，WebE 团队所选择的设计工作流应当适应将要创建的应用的特定需求。因此，并非对于每个 WebApp 界面，所有的 10 个设计任务总是必需的。

9.3 界面设计的前期工作

WebE 过程模型的一个关键原则是：在你试图设计一个方案之前，最好理解问题。就界面设计而言，对问题的理解意味着要理解：1）将通过界面与系统进行交互的人（最终用户）；2）最终用户要完成他们的工作所必须执行的任务；3）作为界面的一部分进行展示的内容；4）完成这些任务的环境。在后面的几节中，我们将针对界面分析[⊖]对这些元素中的每一个进行详细检查，以给后面的设计任务建立一个坚实的基础。

9.3.1 如何理解 WebApp 用户的特征

在关注技术问题之前，术语*用户界面*（user interface）大概是花时间来理解用户的全部理由。每一个用户都对 WebApp 有一个想象，并且和其他人的想象可能并不相同。另外，用户的想象可能与你的设计模型有极大的不同。使想象和设计模型达成一致的唯一方法是理解用户本身以及这些用户使用系统的方式[⊜]。可以使用如下各种各样来源的信息来加以实现：

用户面谈。WebE 团队成员可采用的最直接的方法是与最终用户进行面谈，以更好地理解用户的需要、动机、工作文化和很多其他问题。这可以通过一对一的会面或者小组访谈来实现。

销售输入。销售人员可以定期与顾客和用户会面，并且能够收集有助于 WebE 团队对用户进行分类并更好地理解用户的需求信息。

营销输入。市场分析在确定市场划分和理解每个划分是如何以轻微的不同方式使用软件方面价值非凡。

支持输入。支持人员可以每天都和用户进行讨论。用户是什么起作用、什么不起作用、用户喜欢什么、不喜欢什么、哪些特性会产生问题以及哪些特性易于使用等信息的最可能的来源。

下面的一组问题（改编自［Hac98］）将有助于你更好地在商业环境下理解一个 WebApp 的用户：

- 用户是训练有素的专家、技术员、办事员还是生产工人？
- 用户受正式教育的平均程度如何？
- 用户有能力从书面材料中学习吗？或者他们已经表达对教室或基于辅导的培训愿望了吗？
- 用户是专业打字员还是恐惧键盘的人？
- 用户群的年龄范围如何？
- 用户将只有一种性别吗？
- 如何补偿用户从事的工作？

⊖ 这些活动中的一些或全部可能已经作为沟通活动（第 4 章）或分析建模（第 7 章）的一部分完成了。

⊜ 对于用户和他们需要的信息，即使不是全部，也是大部分确实应该在需求获取和分析中被收集。如果合适地执行了这些 Web 工程动作，那么你将有足够的信息来理解用户。然而，还有很多次你必须得到这些信息。

- 用户在正常上班时间内工作，还是要一直工作到任务完成？
- WebApp 是用户工作的必要组成部分，还是只是偶尔使用？
- 用户之间主要的口语是什么？用到哪些特定领域的语言？
- 如果用户在使用系统时犯了错误，结果是什么？
- 用户对系统所处理的主题熟悉吗？
- 用户想知道界面背后的技术吗？

这些问题和类似问题的答案将使你理解：最终用户是谁，什么可能激励和取悦于他们，他们是如何被归为不同的用户种类或资料的，他们想象的系统模型是什么，以及用户界面必须有哪些特征来满足用户的需要。

9.3.2 如何详细描述已经识别的内容对象

一个界面设计者还要检查使用场景和从用户那里获得的其他信息，而且还要抽取出用户执行任务所操作的内容对象。这些对象可以进行分类。每一类的属性都被定义，并且应用到每个对象行为的评估为你提供一组操作。

在第 7 章中，我们为 SafeHomeAssured. com 的内容对象（称为**组件**）定义了一棵数据树（如图 7-5 所示）。如果你负责界面设计，那么你将审查内容对象并使用组件的描述来帮助布局界面。你应该认识到，所有的用户将对组件这个对象感兴趣，并且将把它视为 SafeHome 某个组件的一个产品说明。必须要通过界面来展示的关键内容包括产品名称、序列号、器件种类、描述、价格、照片、技术描述和其他相关信息。

然后你创建如图 9-3 所展示的页面布局。这个 Web 页面是每个产品说明的模板。注意除了导航菜单[⊖]被扩展开以反映那些将被应用到组件（产品说明）这个内容对象的任务外，页面格式和图 9-1 所示页面布局保持一致。产品名称、种类、器件号码、描述和照片都如图所示。

通过应用本章前面所讨论的设计原则，界面设计促进易理解性和易用性。例如，用户可以通过“技术细节”这个导航链接来导航到技术细节（并非每个用户都有兴趣，而且把技术细节包含进来会使得页面非常繁忙）而不必在页面中直接提供。其他与产品说明相关的导航链接（“产品定价”、“放置到购物车”和“获取另一个规格说明”）展示在页面的左下角。

9.3.3 用户完成哪些任务

在第 4 章和第 7 章中，我们强调了获得下面这些问题的答案的必要性：

- 在特定的情况下，用户将完成哪些工作？
- 在用户完成工作的过程中，将完成哪些任务和子任务？
- 在完成工作的过程中，用户将操作哪些特定的问题域对象？
- 工作任务的序列（工作流）是什么？
- 任务的层次结构如何？

希望这些问题在设计开始之前已经得到了回答。

在前面几章中，我们提到用例描述了一个参与者（在用户界面设计的上下文中，一个参与者总是一个人）与一个系统的交互方式。当用例用作任务分析的一部分时，被开发为显示一个最终用户如何完成一些特定的与工作相关的任务。在大多数情况下，用例由第一个人以一种非

⊖ 你可以选择为子功能（例如，“产品说明”、“安装”、“获得一个报价单”）使用下拉导航菜单，这样就能一直允许用户访问在主页面中展示的主要功能。

正式的风格（一个简单的段落）来描述。

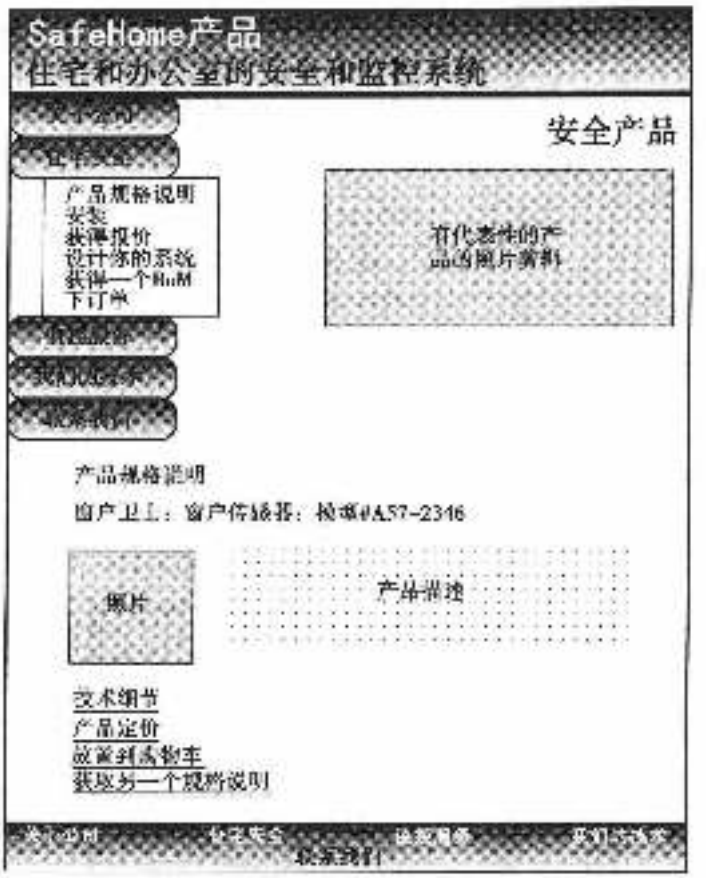


图 9-3 安全产品页面布局

需求获取（第 4 章）之后，SafeHomeAssured. com 的 WebE 团队定义了一个名为“为被监控的空间开发布局”的增量。界面设计者检查这个增量的用例：

用例：为被监控空间开发布局

参与者：任何用户

描述：我想要配置一个安全系统，首先描述一个“空间”（例如，住宅、办公室/家庭空间）的布局，在这个空间中将要安装安全传感器和监控设备。为了完成它，我必须能够使用一个“绘图工具箱”，它将允许我为建筑平面图绘制墙壁、窗户和门。我必须能够指定墙壁、窗户和门的尺寸；还当我确定它们的方向；并描述需要的安全传感器和监控设备。所有的这些绘图元素（墙壁、窗户、门、传感器、监控设备）都必须能够移动和旋转。我希望 SafeHomeAssured. com 能够分析我开发的建筑平面图布局，并为有传感器和监控设备的布置。我想通过名字把一个建筑平面图保存起来，然后检索、编辑和删除它。建筑平面图和传感器的布置一旦完成，我希望能够启动传感器和监控设备的配置。所有的这些动作都必须安全。

这个用例为建筑平面图布局的界面提供了一个基本描述。你可以从中提取任务、对象和交互的总体流程。另外，还可以想出可能取悦用户的扩展 WebApp 特征。例如，在这个用例中没有明确提到多层空间，但是它对于一个完整的建筑平面图布局功能来说却是必不可少的。标识识

到这一点，然后提供额外的功能。产生的建筑平面图布局界面设计如图 9-4 所示。

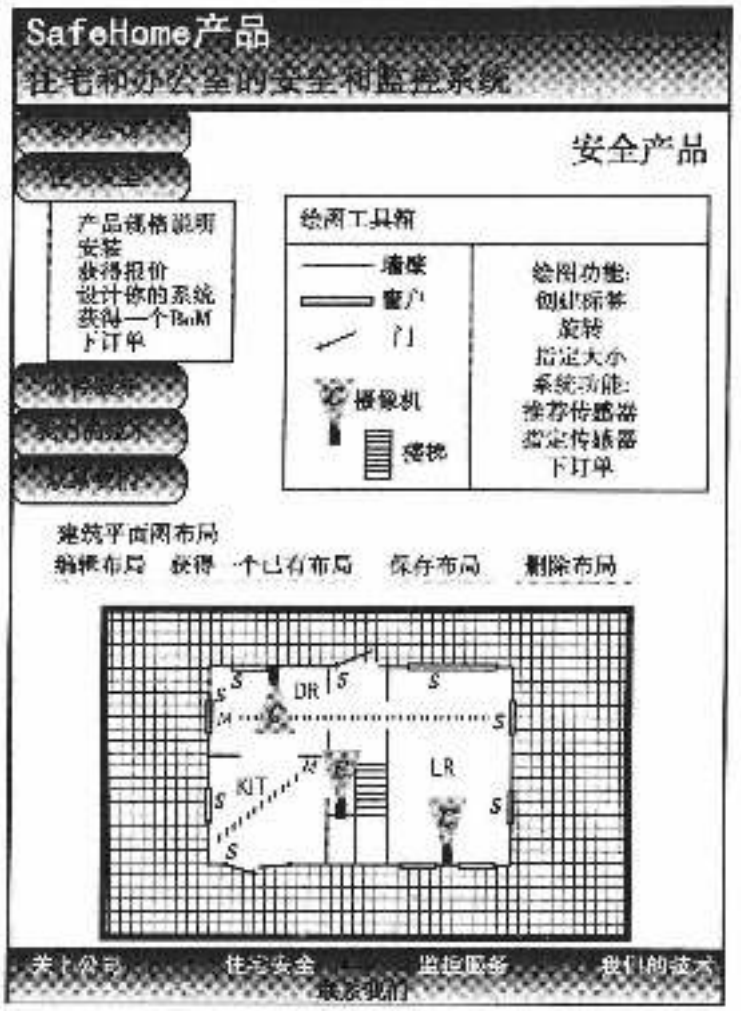


图 9-4 安全产品建筑平面图布局的界面设计

9.3.4 如何详细描述已经确定的任务

任务分析可以通过两种方式加以应用。如同我们已经提到过的，WebApp 常常用来取代新的软件或者手工的或半自动化的活动。为了理解完成活动目标所必须完成的任务，你必须理解人们当前完成（使用现有的方法时）的任务，然后把它们映射到在 WebApp 用户界面中完成的类似（不必是一样的）的一组任务。另外，你也可以研究 WebApp 已有的需求，并得到一组用户任务。

作为一个设计者，即使不考虑任务分析的总体方法，你也必须对任务进行定义和分类。一种方法是逐步精化。例如，我们来考虑用例“为被监控的空间开发布局”。这个用例明确引用了很多主要任务，并且暗含了大量的子任务。每一个主要任务都可以被精化为界面实现的一部分的子任务。例如，考虑下面的用例片段：

> 我必须能够使用一个“绘图工具箱”，它将允许我为建筑平面图绘制墙壁、窗户和门。我必须能够指定墙壁、窗户和门的尺寸；适当地确定它们的方向；并描述需要的安

全传感器和监控设备。所有的这些绘图元素（墙壁、窗户、门、传感器、监控设备）都必须能够移动和旋转。

参见图9-4，一个绘图工具箱（DraftingToolBox，一个内容对象）在界面设计中被实现。在绘图工具箱内部，明确提到了绘图子任务——创建标签、旋转和指定大小。能够使用户“为建筑平面图绘制墙壁、窗户和门”的功能必须实现，但是不在界面中明确命名。为了实现这个绘制功能，要在界面布局的底部对对象墙壁（Wall）、窗户（Window）和门（Door）进行拖放来形成建筑平面图网格。

其他系统功能（推荐传感器和指定传感器）暗含完成这些功能的实现专门算法的一组复杂的功能组件。这些功能组件在用户选择绘图工具箱内合适的链接时进行调用。

链接“下订单”允许用户导航到 SafeHomeAssured. com WebApp 的电子商务部分。但是，必须要创建内部的数据结构来存储已经被手工指定或自动推荐的所有传感器和监控服务的列表。这个数据结构将在后面用于开发完成一个订单所必需的原料账单和报价单。

界面设计必须针对用例进行审查，确保每个显式的和隐式的用户任务都进行了充分的处理。随着设计工作的进行，界面模型必须以和*用户模型*（一个典型的 WebApp 用户描述）以及*系统感知*（用户对 WebApp 的期望）相一致的方式来调节每个任务。

9.3.5 如何为不同角色的不同用户进行设计

当很多扮演不同角色的用户使用同一个用户界面的时候，有时需要逾越任务分析和对象精化，而应用*工作流分析*（workflow analysis）。这种技术可以让你理解当有多个人（角色）参与的时候，如何完成一项工作。考虑一个制药公司，它想实现一个系统，这个系统把开处方和送药的过程完全自动化。整个过程[⊖]将围绕一个基于 Web 的应用，医师（或他们的助理）、药剂师和病人将访问这个应用。工作流可以使用 UML 泳道图（活动图的一种变种）来有效地表示——尤其是当有多个用户或实体参与交互的时候。

我们仅仅考虑工作过程的一小部分——当病人再次要求配药时发生的情况。如图 9-5 所示为一个泳道图，它表示在前面提到的所有三个角色的任务和决定。这些信息也许已经通过面谈或每个参与者写的用例得到了。不论如何，事件流（如图所示）使得界面设计者能够识别出很多关键的界面特征：

1. 每个用户通过界面完成不同的任务；因此，为病人设计的界面观感将和为医生或药剂师设计的界面观感不一样。

2. 为医师和药剂师设计的界面必须适应对来自其他信息源的信息的访问和显示。（例如，药剂师访问药品目录和医师访问替代药物信息）。

3. 在泳道图中提到的很多活动都可以使用任务分析和对象精化（例如，“按处方配药”隐含一个邮购交货、访问药房或访问一个特殊的药物分销中心）来进一步精化。

随着界面的分析，精化过程继续进行。一旦工作流被建立起来，就可以为每个用户类型定义一个任务层次。这个任务层次是通过逐步精化为每个用户确定的每个任务而得到的。例如，考虑用户任务“请求药方再配药”。开发了下面的任务层次：

请求药方再配药

⊖ 这个例子改编自［Hac98］。

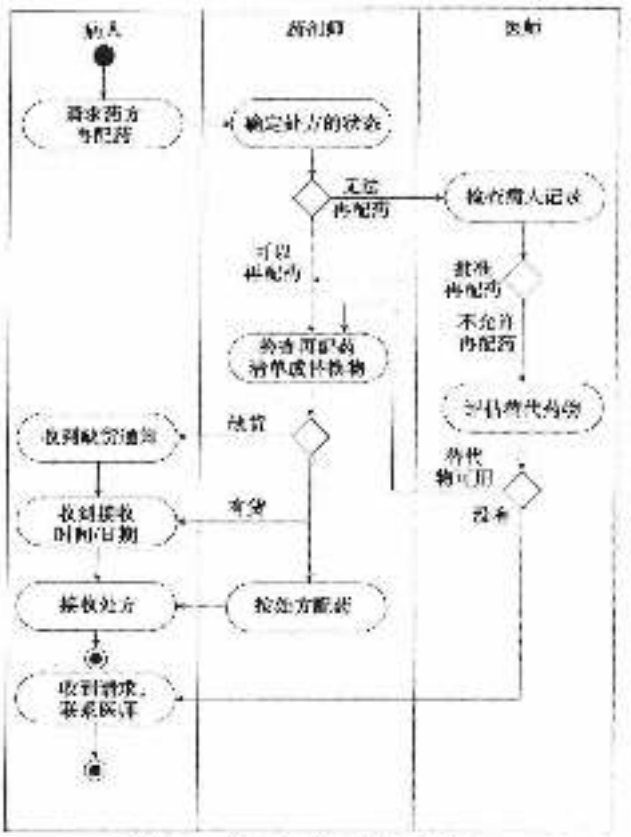


图9-5 药方再配药功能的泳道图

- 提供识别信息
  - 指定姓名
  - 指定 userID（用户 ID）
  - 指定 PIN（个人标识码）和密码
- 指定处方号
- 指定需要处方的日期和时间

为了完成“请求药方再配药”这个任务，定义了三个子任务。其中的一个子任务——提供识别信息，进一步细化为三个子任务的子任务。

9.3.6 内容如何集成到界面描述之中

我们在9.3.5节中确定的用户任务产生对很多不同种类内容的表示。对现代应用来说，显示内容的方式可以是基于字符的报告（例如，电子数据表）、图示表示（例如，柱状图、3D 模型、人的照片）或者专门的信息（例如，音频或视频文件）。内容对象可以是 WebApp 其他部分的组件（与界面无关）所产生的；从应用可以访问的存储在数据库中的数据中获取的；从 WebApp 外部的系统进行传输的。

内容（在它出界面进行展示的时候）的格式和美学必须作为界面设计的一部分加以考虑。下面是你应该考虑的一些问题：

- 不同类型的数据是否被分配到屏幕上一致的位置上（例如，图片是否总是出现在右上角）？
- 用户可以定制内容在屏幕上的位置吗？

- 是否为所有内容都分配了适当的屏幕标识？
- 如果要显示一个大的报告，应当如何对它进行分割才较于理解？
- 对大的数据集来说，是否有可用的机制来在表格移动到综述信息？
- 是否可以对图形化输出进行调节，使其适应所用设备的显示范围？
- 如何使用颜色来加强理解？
- 应该如何给用户显示错误信息和警告？

回答了所有的这些问题，内容展示的设计需求就建立起来了。

9.4 界面设计步骤

一旦完成设计的前期工作，最终用户需要的所有设计任务（或对象和动作）就已经被详细确定了，并且界面设计活动也开始了。和所有的 WebE 设计一样，界面设计是一个迭代的过程。每个用户界面的设计步骤都会重复很多次，每一个步骤都会对在前一个步骤中的信息进行精化和完善。

虽然已经提出了很多不同的用户界面设计模型，但是所有这些模型都是下列步骤的某些结合：

1. 使用在初步设计中开发的信息来定义界面对象和动作（操作）。
2. 定义将引起用户界面状态变化的事件（用户动作）。对这种行为进行建模。
3. 像界面要真正面对最终用户那样对每种界面状态进行描述。创建一个美学原则，并设计所有的导航机制、内容对象和相关信息。
4. 指明用户如何从界面提供的信息来理解系统的状态。

在有些情况下，你可能从绘制每个界面状态的草图（例如，在不同的环境下，用户界面的外观如何）开始，然后反向继续定义对象、动作和其他需要的设计信息。在不考虑设计任务顺序的情况下，你应该：1）总是遵循本章前面讨论的设计原则和指导准则；2）对界面将如何实现进行建模；3）考虑将要使用的环境（例如，显示技术、操作系统、开发工具）。

9.4.1 如何布局界面对象和动作

界面对象和动作一旦完成了迭代的定义和精化，它们就按类型进行了分类。目标、源和应用对象被确定。一个源对象（例如，一个报告图标）被拖放到一个目标对象（例如，打印机图标）之上。这个动作的含义就是创建一个硬拷贝报告。一个应用对象代表特定应用的数据，这些数据本身不作为屏幕交互的一部分被直接操作。例如，用来存储邮件名称的邮件列表。列表本身可能被排序、合并或者清除（基于菜单的动作），但是它并不是通过用户交互来进行拖放的。



设计布局

场景：GH 公司的 Web 工程区域

参与者：作为结对工作的 SafeHomeAssured.com 团队的两个成员，他们正在为用例（通过因特网访问摄像机监控）设计页面布局。

会话：

团队成员 1：我们从分析中得到的这个用例并没有为我们提供很多细节。

团队成员 2：我同意，但它对于开始我们的工作来说可能已经足够了。我们将不得不在需要的时候对它进行精化。

团队成员 1：如果我们有问题，营销部门会来吗？

团队成员2：会来，他们也是团队的一部分。

[团队成员1为摄像机监控页面绘制了一个草图（图9-6）。]

团队成员1：你认为怎么样？

团队成员2：嗯……我认为我们需要做一些修改。首先，看一看这个用例。他们指明要从建筑平面图中选择摄像机、传感器等，而我并没有看到一个建筑平面图。

团队成员1：我本以为它将使页面复杂化，而且，它将需要更多的工作来实现。我使用了摄像机和其他传感器的一个列表，我们可以添加它们的位置……

团队成员2：不，不同的，此外，我们还必须处理我们的这个用例。

团队成员1（叹息）：好的，你说的有道理。那么我们要剖析这个用例，并选出那些我们将要操作的关键内容对象和操作，然后呢？

团队成员2：我们需要确保所有的内容对象都出现在界面设计之内，并且所有的操作都以某种方式得以实现。

团队成员1：你知道的，我刚刚注意到了我们并没有提供任何方法来控制摄像机功能，例如，开/关，缩放/平移。

团队成员2：那是我们必须完成的一个操作。

[两个团队成员继续修订草图，它是图9-7的前身。]

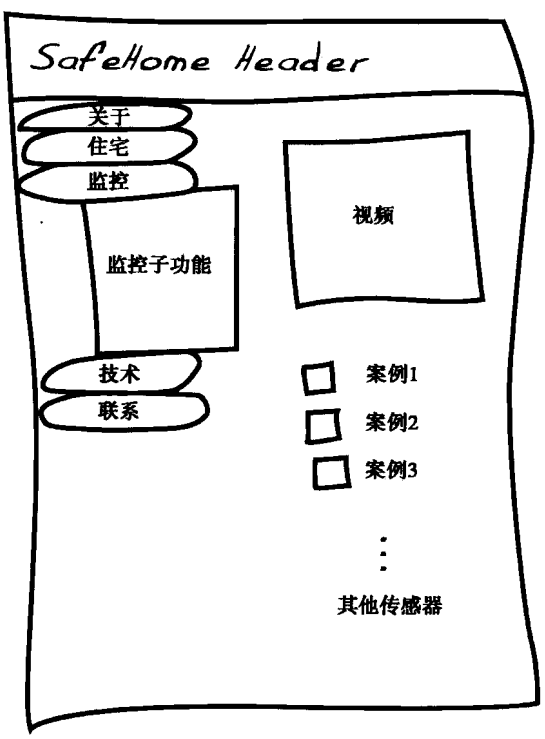


图9-6 摄像机监控界面草图

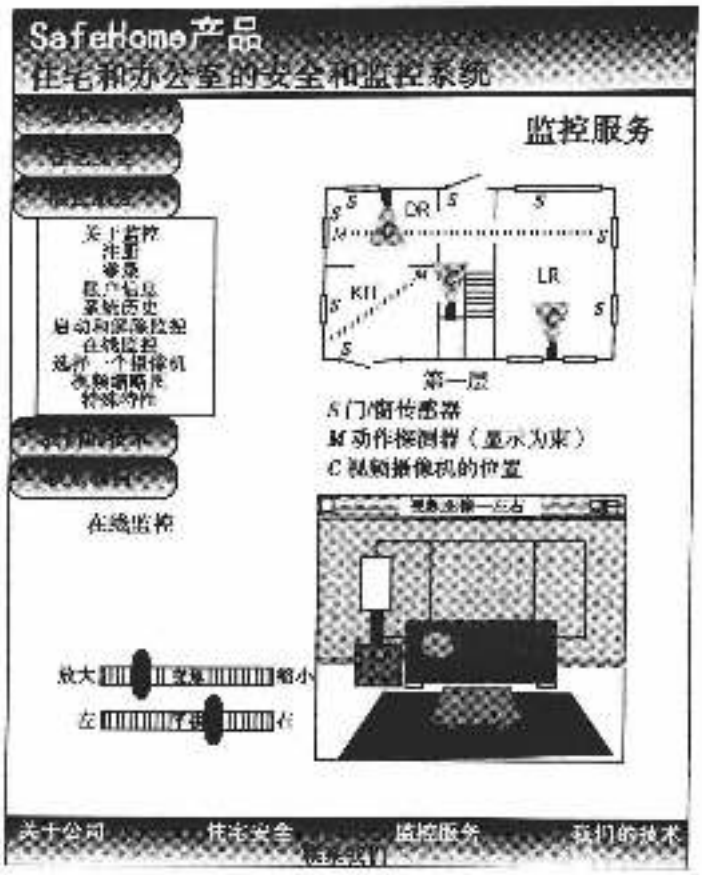


图9-7 监控服务界面的初步设计布局

当你对已经定义的（对于一个设计迭代而言）所有重要的对象和动作都满意时，就完成了屏幕布局。和其他界面设计活动一样，屏幕布局也是一个迭代的过程，在这个过程中，要进行图形设计和图标布置、描述性屏幕文本的定义、导航机制的设计和布置、窗口的详细说明和标题的指定、以及主菜单和次级菜单条目的定义。如果现实世界的一个隐喻适合于这个应用，那么现在就指定它，并且细致这个布局以补充隐喻。

为了简要说明前面提到的设计步骤，我们再次考虑名为“通过因特网访问摄像机监控”的SafeHomeAssured.com功能，假设“房主”参与者角色的利益相关者也许会写下下面的用例：

用例：通过因特网访问摄像机监控

参与者：房主

叙述：如果我在很远的地方，我可以使用具有合适的浏览器软件的任何PC机登录到SafeHomeAssured.com网站。我输入用户ID和两级密码，并且我一旦通过验证，就将获得对我所安装的SafeHome系统的所有功能的访问。要访问一个特定的摄像机图像，我从显示的菜单项中选择“监控”，然后选择“选择一个摄像机”，就出现了住宅的建筑平面图。然后我选择感兴趣的摄像机，或者，我可以选择“所有的摄像机”作为观看选择，同时就可以看到所有摄像机的缩略图快照。一旦选择了一个摄像机，并选择“图像”，一个摄像机图像就会出现在由摄像机ID所识别的一个观看窗口内。如果我想要转换摄像机，我选择“选择一个摄像机”，该画的观看窗口就消失了，同时另一次由

现住宅的建筑平面图。然后我选择感兴趣的摄像机。一个新的观看窗口就出现了。

基于这个用例，识别出下面的房主任务、对象和数据项：

- *访问*SafeHome 系统。
- *输入*一个 ID 和 Passward（密码）来允许远程访问。
- *显示*FloorPlan（建筑平面图）和 SensorLocations（传感器的位置）。
- 在建筑平面图上*显示*VideoCameraLocation（视频摄像机的位置）。
- *选择*用于观看的 VideoCamera（视频摄像机）。
- *观看*VideoImages（视频图像）（每秒 4 帧）。
- 对 VideoCamera（视频摄像机）*进行缩放*或平移。

从这个用户任务列表中抽取出对象（黑体字）和动作（斜体字）。

视频监控屏幕布局的一个初步设计布局被创建出来（图 9-7）[⊖]。为了调用这个视频图像，需要选择监控窗口中显示的建筑平面图中的视频摄像机位置图标——*C*。在这种情况下，选择位于客厅（LR）中的一个摄像机。摄像机图像窗口出现，显示来自位于客厅的摄像机视频。缩放和平移控制滑块用于控制视频图像的放大倍率和方向。要选择来自另一个摄像机的视图，用户从菜单中选择“选择一个摄像机”，然后一个新的视频窗口就取代了原来的窗口。

如图 9-7 所示的界面初步设计布局将使用另外的图形（草图）来进行补充，这些图形展示菜单条展开的每个菜单项，指明对于每个监控服务模式（状态）可用的动作。在对初步设计进行检查之后，出现了两个问题：

- 平移和缩放滑块应当紧挨着置在视频窗口的下方。
- 利益相关者可能希望能够录制视频流并对其进行回放[⊜]。

鉴于这些问题，初步设计被修改为如图 9-8 所示。

9.4.2 如何设计界面导航机制

如我们在本章前面所提到的那样，一个 WebApp 的界面的目标是：1）确立 WebApp 的内容和功能的一个一致的窗口；2）指导用户进行与 WebApp 的一系列交互；3）组织用户可用的导航选项和内容。为了实现一致的界面，你应该首先使用美学设计（在 9.5 节中考虑）来为界面建立一致的外观。这包括很多特征，但是必须强调导航机制的布局和形式。为指导用户进行交互，你可以利用一个能够使用户获得对界面的直观理解的合适的隐喻[⊝]。为了实现导航选项，你可以从如下的交互机制中选择一个：

- **导航菜单**。关键字菜单（垂直或水平组织）列出关键的内容和（或）功能。这些菜单被实现以后，当用户选择了一个主菜单选项的时候，就会出现子主题的一个层次，用户可以从中选择。参见图 9-8，关键字菜单被垂直组织在 Web 页面的左边。选择一个主功能（例如，监控服务）会引起如图所示的一个子菜单的出现。然后可以选择菜单内包含的子功能来满足用户需要。
- **图标**。按钮、开关和类似的图形图像能够使用户选择一些特性或指定一个决定。再次参见图 9-8，视频图像的滑块开关和录像机控制是与一个视频隐喻一致的图标。

⊖ 注意这和在前面章节中的这些特征的实现有些不同。它可以视为一个设计初稿，并且展示了一个可以考虑的候选布局。

⊜ 这代表了 WebApp 的一些特征，同时将必须通过协商来确保增量的交付日期可以被保存或调整。

⊝ 在这个上下文中，一个隐喻是一个可以在界面上下文中被模型化的表示（来自于用户的真实的经历）。一个简单的例子可能是用于控制一个 .mpg 文件音量的滑动开关。

- **图形图像**。图形表示是用户导航的，同时实现了到内容对象或 WebApp 功能的链接。在图 9-7和图9-8 中，建筑平面图布局包含用户可以选择的很多图形图像（例如，摄像机符号）。

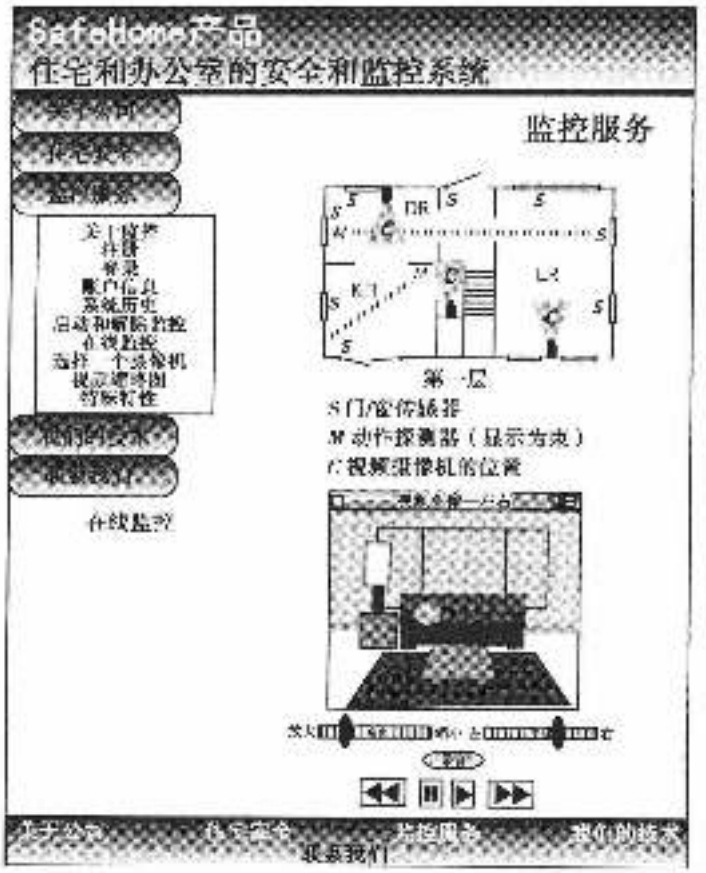


图 9-8 监控服务在线面板的修改后的设计布局

在内容层次的每一层，应该提供这些控制机制中的一个或多个。

把导航机制布局（一个界面设计问题）和 WebApp 的导航设计区分开是很重要的。在第 10 章中讨论的导航设计的重点是内容对象之间的和 WebApp 所有功能的导航流语义。界面设计使用导航机制的语法（菜单、图标和图像）来实现这些语义。

9.4.3 为什么界面的一致性如此重要

一个设计良好的 WebApp 界面不管在内部还是外部都应该是一致的。就像我们曾经提到的，内部一致性需要每个 Web 页面的美学布局的观感、导航机制以一致的方式进行使用、内容对象有着相同的基本设计特征、并且来自 WebApp 的输入和输出展示出可预测的和令人愉悦的灵活性。内部一致性使用户能够建立一个带来高可用性环境的交互节奏。

不过仅有内部一致性是不够的。WebApp 还必须展示出外部一致性。不论是何种情况，到一个 WebApp 的典型访问者都不会在那里花太多的时间。访问者很可能是在访问了其他网站之后到达这个 WebApp 的，而且也将会离开这个 WebApp 而去访问其他的 WebApp。Nielsen [Nie00] 曾经提到："……用户感觉他们正在把 Web 用作一个整体而不是任何特定网站。"正因为这样，你的 WebE 团队必须遵循当前的界面设计约定，使你的 WebApp 界面约定与 Web 上使用的那些是一致

的。如果你不这样做（毕竟，我们每个人都想富有创造性并打破常规），那么你可能会疏远那些可能没有时间或爱好学习一个新交互模式的访问者。

9.5 美学设计

最好的 WebApp 界面会高效地显示其内容和功能，同时它们在美学上也应当是令人愉悦的。美学设计，又称为图形设计，是一种艺术，它是对界面设计和内容设计技术方面的补充。没有它，一个 WebApp 可能有强大的功能，但却不能吸引人；有了它，一个 WebApp 就能够将它的用户吸引到一个以用户为核心的充满智慧的世界。

但是什么是美学呢？有一句谚语，“情人眼里出西施。”当考虑 WebApp 的美学设计时，这句话特别贴切。为了进行有效的美学设计，我们再一次回到作为分析模型的一部分而开发的用户层次（第 7 章），并提出这样的问题，“谁是 WebApp 的用户，他们希望什么样的 WebApp 外观？”

9.5.1 如何创建令人愉悦的布局

每一个 Web 页面中能够用于支持非功能性的美学设计、导航特征、信息内容和用户控制功能的“空间”都是有限的，应该在美学设计时对这种空间的“开发”进行规划。

和所有的美学问题一样，在设计屏幕布局时，没有绝对的规则。然而，很多通用的布局规则是值得考虑的：

- **不要害怕空白。**把 Web 页面中每个地方都塞满信息是不可取的。如果非要这样做，产生的杂乱将使用户难以寻找需要的信息或要素，并造成很不舒服的视觉混乱。参见前面的图 9-3，设计者选择提供链接链到描述技术细节内容，而不是把这些内容包含在现有页面中。原因是为了避免页面变得过于繁忙，同时保留一些空白。
- **重视内容。**毕竟，内容才是用户浏览到网页的根本原因。Nielsen［Nie00］建议，典型的 Web 页面的 80% 应该是内容，剩余的资源用于导航和其他要素。参见前面的图 9-1，SafeHomeAssured. com 的主页面几乎全是内容，为用户提供了一幅 SafeHome 产品的图片。
- **以从左上到右下的顺序组织布局元素。**绝大多数用户浏览 Web 页面的方式和看书的方式一样——从左上到右下。如果布局元素有特定的优先级，那么应该将高优先级的元素布置在页面空间的左上部分。
- **在页面内按照位置对导航、内容和功能进行分组。**人们几乎在所有事情上寻找模式。如果在一个 Web 页面内没有可辨别的模式，那么用户的挫败感就很可能会增加（归咎于对所需信息的不必要查找）。SafeHomeAssured. com 的界面设计（例如，图 9-3）把所有的主要功能布置在左上角，并把所有的内容从左到右、从上到下进行显示。
- **不要用滚动条来扩展你的空间。**虽然说滚动经常是需要的（尤其是当你向下移动获取详细内容的页面的时候），但是大多数的研究表明，用户不喜欢滚动。最好是减少页面的内容，或者在多个页面上显示必要信息。SafeHomeAssured. com 的界面设计者违反了这个规则（可能需要通过滚动来到达某些功能或内容），认为必要的内容优先于对滚动限制的要求。
- **在设计布局时，考虑分辨率和浏览器窗口的尺寸。**设计应该把所有的布局元素指定为可用空间的百分比，而不是在布局中定义固定的尺寸［Nie00］。
- **为自由导航设计布局。**所有 WebApp 页面的通用布局都应当假设用户不会以预期的方式导航到页面（例如，通过来自一个搜索引擎的直接链接）。布局设计应当适应这种不可预测的访问，并且还不能产生困惑（对用户而言）。确保用户在没有通过安全验证的情

况下不能导航到安全页面也很重要。例如，如果一个访问者想直接导航到如图 9-8 所示的 SafeHomeAssured.com 监控服务页面，那么这个访问者将被自动重定向到登录页面。

- 不要假设布局在不同的显示设备和浏览器之间将保持一致。使用那些在绝大多数流行的浏览器上都可以进行正确转换的要求设计布局，使得布局在大的或小的显示设备上都有效。
- 如果使用照片，那么就使它们变成小格式，并提供放大选项。大的 JPEG 文件非常耗时间下载的。大多数用户在有查看大尺寸照片的选择的情况下，对缩略图照片都是满意的。
- 如果你想在所有 WebApp 页面内有内聚的布局、外观、感觉，那么就使用层叠样式表（CSS）。一个 CSS 可以让你在所有 Web 页面内指定一种观感（例如，字体类型、字号和样式）。另外同等重要的是，CSS 使你能够通过修改一个文件而改变所有页面的观感。

Nielson [Nie00] 在如下陈述中建议一个有趣的设计实践："……审查你的设计元素，并且一个一个去掉。如果你的设计没有其个特定元素可工作，就去掉它。简洁总是胜过复杂……"



审查美学设计

场景： CPI 公司的 Web 工程区域

参与者： 为结对工作的 SafeHomeAssured.com 团队的两个成员，他们正在进行产品说明页面（图 9-3）美学设计的一个结对走查

会话：

团队成员 1（看着图 9-3）：我对这基本满意。它看起来可以工作。

团队成员 2：我一直在思考，我们有一些美学问题。

团队成员 1：什么问题？功能在左上角进行了很好的描述；我们使用了足够的空白；产品信息很容易在页面上找到。

团队成员 2（温和地打断）：我对产品信息很容易找到不是那么确定。使用小屏幕的用户将不得不进行滚动来看到它。更糟糕的是，他们将必须滚动才能看到链接到技术细节、产品定价等的链接。

团队成员 1（想了一下）：是的，你的关于滚动的说法是对的。那我们去掉照片剪辑并把产品说明的内容向上移怎么样呢？

团队成员 2：那样应该可以，但是我们必须确保不会让事物变得拥挤。记住，空白。[暂停了一会儿。] 还有问题。

团队成员 1（打断）：在你开始之前，我刚注意到一些问题。我们从来都没有提供一个增加组件照片尺寸的机制。我会记录它，这样我们才能做出变化。你想说什么？

团队成员 2：在美学上，我认为我们可以使用到技术细节、产品定价等的链接来做得更好。现在，那些都是非常基本的，而且看起来很乏味，你认为呢？

团队成员 1：实际上，你是对的，同时那些链接处在页面的下端并且从其他功能那里移来不是最好的。也许我们需要做的是创建一个在选择了"产品说明"时会弹出的第三层功能。

团队成员 2：我喜欢这种做法。所有的功能都出现在左上角。不过我们必须保证功能的层次不会变得过于繁杂。

团队成员 1：它是可行的。但是这个功能层次必须出现在所有需要它的页面上——一致性和其他所有要求。

团队成员 2：我们去做出一些变更。

团队成员 1：这很容易。

9.5.2 什么是好的图形设计

图形设计考虑 WebApp 观感的各个方面。图形设计过程从布局开始，然后考虑合适的颜色配置，文本类型、字号和风格，对补充媒体（例如，音频、视频、动画）的使用，以及一个应用

的所有其他美学元素。

对 WebApp 图形设计问题的全面讨论超出了本书的范围。你可以从很多到这个主题有贡献的网站上，例如：

Design & Publishing eZine	www.graphic-design.com
Grantastic Designs	www.grantasticdesigns.com
Web Page Design for Designers	www.wpdfd.com

或者从一些出版资源（例如，[Bag01]、[Clo01] 或者 [Hei02]）中获得设计技巧和指导准则。

交互设计

设计良好的网站

有时候，要理解好的 WebApp 设计，最好方法就是看一些例子。Marcelle Toor 在他的文章 "The Top Twenty Web Design Tips" 中（www.graphic-design.com/Web/feature/tips.html）[Too02] 建议将下面的网站作为好的图形设计的范例。

Primo Plastics，www.primo.com Primo Angeli 领导的一个设计公司。

Workbook Assignment Profolios，www.workbook.com 一个展示插图画家和设计者作品的网站。

River of Song，www.pbs.org/riverofsong 针对公共电视的电视连续剧和关于美国音乐的无线电广播。

Design Office of RKD，www.RKDINC.com 一个有在线代表作品选和好的设计技巧的设计公司。

Creative Hotlist，www.commarts.com/career/index.html《Communication Arts》（沟通艺术）杂志，一本图形设计师的行业期刊，介绍了其他很多设计良好的网站。

BTD/Beth Tondreau Design，www.btdnyc.com Beth Tondreau 领导的设计公司。

9.6 可用性

在一篇关于可用性方面有着深刻见解的论文中，Larry Constantine [Con95] 提出了一个与可用性主题紧密相关的问题："用户究竟想要什么？"他给出了这样的回答：

> 用户真正想要的是好的工具。所有的软件系统，从操作系统和语言到数据录入和决策支撑应用，都是工具而已。最终用户希望从为他们设计的工具中得到的和我们希望从所使用的工具中得到的一样。他们想要那些易于学习并且能够帮助他们工作的系统。同时，他们想要的软件应该不会降低他们的工作效率，不会妨碍他们或使他们产生困惑，不会使他们易于把错成难于完成工作。

Constantine 指出，可用性并非源于美学、交互机制的发展水平或者内置的界面智能等方面。当界面架构适应于将要使用这些界面的用户的需求时，才获得可用性。

正式的可用性的定义往往令人有些迷惑。Donahue 和他的同事 [Don99] 给出了如下的定义："可用性是一种衡量计算机系统好坏的度量……便于学习；帮助初学者记住他们已经学到的东西；减小出错的可能性；使经用户更加有效率；并且使他们对系统感到满意。"

确定你所创建的 WebApp 内是否存在"可用性"的唯一方法就是：在界面设计建立之后，进行一个可用性评估。观察用户与 WebApp 设计原型的交互，同时回答下面的问题 [Con95]：

- 在没有连续的帮助或说明的情况下，WebApp 是否便于使用？
- 交互和导航规则是否能够帮助一个初学的用户更加高效地工作？
- 当用户具备相关知识后，交互和导航机制是否变得更加灵活？

- WebApp 是否进行了调试，使之适应其将运行的物理环境和社会环境？
- 用户是否能意识到 WebApp 的状态？用户是否一直能知道自己在哪里？
- 界面是否是按照一种合理的并且一致的方式来组织的？
- 交互和导航机制、图标和程序是否与界面相一致？
- 交互是否能够提前发现错误并帮助用户修正它们？
- 界面是否能够容错？
- 交互是否简单？

如果上述每个问题的回答都是“是”，那么可以认为这个 WebApp 达到了可用性。

可用性系统带来的很多可量化的好处在于：提高销售量和顾客的满意度，具有竞争优势，在媒体中获得良好的评论，获得更好的口碑，降低支持成本，改进最终用户的生产率，减少培训成本，减少文档开销和减小来自不满意顾客的诉讼的可能性［Don99］。

9.7 设计问题

在 WebApp 界面设计时，几乎总是会遇到系统响应时间、用户帮助设施、错误信息处理、可访问性和国际化五个方面的设计问题。不幸的是，很多设计者往往在设计过程的很晚阶段才处理这些问题（有时一个问题的初始迹象到一个可运行增量被部署以后才会出现）。不必要的反复、项目延迟和顾客的挫败感常常会发生。最好的办法是，在界面设计的初期，就将每一个问题建立为一个设计问题进行考虑，因为此时比较容易修改（至少与在开发后期解决这些问题所遇到的困难相比是这样），而且代价也低。

9.7.1 哪些因素影响响应时间，怎么改进

WebApp 的系统响应时间被很多因素复杂化：服务器的瞬时负载、客户端 - 服务器连接的带宽、客户端的处理速度、内容和产生内容的功能的复杂性以及浏览器的能力。应当不要奇怪，很多对 WebApp 的主要抱怨就是响应时间。一般来说，系统响应时间的度量是从用户执行某个控制动作（例如，点击返回键或者点击鼠标）到 WebApp 以预期的输出或动作形式来给出响应。

系统响应时间包括两个方面的重要特性：持续时间和可变性。如果系统响应的持续时间过长，用户就会感到挫败和压力。系统响应时间的可变性是指和平均响应时间的偏差，在很多方面这是最重要的响应时间特性。即使响应时间比较长，响应时间的低可变性也可以使用户建立稳定的交互节奏。例如，稳定在 1 秒的命令响应时间常常要比从 0.1 秒到 2.5 秒变化的响应时间好。当可变性很显著的时候，用户总是失衡的，总是在想界面后面是否发生了什么异常。

因为有如此多的因素影响响应时间，所以设计者要创建一个可以保证短的可预测（低变化性）响应时间的设计（在任何层次上）是很困难的。在界面设计层次[⊖]，将 WebApp 的特定需求和它的总体硬件、软件和通信特征进行权衡是很重要的。例如，如果 WebApp 可能服务于大量的用户，那么服务器的配置必须设计为能适应这个负载；如果平均用户的通信带宽预测为相当低，那么就应该尽可能避免使用大的内容对象（如大的照片、图形文件或者视频内容）；如果一个处理功能非常复杂并且是计算密集型的，那么最好把它在客户端和服务器端的功能分离使处理速度最大化，或者也可以在某个选择（处理功能或内容对象）会需要比通常长得多的响应或下载时间时警告用户。也可以提供渐变的时间刻度，为用户提供操作完成比例的指示。在极端情况下，使用诸如 AJAX 这样的技术可以克服 Web 的一些限制并改进响应时间。

⊖ 值得注意的是，系统的响应时间也是内容设计、组件级设计或架构设计的一个功能。

9.7.2 如何设计“帮助”设施

你应当努力创建一个直观的、因此完全可自我解释的 WebApp 界面。但是基于计算机的交互系统的几乎每个用户都会时常需要帮助。因此，你应该为那些需要帮助的用户提供一组明显的机制。用户一旦导航到“帮助”设施，就应该提供分级解决环境：

1. 大多数的常见问题都可以由一个常见问题解答（Frequently Asked Questions，FAQ）功能来回答。但是很多 WebApp 都把这个作为一个事后的想法来设计，使得 FAQ 很少，因此也就没有什么用处。FAQ 应当通过主题来进行分类，有各种常见和不是那么常见的问题（有规律地进行更新的问题），并在其响应级别上进行精化。

2. 应该为特别复杂的网站提供联机帮助。用户指南应该有一个详细目录、一个搜索功能和一个特别详细的索引。

3. 应该提供一个技术支持链接作为最后的手段。

当开发联机帮助设施的时候，必须解决如下诸多设计问题［Rub88］：

- *在系统交互的过程中，对于所有的 WebApp 功能在任何时间都可以得到帮助吗？*有两种选择：提供部分功能和动作的帮助和提供全部功能的帮助。
- *用户将如何请求帮助？*有两种选择：“帮助”菜单或导航图标（例如，一个程式化的问号）。
- *如何展示帮助？*有两种选择：在另一个窗口中指示参考某个 PDF 文档（不是理想方式），或在屏幕特定的位置给出一行或两行简单提示。
- *用户如何通过导航返回到正常交互？*选择包括屏幕上显示返回按钮或者浏览器上的返回按钮（不是理想方式）。
- *将如何构造帮助信息？*选项包括：“平面式”结构，所有的信息都通过一个关键字来访问；或者分层式信息，用户可以通过超链接进入到结构中得到更详细的信息。

纯粹主义者认为，当界面设计者做了正确的工作时，这些都是没有必要的。但是实用主义者（我们的立场）认识到，即使是坚固的设计也无法预见并消除每个问题。

9.7.3 界面应当如何处理错误

一个设计良好的 WebApp 界面应该预见错误并帮助用户来避免错误。然而，当确实发生错误时，界面应该提供一个即时的、可辨认的和可理解的错误通知。

以一个通常会遇到的错误避免和通知都很弱的场景为例，考虑如下场景：

> *商业 WebApp 中在下载或者订购之前，用户必须选中一个复选框以指明他同意遵守许可或者其他规则。这个复选框通常很小，而且位于一个在视觉上偏僻的地方。因此，它常常都被忽略而没有选中。用户提交他的“订单”，但是什么也没有发生。用户感到很奇怪，就会快速浏览 Web 页面，在查找之后发现了如下信息（常常是红色的小文本）：因为有错误发生或者信息丢失，你的请求无法处理。*
>
> *并没有指明发生了什么错误或丢失了哪些信息。用户必须再次检查他的动作，直到他发现没有选中复选框为止。这是一串令人沮丧的和消耗时间的事件序列。*

在这个场景中，我们遇到了两个问题。第一，界面设计者没有首先帮助用户避免错误。设计者把复选框放在一个视觉上偏僻的地方，使某个错误（没有选中复选框）的可能性大大增加。第二，错误信息本身很弱。它应该这样陈述：你忘记选择许可复选框，请现在选中并重新提交。这没有歧义，也不会浪费时间。

通常，WebApp 给出的每个出错信息或警告应具备如下特征：

- 消息应以用户可以理解的语言描述问题。
- 消息应明确指出用户有哪些选择来对问题进行矫正和（或）从错误中恢复。
- 消息应指明错误（例如，可能损坏的数据文件）可能导致的所有不良后果，以便用户检查以确保没有发生错误（或者如果错误发生进行改正）。
- 消息应伴随有听觉的或视觉上的提示。也就是说，显示消息时应该伴有嘟嘟的警告声音，或者消息用闪烁的或以一种明显表示错误的颜色来显示。
- 消息应当是非主客观的，即指问不能指责用户。

因为没有人真正喜欢坏消息，所以不论一个出错消息的设计有多好，用户都不会喜欢。但是从道理上讲，一个有效的出错消息可以在很大程度上帮助提供 WebApp 的质量，而且在问题发生时，将显著减少用户的挫败感。

9.7.4 什么是“可访问性”？如何应用于界面设计中

随着 WebApp 变得无处不在，Web 工程师必须确保界面设计中包含一些使得有特殊需求的用户易于访问的机制。对那些可能在实际上面临挑战的用户（和开发者）而言，基于道德、法律和业务等方面的原因，可访问性是必不可少的。很多可访问性的指导准则（如［W3C03］）为实现不同等级可访问性界面的设计提供了详细建议。其他的准则（如［App07］和［Mic07］）为“辅助技术”提供了专门的指导，这些辅助技术用来解决那些在视觉、听觉、活动、语言和学习等方面有障碍的人员的需要。开发可访问软件的准则也可以在 IBM 的 Human Ability and Accessibility Center（人类能力和无障碍中心）（www-03.ibm.com/able/access_ibm/disability.html）上找到。



界面开发中的设计问题

场景：CPI 公司的 Web 工程区域

参与者：作为结对工作的 SafeHomeAssured.com 团队的两个成员，他们已经完成了“摄像机监控”这个增量的界面设计

会话：

团队成员 1（看着图 9-8 并审查相关的工作产品）：你知道，在我完成文档时，一些事情发生了。

团队成员 2：什么事？

团队成员 1：我们忘了指定对任何特定增量的帮助设施或对任何错误处理特征的设计。

团队成员 2（做了个鬼脸）：嗯，你说得对，但是我们必须要进入部署阶段了，否则我们将永远无法按时交付这个增量。

团队成员 1（想了一下）：我同意。我们为什么不花一个小时来做出摄像机监控的具体错误条件的一份列表呢，这样我们就可以在构建过程中设计和实现错误处理。

团队成员 2：好策略。我们可以为摄像机监控“帮助”做同样的事情，然后在构建过程中完成帮助设施。

团队成员 1：也许我们应该和营销部门商量一下，确定他们希望帮助设施有多复杂。没有人讲过太多关于帮助的事情，而且在用例中也没有提到。

团队成员 2：但是不管他们是否提到，我们都需要一个帮助设施。

团队成员 1：同意。但是我们可以做一个复杂的或简单的设施。

团队成员 2：既然没有人提及，那么我们为什么不为这个增量使用一个简单的 FAQ 呢。如果我们得到的反馈要求得更多，我们将在另一个 WebApp 增量中进行解决。

团队成员 1：在给定的时间线下，这是一个合理的方法。我们就这样做吧。

团队成员 2：好的。我们首先做出可能出错条件的列表。

9.7.5 什么是“国际化”？如何应用到 WebApp 中

Web 工程师和他们的经理往往会低估创建适应于不同国家和语言需要的用户界面所需要的工作和技能。界面经常是为一个国家和一种语言所设计的，然后在面对其他国家时只好应急对付。界面设计者面临的挑战就是设计出“全球化”的 WebApp。也就是说，用户界面应该被设计成能够容纳需要交付给所有 WebApp 用户的通用核心功能。本地化特征使得界面能够针对特定文化市场进行定制。

Web 设计者有很多可用的国际化指导准则（如［IBM07］）。这些准则解决很多设计问题（例如，屏幕布局在不同的市场情况下可能是不同的），同时离散地实现问题（例如，不同的字母表可能生成特定的标记和间距）。已经开发出了统一字符编码标准（Unicode）［Uni07］来解决如何管理具有成千上万的字符和符号的很多自然语言所带来的令人生畏的挑战。

9.8 本章小结

用户界面是访问 WebApp 内容和功能的窗户。在很多情况下，界面塑造了用户对系统质量的看法。如果“窗户”污脏、不平或者破损，那么用户可能会拒绝强大的功能和内容。事实上，一个设计拙劣的界面可能造成一个具有良好的设计和稳固的实现的应用的失败。

大量原则和指导准则可以帮助设计者创建有效的用户界面。可以总结为如下规则：1）置用户于控制之中；2）让界面易于交互；3）保持界面的艺术性和一致性。应当进行一个有组织的设计过程以实现遵守这些规则的界面。

界面设计从一系列的分析任务开始，分析任务定义了各种最终用户的资料，并描述与用例、任务和对象精化、工作流分析和分层任务表示相关的用户任务和动作。

一旦任务识别完毕，就可以从描述每个任务的用户场景中抽取一组界面对象和动作。这为创建屏幕布局提供了基础，屏幕布局描述了图形设计和图标的布局、描述性屏幕文本的定义、窗口的规格说明和标题以及主、子菜单项的规格说明。在对设计模型进行精化的过程中，需要考虑诸如响应时间、命令和动作结构、错误处理和帮助设施等设计问题。

在后续的设计章节中，我们将分析 WebApp 的设计方面。WebApp 设计将从用户的世界慢慢移出，转向作为基于计算机的系统的 WebApp 的技术领域。在第 10 章中，我们将从考虑信息设计（对内容对象的表示以及允许用户在内容对象之间进行导航的机制）开始考虑 WebApp 的设计。

参考文献

［App07］ Apple Inc., Accessibility, 2007, www.apple.com/accessibility/ (accessed August 6, 2007).

［Bag01］ Baggerman, L., and S. Bowman, Web Design That Works, Rockport Publishers, 2001.

［Clo01］ Cloninger, C., Fresh Styles for Web Designers, New Riders Publishing, 2001.

［Con95］ Constantine, L., “What DO Users What? Engineering Usability in Software,” Windows Tech Journal, December 1995, www.forUse.com (accessed August 6, 2007).

［Dix99］ Dix, A., “Design of User Interfaces for the Web,” Proc. of User Interfaces to Data Systems Conference, September 1999, www.comp.lancs.ac.uk/computing/users/dixa/topics/webarch/ (accessed August 6, 2007).

［Don99］ Donahue, G., S. Weinschenck, and J. Nowicki, “Usability Is Good Business,” Compuware Corp., July 1999, http://www.half-tide.com/ResourcesFiles/UsabilityCost-BenefitPaper.pdf (accessed August 6, 2007).

[Fit54] Fitts, P., "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," Journal of Experimental Psychology, vol. 47, 1954, pp. 381-391.

[Hac98] Hackos, J., and J. Redish, User and Task Analysis for Interface Design, Wiley, 1998.

[Hei02] Heinicke, E., Layout: Fast Solutions for Hands-on Design, Rockport Publishers, 2002.

[IBM07] IBM, "Overview of Software Globalization," 2007, www-128. ibm. com/developerworks/xml/library/x-i18n1. html (accessed August 6, 2007).

[Mic07] Microsoft, Accessibility, 2007, www. microsoft. com/enable/ (accessed August 6, 2007).

[Nie00] Nielsen, J., Designing Web Usability, New Riders Publishing, 2000.

[Nie96] Nielsen, J., and A. Wagner, "User Interface Design for the WWW," Proc. CHI'96 Conf. On Human Factors in Computing Systems, ACM Press, 1996, pp. 330-331.

[Rub88] Rubin, T., User Interface Design for Computer Systems, Halstead Press (Wiley), 1998.

[Tog01] Tognozzi, B., "First Principles," askTOG, 2001, www. asktog. com/basics/firstPrinciples. html (accessed August 6, 2007).

[Too02] Toor, M., "The Top Twenty Web Design Tips," 2002, www. graphic-design. com/Web/feature/tips. html (accessed August 6, 2007).

[Uni07] Unicode, Inc., The Unicode Home Page, 2007, www. unicode. org/ (accessed August 6, 2007).

[W3C03] World Wide Web Consortium, Web Content Accessibility Guidelines, 2003, www. w3. org/TR/2003/WD-WCAG20-20030624/ (accessed August 6, 2007).

第10章 信息设计

内容是几乎每个 WebApp 的核心。的确，对于大多数基于 Web 的系统和应用来说，“内容为王。”[⊖]有效的用户界面、直观的导航和丰富的功能都很重要，但是很多展示了这些特征的 WebApp 最终却失败了，因为这些 WebApp 缺乏有意义的内容或内容无法被定位。

第 9 章分析了用户界面设计。大多数 WebApp 的用户界面是到达内容的通路，同时内容（和操作它的功能）使得用户能够实现在沟通活动（第 4 章）中确定的目标。在本章，我们考虑如何提供用户界面下的内容——我们将关注信息设计（information design）。

当 WebE 团队进行信息设计的时候，实际上只有三个关键问题[⊜]应当考虑：

- **内容**。哪些内容是可用的？
- **组成**。你想为用户提供关于内容的哪些视图？
- **导航**。用户如何获得对那些视图的访问？

你可以在不同的抽象层次考虑这些信息设计问题。例如，你是正在考虑 WebApp 的高层信息架构还是每个页面上具体内容的低层详细设计？你正在设计主要信息种类之间的访问策略还是很小一组内容选项内的具体导航选项？

WebApp 特定的内容需求首先在沟通活动（第 4 章）期间进行考虑，然后在分析建模（第 7 章）期间进行精化和组织。在沟通活动早期引出需求会话时，Web 工程师和其他利益相关者确定了初始的内容对象（一个被命名的相关信息的集合）。引出需求同时会考虑用户访问内容的总体目标，并且会在他们的背景与内容的解释有关系时考虑他们的背景。最终，沟通活动产生了对用户场景或阐述内容用法的用例的开发。

在分析建模期间，WebE 团队的成员常常构建一个内容模型。内容模型会利用有助于对内容对象进行精化的 Web 信息交换图（第 7 章），产生 WebApp 的一组清晰的内容说明，但不一定是将在 WebApp 中安排这些信息的方式。

在本章，我们考虑这些内容如何组织、访问和管理，即 WebApp 的信息设计。我们从审查高层信息架构开始——它的角色是什么，它包含了什么以及它可以如何开发。然后我们将考虑详细的信息设计和对内容进行访问所需的特定导航结构。最后，我们将把高层信息架构设计和低层导航设计方法一起融入一个可以应用于 WebApp 项目的信息设计工作流中。

10.1 信息架构

信息设计的总体目标是：把通常以一组详细的内容对象表示的内容需求转化为 WebApp 具体的信息设计。信息设计的总体策略通常会结合自底向上方法和自顶向下两种方法。

自底向上的信息设计一般用于小型 WebApp，而且仅仅涉及实际 Web 页面的设计和构建，并渐进地把这些页面链接起来，这样 WebApp 的信息视图和结构就能够有机地出现。然而，对于大型的 WebApp 来说，自底向上方法产生的解决方案会阻挠用户定位信息的意图，表示信息时断章取义（当它被发现的时候），并且限制对变化的易适应性。

⊖ 参见 Bill Gates，“Content Is Where the Money Is on the Internet”（内容就是因特网上的财富所在地），1996，www. microsoft. com/billgates/columns/1996essay/essay960103. asp（2006 年 1 月 23 日）和 David Callan，“Content is King，”没有日期，www. akamarketing. com/content-is-king. html（2007 年 8 月 7 日）

⊜ 第四个问题（内容如何管理？）将在第 12 章考虑。

自顶向下的信息设计方法强调 WebApp 内主要内容种类的总体组织、相互关系和结构。这种高层设计方法针对 WebApp 应用的总体结构、组织信息的方式以及用户可能的访问信息的方法等方面。这是信息架构的范畴。

10.1.1 什么是信息架构

术语信息架构（IA）广泛应用于 Web 工程团体，然而，尽管每个人都似乎知道它的含义，但是每个人对它的定义的差异却相当大。为了消除这种一致性的缺失，很多组织（例如，IA 协会[㊀]）、网站（例如，IAWiKi www. iawiki. net/）、出版物（例如，《Boxes and Arrows》[㊁]）和会议（例如，每年的 IA 峰会[㊂]）都对信息架构做出了贡献。在开发大型项目的 WebE 团队内部，一般都会遇到信息架构师的角色。

信息架构可能就像表示 WebApp 基本导航结构的站点地图那么简单。IA 也可能是提供 WebApp 内的组织、管理和信息访问方法的综合概述的详细模型。一个表达了关注用户任务需要的 IA 的定义[㊃]是："一个信息空间的结构设计，有助于任务的完成和对内容的直观访问"［Ros02］。

IA 提供了一个框架，WebApp 信息的各个方面都是围绕这个框架来构建的。为了进一步说明，我们转移一下注意力，考虑一个来自住宅架构的例子。

考虑一所住宅的哪些建筑规划图会真正展示给你。这些规划图肯定会指明住宅主要的结构元素和构建这个结构所使用的材料。从主要的结构元素（房间布局、楼梯、墙壁、门、窗）的内部空间的位置，可以对结构方案的有效性进行评估。这些图还将展示这所住宅如何与其环境相适应，环境包括住宅的方向、与房产主要环境元素（例如，一个池塘或小溪、露出地表的岩石）之间的关系和住宅所处地面的倾斜度。最后，建筑规划图将显示住宅主要通路（例如，车道、人行道）的位置和其他围绕住宅所在房地产的事物。

换句话说，住宅的建筑规划图显示了其基本结构及其所处环境——那些定义了总体形状和住所可用性的事物。更加详细的架构信息（例如，水管和电路的布局）可以作为额外信息层包含进来。

因为建筑规划图显示了住宅与其周边环境的关系，所以有可能去评估设计是否适合周围环境。构建出的住宅的效用不仅仅取决于它的结构，也依赖于结构和环境的关系。

WebApp 的 IA 相当于（或者应当是）住宅的一组建筑规划图。它应当描述解决方案的基本信息"结构"，也应该决定这个结构在 WebApp 所在总体信息"风景"中的位置。

10.1.2 信息架构的元素有哪些

Rosenfeld 和 Morville［Ros02］在关于信息架构的有重大影响的一书中，以一个包含了表示背景、内容和用户三个相交圆的维恩（Venn）图形对 IA 进行了描述。他们在第 1 章中写道：

> 有些 Web 站点提供有助于我们发现答案和完成任务的逻辑结构，而另一些站点就缺乏易于理解的结构，阻碍我们在网站中导航。我们无法找到需要的产品；无法找到上周发现过的报告；在一个在线购物流程中感到迷惑。这些网站可能使我们想起那些失败的建筑：屋顶平台漏水的房子、没有橱柜空间的厨房、窗口无法打开的办公大楼和有着误导标志的像迷宫一样的飞机场。

㊀ iainstitute. org.

㊁ www. boxesandarrows. com/.

㊂ ASIS&T Information Architecture Summit 2007, www. iasummit. org/.

㊃ 在 IAWiki, www. iawiki. net/DefiningTheDamnThing（2007 年 8 月 8 日）上面可以看到对 IA 多个定义的很好的讨论。

然而，作为网站的设计者，我们不应当受限于建筑架构的隐喻……还将谈论信息生态学、知识经济、数字图书馆和虚拟社区。我们将从每个类似的事物中学到可以学到的知识，把过时的观念丢弃。

然后他们继续定义 IA：

信息架构（IA）

1. 信息系统内的组织、标记和导航策略的结合。
2. 一个信息空间的结构设计，有助于任务的完成和对内容的直观访问。
3. 对网站和内联网进行组织和分类的艺术与科学，用于帮助人们发现和管理信息。
4. 一种新兴的实践的学科和实践团体，致力于把设计和架构的原则运用到数字场景。

换句话说，IA 是一个创建在 WebApp 内使用的信息空间结构的高层设计。

10.1.3 好的信息架构有什么特征

Hardman、Bulterman 和 van Rossum［Har93］给出了一系列超媒体信息模型的理想需求，其中大多数同样可以很好地应用于 WebApp 的 IA。这些需求是：

- **与多种动态数据进行组合**。模型必须支持把不同的信息项归为一种表示并表达这些条目之间的限制的能力。

例如：SafeHomeAssured. com 可能包含使用各种传感器的多媒体表示。这些表示应该发生在一个新窗口中吗？在结束的时候应当自动重播吗？

- **高层表示的规格说明**。模型应该能够指明多个信息项之间的约束。例如，我们可能想指明某种图像种类（例如，艺术作品）在没有包含艺术家名字的情况下是绝不应该被表示的。

例如：在 SafeHomeAssured. com 中，个人信息（例如，账户信息或住宅安全细节）不论何时展示给用户，用户的姓名和登录状况都必须作为树立用户自信的策略的一部分进行显示。

- **时间关系**。有些信息项可能有基于时间的关系，这对其显示可能非常重要（例如，到一个事件信息的链接可能只有在事件被挂起时才是可用的）。

例如：对 SafeHomeAssured. com 而言，可能需要考虑对最近传感器监控事件的信息进行纳入和排除。应当显示给用户多少？在哪一点它们将不再被认为是相关的？它们如何成为架构的一部分？

- **链接的上下文和链接语义**。很多 WebApp，尤其是那些支持适应性和个性化的 WebApp 的一个重要特征是能够根据选择的链接来控制显示。这可以描述为如下的用例，“如果我沿着链接 A，那么结果将显示在一个新窗口中。如果我沿着链接 B，那么用结果取代现有信息。”

例如：当用户选择观看在 SafeHomeAssured. com 中的当前传感器布局时，建筑平面图应当显示在一个新窗口中。

除了这些需求，增加一些其他的需求也是合理的：

- **内容和信息分离**。内容是可用数据源的集合。信息是对 WebApp 用户有用的那些内容。一个好的 IA 应当区分它们。

例如：对 SafeHomeAssured. com 而言，产品目录包含不同定价结构的一组很丰富的数据，而这些信息的大部分可能从来都不会显示给大多数用户。对一个特定用户而言，显示的产品信息很可能是这些数据修改后的子集和用户的信息。

- **信息和应用分离**。一个 WebApp IA 应当区分一个用户可能发现的有意义信息和这些信息可能被安排和访问的结构方式。

例如：在 SafeHomeAssured.com 中，我们不能把产品的分类直接嵌入到产品信息中，而是把它作为一个单独的分类模型。

- **应用和表示分离**。如果我们把表示机制和应用分离，那么应用的可移植性和通用性（被用于其他应用或问题而变化极少的能力）就将大幅改进。

例如：在 SafeHomeAssured.com 中，产品的格式将不会被嵌入到产品目录中，而是表示为显示模板。

我们使用在这一节讨论的每一个需求来评估 IA 的质量和有用性。



信息架构

场景：沟通活动开始时，SafeHomeAssured.com 团队负责人的办公室

参与者：团队负责人和 SafeHomeAssured.com WebE 团队的两个成员

会话：

团队负责人：你们的初步沟通会议进行得怎么样？

团队成员 1：我们已经让利益相关者开发用例，进展顺利。

团队负责人：我们到目前知道些什么？

团队成员 2：一件事情是确定的——这个 WebApp 将有一个相当复杂的信息架构。

团队成员 1：你所谓的“信息架构”是什么意思？

团队成员 2：内容的结构、我们组织内容对象的方式、我们导航到内容的方式等。

团队负责人：我们不希望这个架构仅仅以某种特定的方式设计。它应当是我们真正应该设计的第一件事，而且我认为应该有个人充当团队这个角色。

团队成员 2：信息架构后。

团队负责人（点头表示同意）：我希望你能充当这个角色。你对这个职责满意吗？

团队成员 2：当然。我在上一项工作中做了一些 IA，阅读了一些关于 IA 的书籍。我想我能完成它。

团队成员 1：你将使用什么标准？我的意思是去做设计。

团队成员 2：嗯，首先我们必须要等到获得所有的用例，然后我们分析用例，得到将需要的内容对象、理解那些应用于对象的功能，并且理解哪一组用户需要哪些内容以及他们需要如何得到它——导航。

团队成员 1：这些是分析的所有部分，是吗？

[团队成员 2 点头。]

团队负责人：但是你并没有真正回答这个问题：怎样得到一个好的信息架构？

[团队成员 2 讨论在前面一节中介绍的特征。]

10.1.4 如何开发信息架构

正如我们在前面小节中提到的，IA 应当描述 WebApp 的基本信息“结构”以及用户可能如何与这个结构进行交互。这个结构通常都会围绕站点的一个宽广蓝图（blueprint），结合有助于形式化建模的分类学、本体[1]和受控词汇。用户与 WebApp 的交互方式常常是通过线框图[2]（wireframe）获取的，并且通过考虑其他访问机制（比如搜索工具）的角色来支持。

用于表示 WebApp IA 的复杂度有很大的差别。在某些情况下，WebE 团队将开发一个站点地

(1) 本体是一个正式的领域模型，表示领域内的对象，这些对象的属性和对象之间的关系，这个模型支持对领域的推理。

(2) 线框图展示了构成一个 WebApp 页面的组件的基本尺寸、位置和布局。

图，并认为站点地图表示 WebApp 的一个信息架构。即使这个站点地图在结构上非常丰富（就像图 10-1 所示的那样），它也仅仅是一个站点地图，常常无法提供足够的关于 WebApp 总体设计的细节，而且也无法针对用户目标对总体结构进行全面评估。

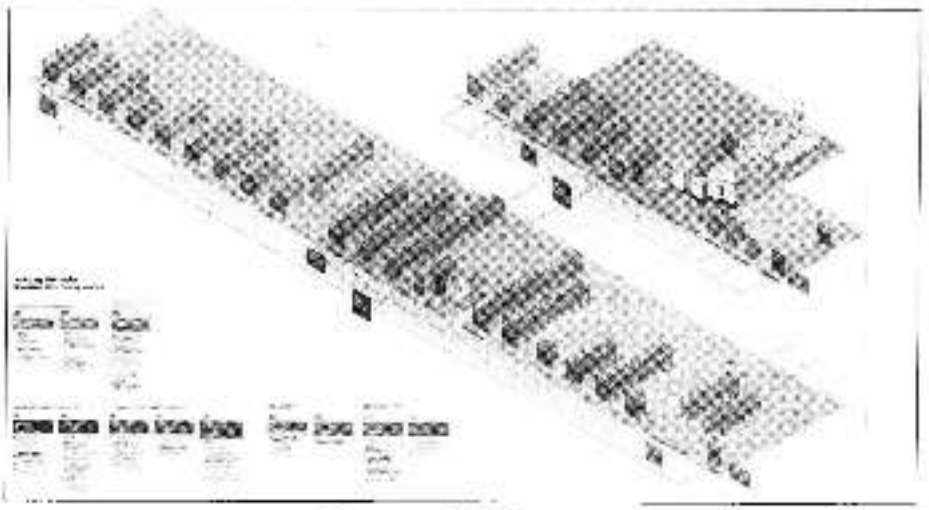


图 10-1 一个示例站点地图

来源：Copyright Dynamic Diagrams，www.dynamicdiagrams.com/case_studies/sec_web.html，转载得到了许可。

在有些情况下，一个站点地图可能对用户与 WebApp 交互的讨论和这些交互对信息组织的影响方式进行补充。虽然这种方法提供了进一步的理解，但是它没有明确地对实际发生的信息交换的性质和 WebApp 内信息域之间的内部相互关系进行建模。对用户交互的讨论也没有提供对系统所在信息环境的有效考虑。

Rosenfeld 和 Morville [Ros02] 讨论了开发 IA 的综合方法，涉及五个关键阶段：调查、计划、设计、实现和管理。前两个阶段对于成功地开发 IA 来说是最关键的。

第一个阶段，调查，关注于对 IA 上下文的理解——背景材料、目标、商业环境、用户和可能的内容。一个 WebApp 所有的这些方面都已经在前面的沟通活动（第 4 章）和分析建模（第 7 章）中进行了讨论。

第二个阶段，设计，是 IA 开发过程的核心。Rosenfeld 和 Morville [Ros02] 对它的讨论如下：

> 设计是把一个商务策略变成为一个信息架构，创建将被网页设计者、程序员、内容作者和构建团队使用的详细范围、线框图和元数据模式。在这个阶段，通常信息架构师要做文多数的工作，然而创新不能取代质量。如果设计的执行将会损害整体策略。对于信息架构师来说，主要的部分是责任，细节是魔鬼。

整句话说，IA 随着信息结构合适的设计模型的开发而渐进地出现。在下面几节中，我们将考虑 IA 开发过程中的一些问题，然后展示解决这些问题的模型。首先，我们将看信息组织，然后讨论描述如何访问信息的其他模型。

10.2 组织内容

设想回到 20 年前，还不能通过网络对大量信息仓库进行访问。如果要求你确定一个特定检察官关于某一项法律条款的具体看法，你将如何进行呢？你可能到当地的图书馆找到法律记录的副本，然后查阅关于那项法律的辩论。现在，假想一下如果你到达了图书馆，却发现有人已经

把所有的书籍都从书架上搬了下来，并且把它们扔进了地上的一个巨大的文件堆中，那么这件事会变得很困难。或者更糟——如果他们把每本书的页面取了出来，并混在一起。你几乎不可能查找！

一个 WebApp 内信息空间组织的目的和一个图书馆内书籍的组织以及一本书内页面的组织几乎一样——它使得用户可以找到需要的东西。但是信息组织还有其他目的。例如，一个电子商务应用可能把信息结构设计为确保用户能够看见那些鼓励他们购买的更多信息。考虑 Amazon. com 内专门的用户适应功能的角色。这些功能确保根据 Amazon 的每个顾客之前的导航和购买，展示给他们特定的图书。

在 SafeHomeAssured. com 中，我们可以对内容进行适应，这样当用户登录之后，将给他们提供一个“主页面”，其中包含了可以很好地与已经安装的传感器一起工作，或者是用来取代那些已经安装的传感器的更新、更安全的版本的新传感器的建议。

传统的图书馆使用那些已经演化了几百年的信息分类（结构）[⊖]。信息架构师可能设计一个特定 WebApp 独有的定制结构。这些结构应该是直观的，并且应该考虑用户最可能希望的信息组织方式。在某些情况下，这种组织是信息本身所固有的——考虑一个在线电话簿中信息的自然排序。在其他情况下，可以采用一个可以提供自然组织的隐喻。用户一旦了解了这个隐喻（例如，一个购物车），信息结构就很容易理解。Rosenfeld 和 Morville［Ros02］在下面的叙述中对这些结构的需求进行了很好的表达：“我们对信息的组织、标注和叙述的方式将会影响人们理解这些信息的方式。作为信息架构师，我们要为人们可以发现他们问题的正确答案而组织信息。”

不考虑 WebApp 的界面和美学的复杂性，底线几乎总是信息获取。如果内容的内部结构混乱，那么对 WebE 团队来说，要完成设计和实现为用户提供他们所需要信息的导航和搜索机制，即使有可能也非常困难。

10.3 设计信息空间

在我们看一个具体的信息模型之前，让我们先考虑一下有哪些可能的信息结构的一般类型以及这些结构将如何用于不同的应用。当在信息设计中构造信息时，你是在试图明确地定义用户在心理上对信息进行管理的方式——他们已经建立的、学到的或假设的信息结构。

例如，考虑一个包含了著名人物传记信息的一个仓库型 WebApp，这些人物包括比尔·克林顿、乔治·W. 布什、梅格·瑞恩和比利·鲍伯·松顿。这些信息应当如何归类呢？克林顿和布什（美国总统）应当和瑞恩和松顿（演员）分属不同类吗？或者克林顿和松顿（都出生在阿肯色州）建立一个组，而布什和瑞恩（都出生在康涅狄格州）建立另一个组？很明显，这将取决于用户可能具备的知识和他们使用 WebApp 的原因。

10.3.1 有哪些可能的信息结构

在进行信息设计时，创建的信息结构可以以很多种方式进行分类，如图 10-2 所示。值得注意的是，WebApp 可以使用不止一种信息结构。例如，一个有关教育的 WebApp 可能使用一个线性结构来对高层培训信息进行组织。用户在进入主题 2 之前不得不首先完成主题 1。但是在一个给定的主题内，信息可以被组织为一个层次化结构，允许用户以任何顺序学习不同的子主题。另外，你也可以在相同的基础信息之上覆盖不同的结构，这样就可以根据上下文进行不同的访问和使用。使用的结构常常能够反映你试图展示的信息的类型。

⊖ 最初的杜威（Deway）十进制分类法是 Melvil Dewey 在 1876 年开发的。

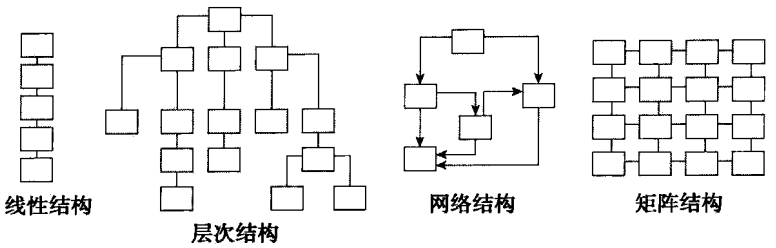


图 10-2 示例信息结构

来源：来自［Low99］。转载经过了许可。

线性结构。这种信息结构可以在一个 WebApp 内以多种方式使用。例如，当用户应该由一个规定的事件序列所要求的步骤顺序进行交互时，线性结构可以用于保存一个原始文档的顺序结构或用于控制访问。当交互的可预测顺序（有一些变化和转移）很常见时，常选择线性结构。帮助文档的展示可能是一个典型的例子，只有当必要的前题信息展示完毕之后，才会展示有着相关图片、短视频或音频的信息页面。内容展示的顺序是预先定义的，而且通常是线性的。另一个例子是一个产品订单录入工作流，其中信息必须以一种特定的顺序来说明。当内容和处理变得越来越复杂时，一个纯线性流（见图 10-2）就要被更复杂的分支结构（虽然仍然是直接顺着特定的路径）所取代，在分支结构中，可替换的内容可以被触发或者发生转换，来获取补充的内容。

矩阵（或网格）结构。当 WebApp 的内容按类别组织成两维（或更多维）时，可以采用这种层次化的结构。例如，考虑这样的情况：一个提供了汽车维修过程细节的 WebApp。选择一个矩阵结构。矩阵的水平方向分成不同问题（转向错误、点火问题等）上的信息，矩阵的垂直方向分成描述症状、原因、解决方案、工具和过程的信息。因此，当一个用户试图解决一个具体问题的时候，他们可以对网格进行水平导航，然后进行垂直导航来理解问题的性质。这种 WebApp 结构只有在内容高度有规则的情况下才会使用。

层次结构。层次结构毫无疑问是最常用的 WebApp 结构，而且用于反映自然的信息分类法。选择正确的分类方法是一个相当大的挑战，尤其是当信息的性质不一致的时候。在很多情况下，多个不同的层次可能都合适。前面我们给出了一个针对名人网站的例子。我们可以使用一层以用职业来定位（名人→职业→演艺人员→演员→电影→喜剧→亚当·桑德勒），而同一个人将在第二个层次中基于其出身进行安排（名人→出身→出生地→纽约→亚当·桑德勒）。

网络或图结构。这些架构元素是由把信息空间中常见的或相关的概念绑定在一起的关联的链接组成的。一个网络结构在 WebApp 内交叉连接相关方面非常有效，并且提供给用户相当灵活的导航。这些结构通常作为一个层次化结构之上的覆盖层来使用。如果使用得当，一个网络结构可以帮助用户进行有效的浏览。如果过度使用，用户将会被选择所迷惑或迷失他们在哪里以及哪些信息可用的感觉。网络结构的一个极好的例子是 www. wikipedia. com 。Wikipedia 的内容有一些分类，但是它的基础是内容之间的一组非常丰富的交叉引用，而不是一个自然的层次。

可以对前面段落中所讨论的设计结构进行组合，形成组合结构。WebApp 的总体结构可能是层次结构，但是部分结构可能会展示出线性特征，而结构的另一部分可能是网络结构。架构设计人员的目标是使 WebApp 的结构和将要展示的内容以及将要进行的处理相匹配。考虑下面的说明性例子：

- **培训材料**。线性结构可能更加合适，因为一个培训生将以顺序的方式对模块进行学习。

- 产品目录。考虑到有一个自然的产品分类，层次化的结构是最合适的。
- Wikis。在这种情况下，信息几乎没有在语义联系和交叉引用之外的固有组织进行相互链接，网络结构是最合适的。

SafeHomeAssured.com采用什么IA呢？在第9章，我们展示了一个界面设计审查的片段。下面的会话摘自那个片段：

> 团队负责人（为团队成员展示界面原型的一个脚本序列）：这就是显示在主页面上的主要功能菜单……问题是我们忽略了一些东西，而它们又不在一个抽象层次上。
>
> 团队成员2：它们都是主要功能，是吗？
>
> 团队负责人：是的，但是还有一些事情……你可以通过输入一个组件列表来购买一个系统……如果你不熟悉，就不需要真正对住宅进行描述。我建议主页面上采用一个不同的选项集合……这些主要选项的下面都有很多子功能。可以想象当选择了某个高层选项的时候，就会为我们弹出一个子菜单，子菜单的每个选项都将把用户带到一个单独的Web页面，这些页面有自己的布局、内容对象和功能。

这个讨论虽然反映了功能上的一个分割，但是它同样可以指导信息结构的后续讨论。考虑SafeHomeAssured.com团队引导的初始IA会议。



信息架构的初始会议

场景：SafeHomeAssured.com团队负责人的办公室

参与者：团队负责人、信息架构师和Web团队的一名成员

会话：

信息架构师：我听说在昨天的界面设计审查会议上，提出了主页面菜单的一个不同设计。对此我并不是那么高兴。

团队负责人：是的，我们确实建议了。但是我不明白你为什么不接受。在考虑到站点将提供的功能的情况下，提出的这个新设计是很合理的。

信息架构师（皱眉）：我对此表示怀疑。问题是我们确实应该了解用户想要访问的信息的上下文。注意，这个站点将会是所提供信息（比如产品细节）和功能（如建议一个安全布局）的混合。我只是认为我早应当参与进来。

团队负责人：好的——我接受。那么现在你对总体结构和在站点较低层次看到的外观方面有什么建议吗？它会产生与我们的总体界面设计的任何不同吗？

信息架构师：好的，如果我们回到前面的分析，我们确定了这个内容对象列表［拍出一张纸］。它很好地映射到了我们已经开发的结构。在最高层，我们有一个功能层，但是我们一旦下降一层，那么它就会变得更加复杂。可能会像这样……

［信息架构师在纸上画出了如图10-3所示的草图。］

信息架构师（继续）：除了住宅安全和监控服务上的两个部分外，其他大部分都很简单。对这两个来说，将会有一个信息层次，而且对相关的功能步骤会有一组工作流或过程。

团队负责人：对——那些看起来都很好。很显然，它应当被进一步充实为一个草图，然后进行审查。但是它基本上是很好的。

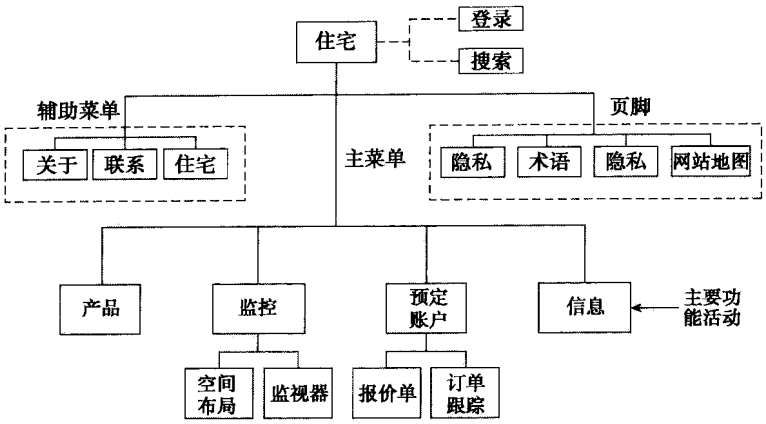
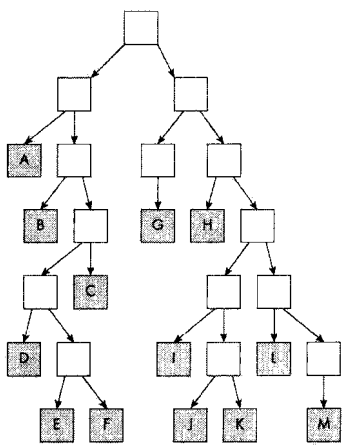


图 10-3 SafeHomeAssured. com 的初步信息结构草图示例

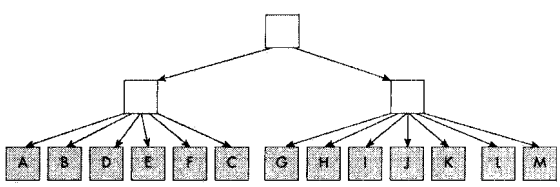
10.3.2 什么可以产生好的结构

我们将侧重于层次结构来回答这个问题。一般来说，一个好的结构满足用户的信息要求，并且易于导航。

对层次结构来说，信息结构的广度和深度将极大地影响用户导航到他们需要的信息所要付出的工作。图 10-4 展示了相同内容的两个差异很大的层次结构。一个窄而深的结构（图 10-4a）很可能使得导航选择更加简单（因为在任何给定页面，可以选择的链接更少一些），但是导航路径要长得多。另一个宽而浅的结构（图 10-4b）将产生更短的导航路径，但是在每个页面的选择就很可能复杂得多。



a）窄而深的层次结构



b）宽而浅的层次结构

图 10-4 相同内容的不同层次结构

作为一个通用指导准则，一些作者提议采用一个 7 ±2 规则[⊖]，并推荐结构内的一个设计应该只有 5 ~ 9 个主要链接（或链接组）。然而，这过于简单化了。我们建议层次结构的适当扇出[⊜]应该和 WebApp 的选项的复杂性和选择之间的相异程度相关。例如，一个在线字典提供一个初始的扇出为 26 个——每个字母一个。考虑到 WebApp 的性质，这个扇出是必要而合理的。

另一个经验法则就是避免创建那些超过 3 ~ 4 个层次深度的结构。用户导航得越深，他们就越可能变得不确定和沮丧（尤其是当他们在期望的地方发现没有信息时）。

层次的设计应当保证种类之间的区别尽可能明显。当用户不清楚在哪个种类内存在哪个特定信息块时，就会产生困惑。例如，考虑 SafeHomeAssured. com 初始的主页面结构：

关于公司
描述你的住宅
获得 SafeHome 组件的推荐
购买商品
获得技术支持

你期望在上述结构中的哪个部分找到一件特定产品的技术说明？它应当在“购买商品”（链接自产品描述）的下面，还是“技术支持”的下面？两个选项都有可能。产生这种困惑是因为正交的层次不清晰。即使有一种方案是从两个地方同时链接到信息（从技术和设计角度来看，这是毫无意义的），仍然会让用户疑惑到哪里去看，这浪费了时间，并且可能造成对 WebApp 负面的看法。

基于准确价值和清晰分类的层次一般不会有歧义。例子包括地理位置、时间、按次序或数字或字母排序的信息子集（如一本字典）。相反地，基于定义分类的层次常常是有歧义的。例如主题和观众（这常见于那些含有针对人或某个组织信息链接的结构，例如，“针对小的业务”或“针对企业”）。如果必须要使用有歧义的层次，那么就要特别注意确保导航选项将允许每类用户在犹豫最少的情况下找到他们需要的信息。在这些情况下，搜索功能同样变得更加重要。

10.4 蓝图：为结构添加细节

识别信息架构整体性质为你提供了一个起点。然而，如果 IA 要作为完整 WebApp 设计的基础，那么就必须提供更多的细节。

实现它的一种方法是使用蓝图。一幅蓝图展示各种内容对象如何映射到一个具体信息结构。因为一幅蓝图捕获信息结构的表面形状，所以它看起来常常像一个站点地图。然而，蓝图还包含了有助于理解设计的其他信息。例如，一幅蓝图可能还会讨论：

- 内容是动态的还是静态的。
- 内容是否为单个用户进行个性化（以何种方式）。
- 什么内容对象被映射到哪些 Web 页面。
- 什么导航路径处理给定的任务。

蓝图可以自顶向下地进行开发，从主页面开始，然后逐步增加附属页面。相反地，蓝图也可以自底向上进行开发，考虑内容对象，并审查它们如何被聚类（对层次而言）或排序（对一个线性结构而言）。最常见的情况是，蓝图的开发将是一个包含这两种方法的迭代过程。

蓝图使你可以想象 WebApp 如何组成一个整体以及由此产生的用户对它的响应。它是最早的设计模型之一（和界面设计一起），与用户和其他利益相关者交互，带来这些支持者的有价值的反馈。

⊖ 虽然有些争议，但是这个准则是和普通人能够立刻想起和管理的分离思想或概念的数目有关的。

⊜ 扇出是对单一结点下面导航结构的广度的一个度量。例如，图 10-4a 的扇出在 1 和 2 之间变动，而图 10-4b 第二层的扇出是 6 和 7。

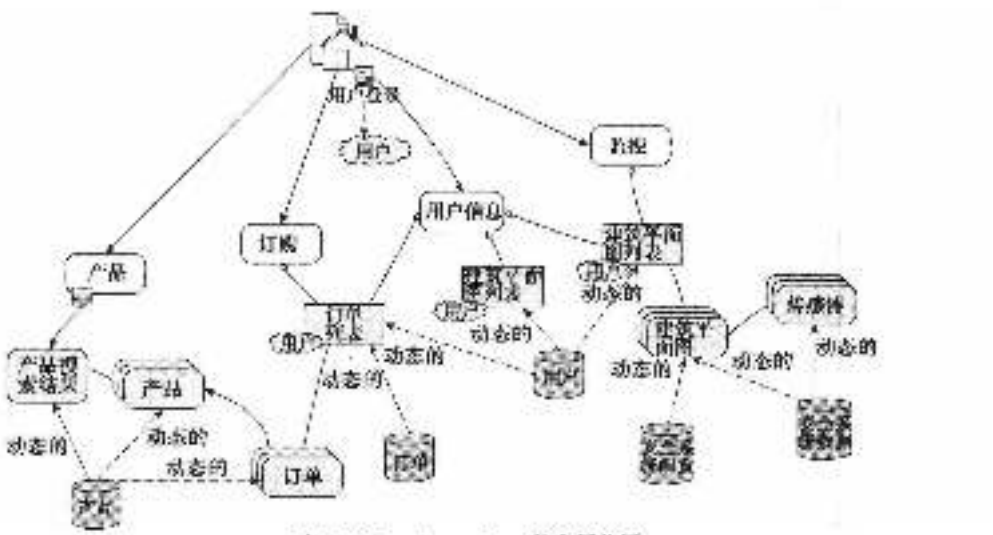
蓝图是什么形式

一幅蓝图的格式取决于你试图获取的IA特征。一个初始的高层蓝图可能关注于总体的信息结构；第二次迭代可能会详细说明结构信息，并提供对一个搜索界面的特质及其如何允许对内容访问的更深入的理解（WebApp的一些部分可以通过一个搜索直接访问，而其他部分则不能）；第三次迭代可能会强调内容的个性化。

虽然绘制蓝图没有普遍可接受的格式，但是在本章后面讨论的一些设计模型和记号可能会有用。为了阐明一种表示，图10-5a提供了一个可采用的示例符号，图10-5b展示了把符号应用到SafeHomeAssured.com初始高层蓝图的示例。图中，深色圆柱体表示在页面构建中使用的基础数据源。浅色符号表示动态构建的页面。这些动态页面（或包含在页面中的组件）可以基于一个内容源进行动态构建，或者可以基于某个其他的数据源（例如，在个性化中的用户信息）进行适应。注意，根据上下文的不同，信息组件（如订单列表）可以包含在不同的页面中。在所示的例子中，订单列表可以被包含在订购页面（可能还包含下新订单的信息）和用户信息页面（可能还包含基本的用户账号信息）之中。



a）蓝图中使用的示例符号



b）SafeHomeAssured.com的高层蓝图

图10-5 高层蓝图和示例符号

如果你想了解更多关于蓝图的信息，参考Christina Wodtke［Wod06］和IAWiki[1]已经开发的两个很好的补充材料的来源。

[1] 访问 http://iawiki.net/SiteMaps。

10.5 访问信息

除了 WebApp 实际的信息结构，还有很多其他因素会影响用户达到他们目标的能力。这些因素通常和导航机制以及特征有关：

- WebApp 机制——使用户在任何给定的时间知道哪些导航选项可用。
- 界面机制——为用户提供他们在哪里以及他们正在看什么的说明。
- 导航机制——允许用户在信息结构内穿行。

我们将在接下来的几个小节中考虑这些机制。

10.5.1 如何确保用户可以理解上下文并且不会迷路

你曾导航到一个复杂 WebApp 的内部，并感觉“迷失在超空间之中”吗？当这种情况发生时，你就失去了你在 WebApp 中（或外）位置的踪迹。这是一个很常见的问题，会让用户迷失方向，并无法获取需要的信息或者调用合适的功能；还会使已经获取的信息难以理解。这个问题的一个非常好的例子是一个学生在一堂 Web 设计课上告诉给 David Lowe 的：

> 几年前，我在为几个拜访我的朋友安排花式晚餐的时候，认为晚餐的一个理想的开始是一份海鲜浓汤。由于在书中无法找到一个好的食谱，我在 Web 上进行搜索。在看了几个选项之后，我选择了一个看起来不寻常的选项，并且做了汤。我的客人们对汤和晚餐都给予了高度赞扬，而且我们都过得很愉快。不幸的是，第二天出席晚餐的人都病得很厉害，包括我自己。
>
> 康复之后，我开始想——晚餐的哪个部分出问题了呢？我无法记起我放进浓汤里的所有配料，所以我就重新搜索了一下，发现了原来的那个网站。事实证明，我从搜索引擎上直接跳转到的页面仅仅对浓汤进行了描述——但是当我回溯到那个网站的主页面时（通过剪裁 URL），我发现它是一个包含了各种医疗问题的家庭治疗方法的网站，只不过把它们伪装成更加可口的食物。噢，天哪！我竟然给客人用了治疗严重便秘的家庭疗法。

虽然这是一个有些“严重的”的例子（很可能是杜撰的——虽然那个学生坚称那确实发生在他身上），但是它确实说明了确保信息总是展示在一个清晰上下文中的重要性。它还指明这个问题的一个显著原因是跳到 WebApp 的中间部分来作为搜索查询的结果。

下面的这些准则可以帮助用户理解他们在哪里并确定他们正在看的信息的上下文：

明确标记。WebE 团队应当开发一组局部标准，它可以生成一组可用于所有链接锚点清晰的标签。这些标签应当准确地描述链接的目的地，并且还确保用户在跟随一个链接时能够知道自己已经到过哪里的关键。实际上，确保标签不仅明确，并且在整个网站内一致也是非常有帮助的。很多信息架构师建立一个 WebApp 术语“字典”，用于 WebApp 设计的整个过程。

层级（Breadcrumb）。当你导航到一个信息架构深层的时候，能够知道你来自哪里总是一个好主意[⊖]。层级提供了包含在每个 Web 页面中的一个导航路径描述。这个路径描述了当前页面在信息结构内部位于哪里。作为例子，考虑 SafeHomeAssured. com 的用户的一个导航层级路径，这个用户已经访问了远程监控功能，并且正处在从一个住宅建筑平面图（监控视频将显示在那里）中选择一个摄像机的过程中。把它和图 10-5b 中的蓝图进行比较：

⊖ 对用户为了获取需要的信息所必须导航的深度进行限制是一个好主意。

监控（Monitoring）→建筑平面图（Floor Plan）（ID：34）→传感器（Sensor）（ID：653）

这个导航层次路径指明用户为了到达建筑平面图所做出（或可能做出——如果他通过搜索直接跳到某个位置）的导航选择。

标识。每个 Web 页面都应该明确表明网站的性质或显示的信息所属的子网站[⊖]。它同时应该（在可能的时候）提供对上下文和 WebApp 意图的提示，还应当提供到“首”页的链接。



创建一个数据字典

场景：SafeHomeAssured.com 的工作区

参与者：信息架构师和其他 WebE 团队成员

会话：

信息架构师：我发现在开发初始设计的各个方面时，并非每个人使用一致的术语。

团队成员 1：例如，我们当中一些人使用术语传感器来表示我们出售的所有设备；而其他人却只把它用于检测设备而不是其他设备。

团队成员 2：我们最终将会进行整理……

信息架构师：不够好。我敢保证如果在命名时不一致，那么我们将犯错误，或许其中有一些错误是很轻微的，但是它们将反过来制约我们。

团队成员 1：你有什么提议？

信息架构师：我的一部分工作是开发一个数据字典——这样我们就能够对内容对象和类似事物有一致的定义。这里，看看这个……它是一个开始。

团队审查了一个两页的文档，它在开头写道：

下面这些术语是用于 SafeHomeAssured.com WebApp 内公认的标准术语：

产品。一种可供购买和安装的住宅安全设备。

传感器。可以监控住宅安全的某些方面的产品。

联网的传感器。可以在配置之后在线监控的产品。

连接的传感器。已经安装和配置了的用于在线监控的产品。

建筑平面图。物理空间的安排细节，以及空间内所有传感器的位置和 ID。

账户。与一个给定的账户持有者相关的所有订单（过去的和待定的）细节，以及当前支付细节等。

订单。对服务和（或）产品的请求（临时的、确认的、完成的）。

团队成员提出修改和添加意见。

10.5.2 如何帮助用户在信息结构内移动

随着 WebApp 信息结构复杂性和大小的增加，用户很易于获取信息的能力会被削弱。为了避免这个问题，一个信息架构师应该调整导航以支持 IA 的具体特征，并且设计可以把用户带到所期望的信息处而把无关的内容过滤掉的搜索机制。目的是帮助有经验的用户更快地完成他们的导航目标[⊖]，并为经验不足的用户提供其他导航支持。

如果信息架构师做得够好，那么 WebApp 信息结构的最终设计将非常清晰，能使用户在没有任何怀疑和疑惑的情况下导航到期望的信息。但是随着用户变得更加有经验，设计者就可以考虑提供一种能使用户可以在不必穿越很长（同时乏味[⊖]）导航路径的情况下直接“跳到”期望的信息系统的机制。书签（bookmark）之类的浏览器技术是在用户端对此提供支持，而信息架构师可

⊖ 如果已经遵循了这个准则，那么就可以避免在本节的引言中提到的令人不快的情况。

⊖ 这类似于大部分软件应用中的各种快捷键——组合使用让有经验的用户能够更快地完成任务。

⊖ 至少从“有经验的”用户的角度来看。

以做一些其他的工作。一些简单的例子如：

全局链接。这些链接在每个 Web 页面上都有，链接到经常访问到的 WebApp 位置或功能。这些链接使用户可以跳到那些位置，而不需要返回到首页或遵循其他规定的导航路径。典型的例子包括：首页、帮助、联系、网站地图、索引、搜索、新闻、关于、注册和登录。

快捷方式。有很多方法可以绕过常规导航路线并跳过中间步骤，直接到达信息空间内的一个特定位置。最常见的例子是使用子菜单来使用户绕过中间页面。

层级和轨迹。我们已经提到了浏览层级对于帮助用户对自身进行定位是有用的。但是浏览层级（显示为激活的链接）还可以使用户快速返回到首页和他们当前位置之间的导航路径上的任何中间点。

图 10-3 所示的是 SafeHomeAssured. com 的草稿结构，给出了这些类型链接的几个例子。

从某种程度上来说，前面的每种机制都可以使用户“绕过”规定的信息访问方式。信息的受控访问的好处必须总是与易用性和访问能力进行权衡。

10.5.3 实现搜索机制的指导准则

大多数大 WebApp 都有搜索机制。事实上，搜索允许用户绕过强制的导航结构而直接跳到 WebApp 内的特定位置，结果，它常常是那些“现在”就需要信息的用户的首选。

在很多情况下，搜索引擎是在设计和实现完成之后才连接到 WebApp 的。虽然它可以为用户提供有用的支持，但是它也会引起一些问题。一个附加的搜索引擎并不会考虑应该为 WebApp 的哪些组件建立索引。这意味着可能给用户指向不合适的信息（例如，为一类用户设计的组件可能变得对另一类用户也可用），并且将以不受约束的方式在信息空间游走。它还意味着搜索功能并非总体应用的一个正确集成的组件。

图 10-6 阐明一种把一个搜索引擎受控地集成到一个应用中的方法。搜索引擎常常可通过限制其范围来使其使用更加有益。也就是说，搜索可以作为访问结构不好的或复杂的信息空间范围的机制。参照图 10-6，使用传统的导航机制访问右边组织良好的信息。这些页面处于搜索范围之外。然而，右边图中的页面（表示没有组织的信息）更难以限制在规定的导航，因此也就成为搜索功能的焦点。

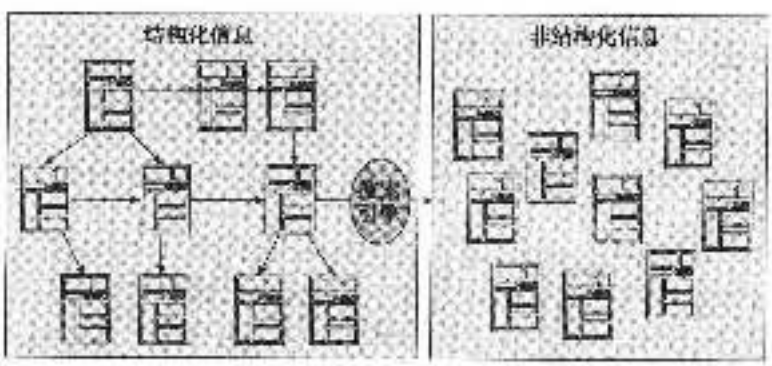


图 10-6 搜索引擎受控使用的一个例子

以 SafeHomeAssured. com 中对产品信息的访问为例。WebApp 提供的很多内容和功能都可以通过规定的导航路径最好地访问。然而，对于寻找特定安全或监控产品的用户来说，一个搜索功能可能有益。应该实现一个仅限于 SafeHomeAssured. com 这部分信息域的搜索功能。

当然，搜索引擎可以在整个 WebApp 内有效地使用。然而，问题是一种全局的方法常常会带来太多而非太少的搜索“提示”。用户淹没在他们所期望的信息可能的位置中，而且常常会因为导航到没有收获的位置而浪费时间。

对搜索引擎的合理使用可以通过考虑可能的使用模式的类型和将会存在的信息结构来理解。根据这两个因素，就能够确定搜索引擎的适宜性和操作范围。当底层信息有组织（例如，存储在数据库或 XML 文件中）和（或）包含可访问元数据时，搜索就会变得特别有用。在这种情况下，搜索可以被限制到数据的特定字段中。

在搜索引擎的设计中，还应当考虑很多其他因素。包括对基础索引进行更新的机制和频率，以及信息获取的过程和获取效率（特别是对于大的索引）。这些一般不会影响到搜索功能（除非有性能约束限制），并且常常可以归到 WebApp 开发的技术设计阶段。

10.5.4 搜索机制会产生问题吗

一个搜索引擎将绕过信息结构，确定一个给定信息空间范围内可能的目的地。这就引入了两个可能的问题。

第一个问题和强加在导航结构之上的用户要遵循的信息中的某个逻辑顺序的可能性有关。破坏这个顺序（通过使用搜索引擎跳出链接结构）可能会破坏内容和思想的逻辑流程，由此也就损害了 WebApp 的有效性。这个问题的严重程度将取决于应用的具体种类（例如，和参考应用相比，教育应用的问题就更加严重）。解决方案是为搜索结果谨慎生成索引和允许的目的地。在 SafeHomeAssured. com 中，很明显我们不希望在搜索结果中包含订单处理流程的中间页面。

当用户在搜索帮助下的不受限跳转中迷失了方向（“迷失在超空间中”）时，第二个问题出现了。这个问题一部分是内容间断性（瞬时穿越长的“概念性”距离）的结果，一部分是上下文突然变化的结果。同样，唯一合理的方案是在设计搜索引擎（只允许合理的跳转）和每个可能成为基于搜索的跳转（这样上下文就有助于用户的再定位）目的地的页面或结点的上下文时保持小心谨慎。确保信息空间中所有潜在的目的地的位置上都包含清楚的信息也很重要。层级是完成这一目标的有效机制。

10.6 线框模型

我们在前面把蓝图作为信息结构设计模型的一个元素的角色进行了讨论。线框图（Wireframe）描述一个 WebApp 内的一个单独页面（或多个页面）如何从概念上来看达到并行的目的。虽然一个线框图并不描述一个页面的图形设计，但它确实捕获那些应该显示在页面上的核心信息和导航元素以及这些元素的大概布置。本质上，一个线框图就是一个没有图形设计或实际实例内容的人工屏幕转储。

图 10-7 说明了 SafeHomeAssured. com 首页的很多设计选择中的一个的线框图。参照这个图，线框模型帮助对各种信息和导航元素设定优先级，并为最大化理解而定位这些元素。一般来说，线框图是为一个 WebApp 内的关键页面（那些可以作为设计范例或包含特别关键元素的页面）而创建的。

一个常见的问题是一个线框图是否是一种信息设计工具或一种交互设计工具。答案是，它可以是任何一种，但应该两种都是。我们可以开发一个强调从 WebApp 内的位置触发的关键交互的线框图。但是同样的线框图也可以用于探究内容的集成以及信息结构从 WebApp 内的一点来看像什么。

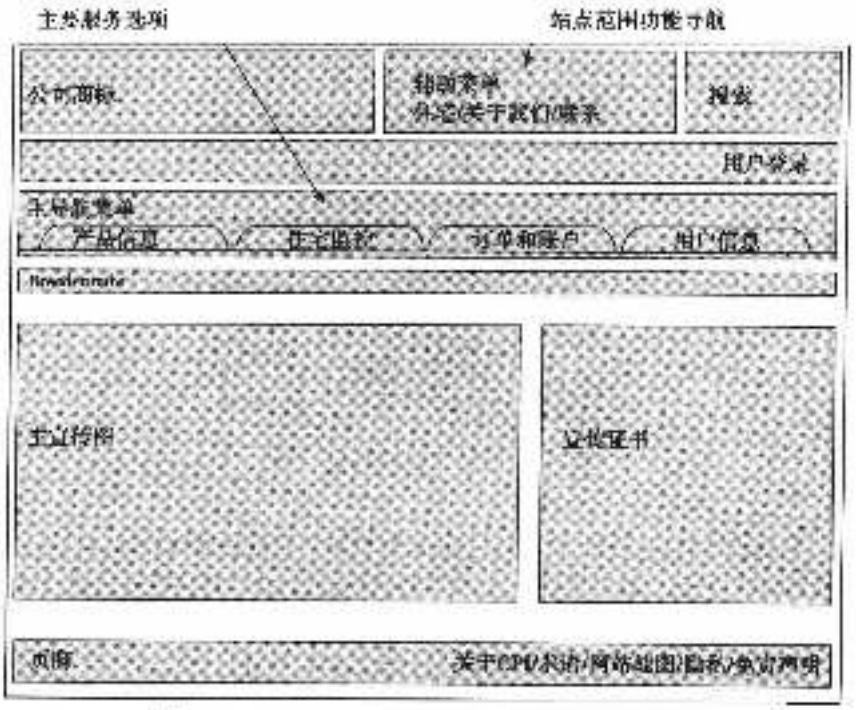


图 10-7 SafeHomeAssured.com 主页面的线框图示例

10.7 导航设计：创建详细的结构

信息架构描述 WebApp 的结构和组织，也隐含地描述查找和访问内容所需要的机制以及所有这些是如何组成一个整体的。然而，你仍然必须在局部层次创建具体的导航设计。在这一节，我们将研究一些完成导航设计的可能的方法。在第 10.8 节，我们将把所有的方面放在一起，推荐一个总体的信息设计过程。

10.7.1 信息设计和导航设计模型是如何演化的

信息空间导航的详细设计方法已经在过去的 20 年中进行了演化。在 20 世纪八九十年代，超文本的研究者们考虑了复杂信息空间中的导航结构问题。Web 出现后，他们就很自然地把工作进行了扩展并应用到 WebApp[①]。

Dexter 超文本参考模型（Dexter Hypertext Reference Model）［Mal94］（一种早于 Web 的模型）试图获取超媒体应用范围内的主要抽象。Dexter 模型把超媒体应用表示为三层。最底层称为组件内层（within-component layer），表示应用中组件内的内容和结构。中间层称为存储层（storage layer），是 Dexter 模型的核心，描述由结点（可以是原子的，也可以包含其他结点）和结点之间的链接组成的一个多媒体应用的基本结构。结点被固定在最底层的内容之上。最高层称为运行时层（run-time layer），捕获访问和操作数据结构的功能。这些都以示意图的形式显示在图 10-8 中。

① 这是不管早期超文本社区内认为 Web 是不值得做的超文本系统的担忧。实际上，Web 的发明者——Tim Bonners-Lee 给 1991 Hypertext Conference 的提议被拒绝了！



图 10-8 Dexter 超文本参考模型

来源：源于 [Hal94]。

Dexter模型本意不是作为一种设计工具，而是作为比较超媒体系统和模型的一个参考模型。然而，它确实强调建模基础内容和考虑这些内容为了访问而如何进行组织的重要性，还暗示了表示与结构是不同的问题。

这些概念已经被其他超媒体设计模型采用了，随后也被应用到了 Web。这些模型中很多是基于实体－关系建模或面向对象建模的[⊖]。

一般来说，这些技术或者是明确地出于对 Web 环境中的信息进行建模的目的而开发的，或者已经被适应到了这个领域。每种建模方法的具体结构、符号和重点各不相同。但是在每种方法中，设计都会涉及对基础信息的建模和理解、对用户将要使用信息的方法的建模，以及使后基础信息源之间的固有关系来设计合适的导航链接。

很多早期模型（特别是那些借鉴超媒体的概念和技术的模型）的一个关键的局限性是，它们的重点几乎全部集中在详细的信息结构上，忽略了更高层架构（就像我们在本章前面讨论的那些内容）和集成复杂功能。虽然说这一限制对于小规模的信息型 WebApp（第 1 章）来说不算是一个主要限制，但是对于那些含有丰富功能的 WebApp（例如，电子商务应用、在线银行、拍卖网站）来说却是一个问题。

Web 应用扩展（Web Application Extensions，WAE）[⊖] [Con99] 是一种试图避免这个问题的方法，它的做法是通过调整 UML 来适应 Web 的功能。另外，它试图把一个用户对系统的视图（像通过与交互相关的 Web 页面看到的）链接到支持这个交互的后台处理。在下面的小节中，我们将审查两个说明性的导航设计方法（RMM 和 WebML），同时看一看它们应当如何应用。

10.7.2 RMM 模型如何用于导航设计

关系管理方法论（Relationship Management Meebodology，RMM）[Isa95] 是一种早期的导航

⊖ 这些早期方法的例子包括关系管理方法论（Relationship Management Methodology，RMM）[Isa95]，面向对象超媒体设计模型（Object-Oriented Hypermedia Design Model，OOHDM）[Sch98]，RMDM [Lan94]，Lee [Lee97]，和 WSDM [Tro97] 等方法，以及 Web 建模语言（Web Modeling Language，WebML）[Ceri00]。其他方法则讲一个系统可以如何被使用 [Cue00]，及其将会如何影响信息域和由此产生的导航，以及使用 Z 等记号形式化方法来描述概念的、结构的和概念的模型 [Hals2，Gu99]。

⊖ WAE 将在第 10.7.4 节中进行讨论。

设计方法，包含如下一组活动：可行性分析、实体－关系（E-R）建模、切片设计、导航设计和构建。每个活动都由一系列步骤和一组工作产品（例如，一个“可行性文档”是在可行性分析结束时产生的产品）组成。RMM 有三个活动和我们对信息设计的讨论有关：ER 建模、切片设计和导航设计。

E-R 建模[⊖]活动通过识别内容（数据）对象、属性、关系和组成 WebApp 信息空间的各种类型指标来定义应用的信息域。内容对象用一个有标号的矩形表示。关系表示为一条连接对象的有标号的线。数据对象和关系之间连接性质的建立使用了各种特定的符号——虽然有很多不同的 ER 记号在普遍使用。在 SafeHomeAssured. com 中，我们将对用户、账户、产品、传感器和其他对象进行建模。图 10-9a 给出 SafeHomeAssured. com 的部分 ER 模型。值得注意的是，即使在现阶段，建模也可能会导致识别重要的不明确问题。在创建 SafeHomeAssured. com 的 ER 模型时，要考虑传感器和建筑平面图之间的关系的基数——它产生的问题是一个传感器能否存在于建筑平面图上。这个决议（在和客户讨论了之后）是建筑平面图不需要是正交的（例如，它们可以重叠，因此两个建筑平面图可以包含同一个传感器）。这是因为用户可能希望创建不同的但是重叠的建筑平面图，用来支持在不同时间对一个住宅的不同部分进行监控。

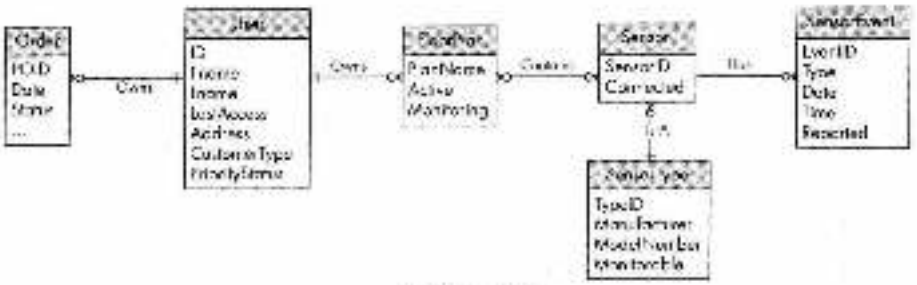
“切片设计”活动在 ER 建模之后，确定详细的信息结构和访问机制。本质上，这一步是把这个域的内容（在 ER 模型中获取的）分成不同的集合，这些集合为使自身有用和有意义而能够或应该显示在一起。信息的一个切片可能是一个实体的一小部分（例如，一个用户实体的名字或一个产品实体的照片）或者被分在一起的几个实体的信息片（例如，包含传感器的名字和类型的建筑平面图）。切片表示组织在一起时对用户有用的信息集。它们常常是分层进行构建，逐渐把小的信息项和其他信息集合融入渐渐丰富的集合之中。

图 10-9b 中展示了 SafeHomeAssured. com 的多切片的一个不错的例子。图中，圆角矩形表示基础信息实体，椭圆是这些实体的属性，扇形区则是用于联合这些实体属性的切片。图中显示获取了显示给登录用户的信息（作为一个单一页面）的一个切片。这个切片含有用户的详细信息（例如，名字和最后一次访问——包含在一个事先定义的切片内）、所有近期购买的订单和这些订单的当前状态（再次使用另一个切片）、与此用户相关的任何建筑平面图的名字以及和每个建筑平面图相关的最后五个监控事件。注意，这个切片是由其他最终从基础域模型中得出的信息切片组成。实际上，这些域实体很可能实现为一个数据库中的记录（或者信息存储的其他形式，如 XML 页面），而且切片会是对这个内容的一个合适查询。

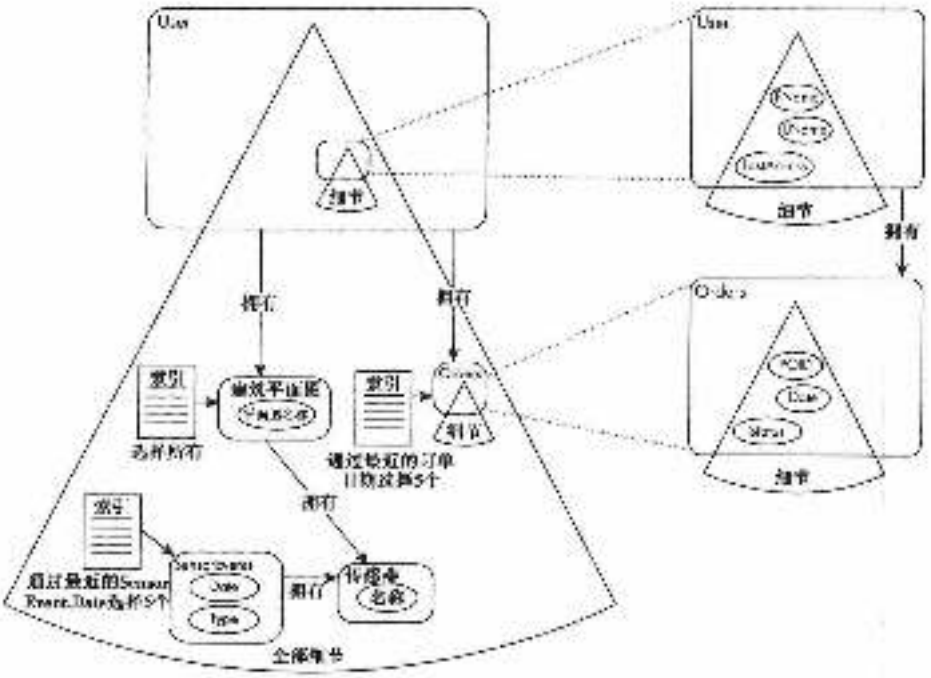
即使没有理解这个切片设计的具体记号，也应该清楚这种方法允许设计者把部分信息收集为有意义的单元。切片设计准则强调，每个切片表示对用户来说是一个完整的信息单元，而且前提是域分析。

RMM 的第三个设计步骤是导航设计——确定了各种切片之间的链接。切片设计创建了对各类用户有意义的信息单元。最终，这些信息单元将聚集并转化为 Web 页面。导航设计通过对来自 ER 图的链接标签的所有切片进行选择来对这些页面进行连接。图 10-9c 显示了 SafeHomeAssured. com 的一个导航设计片段。

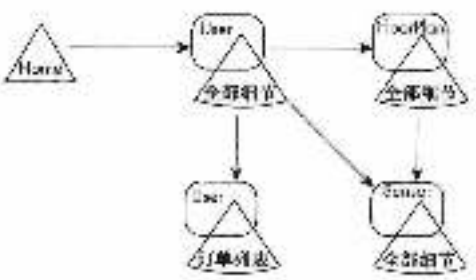
⊖ 对 E-R 建模的全面讨论超出了本书的范围。要获得到更多的信息，请参看 SmartDraw. com 上的教程“How to Draw Entity Relationship Diagrams”、http：//www. smartdraw. com/tutorials/software-erd. htm（2007 年 8 月 8 日）或者 Webopedia，http：//itmanagement. webopedia. com/TERM/E/entity_ relationship_ diagram. htm（2007 年 8 月 8 日）上面关于实体－关系图的讨论。



a）部分ER模型



b）主要用户信息的m-切片视



c）部分导航设计

图 10-9 SafeHorseInsured. com 的 HMM 建模

HMM 设计活动可以自顶向下，也可以自底向上，或者两者结合。例如，在自底向上方法中，设计者先关注每个信息实体，然后关注更一般的访问机制。而在自顶向下方法中，设计者首先关注切片设计的一般结构，然后把结构转化为更低层的表示单元。实际上，这个过程可能是一种

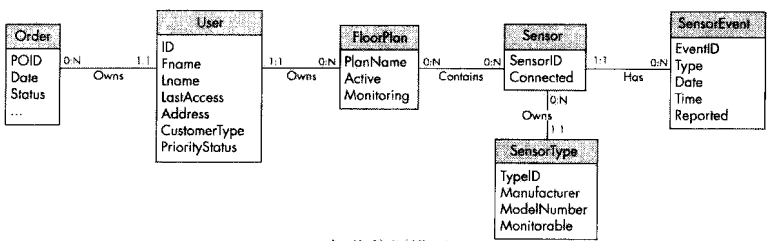
结合了自顶向下和自底向上的迭代方法。还值得注意的是，在这一步中，设计者必须要识别组件及其可能的访问方式。这就意味着对用户和他们背景的理解相当重要。

RMM 的重点在于建模基本内容、这个内容的用户视图和对内容进行链接的导航结构。RMM 设计过程是有些主观，而且基本没有提供指导准则来确保切片（以及由此产生的导航结构）的分组是合适的。它假定设计者知道（直观地）哪些切片最合适并且对于用户来是有用的。

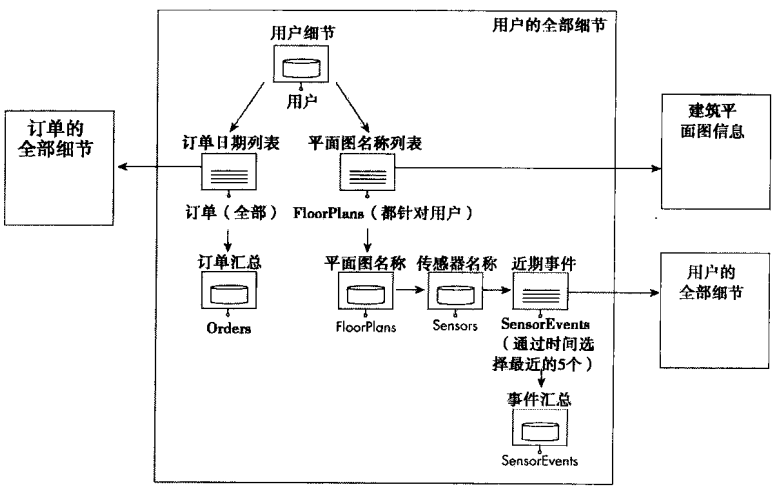
10.7.3 如何使用 WebML 创建导航设计

Web 建模语言（Web Modeling Language，WebML）［Cer00］是比 RMM 出现更晚、更加丰富的表示法。这种建模语言对工作流建模、表示和内容的适应、个性化和设计模式具有强有力的支持。由一个 CASE（计算机辅助软件工程）工具（虽然是相当粗糙的一个）、建模模板和丰富的技术资源来支持[⊖]。

图 10-10 展示了 WebML 模型，和图 10-9 中展示的 SafeHomeAssured. com 的 RMM 例子相同。如图所示，WebML 中的基础模型和 RMM 以及其他设计模型中的集成模型很相似。WebML 从一个数据模型（和 RMM 的 ER 模型大体相当）开始，然后应用于构建描述内容组成和网站导航的超文本模型。组成（composition）详细说明哪些页面组成 WebApp 以及哪些内容单元组成一个页面（类似于 RMM 中的切片图）。导航模型表示页面和内容单元如何链接而形成 WebApp。WebML 还包括一个表达页面布局和图形外观（与输出设备和表达语言无关）的表示模型，以及一个表示其他模型如何为不同类用户进行变化的个性化模型。



a) 部分数据模型



b) 部分超文本模型

图 10-10 SafeHomeAssured. com 的 WebML 建模

⊖ 可以在 WebML. org，www. webml. org（2007 年 8 月 8 日）上找到关于 WebML 详细信息。

10.7.4 可能创建链接了内容和功能的模型吗

RMM 和 WebML 的重点几乎全在信息结构上，而且很大程度上忽略了复杂功能与这些结构的集成。WebML 确实包含一些可以用在导航模型中操作单元，但是这些操作单元最适合于展示如何把功能编织于用户交互和内容表示中，而不是详细的功能设计之中。虽然这不是信息型 WebApp 的一个主要约束，但是对于包含复杂功能的 WebApp 来说，这就是问题[⊖]。

UML 的 Web 应用扩展（Web Application Extension，WAE）［Con99］是一种把信息视图和功能型 WebApp 组件进行链接的设计方法。它阐明功能组件如何生成和提供信息以及信息（通过诸如链接激活或表单提交等方面）如何触发功能组件。特别地，WAE 对客户端功能与行为和服务器端功能之间的连接进行建模。

图 10-11 提供了 SafeHomeAssured. com 的一个 WAE 模型的一小部分。在这个例子中，既包含信息型元素，又包含功能型元素。功能型元素将在第 11 章中进行详细讨论。信息型元素包括边界对象（例如，和用户进行交互的 Web 页面或其他组件——如报警框）和实体对象（例如，为页面上包含的项目提供内容的数据源）。其他组件提供实体对象如何连接的细节。

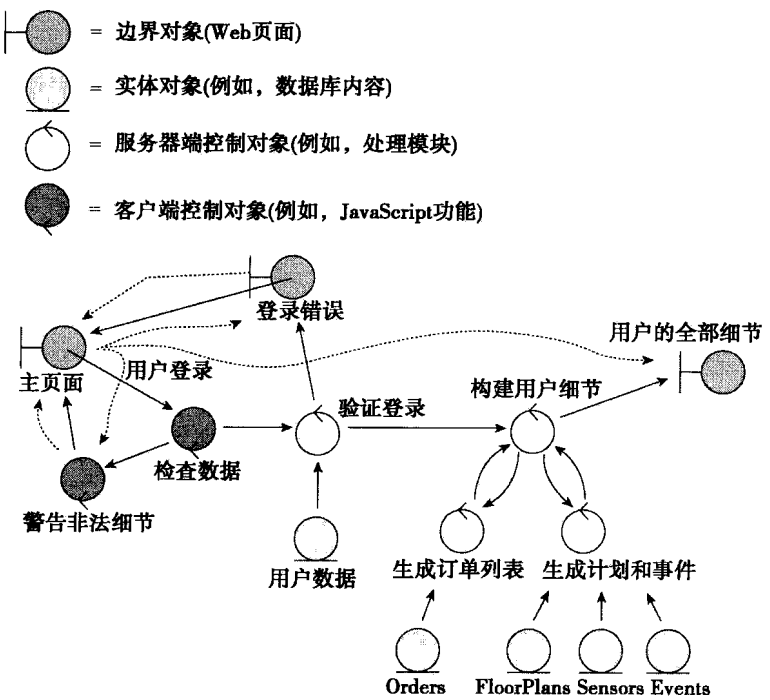


图 10-11 SafeHomeAssured. com 的 WAE 模型示例片段

10.7.5 Web 自身的结构有影响吗

万维网最初的概念是为不同应用提供公共界面的一组协议，并不是作为信息应用本身。分布式客户端－服务器模型继续用于访问信息和其他资源，但是 Web 并没有提供一个用于信息管理和访问的形式化模型。

信息设计可以产生相当复杂的信息结构，但是 Web 自身只使用信息结构的一个简单的“结

⊖ 如今，几乎所有主流的 WebApp（例如，电子商务应用、在线银行、拍卖网站）都含有许多复杂的功能。

点-链接”模型。信息显示在结点，而结点之间通过简单的点对点、单向的、上下文无关的、无类型的链接来互相连接。结点保持基本信息单元，而链接则提供丰富的导航功能。

随着 Web 的演化，为了避免超媒体的一个简单模型引起的约束，产生了大量复杂的修改和加载项。WebApp 构建阶段的复杂性很大程度上发生于 WebE 团队试图把信息设计映射到 Web 提供的功能支持的时候。例如，用于表示信息的机制从最初的单一文本结点演化为各种浏览器都支持的一组非常复杂的媒体要素。早期的例子包括使用帧来改进信息的语境，以及使用 Java Applet 来改进媒体的集成。更近期的例子包括使用 AJAX 这样的技术。目前，多种修改扩展了 Web 支持的受限超媒体功能。例如，提供多个目标链接的插件和提供了一个网站内信息之间相互关系图的应用。然而，这些修改的大多数都致力于诸如界面改进和更好的媒体处理这样的非超媒体方面。

10.8 总结设计过程

在本章中，我们已经讨论了信息架构、蓝图、线框和信息导航模型。但是我们如何把所有这些应用于一个实际的设计过程呢?

关键的设计任务（域建模、信息架构设计和详细设计）以及由此产生的工作产品都展示在图 10-12 中。

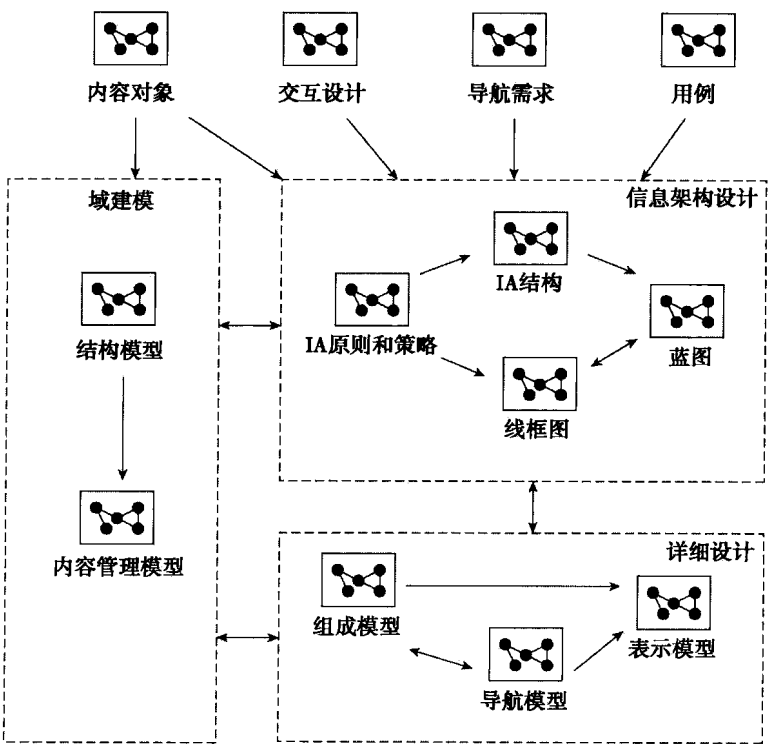


图 10-12 信息设计过程

域建模的重点是把内容对象转化为一个清晰的、可以作为信息设计的基础的模型。域模型包括一个诸如在 WebML 或 WAE 中使用的结构模型和一个把内容映射到可管理形式的管理模型（第 16 章）。

在本章前面所讨论的原则、特征和技术的指导下，设计人员创建一个信息架构模型。信息架构师使用作为交互设计的一部分而生成的信息和内容对象之间的关系所驱动的总体导航需求。

然后指定了信息空间的结构，同时定义了组织和管理内容的策略。这个策略确定：1）管理内容将使用的方法；2）元数据是否是一个问题；3）如何维护这个架构；4）将考虑使用哪些工具。接着开发了网站的蓝图和线框图，并且迭代地进行精化。

然后把信息架构设计用作详细设计的基础（必要时）。信息设计包含一个组成（或切片）模型，这个模型把来自结构模型的内容划分为具体的用户层组件，将使用详细定义了链接结构的低层导航模型来联合开发。最终，表示模型将把具体的信息组件映射到一个线框模型内展示的界面。

要注意设计过程的正式程度应该适合所构建 WebApp 的特征。在你开始信息设计并创建一组综合的复杂模型之前，应该认识到，进行信息设计的正式程度取决于如下多个方面：

- **应用规模**。随着大小和复杂性（WebApp 的、内容对象及其之间的关系）的增加，以一种可以让 WebE 团队在构建开始之前对设计的质量进行评估的方式来表示事物就变得越来越重要。
- **信息易变性**。如果内容是高度动态的（例如，一个在线拍卖网站的内容），那么建立一组清晰的结构模型和一组健全的 IA 原则与策略就变得愈加重要。相反，详细设计模型的有用性就会降低，因为它们不得不进行频繁的更新来变得有用，而且详细的模型可能会不适宜地限制 WebApp 的演化。另外，在蓝图中捕获的访问机制必须考虑内容的动态本质。蓝图和访问机制成为信息设计过程的中心。
- **应用易变性**。如果 WebApp 的总体需求很可能频繁地发生变化（例如，商务环境是变化的和不可预测的），那么 WebApp 设计者就应该把重点放在 WebApp 的这些方面和那些已知稳定的信息空间[⊖]。详细设计的有用性不高，因为很难维护它们的有效性。
- **用户多样性**。只有用户上下文保持相对一致，支持单类用户的 WebApp 的设计将简单得多。在这种情况下，IA 结构和蓝图的重要性将有所降低，而线框图将变得更重要。相反，随着最终用户多样性的增加（由此，用户目标和任务就变得更加多样），确保信息结构和信息访问路径总体的一致性就变得更加困难，因此，蓝图的重要性就会增加。
- **应用关键性**。虽然 WebApp 的质量总是很重要，但是当一个 WebApp 是关键任务的时候，它就成为 WebE 团队的中心焦点。为了确保质量，团队应当针对设计的工作产品进行一系列的审查。评估作为信息设计的结果所创建的模型，以确保在创建 WebApp 的时候将会构建出的质量。

一个具体 WebApp 项目应该在设计过程的早期决定合适的建模深度，不要等到由时间压力驱动才特别决定。

10.9 本章小结

对几乎所有的 WebApp 来说，内容是关键。事实上，它处于应用的核心地位，可以看作是构建 WebApp 时所基于的框架。即使一个 WebApp 有设计良好的界面和有效的功能，它也会难以使用，也将为误解敞开大门（如果信息设计拙劣，回想本章前面讨论的海鲜浓汤的例子）。

WebApp 信息设计的核心是信息架构（IA）。IA 描述信息如何进行组织、访问和管理。实际上，IA 常常是仅有的需要进行明确文档化的设计部分——考虑到详细信息设计的流动性要大得多以及常常会持续地演化。事实上，这种处于低层的流动性使得一个稳定的、谨慎构建的 IA 更为重要。如果 IA 有缺陷，那么不可避免的低层演化将会破坏这个应用。

⊖ 如果没有任何稳定的东西，那么就要问一问项目为什么已经被启动了。

虽然有很多可用的模型，但是蓝图和线框图（或它们的变种）通常是需要构建的最重要的模型。蓝图为信息访问和管理建立了一个稳定的基线。线框图搭建了到交互设计的桥梁。其他模型对它们进行补充，并且阐明具体的细节或方法。

如果信息设计形成了一个 WebApp 的“骨架”，那么 WebApp 的功能组件就是使得骨架可以移动并完成某些事情的肌肉、肌腱和关节。在第 11 章中，我们将考虑如何设计功能元素，然后研究功能元素如何绑定到信息设计来创建一个有效整体。

参考文献

[Cer00] Ceri, S., P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): A Modeling Language for Designing Web Sites," in *Proc. WWW9 Conference*, Amsterdam, 2000, pp. 137-157.

[Che96] Chen, M. S., J. S. Park, and P. S. Yu, "Data Mining for Path Traversal Patterns in a Web Environment," in *Proc. International Conference on Distributed Computing Systems*, Chicago, October 1-4, 1996, pp. 385-393.

[Con99] Conallen, J., *Building Web Applications with UML*, Object Technology Series, Addison-Wesley, 1999, p. 336.

[Coo99] Cooley, R., B. Mobasher, and J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Pattern," *Knowledge and Information Systems* vol. 1, no. 1, 1999, pp. 5-32.

[Det97] De Troyer, O., and C. Leune, "WSDM: A User-Centered Design Method for Web Sites," In *Proc. 7th International World Wide Web Conference*, Elsevier, Brisbane, Australia, 1997, pp. 85-94.

[Ger99] German, D. M., and D. D. Cowan, "Formalizing the Specification of Web Applications," *Lecture Notes in Computer Science*, Issue 1727, Springer Verlag, 1999, pp. 281-292.

[Gue00] Guell, N., D. Schwabe, and P. Vilain, "Modeling Interactions and Navigation in Web Applications," In *Proc. World Wide Web and Conceptual Modeling'00 Workshop—ER'00 Conference*, Salt Lake City, 2000, pp. 115-127.

[Hal94] Halasz, F., and M. Schwartz, "The Dexter Hypertext Reference Model," *Communications of the ACM* vol. 37, no. 2, 1994, pp. 30-39.

[Har93] Hardman, L., D. Bulterman, and G. van Rossum, "The Amsterdam Hypermedia Model: Extending Hypertext to Support Real Multimedia," *Hypermedia Journal* vol. 5, no. 1, 1993, pp. 47-69.

[Isa95] Isakowitz, T., E. Stohr, and P. Balasubramanian, "RMM: A Methodology for Structured Hypermedia Design," *Communications of the ACM* vol. 38, no. 8, 1995, pp. 34-44.

[Lan94] Lange, D., "An Object-Oriented Design Method for Hypermedia Information Systems," *HICSS-27: Proc 27 th Hawaii International Conference on System Sciences*, Maui, Hawaii, 1994, pp. 366-375.

[Lee97] Lee, S. C., "A Structured Navigation Design Method for Intranets," In *Proc. Third Americas Conference on Information Systems*, Association for Information Systems (AIS), Indianapolis, 1997.

[Low99] Lowe, D., and W. Hall, *Hypermedia and the Web: An Engineering Approach*, Wiley, 1999.

[Pow00] Powell, T., *Web Design*, McGraw-Hill/Osborne, 2000.

[Ros02] Rosenfeld, L., and P. Morville, *Information Architecture for the World Wide Web*, 2nd ed., O'Reilly, 2002.

[Sch98] Schwabe, D., and G. Rossi, "Developing Hypermedia Applications Using OOHDM,"

presented at *Workshop on Hypermedia Development Processes, Methods and Models* (*Hypertext'98*), Pittsburgh, 1998.

[Spi98] Spiliopoulou, M., and L. C. Faulstich, "WUM: A Tool for Web Utilization Analysis," *EDBT, Workshop WebDB'98*, *LNCS1590*, Springer Verlag, March 1998, pp. 184-203.

[Tak97] Takahashi, K., and E. Liang, "Analysis and Design of Web-based Information Systems," presented at *7th International World Wide Web Conference*, Brisbane, Australia, 1997.

[Ton03] Tongrungrojana, R., and D. Lowe, "WebML +: Connecting Business Models to Information Designs," *SEKE: 15th International Conference on Software Engineering and Knowledge Engineering*, eds., K. Zhang and J. Debenham, Knowledge Systems Institute, 2003, pp. 17-24.

[Wod06] Christina Wodtke, "Information Architecture: Blueprints for the Web," 2006, www.eleganthack.com/blueprint/author.php (accessed August 8, 2007).

第11章 功能设计

内容或许是 WebApp 的核心，但是功能却可以使内容变得丰富多彩。现代 WebApp 的用户希望充实的内容可以与复杂的功能结合起来，希望这些功能能够增进他们对内容的理解，能够以不同方式描绘内容，能够使交互个性化，并为他们访问站点提供一些附加价值。为此，WebApp 设计者把越来越多的工作用在功能设计上，这些功能将被纳入每一个 WebApp 增量。

11.1 WebApp 功能

现代 WebApp 已经和早期 Web 有了根本的不同，最大的改变可能是越来越多的高级而复杂的功能被包含进来。在第 1 章提到过，最早的基于 Web 的应用主要是为了提供对内容的访问，这样的 WebApp 可以使用核心的 Web 架构、最早的 HTML 表单或者超文本传输协议（HTTP）来实现。随着时间的流逝，HTML 已经被扩展为允许更多高级的格式并引入样式表。

通过使用服务器端脚本[⊖]和客户端插件，简单功能开始出现。更高级的和完美的用户交互的强烈要求促使了对客户端脚本的引入。在 Web 的第一个十年结束时，WebApp 已经发生了很大的变化。技术基础的提高使得 WebApp 在功能上非常丰富。近几年来，我们看到一些概念和技术使得这个趋势得以延续——例如，Web 2.0、AJAX 和 Web 服务。这些技术都强调对丰富的功能和高度交互的应用的支持（典型的 WebApp 有博客、百科、RSS 驱动应用）。

因为早期 WebApp 主要关注信息的管理和访问，所以设计过程（当其被应用时）、实践、模型的重点自然落在了信息设计上。随着 WebApp 的重心转为越来越复杂的功能，设计方法就落后了。WebE 团队的挑战是以一种可以弥合其他设计活动的无缝的方式（例如，交互设计和信息设计）来进行功能设计。

在某些方面，WebApp 的功能设计比传统应用的功能设计更简单，WebApp 的功能设计几乎总是基于组件的和划分成模块的。这使得许多功能能够被设计成简单的 WebApp 增量，而不用太关注组件之间的耦合[⊜]。功能设计权衡那些能很好地被集成到 Web 环境中的复杂语言（例如，JavaScript 和 HTML 之间有很紧密的联系），同时可以应用高级的架构和环境（第 14 章）。如果功能设计做得好，那么无论是采用一种特殊架构还是协调那些为了构造 WebApp 而内嵌的功能都将非常简单。例如，一个好的内容管理系统（第 16 章）可能提供所需要的大部分功能，从而实现作为 WebApp 的一部分的一个丰富的产品目录。

然而，在其他方面，WebApp 的功能设计会比传统应用的功能设计复杂得多。设计者必须考虑由 Web 基础架构所赋予的一些必不可少的约束——如分布式模型（使得信息处理和用户响应变得复杂）、安全问题、从 Web 浏览器继承来的受限的接口模型。同时，如果相关信息架构比较复杂，那么想要做到有效的集成就会更困难。

在这一章，我们将尝试提供一种方法来研究 WebApp 的功能设计。我们以考虑 WebApp 的功能特性和 WebApp 中需要处理多少种不同类型的功能作为开始。从而引入对设计 WebApp 的功能组件的过程的描述。接下来，我们考虑 WebApp 的整体功能架构及其开发。最后，我们将考虑详细功能设计的各个方面。

⊖ 服务器端功能的最早的形式之一，即通用关口接口（CGI），出现于 1993 年（Web 发展的初期）。

⊜ 然而，重要的是要注意耦合（组件之间相互连接程度的定性度量）也许在功能复杂的 WebApp 中仍然是关注的焦点。

11.2 WebApp 功能的性质

各个 WebApp 的功能的规模、复杂性和类型都有很大的区别。因此，WebE 团队的功能设计方法必须和利益相关者所需的功能以及在沟通活动（第 4 章）中识别的功能类型相一致。在我们讨论如何完成功能设计之前，有必要考虑可能遇到的 WebApp 不同类型的功能。我们以一些例子来说明。

11.2.1 功能示例

一个 WebApp 的功能可能像实现一个下拉菜单那样简单，或者像完成支持一个业务工作流所需的复杂计算那样复杂。为了说明不同的功能复杂性，考虑如下一些例子：

例 1：客户端交互支持。Web 应用常常包含一些提供用户导航和信息访问的客户端功能（参见图 11-1a），最常见的例子有下拉菜单、滚动条以及图片加载。

例 2：客户端信息管理。WebApp 有时实现一些复杂的客户端信息操作（参见图 11-1b）。例子包括从基于浏览器的界面到底层功能应用，以及包含复杂信息操作的应用，包括预加载数据以及 AJAX 应用。

例 3：服务器端内容处理。WebApp 可以包含服务器端功能来管理动态和快速变化的内容（参见图 11-1c）。例子有：讨论板、维基（wiki）以及一些展现来自实时事件内容的站点（例如，新闻和体育站点）。

例 4：服务器端大数据集管理。WebApp 可以使用服务器端功能来管理大而复杂的数据集（参见图 11-1d）。例如，产品目录、文档仓库、图书馆、人员列表。

例 5：过程和工作流支持。WebApp 常常完成一个特定的计算的或者传输的过程，这个过程支持过程化的数据录入、事务管理或者业务工作流（参见图 11-1e）。例如在线购物、注册过程以及在线调查。

11.2.2 功能分类

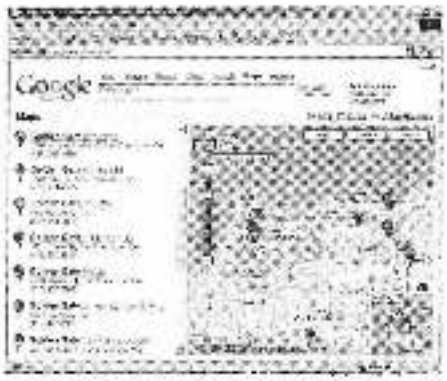
我们刚刚讨论的例子说明了 WebApp 功能的多样性，但是仅仅有这些例子是不足以指导我们设计功能的。作为讨论功能设计的第一步，我们将建立功能分类，理解每一类的属性，然后考虑作为设计过程一部分的每个种类是如何进行的。WebApp 的功能可以分成六大类，总结在表 11-1 中（每一类都有一个来自 SafeHomeAssured. com 的简短例子）。

应该注意用户层功能种类并不意味着客户端功能（例如，为用户实现的“关闭”）。类似地，应用层功能种类也不意味着服务器端功能。功能在哪里及其如何实现是一个本质问题，将在功能架构设计时进行解决。这两个主要分组与功能的意图——以及由此你应该何时（而不是怎样和在哪里）考虑它相关。

例如，在 SafeHomeAssured. com 中，要展示给用户的产品信息格式应该考虑为 1B 类（用户信息支持），但也可能发生在服务器端的内容管理系统（CMS）内。相反地，为用户与应用交互（既包括选择哪一个链接，又包括鼠标移动到哪里）所记录的日志应该考虑为 2A 类（应用交互支持），但这可能是由客户端脚本跟踪的。



a）作为客户端信息访问支持例子的下拉菜单（www.smh.com）



b）作为客户端数据操作例子的图像缩放和滚动（maps.google.com）



c）作为处理动态内容功能例子的实况记分更新（www.wimbledon.org）



d）作为管理大量数据集例子的搜索产品库（www.amazon.com）



e）作为WebApp过程支持例子的一个工作流过程（www.amazon.com）

图 11-1　不同 WebApp 功能示例

表 11-1　WebApp 功能种类

第 1 组：用户级（外部的）功能。这些种类包括直接影响用户对 WebApp 的体验并且通常直接被他们感知的功能。

种类 1A：用户交互支持。这类功能包括任何影响用户可能直接与 WebApp 交互的方式。包括：动态调整访问机制（例如，当鼠标在链接上面时突出显示链接），简化访问导航机制（例如，下拉菜单），导航结构和可用选项指定（在线或离线，个人或全局），并提供全局导航选项（例如，站点上菜单）和位置支持（例如，层级）。

（续）

SafeHomeAssured. com 的例子：如图 9-4 所示，一个用户使用一套工具绘制建筑空间的布局，这一特定的增量需要非常复杂的绘图支持，比如从绘图工具箱中拖放元素。

种类 1B：用户信息支持。这类包括任何影响信息本质和（或）如何展示给用户的功能，以及内容和展示适应，实时或动态内容更新，用户内容修改，并提供全局情境有关的信息（例如，自动包括一个站点旗帜或强加特定的样式表功能）。

SafeHomeAssured. com 的例子：给用户展示安全监控实时传感器数据（参见图 9-8）。

种类 1C：用户任务支持。这包括任何支持、指导或控制用户完成特定任务的功能。最常见的例子是搜索和工作流管理（例如，处理一个用户之后的工作流路径，包括各种选择和例外），但是它也包含以下功能：生成跟踪和层级、动态检查和反馈用户提供的信息、登录机制和访问控制。

SafeHomeAssured. com 的例子：产品订单管理。

第 2 组：应用级（内部的）功能。这些种类与支持 WebApp 必要的功能有关，但是只对用户间接可见。

种类 2A：应用交互支持。这类包括对管理和维护交互机制必要的有关功能，但不直接控制用户交互的特定元素。例如，这将包括跟踪和记录特定用户的交互（因此可以发生后续的评估和适应）。其他方面包括以下功能：搜索所有生成和维护、使用日志和分析以及监控断的链接和其他可能的交互错误。

SafeHomeAssured. com 的例子：根据用户的产品历史进行对他们进行资料分析。

种类 2B：应用信息支持。这类包括任何应用级信息管理。包括以下功能：开发和维护内容管理系统（CMS）、数据库维护以及监控内容过期或样式不一致。

SafeHomeAssured. com 的例子：在客户没有登录期间客户安全传感器后台监控。

种类 2C：应用任务支持。这类包括 WebApp 整体用户任务的内部支持。这包括到其他应用的网关（例如，电子商务 WebApp 中的支付系统）、用户认证、批处理、离线或时间紧迫的任务处理（如提交订单的处理或在线拍卖过程的监控）等方面功能。

SafeHomeAssured. com 的例子：批处理客户订单。

虽然这些类型大致正交，但是并不是完全独立的，注意这一点也是很有用的。事实上，在大多数情况下，可能有很多种不同的方式看待同一个功能。可以用 SafeHomeAssured. com 中提供来自活动的安全传感器实时信息的更新功能来说明这个问题。我们可能通过用户层功能来实现实时更新。这些功能可以周期性地使用户请求更新，从而动态地从底层数据仓库生成所请求的更新。相对应地，我们也可以将这作为一个应用层问题来解决，即存在一个后台处理，从底层数据流中定期重新生成传感器页面。在这种特殊情况下，这两种方法应该同时使用。定期地对可用信息进行更新以反映出当前的传感器数据，这属于应用层功能。重复地刷新所要显示的信息是用户层功能。

11.2.3 信息和功能总是可区分的吗

信息域和功能域的边界通常是模糊不清的。考虑一个为了使用户可以在信息空间中维护访问方向的 WebApp 信息架构，在其设计中包括设计导航层级（第 10 章）。功能设计人员可能会在 Web 服务器上加一些代码，用自动为每一个要发布的页面生成并添加导航来实现这一特殊的信息架构特性。在这种情况下，信息设计上的一个特殊需求就驱动了功能设计。

有时也会有相反的情况发生（虽然这不太常见）。举一个简单的例子，某个特殊工作流包含一系列步骤。每一个步骤都实现成导航路径中的一组页面。在这种情况下，功能设计中的一个特殊需求就触发了信息设计。

这说明信息和功能设计是相辅相成的，并且可能需要同步进行。我们将在下面的几节进行详细讨论。

11.3 设计过程中的功能设计

功能设计不是设计过程中一个具体可执行的任务的一个点，而是与其他设计活动交织在一起。第11.2节给我们提供了一种重点考虑所需功能的不同方面，以及在设计过程中处理这些方面的位置的方法。

综观WebApp设计，用户层功能和应用层功能之间有一个重要的区别。用户层功能是WebApp支持用户达到其目的的能力的描述。应用层功能表示了较低层内部功能的设计，这些功能可能不是直接对用户可见（除了以间接的方式）。有了这些定义，就可以说用户层功能和WebApp的核心需求以及随后的分析是紧紧密耦合的。应用层功能是嵌入在WebApp结构中较深层次的功能，将随着用户层功能设计的逐步进行而不断出现。

参见图11-2，外边几层表示逐步完善的WebApp规格说明，内部几层与设计相关。在设计中与需求描述关系最密切的方面是交互设计，它是用来捕获用户和应用交互方式的一个任务。交互（界面）设计既有信息元素，又有功能元素，还常常涉及特定的信息设计和功能设计任务。信息设计和功能设计可以并行进行，而且经常是反复进行的。用户层功能设计响应已设计的交互，并决定如何实现这些交互。应用层功能设计提供实现用户功能的内部支持。因此，应用层功能设计一般随着用户层设计的进行而进行，并且需要对其响应。

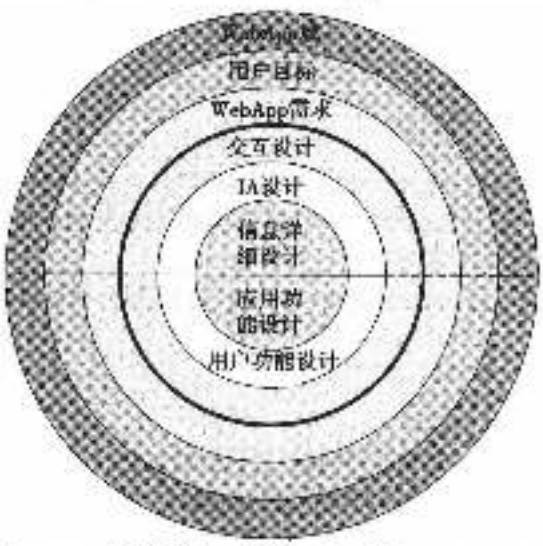


图11-2 不同功能层和其他相关设计任务之间的关系

11.3.1 功能设计过程包含哪些要素

功能设计包含的要素如图11-3所示。参照这幅图，功能设计是从用户功能设计开始的，而用户功能是从用户目标定义中得来的（以自然语言描述和从沟通活动中的用例进行记录）。用户功能以交互模型（UML序列图和状态图）和功能模型（UML活动图）来描述。

用户功能设计与信息架构设计并行进行，以使得它们保持一致。实际上，我们在开始设计时考虑了分析模型（第7章）和初步信息架构（第10章），然后审查功能是如何影响如下几个方面的：

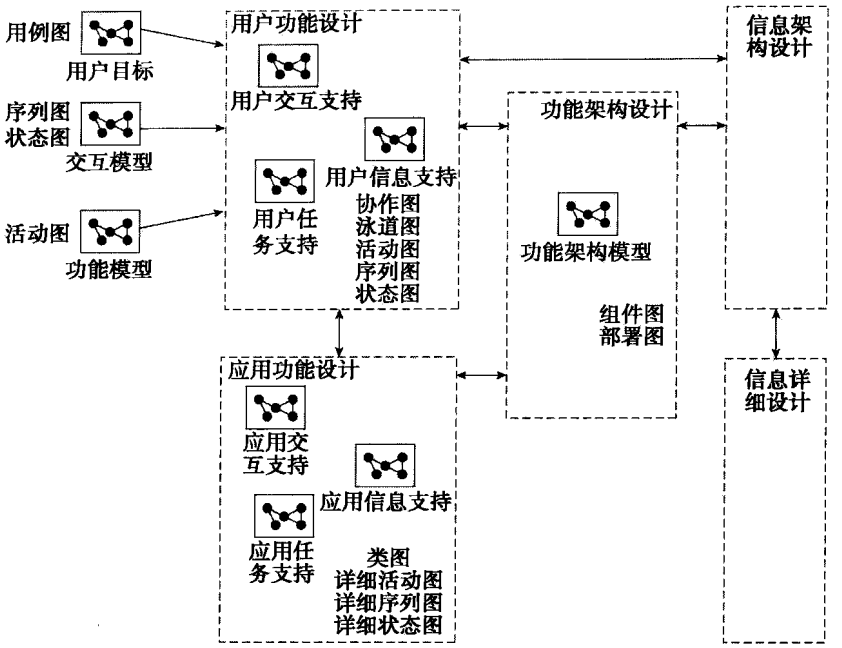


图 11-3 WebApp 功能设计过程

- 用户与应用的交互
- 要显示的信息
- 要实现的用户任务

用户的功能设计和信息设计结合在一起创建功能架构。*功能架构*（functional architecture）是 WebApp 的功能域的表述，描述 WebApp 中的关键功能组件以及这些组件之间如何交互。鉴于它的核心地位，我们将在第 11.4 节进行详细讨论。

应用功能设计定义 WebApp 的内部支持功能。它通常伴随着用户功能，而且也将影响功能架构。

11.3.2 需要多少功能设计

像所有其他设计任务一样，功能设计的格式和范围取决于 WebApp 的性质。对一个大规模的、具有对整个组织的业务过程有很大影响的复杂功能的 WebApp 来说，通常需要完成如图 11-3[⊖]所示的所有功能设计建模任务。相反地，一个只补充组织的业务操作的较小的 WebApp，需要少得多的形式化设计。在这种情况下，功能设计可能只限于用序列图来表示用户功能，用组件图来表示功能架构。

对 SafeHomeAssured. com 来说，不同增量需要的功能设计层次不同，WebApp 第一个增量（与公司及其产品相关）的功能复杂度非常低（可能仅限于产品数据库的一些内容的管理），因此功能设计会是最低限度的。对于增量 6（与传感器的控制和监控相关），功能将会复杂得多，因此功能设计将会非常详细。

11.3.3 SafeHomeAssured. com 的初始功能设计应该如何进行

SafeHomeAssured. com 具有一个关注信息的部分和关注功能的部分的有趣组合。在初始沟通

⊖ 当 WebApp 对潜在的或者实际的客户来说是关键业务接口时，这是非常正确的。

活动中（第4章），我们为 SafeHomeAssured. com 识别出了一组初始信息及所需目标，下面是其中的一部分：

- 为用户提供请求的产品规格说明。
- 为用户提供使他们可以展示要保护的空间（例如，住宅、办公室/零售区）的布局的工具。
- 制定可以在用户空间中进行使用的安全和监控产品的定制建议。
- 使用户可以获得产品成本的报价。
- 使用户下订单订购安全硬件。
- 允许用户控制其空间内的监控设备（例如，照相机、麦克风等）。
- 使用户可以签订监控服务的合同。
- 允许客户查询监控数据库中有关他们账户的活动。

然后这些目标被简化成如下要实现的功能：

- 提供产品报价。
- 处理安全系统订单。
- 处理用户数据。
- 创建用户资料。
- 绘制用户空间布局。
- 为布局推荐安全系统。
- 处理监控订单。
- 获得并显示账户信息。
- 获得并显示监控信息。
- 客户服务功能（在后面进行定义）。
- 技术支持功能（在后面进行定义）。

最终，这些功能被精化为一组用例，这些用例获取了关键的用户信息和功能性交互。

正如我们前面所述的，SafeHomeAssured. com 的第一个增量是信息占主导地位（第4章），后续的增量将开始逐渐引入越来越复杂的功能。我们来考虑增量2，它包含如下的使用场景或用例：

卡片号	卡片名称
2	下载产品规格介绍
3	获取为我的用户种类定制的信息
4	查询一个特定的传感器

在分析建模期间，WebE 团队为这个增量开发了简单的内容和交互模型，但是又认为这些功能太简单而不需要一个单独的功能模型（尤其是当大部分功能能够很清楚地从交互模型的序列图中表现出来的时候）。

在这种情况下，因为这三个使用场景都和信息访问有关，设计将和信息架构相一致。然而一旦信息架构的草案开发出来，团队就能够提出和回答如下问题：

问：是否有任何需要的功能会影响用户与应用的交互？

答：对场景4来说，用户通过展开产品类别浏览产品目录。这个场景表明用户从所有类别列表中选择一个特定的类别，然后就会有子类别列表显示出来。当用户选择一个子类别时，此类别中的所有产品都会列出来，然后就可以选择其中一个特定的产品。

这在图11-4的线框图方法行展示。

问：是否有任何需要的功能影响所显示的信息？

答：对场景3而言，显示的产品是为用户类别（例如，游客、注册用户或房主）定制的信息。之前开发的领域模型表明，某些特定的产品仅仅对注册用户可用，某些特定的信息（如价格）根据用户类型的不同而改变。所以需要支持内容调整的功能。

问：是否有任何需要的功能会影响用户正在进行的任务？

答：访问产品需要随意支持。



图11-4 SafeHomeAssured.com产品浏览的示例线框

为这些用户功能所进行的初始功能设计的其他方面可能会被立即实现，但是实现这些功能的其他方面将关系着或者影响整体功能架构的开发。例如，当一个类别被选择时应该改变子类别列表。而当一个子类别被选中时应该改变产品列表，与此相关的功能将需要客户端功能，可以被立即设计，或者更可能从已存在的此类功能的Java脚本代码库中提取。然而，这些功能也需要合适的服务器端功能来生成产品列表以及构造相应的客户页面。整体架构决策也是从这里开始做出的。

我们在这一小节讨论说明某些功能性设计是自顶进行的，至少有一部分是这样，因为有固有的可以被广泛的Web架构所使用的结构和技术。然而其他方面需要慎重考虑，而且需要开发一个整个功能架构。

11.4 功能架构

功能架构是WebApp的功能域的一种表示，这种架构表示回答了两个关键问题：

- 我们如何把功能分成定义清晰的角色和界面组件？
- 每一个功能组件存在于哪里，它与什么交互？

换句话说，功能架构把WebApp分成多个功能组件。

11.4.1 功能架构是什么样的

有很多方法来建模和文档化功能架构[⊖]，然而，因为大多数 WebApp 是高度模块化的，一个好的基本模型是 UML 组件图［OMG04］。组件图说明 WebApp 将如何被组成一组主要的系统组件，并且表示这些组件[⊜]之间的相互关联。

图 11-5 展示了 SafeHomeAssured.com 的增量 2 的一个初步功能架构。注意，功能组件的位置描述功能组件在哪里被执行而不是被存储在哪里（或者哪里产生，如果功能是动态生成的）。这个架构的决策是动态生成产品信息页面（包括中间类别页面）。之所以如此设计是因为实际的内容是基于用户类别的。在这个特定的架构中，产品访问、类别及产品信息页（这些页将被设计成信息架构的一部分）是生动态页面编辑组件动态产生的。反过来，这将使用搜索结果生成或者产品页面生成组件来创建页面内容，使用产品菜单生成组件产生相关的菜单和其他的导航帮助（如导航层级）。后台组件搜索索引器将负责生成搜索索引。

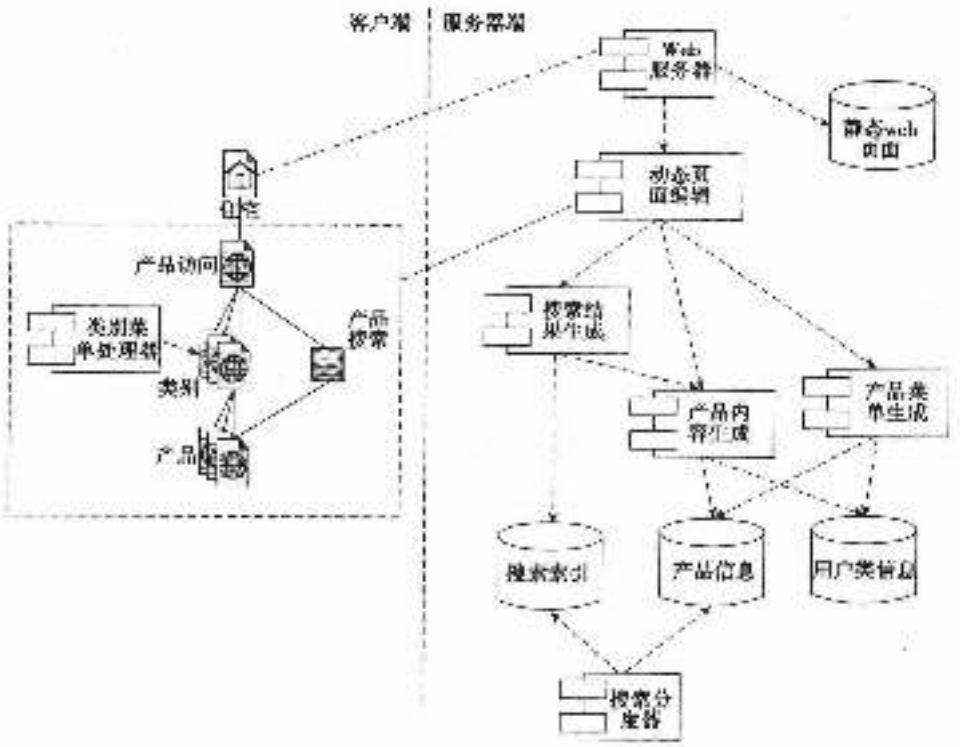


图 11-5 SafeHomeAssured.com 增量 2 的功能架构

要注意的是，这个架构模型已经进行了扩展，用来展示与信息架构[⊝]的关系。例如，动态页面编辑组件负责创建产品访问页面和在这个信息层中此页面下的所有子页面。当 IA 发生变化时，我们可以看到哪些功能组件可能受到影响，反之亦然。

⊖ 推荐查看［Dus03］，这是对软件架构非常好的整体介绍，它会让你对许多 Web 工程师需要关注的架构问题有一个很好的理解。

⊜ 查看敏捷建模站点 www.agilemodeling.com/artifacts/componentDiagram.htm（2007 年 8 月 7 日）中对 UML 2 组件图的简单介绍。

⊝ 它的意思是这幅图不再是严格意义上的 UML 组件图，而只是为了描述信息和功能架构之间的关系，我们认为它是合适的。

11.4.2 如何开发功能架构

明显的启动功能设计的地方是考虑 WebApp 的分析模型（包括相关的任何规格说明书）和初始的信息架构。我们已经提到要识别存在于表 11-1 中的每一类功能是可以做到的，每一个场景或者部分场景（例如，基于用户类型定制产品内容）可以被分解成如下通用的场景组件类型：

- 信息选择（例如，与识别和对展示给用户的信息的选择相关的功能）
- 信息组织（例如，和把信息整合成一个展示给用户的组合体相关的功能）
- 信息处理（例如，数据的分析或计算）
- 系统交互（例如，和 WebApp 外部的其他系统进行交互的相关功能）

然后，我们考虑特定的场景中，在事件初始化自动或者手动启动时，组件是否将在用户请求下被动态地调用。最后，我们将对架构组件进行聚类、合并以及精化。下面以与 SafeHomeAssured.com 的一个功能架构开发相关的工具讨论为例。



功能架构建模

场景：SafeHomeAssured.com 的 WebE 团队会议室，第二个增量的 IA 初步开发之后（产品评估）。

参与者：WebE 团队负责人以及担任系统架构师的小组成员

会话：

团队负责人：我们已经对第二个增量有了一个相对清楚的 IA，对于怎样将它与所需的功能联系起来，你们有什么想法？

系统架构师：嗯……我一直在想总体功能架构。如果是一个简单的小的产品目录，我们可以仅用静态页面来管理，即使目录比较大，我们仍然可以用一个单独的数据库存储所有的产品信息，数据库中有另外一张表来将产品和一个或多个类别链接起来。然后我们可以通过动态生成中间页面以及产品页面的简单脚本来访问产品。这非常简单，因为客户端已经有了在这一组可以合并的数据库中的所有产品信息。但是……

团队负责人：那么，会引起什么复杂性呢？

系统架构师：这个……在这种情况下，我们需要基于用户身份调整产品内容。知道用户类型是非常容易的——我们只需用一个 Web 服务器会话变量获得用户登录时的设置——但是，麻烦的问题是我们事实上改变了内容以及其他方面。

团队负责人：你的意思是……？

系统架构师：事实上，我们仍然可以动态提取内容，只是有一些额外功能和第一次调整内容来修改价格之类的信息相关。记得营销人员说过，特定的用户在标准价格上有自动折扣。我们还必须确保只能链接到而且只能访问与用户类别对应的页面。

团队负责人（指着图 11-5 所示的图）：那好，那么在你的概要设计图中……产品内容生成组件是用来为用户生成正确的产品信息的，产品菜单生成组件生成相关产品的结构和链接，同时，动态页面编辑组件将它们组成整体……

系统架构师：是的，你说得对。同时编辑组件还做了所有的整理工作，例如加上正确的标题和页脚。

[他们继续讨论这张图，包括如何将搜索功能集成到架构中去。]

除了在非内讨论的因素之外，许多其他因素也会影响功能架构的设计。例如，允许多个管理员用户来管理和更新内容可能意味着使用一个 CMS 是合适的。这个步给部分功能架构提出或者甚至是强加一种特定的形式（例如，CMS 可能会有助于自动进行内容调整）。

11.4.3 处理异常的功能如何开发

由于在功能设计中的一个重要因素通常是异常处理，所以使系统能处理异常情况对设计功能来说非常重要。对 SafeHomeAssured. com 的大多数增量来说，这不是主要的问题，但是异常处理必须仍然被考虑。例如，产品信息是在动态页面编辑组件中被评估，这个组件生成 SafeHome 产品的访问页面，还生成中间的类别页面，这些中间的类别页面形成了对产品的导航路径（就像在信息架构中定义的那样）。这允许这些中间页面是以确保它们包含的链接只能链接到对当前用户可用的产品上的方式生成。考虑如下的两个用于同类安全摄像机页面的 HTML 片段。

Web 页：安全摄像机类别（对游客）

```
...
<a href="/products/genprod.exe? type=prod&prod-id=3467"> <div class = "Prod-
uct">Camera Model XF67 </div> </a>
<a href="/products/genprod.exe? type=prod&prod-id=3470"> <div class = "Prod-
uct">Camera Model XQ99 </div> </a>
...
```

Web 页：安全摄像机类别（对账户持有者）

```
...
<a href="/products/genprod.exe? type=prod&prod-id=3467"> <div class = "Prod-
uct">Camera Model XF67 </div> </a>
<a href="/products/genprod.exe? type=prod&prod-id=3468"> <div class = "Prod-
uct">Camera Model XF69 </div> </a>
<a href="/products/genprod.exe? type=prod&prod-id=3470"> <div class = "Prod-
uct">Camera Model XQ99 </div> </a>
...
```

动态页面编辑组件已经排除了（从为游客提供的列表中）只对账户持有者可见的安全摄像机产品（Model XF69）中的一个。然而，关键在于，如果产品信息的 URL（或更进一步，动态生成的产品信息的 URL）被一个账户持有者发送给一个游客，会出现什么问题？这就是说，如果一个游客直接进入 www. safehomeasssured. com/products/genprod. exe? type = prod&prod- id = 3469，会发生什么？

如果 SafeHomeAssured. com WebApp 设计成依赖于产品菜单生成组件[⊖]来控制对产品的访问，那么用户就可能会很容易地绕过这个控制。要避免这个问题，动态页面编辑组件还必须检查合适的访问，在必要时以一个恰当的错误信息页面作为响应。这就是确保功能架构解决可能的异常而没有产生“功能漏洞”的一个例子。

总体来说，异常处理是简单信息型 WebApp 中的一个相对简单的设计问题。当 WebApp 有复杂的工作流、在工作流中时间的顺序可以被组合或引起难以处理的情况时，异常处理就会复杂得多得多。例如，考虑在线旅馆 WebApp 的一个场景。一个用户希望为某个晚上预定一个房间，用户输入日期，WebApp 检查可用房间并找到一间可用的房间。这是和用户交互的，用户然后请求预定房间，并提供付款详细信息。WebApp 确认此预定。

这个工作流看起来足够简单。但是，如果在检查房间的可用性（只有一个房间可用）和请求预定之间，有其他人预定了这个房间，会发生什么事情？如果这个工作流设计得不好，WebApp 可能不会在付款时再次检查房间的可用性，然后这个房间可能就会被预定两次！诸如此类

⊖ 产品菜单生成组件将为一个具体用户产生他可访问产品的链接。

的问题在 Web 环境中因为 WebApp 具有的分布式特性而更加严重，在第 11.5 节中，我们将简要研究如何使用状态图模型来处理这个问题。

功能架构理论上可以先于选择支撑技术而设计（然后用来驱动技术的选择）。实际上，这种情况很少发生，在技术框架和采用的架构之间有很强的相互作用——因为框架强加了某些约束，同时支持某些方法。我们将在第 14 章详细讨论这些问题。

11.4.4 在功能设计时可以使用架构模式吗

第 13 章将详细讨论模式。然而，一种特殊的模式（模型 - 视图 - 控制器）对 WebApp 架构来说是非常常见的，所以值得在此简要讨论。

Jacyntho 和他的同事［Jac02］提出了一个三层设计架构，这个架构分离了一个应用界面。他们认为，使界面、应用和导航分离简化了实现并增强了重用性，这种模型 - 视图 - 控制器（MVC）架构［Kra88］是许多被提议的把用户界面从 WebApp 功能和信息内容中分离的 WebApp 架构中的一个。MVC 架构中的**模型**部分（有时又指模型对象）包含所有应用特定的内容和处理逻辑，包括所用的内容对象、对外部数据和信息源的访问以及应用特定的处理功能。换句话说，模型本质上是内容以及任何元数据、导航结构、用户资料等。

MVC 架构的**视图**部分包含了所有界面特定的功能，同时使得能够展示内容和处理逻辑，包括所有的内容对象、对外部数据和信息源的访问以及所有最终用户所需的处理功能。

MVC 架构的**控制器**部分管理对模型和视图的访问和操作。同时也协调它们之间的数据流。在一个 WebApp 中，控制器监视用户的交互，并以此为基础从模型中取出数据，然后使用这些数据更新或构建视图。如图 11-6 所示是 MVC 架构的一个图示。

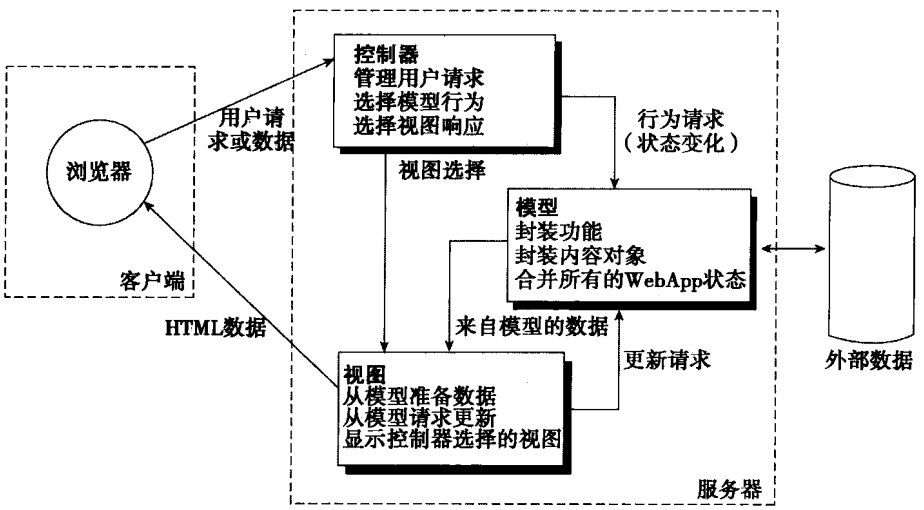


图 11-6 MVC 架构

来源：摘自［Jac02］。

参照此图，由控制器来处理用户请求或数据。控制器还根据用户请求选择可用的视图对象。一旦请求的类型确定，一个行为请求就传送给模型，由模型实现功能或者检索适应于请求的所需内容。模型对象可以访问存储于作为本地数据存储一部分的企业数据库或者一组独立文件中的数据。由模型生成的数据必须由合适的视图对象进行格式化或者组织，然后从应用服务器传送回客户端的浏览器，从而在客户端的机器上显示出来。

比较图11-5和图11-6，你会发现，虽然视图和控制部分是相互影响的，但是为SafeHomeAssured. com的增量2开发的功能架构非常类似于MVC模型。

在许多情况下，一个WebApp的功能架构是在开发环境的上下文中定义的，应用要在开发环境中实现（例如，ASP. net、JWAA或J2EE）。如果你有更多的兴趣，可以参看［Fow03］中对开发环境的讨论以及它们在WebApp架构中的地位。

11.5 详细功能设计

由于WebE过程是增量的，而且WebApp的构建通常大量使用基于组件的开发，详细功能设计将产生较少的形式化模型或详细的文档。当需要建模时，它很大程度上借鉴传统的诸如UML［OMG 04］［Pil05］的功能设计方法。UML为功能设计者提供了大量的设计表示（例如，序列图、状态图），但是没有提供与详细信息设计之间的好的集成。不过，WAE和WebML（第10章讨论的信息设计方法）两者都可以扩展到功能领域。

11.5.1 WAE建模如何用于详细设计中

WAE对UML建立了一组扩展，这些扩展易于对WebApp的低层设计进行建模［Con99］。在第10章中，我们以SafeHomeAssured. com为例在图10-11中说明了WAE的使用。WAE特别适合于为WebApp的典型的客户端－服务器交互建模。

图11-7列出了一个WAE模型中表示对象的主要图标。将这些图标进行组合就可能表示数据对象、功能对象和对象之间的复杂交互——还可以表明这些对象之间的交互发生的位置。分析视图已在第10章中进行了说明；逻辑视图描述WebApp中的概念元素；而物理视图则表明逻辑视图是如何映射到实际的实现组件。

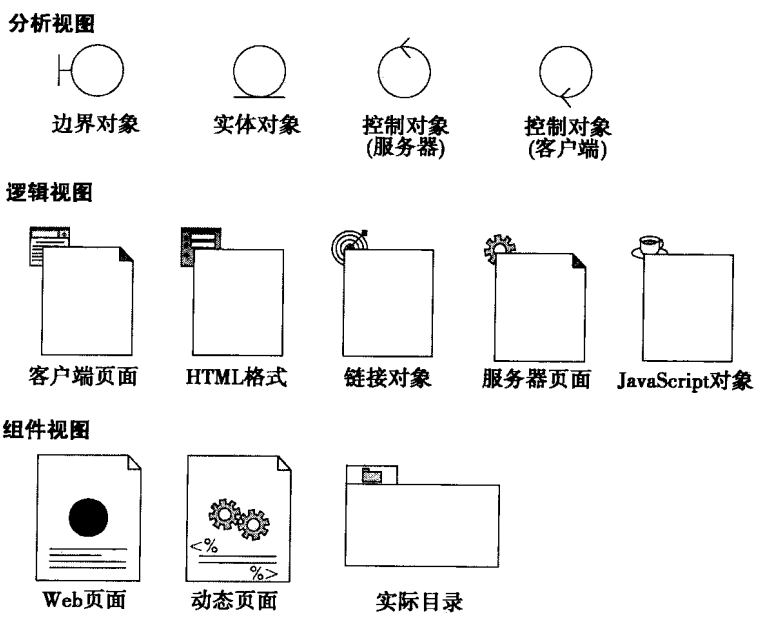


图11-7 WAE模型关键符号

回到图10-11，你可以看到，它捕获了信息架构和功能处理两者，但是没有表示将信息和功能混合的逻辑组件的具体设计。图11-8表示了从图10-11获得的逻辑设计，并强调了功能产生信息的方式，反过来也是这样。

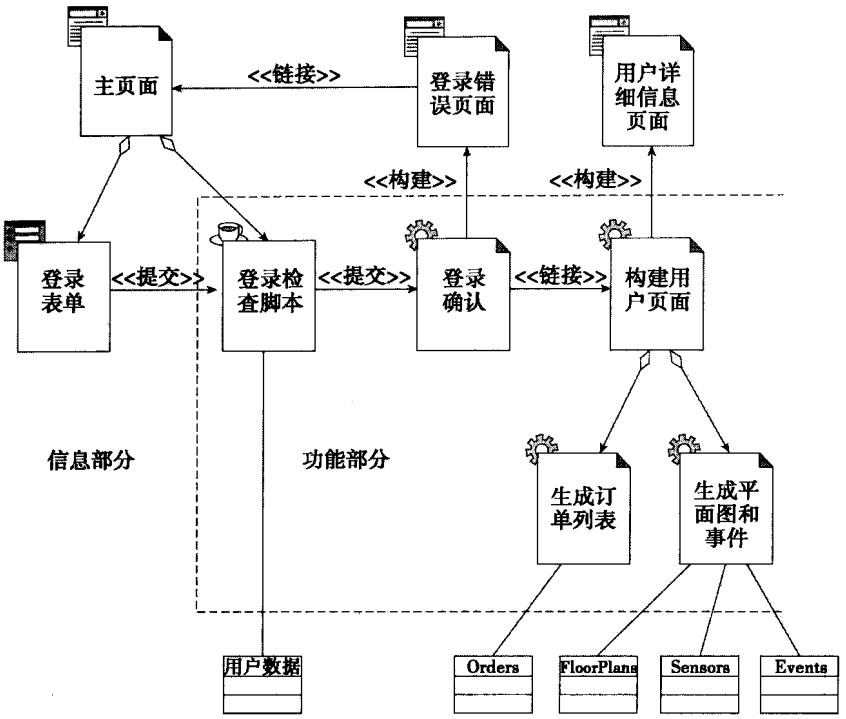


图 11-8 由 WAE 模型（来自图 10-11）导出的详细信息和功能设计

WAE 模型在研究功能 - 信息边界时很有用。但是通常情况下这些模型对处理整个 WebApp 的详细功能设计来说是相当麻烦的方法。我们推荐限制对这些模型的使用来快速地为研究特定的交互建立草图[⊖]。在开发过程中保持一个最新的 WAE 模型以及与构建的系统同步不会有任何明显的益处（极少数情况除外）。

11.5.2 WebML 为什么适合于工作流建模

如果你在开发面向工作流的应用，我们推荐采用 WebML，它可以更有效地支持工作流 [Bra03] [Bra06]。很多 WebApp 通常提供一个前端界面以组织过程，这些过程有一系列预定义的步骤。例子非常多，但是简单一些的有电子商务应用中的付款流程（例如，选择物品、确认订单、提供付款详细信息、确认付款）、比赛或者旅行订票（例如，选择比赛或者旅行路线、选择日期和时间、选择座位、确认付款）、在线拍卖、应用提交等——例子太多了。

WebML 已经被适用于面向工作流的应用的建模 [Bra06]。为业务工作流[⊖]建立模型，然后这些过程都被映射到一个 WebML 设计模型。WebML 设计模型已经扩展为能够支持更有效的对过程的建模。这种建模方法表示在应用过程中的不同用例或场景（例如，一个特定的借贷应用实例）。每一个用例都将执行多种可能会在不同的时刻发生或被不同的人员进行的活动或者过程步骤。这些过程步骤可以使用 WebML 来建模。例如，管理一个特定的借贷应用过程可能会包含如下步骤：1）客户首次提交一个应用；2）借贷管理员验证；3）银行员工完成信用检查；4）被授权的借贷批准者最终同意（或者拒绝）。

⊖ 熟悉这些符号并能够画出具体的详细设计以便审查对 WebE 团队而言是非常有用的。

⊖ 对业务工作流进行建模的方法有很多。在 WebML 中加以适用的方法称为 BPMN（Business Process Management Notation，业务过程管理标记）[Whi04a] [Whi04b]，同时它基于建模事件、任务、工件和决策点。BPMN 的详细信息可以在对象管理组织（OMG）的 BPMN 站点 www. bpmn. org/上找到。

图 11-9 表示一个和借贷请求相关工作流的 WebML 超文本设计模型。参照此图，可以看到多个工作流路径。例如，在位于图顶部的工作流中，开始时提交给借贷管理员一系列的借贷请求（LoanRequest）用例，这些用例目前被交给初始验证（PreliminaryValidation）活动（左上角方程）。如果某一个借贷请求用例被选择，那么这个工作流过程就会开始［如开始图标（有三角形的）］，同时提交给管理员这个借贷请求用例的相关细节以及一个要完成的表单（例如，初始验证页面内的数据项）。如果这个用例无效（图中“无效”），那么这个特定的活动就终止了［工作流指向“结束”图标（有一个正方形以及左上方边框有黑点）］，同时这个特定的借贷请求用例就终止了，并结束这个用例所需的任何进一步的处理步骤。如果管理员确认这个借贷请求用例是有效的，借贷请求用例的细节就将被修改（例如，“修改”图标），同时这个用例被分配给过程中的接下来的两个活动（例如，工作检查（JobCheck）和财务检查（FinancialCheck）），然后这个活动就终止了，但是很明显这个特定用例没有终止，因为更进一步的处理活动仍然在继续（用中间带大黑点的“结束”图标表示）。

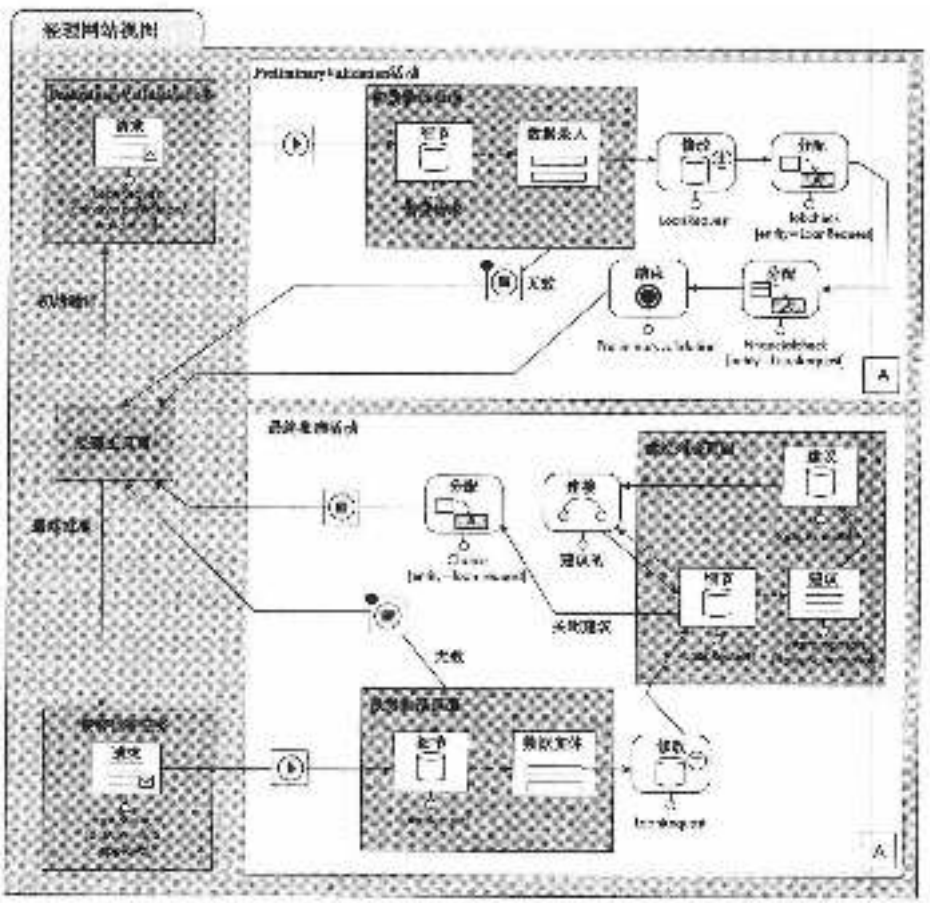


图 11-9 WebML 用于一个贷款请求过程的业务工作流建模的示例应用

来源：来自 WebML 的培训材料，特别获得了 Marco Brambilla 和 Politecnico di Milano 的件子。

像图 11-9 所示的那样，在我们对复杂工作流进行合理建模时，WebML 模型可以提供大量的帮助，使设计者在完成 WebApp 功能架构前发现错误、遗漏或者不一致。如我们已经讨论过的那样，WebML 建模方法对简单的功能有些过度，例如，SafeHomeAssured.com 的增量 2 是个典型的例子（它的功能几乎完全由用户浏览事件触发，而且在用户之间的事件或交互中很少有处理

或者“历史”)。然而，当 WebApp 中的功能是要及时分发的或者在多个用户之间分发时[⊖]，建议使用 WebML 建模。

对 SafeHomeAssured. com 来说，任何包含用户和 WebApp 之间异步交互的 WebApp 增量都将是使用 WebML 进行建模的一个候选，例如，SafeHomeAssured. com 的增量 4 包含了场景“获取为我的空间所推荐的传感器布局”。如果一个用户输入一个布局，WebApp 就立即生成一个推荐方案而不需要相关工作流，可以设计一个简单的功能组件。相反地，如果布局推荐方案需要 SafeHomeAssured. com 员工手动生成（甚至在发布给用户之前进行检查），那么就应该包含一个合适的工作流来管理对推荐方案的请求。虽然这个推荐方案可以自动生成，但是所请求的处理工作意味着它不能凭空产生（在推荐方案生成之前，请求需要被放在一个队列里），这就需要设计一个工作流，一旦推荐方案生成就把结果返回给用户。

你可以在 WebML. org 网站 www. webml. org 上找到更多更好的关于 WebML 的信息，包括过程和工作流建模的扩展。

11.6 状态建模

随着 WebApp 越来越复杂，你很可能必须适应交互过程，尤其是有多个并发用户的（或至少有多个用户，他们和 Web 服务器的交互是交叉进行的）。WebApp 设计的一个重要问题就是确保底层信息状态在我们进行复杂交互处理时被正确保存。

在前面的例子中，我们讨论了一个在线旅馆预订的应用。假定 WebApp 中的交互处理是正常的，那么重复预定（或者指示一个房间是可用的，然后不得不将其指示为不可用）的潜在性就是一个重要的设计问题。为此类情况做出有效的设计，可以通过使用状态建模来进行管理。

状态是行为的外部可观察的模式。外部刺激引起状态之间的转换。状态模型通过描述 WebApp 的行为的状态和引起 WebApp 的状态改变的事件来表示 WebApp 的行为。而且，状态模型还表明一个特定事件发生的结果是执行哪些动作（例如，过程被激活）。可以用状态图来创建状态模型。

在状态图中，设计者可以表示 WebApp 的行为。当不同行为可以被多种不同事件触发时，或者在当前 WebApp 处理依赖于先前的一个或多个用户的动作的情况下，尤其是对一个用户的响应会被另一个用户的行为影响时[⊜]，状态图是非常有用的。

图 11-10 说明了 SafeHomeAssured. com 的新客户交互的状态图。图中标明了六个外部可观察的状态（由圆角矩形表示）。这个状态图显示了使客户从一个状态转移到另一个状态所需的事件、当进入一个状态后显示的信息、在一个状态中发生的处理以及允许转换发生的退出条件。

SafeHomeAssured. com 通过用户观察的行为和其他用户正在做的动作、客户的异常以及在家庭监控中交互的 SafeHomeAssured. com 员工毫无关系。考虑一个客户已经为监控服务签署了使用合同这种情况，SafeHomeAssured. com 监控软件可能会自动地监控摄像机（通过图像处理算法）以及客户住宅里的其他传感器，并且识别异常状况，例如：

1. 安全系统已经启动预警，并且入口传感器指示一个少于 x 秒的安全漏洞。这种情况可能发生在当一个房主进入住宅时没有首先解除安全系统的预警。一个声音报警应该在给定的时间间隔内提醒房间主人解除预警系统。

2. 安全系统已经启动预警，入口传感器指示一个可能的多于 x 秒的安全漏洞，或者在其他传感器上多于 y 秒的安全漏洞。这可能是指示一个真正的安全漏洞。

⊖ 例如，如果一个用户提交了一个请求，而这个请求必须接着被另一个用户（或者另一类用户）处理，那么采用这种方法来对过程用例进行建模，并把组成过程的部分的活动分割是非常有益的。

⊜ 例如，旅馆房间的可用性依赖于其他用户是否已经预定了这个房间。

3. 安全系统中的传感器指示可能的火灾或者其他紧急情况。

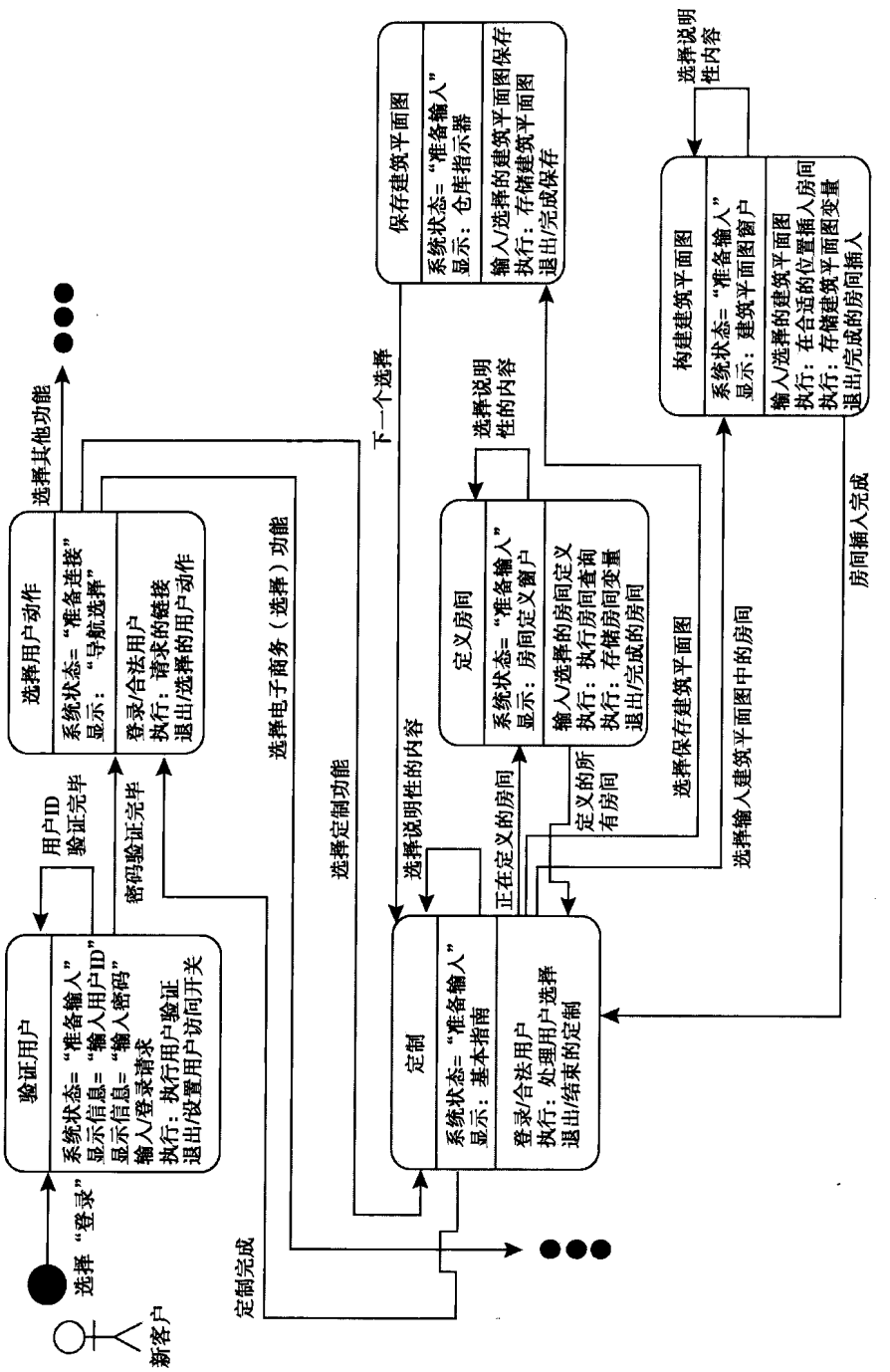


图 11-10 SafeHomeAssured. com 的状态图示例

在状况 1 中，事件应该被记录下来，但是不采取进一步的动作，在状况 2 中，SafeHomeAssured. com 应该开始它的安全漏洞处理。这可能包括一个 SafeHomeAssured. com 员工被警告或者告知一个可能的漏洞（通过相关的工作流）。然后，这个员工可以查看漏洞信息（例如，视频片段或者跟踪传感器的位置）以及客户指明的要在漏洞事件中被警告的人员的信息（例如，警察

或者私人保安）。工作流系统可能会把可能的漏洞排成队列来处理，如果没有在60秒内按照预先安排的流程执行，就会把这个漏洞升级为一个关键状态。在状况3中，过程稍微有点不同，不仅包含即时联系紧急服务，还包含SafeHomeAssured.com与房主进行联系，警告他这个危险事件。

这些场景立即引出一些问题，比如一个漏洞应该被跟踪多长时间？如果这个漏洞持续（例如，传感器坏了）将发生什么？答案可能已经在沟通活动中或者创建分析模型时得到了，但是还有一些问题可能要在详细设计中才能解决。

11.7 本章小结

目前，许多WebApp包含了丰富的动态内容、对个体用户的适应性和个性化、强大的交互性，以及复杂的工作流和过程的支持。所有这些都已经使得支撑大多数WebApp的内容和带给这些内容生命并使得这些内容可访问和有意义的功能之间的互相影响更加复杂。

在这一章，我们探究了如何设计功能以能够支持目前一代的WebApp。功能设计不会独立出现，而是和分析模型、信息设计以及用来实现设计的技术选择紧密相连。在本章中有些关键信息不是关于具体的符号或方法，而是关于我们如何将分析和设计元素联系起来。

功能架构很大程度上受信息架构的影响。WAE提供了一种用于考虑如何把精化的导航路径和生成路径中这些页面所需的功能联系起来的工具。WebML使设计者以一种可以产生更加有效的功能设计的方式对复杂的过程工作流进行建模。

现在我们已经研究了信息设计和功能设计，下一步是了解它们如何在实际的技术设计和WebApp构建中一起使用。在第12章，我们将研究把设计转换成可工作的WebApp所需的任务和技术。

参考文献

[Bas03] Bass, L., P. Clements, and R. Kazman, *Software Architecture in Practice*, 2nd ed., Addison-Wesley Professional, 2003.

[Bra03] Brambilla, M., S. Ceri, S. Comai, P. Fraternali, and I. Manolescu, "Specification and Design of Workflow-Driven Hypertexts," *Journal of Web Engineering* (*JWE*) vol. 1, no. 2, April 2003, pp. 163-182.

[Bra06] Brambilla, M., S. Ceri, P. Fraternali, and I. Manolescu, "Process Modeling in Web Applications," *ACM Transactions on Software Engineering and Methodology* (*TOSEM*) vol. 15, no. 4, 2006, pp. 360-409.

[Cer00] Ceri, S., P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): A Modeling Language for Designing Web Sites," *Proceedings of WWW9 Conference*, Amsterdam, 2000, pp. 137-157.

[Cer02] Ceri, S., P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera, Designing *Data-Intensive Web Applications*, Morgan Kaufman, 2002.

[Con99] Conallen, J., *Building Web Applications with UML*, Addison-Wesley Object Technology Series: Addison-Wesley, 1999, p. 336.

[Fow03] Fowler, M., et al., *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2003.

[Jac02] Jacyntho, D., D. Schwabe, and G. Rossi, "An Architecture for Structuring Complex Web Applications," 2002, www2002.org/CDROM/alternate/478/ (accessed August 7, 2007).

[Kra88] Krasner, G., and S. Pope, "A Cookbook for Using the Model-View Controller User Interface Paradigm in Smalltalk-80," *Journal of Object-Oriented Programming* vol. 1, no. 3, 1988, pp. 24-69.

[Low03] Lowe, D., and B. Henderson-Sellers, "Characterising Web Systems: Merging Information and Functional Architectures," in V. K. Murthy and N. Shi (eds.), *Architectural Issues of Web-Enabled Electronic Business*, Idea Group Publishing, 2003.

[OMG04] Object Management Group (OMG), "UML 2.0 OCL Specification," 2004, www.omg.org/cgi-bin/doc? ptc/2003-10-14 (accessed August 7, 2007)

[Pil05] Pilone, D., and N. Pitman, *UML2.0 in a Nutshell*, O'Reilly, 2005.

[Whi04a] White, S., "Business Processing Modeling Notation (BPMN), Version 1.0," 2004, www.bpmn.org (accessed August 7, 2007).

[Whi04b] White, S., "Process Modeling Notations and Workflow Patterns," IBM Corporation BPTrends, 2004, www.bptrends.com/ (accessed August 7, 2007).

第 12 章　构建和部署

无论要构建的 WebApp 的类型和用来构建它的过程（或者没有过程）的类型如何，WebE 团队最终必须以设计为向导，以可用的技术为工具集来构建应用。团队还必须在尽可能多的要求的环境中部署 WebApp（还有一些环境直到最后才考虑到）。在第 3 章描述的 WebE 过程的上下文中，WebApp 被设计为一系列增量。此 WebE 过程的最后两个活动是：

构建。使用 WebE 工具和技术来构建已经模型化的 WebApp，一旦构建了 WebApp 增量，就会执行一系列的快速测试，以保证发现设计中的错误（例如，内容、架构、界面、导航）。还会有一些测试用来处理其他的 WebApp 特性。

部署。对 WebApp 的运行环境进行配置，然后将 WebApp 交付给最终用户，并且开始评估阶段。WebE 团队得到评估反馈，然后对这个增量进行按需修改。

构建创建 WebApp 增量中必须要交付的所有信息和功能组件。这一活动包括代码生成和测试。由于测试对 WebE 过程和最终用户感受到的质量都非常重要，因此将在第 15 章中单独介绍测试。

部署在 Web 应用增量被 WebE 团队验证之后发生，包括的任务是将 WebApp 安装在每一个需要的环境中。

很自然地，有人会问为什么将构建和部署放在同一章讨论。尽管这两个活动在概念上不同，但是 Web 的分布式特性以及 WebE 过程的增量特性使得它们紧密相连。一旦定义一个增量的组件被构建（编码及测试）完成，每一个组件都可以立即部署，以尽可能快地为利益相关者提供收益。

甚至有可能把一个单独组件定义为一个 WebApp 增量，允许一些单独组件逐步进行部署。尽管这样做可能并不太好[⊖]，但如果处理得恰当，确实能提供很大的灵活性并成为一种强大的工具。实际上，一个不断进行的内容升级、编辑改变、界面调整以及相关任务的过程都会导致 WebApp 更加有组织的演化，并且构建和部署之间的边界更加模糊。

为了在 WebE 过程中保持敏捷性，组件级设计通常不作为信息设计或功能设计的一部分进行（第 11 章）[⊜]，详细的组件级设计也通常被推迟到构建活动开始才进行。它是不太正规的，而且与代码生成和测试一起进行。正规的考虑组件设计往往受到诸如组件的选择、调整及管理等问题的限制。

在这一章，我们先介绍构建和部署活动以及进行这些活动时要用到的环境。然后，我们将研究具体的构建和部署任务并推荐一些好的实践。最后，我们将讨论组件以及它们的开发和管理如何变成构建和部署活动的不可分割的构成部分。

12.1　WebE 过程中的构建和部署

如图 12-1 所示，WebE 过程中构建和部署活动被突出显示出来，与这些活动相关的 WebE 动作也被突出显示。这幅图是对前面章节中的图形稍作修改得到的，以强调构建和部署之间的迭代。

⊖ 你有多少次遇到有着令人沮丧的死胡同的网站，它们通向“构建中的”页面，或者含有没有经过仔细检查就发布的编辑粗糙的页面？

⊜ 在这些设计活动中发生的建模针对于高层架构问题。

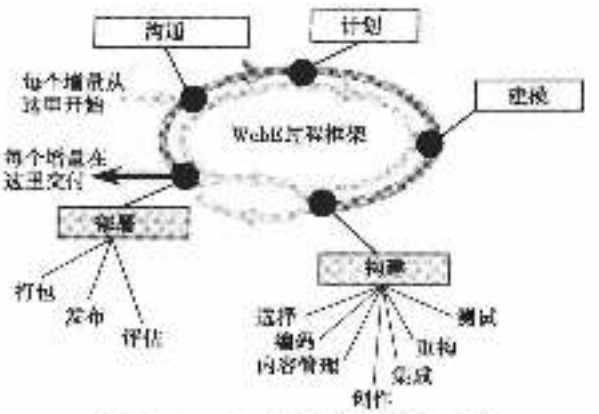


图 12-1 WebE 过程流中的部署和构建

12.1.1 构建和部署之间的相互作用是什么

尽管对每一个 WebApp 增量来说，每个过程周期是重复的，然而，对某个特定增量来说，构建与部署之间仍然有着非常重要的迭代和相互作用。在某些情况下，在构建和部署中发现一个小问题需要快速经过其他 WebE 活动（例如，建模）；有时，这样的内部迭代循环（如图 12-1 所示）是专门设置和计划的；而在其他时候，它可能只是偶然发现的或者只是开发顺序的一个结果。

例如，SafeHomeAssured.com 的增量 2 给用户提供产品信息，包括查询某个特定传感器信息的能力，这个增量还为各类用户定制信息。第 11 章所讨论的功能架构（图 11-5）包括产品内容生成（例如，为一个特定页面取出请求的内容）组件和动态页面编辑（例如，动态地将内容、页眉、页脚、菜单和任何支持交互的脚本嵌入生成的页面中）组件，尽管整个增量 2 的设计可能已经作为一个整体完成了，但是构建可能通过创建内容生成的一个版本而开始，这个版本提供了相关内容，但是没有实现把它适应于特定用户的功能。如果客户接受了这个"简化"版本，那么当辅助功能的编码完成之后，就可以部署这个中间构建了（例如，使得用户可以使用）。

现在，我们考虑这样一种需要在整个 WebE 过程中做一个快速循环的情况。假定增量 2 现在已经被部署了，而且对此增量进行了最终的审查（由所有有兴趣的利益相关者进行），审查人员遇到如下情况：

1. 一个用户正在浏览在某个类别中的产品信息。
2. 此类别不包含任何对此用户可用的产品。

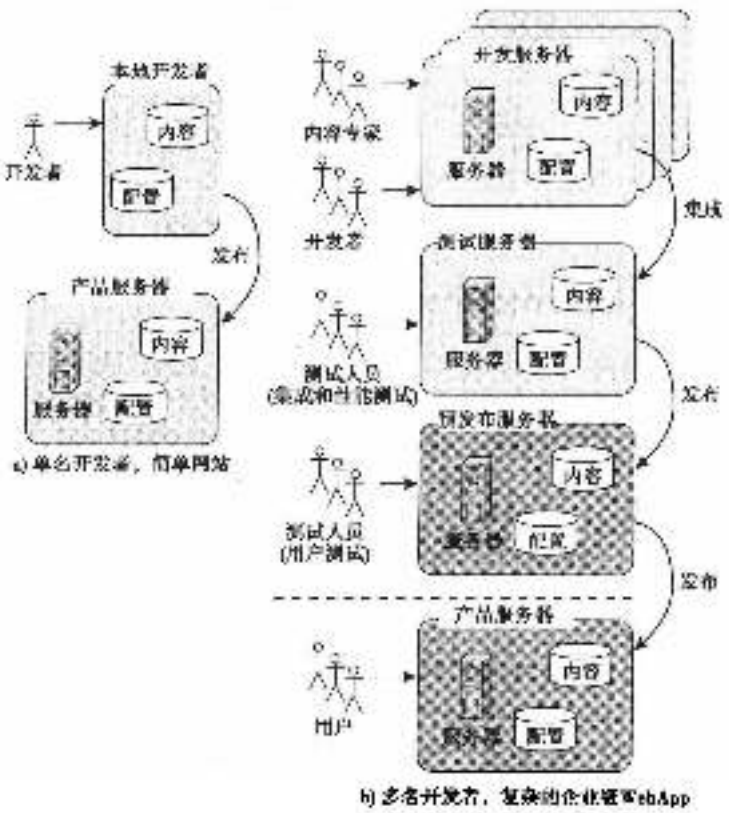
不清楚这是一个所需的功能（尽管这个用户不访问此类别中的任何产品，客户端是否想让用户了解这个类别的存在？）还是只是一个疏忽。对合适的利益相关者做了一个快速调查（例如，重新进行沟通活动），结果表明，用户可以了解这个类别，但这个类别应该包含一些文本说明，告知用户只有 SafeHomeAssured.com 已有的客户才可以访问那些产品。WebE 团队核对这个功能和当前设计（例如，建模活动）是一致的之后，实现并发布这个变化，所有这些功能可以在几个小时内完成。

这个例子说明了部署和构建可以互相作用而使得 WebE 过程更快、更敏捷的方式。

12.1.2 部署环境扮演什么角色

WebE 相关的技术以及这些技术如何影响 WebApp 的开发将会在第 14 章详细讨论。但值得注意的是，WebE 开发环境确实对如何以及何时构建与部署各种 WebApp 增量及其相关组件有着非常重要的影响。

在非常简单的没有关键任务的 WebApp 中，由单个开发者完成本地内容是可以接受的（例如，在本地机器上编辑内容而不是置于一个服务器上），然后把内容发布到产品服务器上立即供用户直接访问，如图 12-2a 所示。但是，对较复杂的 WebApp（发布前要确保 WebApp 的质量是至关重要的）来说，可能需要较复杂的开发环境。考虑如图 12-2b 所示的情况，一个开发环境可能包含如下 4 种不同的服务器类型：



a) 单名开发者，简单网站

b) 多名开发者，复杂的企业级WebApp

图 12-2 WebApp 开发环境

开发服务器（Development Server）。开发人员和内容创作者使用这些服务器来执行所有的创作和单元级测试[1]。这些服务器实际上是一些沙箱，开发者可以在其中创建、执行以及测试他们自己的 WebApp 组件。在有些情况下，开发者可能会有自己的本地环境，这个环境包含所有或部分相关的 WebApp 组件集以及合适的开发工具的副本［例如，编辑器、集成开发环境（IDE）、参考资料、配置文件等］。通常，这种开发配置不会和预发布或产品服务器一样。

测试服务器（Test Server）。一旦开发人员完成了自己组件的单元测试，他们就可以在完全的 WebApp 环境中集成这些组件以供验证。一般来说，测试服务器不安装开发工具，因为这些工具可能会影响 WebApp 的有效测试。集成和性能测试（第 15 章）都将在这个服务器上完成。

预发布服务器（Staging Server）。这个服务器旨在提供一个完全的产品服务器的镜像，从而不必将 WebApp 发布到产品服务器和由此提供充足的用户数量就可以进行 WebApp 综合用户测试。这个服务器应尽可能地和产品环境相似，包括操作系统、支持应用、硬件、驱动、集群等。

产品服务器（Product Server）。当 WebApp 已准备好发布产品给所有用户使用时，就被置于此服务

[1] 在这个上下文中，单元是从测试意味着在作为一个增量的、部分的、个信息或功能组件发布之前，一个发现其中存在的错误的有组织的尝试。

器上。至关重要的是，变更部分必须在预发布服务器上被全面测试后才可以发布到这个服务器上。

某个特定 WebApp 所要采用的实际环境将取决于项目的规模、开发人员和相关人员的数量以及 WebApp 的业务关键性。在很多情况下，上述的 4 种服务器可能被缩减为 3 种（开发、预发布、产品）甚至两种（开发、产品）。在有些情况下，WebE 团队会使用一个单独的网络开发环境；而在其他情况下，不同的 Web 工程师可能会使用不同的开发服务器。产品"服务器"（因此也成为预发布服务器）可能会包含支持负载平衡的一组计算机。在有些情况下，我们可能会有一个开发服务器和一个测试服务器，它们在逻辑上不同，但运行在同一物理服务器上。尽管有这些不同的变体，如下基本原则[6]却保持不变：

- 将开发环境和产品环境分离。不要在你的用户能访问的服务器上直接进行开发。
- 为开发人员提供提高他们生产效率的环境。
- 可能的情况下，在用户可以看到的相同环境中进行测试。

注意，特定的开发和产品环境应当集成在功能架构中。例如，如果一个内容管理系统（CMS）被采用了（第 16 章），那么开发人员如何使用 CMS 生成、集成和发布组件的方式，都应当进行认真考虑。

下面的框注中表述了对 SafeHomeAssured.com 部署环境的讨论。



部署环境

场景：SafeHomeAssured.com WebE 团队的会议室，在初始项目计划阶段，团队在讨论部署环境。

参与者：WebE 团队负责人和 WebE 技术经理

会话：

团队负责人：嗯，我们了解到客户有他们自己的产品服务器，同时有一家被管理的托管服务提供商，他们不仅允许我们使用此服务器安装任何 Web 服务器、数据库、应用服务器以及需要的任何其他应用，还允许我们通过远程访问来进行增量发布。

技术经理：是的——他们说关于这一点的文档大概明天就可以给我们。一拿到文档，我就会检查以确保我们可以解决访问的问题。

团队负责人：好——但是真正的问题是决定我们自己需要什么。

技术经理：这样吧，我会保证我们正常的不被运行良好。换句话说，我将开发一个开发环境的镜像，包括我们标准的开发工具和其他任何与项目相关的工具，并且将它映像到开发人员的每一台电脑上。我还会设置一个预发布服务器来进行测试。我认为我们不需要一个单独的测试服务器，因为这个项目并不太大，而且这个预发布服务器应该能够处理一般的测试负载——同时每个人都可以访问它。

团队负责人：好——这听起来是合理的。我们可以在本周末之前设置好吗？我们需要检查一下，确保所有的连接都没有问题。

---

[6] 关于这类问题的一些有益讨论，参见下面的资源（其中的一些针对具体技术的，但基本原则将在很多的情况中仍然适用）：

- Phillip Greenspun, "Using CVS for Web Development," （使用 CVS 进行 Web 开发）http://philip.greenspun.com/wtr/cvs.html（2007 年 8 月 13 日）.
- InformIT, "Approaches to Content Management," （内容管理方法）www.informit.com/articles/article.aspx?p=23469（2007 年 8 月 13 日）.
- IBM WebSphere Developer Technical Journal, "The Ideal WebSphere Development Environment," （理想的 WebSphere 开发环境）www-128.ibm.com/developerworks/websphere/techjournal/0312_beaton/beaton.html（2007 年 8 月 13 日）.
- Dell Power Solutions, "Keys to Successful Deployment of .NET Applications," （成功部署 .NET 应用的关键）www.dell.com/content/topics/global.aspx/power/en/ps2q03nc_jamison?c=us&cs=555&l=en&s=biz（2007 年 8 月 13 日）.

12.2 构建

构建活动包括一系列选择、编码、创作、集成、重构以及测试动作，通过这些活动产生一个准备交付给最终用户的可运行的 WebApp。选择包括识别可以在当前设计中重用的相关的现有组件（或含有组件的对象）。编码包括修改现有组件或构建新组件，还可能包括直接创建 HTML 和脚本语言源代码或者用来构建构建的组件的一种中间设计表示来自动生成代码。

内容管理（第16章）涉及内容的生成、迁移以及结构化。这包括内容的创建、数据库模式的实现、遗留内容转换成 XML。创作包括对图形化设计（布局）原始内容的集成以及将内容映射到屏幕和页面的过程，还可能包括样式表的实现。集成包括将代码、内容及表示连接到最终要发布的组件上。重构是对已实现的组件进行“调整”以改善组件结构、简化代码更清晰并去掉冗余代码的一项迭代活动。最后，测试包括验证各个组件及对象的正确性。

12.2.1 有通用的构建任务吗

在 Web 的早期，开发人员避免为构建 WebApp 制定特定的任务，毕竟，他们认为 WebApp 是“有机地进行演化的”，而定义它们的任务则是“守旧的”。如果定义了一组任务，那么 WebApp 就不可能是敏捷的了。真的是这样吗？我们不这么认为。

如果你想有组织地建造一栋住宅，会怎么样？如果拒绝定义一个任务序列，会怎么样？有理由定义构建任务——它们帮助构建者有效地管理项目并且避免错误。例如，建造一栋住宅时，必须打地基，必须搭建住宅框架，必须封顶，必须安装窗户，必须布置好基本电线，还必须安装基本的水管等。但是，如果为了敏捷，构建者决定在安装窗户之前粉刷墙壁会怎么样呢？初看起来好像是合理的，但你考虑一下，如果下雨呢？有理由需要一个任务序列。值得注意的是，许多构建任务是可以并行的，其他任务可能会被迭代执行。还有一些可能在发生错误时必须返工。

WebApp 的构建任务集比模拟建造房子的任务集要小，但是 WebApp 构建的敏捷性毋一样复杂的，而且可能更难理解。Web 工程师不是从一组静态设计图来构建 WebApp，而是从一个随着工作的进行必须被简化且适应的设计来构建应用。他们必须在每一步测试他们已经完成的工作，然后把他们的工作和其他的 WebApp 组件集成起来。[Fit01] 是软件构建的一个好的任务参考，并且已经在 [Her02] 和 [Hui01] 中被扩展到了 Web 领域。具体的任务集将依赖于 WebApp 的性质。例如，任务“建立站点地图”的需求将依赖于信息设计是否包含一个站点地图。然而，下面的框注列出了有关 WebApp 的一个特定构建活动的一组任务的样例。

任务集



构建任务集样例

基础任务：这些任务包括为 WebApp 的有效实现建立基础。这将保证一致性和可维护性。

实现品牌标识组件。

创造评估和设计产品。

- 创建白色网站[1]、骨架网站[2]和线框图。
- 原型化人机界面。

架构任务：这些任务针对于 WebApp 总体结构的构建。

完成网站架构和基础结构。

- 检查架构设计。
- 对实现 WebApp 架构的组件进行编码和测试。
- 获取可重用的架构模式。
- 对基础结构进行测试来确保接口的完整性。

选择一个合适的组件框架。

---

① 一个白色网站是一个完成了内容的，对用户来说，从表面上看是完整的，但是缺少一些或大多数的基础功能和内容。

② 一个骨架网站是一个含有所有主要的信息架构和功能条件（但缺少关键界面组件的实现）的网站。

- 评估可能的组件框架。
- 筛选候选组件框架构列表。

低层任务：这些任务强调组成 WebApp 的功能和信息组件的构建。

构建功能组件。

- 检查组件级设计。
- 为组件创建一组单元测试。
- 对组件数据结构和接口进行编码。
- 对内部算法和相关处理函数进行编码。
- 在编写代码的同时对它进行审查。
  - 检查正确性。
  - 确保保持了编码标准。
  - 确保代码是自文档化的（self-documenting）。

管理内容。

- 实现内容管理。
- 创作内容。
- 将内容和用户界面进行集成。

创建导航支持。

- 创建导航地图。
- 创建导航层级支持机制。

执行 WebApp 组件的单元测试。

集成任务：这些任务涉及 WebApp 的总体构建。

集成组件。

实现内容管理策略。

实现个性化策略。

12.2.2 什么是重构？如何应用它

一个组件一旦构建出来，它必须被加以评估以确定它与设计的一致性以及在 WebApp 架构中的有效性。评估后对组件进行精化，优化其结构使其更简洁，并去掉或修改冗余和无效的代码。这就叫做重构（refactoring）。Fowler［Fow99］以如下方式描述重构：

> *源码重构是以不改变代码的外部行为而改进其内部结构的方式修改软件系统的过程。这是一种净化代码以减少引入错误的可能性的科学的方法。从本质上说，重构就是在代码编写完成之后改进代码设计。*

重构在许多敏捷方法学包括极限编程（XP）[⊖]中是一个关键要素。这主要是因为敏捷方法反对“预测”。也就是说，大多数敏捷方法建议 WebE 团队不应该试图把一个或多个团队成员“了解”的、将在后面需要的（在另一个增量中）设计特点，构建在 WebApp 增量中。如果没有认真做设计计划，这种限制可能导致当前增量上进行的构建工作可能会导致为前一个增量建立的设计或者架构产生重大改变。

当重构 WebApp 时，检查现有设计中的冗余性、没有使用的设计元素、低效或不必要的算法、拙劣的或不恰当的内容和数据结构以及任何其他设计不足，修改这些不足以获得更好的设计。例如，第一次设计迭代可能产生一个组件，表现出很低的内聚度（例如，执行3个功能但是相互之间仅有有限的联系）。设计者可以决定把此组件重构成3个独立的组件，每一个都表现出较高的内聚性。这样的处理结果将实现一个更容易集成、测试和维护的 WebApp[⊜]。

对 WebApp 来说，重构既包括优化功能性组件，还包括用户交互。例如，对 SafeHomeAssured. com 来说，当我们完成增量1、2和3时，我们会有一个界面，可以查看产品信息和订购，

⊖ 参见 ExtremeProgramming. org 网站 www. extremeprogramming. org/rules/refactor. html，获得关于 XP 中重构角色的讨论。

⊜ 为了获得更多关于重构的信息，可以从阅读 Martin Fowler［Fow99］关于这个领域的经典书本开始。你还可以参看 Cunningham & Cunningham, Inc. , site, http: //c2. com/cgi/wiki? WhatIsRefactoring 上的“What Is Refactoring”（什么是重构）页面和 Refactoring（重构）主页，www. refactoring. com/。要获得一些简单的例子，参见 www. perl. com/pub/a/2003/10/09/refactoring. html。

但是没有任何监控布局或服务。当我们增加增量4时，就加上相关的菜单选项，但这可能还需要重构来整理整体菜单结构。

12.3 构建的原则和概念

一组基本的原则和概念将使WebE团队开发出可维护的、可测试的WebApp。指导构建活动的这些原则和概念与编程风格、编程语言以及编程方法紧密联系。但是，还有很多与Web相关的基本原则非常有用，应该在构建活动中时刻谨记。

准备原则。在创建一个简单的Web页面或写一行代码之前，你需要：

- 理解你正要解决的问题。
- 理解基本的WebApp设计原则和概念。
- 挑选一种适合要构建的组件需要的语言和组件运行环境。
- 选择一个提供可以简化工作的工具的环境。
- 构建在组件完成时需要用到的一组单元测试。

选择原则。当你选择已有的可重用组件和对象时，需要：

- 考虑技术环境的约束。
- 使组件与信息和功能的环境相匹配。
- 考虑开发人员和可能的维护人员的技能和知识。
- 考虑知识产权问题、组件的所有权及其可移植性。

编码原则。当你开始写代码时，需要：

- 编写自文档化的代码。
- 遵守结构化编程［Boh00］习惯来约束算法。
- 选择符合设计需求的数据结构。
- 理解功能架构，构建符合它的接口。
- 保持条件逻辑尽可能简单，并确保是可测试的。
- 采用可读性好的编码风格（例如，选择有意义的标识符命名和遵循其他的局部编码标准）。可以在Literate Programmer网站www. literateprogramming. com/fpstyle. html上找到更多的编码标准链接。

关于编程（编码）以及指导编程的原则和概念的书比软件工程工程其他任何主题的书都多。关于这个主题的书有早期的关于编程风格［Ker78］的著作、实用软件构建［McC93］、编程珠玑［Ben99］、编程的艺术［Knu98］、实用编程问题［Hun99］以及许多其他书籍。尽管所有这些都是为传统软件所写，但是它们提供的指南同样适用于WebApp。

内容管理。要管理内容，需要：

- 选择符合设计需要的数据结构。
- 理解信息架构并创建与之一致的内容和导航结构。
- 确保格式和数据结构的一致性。
- 避免依赖专有数据格式。
- 把内容看成可发行资料而不是软件。

创作原则。当你创建Web页面（或页面模板）时，需要：

- 不断地考虑可用性问题。
- 记住解决可访问性问题。
- 理解你的用户如何交互，而不是你想要他们如何交互。

- 向竞争者学习。

和编码一样，关于内容管理和 Web 创建风格的文章已经有很多。W3C 网站（www. w3. org/TR/webarch/#app-principles）是很好的入门指导。另外一个很有用的信息源是 Jakob Nielsen 的可用性网站（www. useit. com），还有一篇非常优秀的关于管理内容的文章是［McG01］。

集成原则。当你集成组件和对象时，你需要：

- 保留备份——在某种版本控制系统中更适用。你需要能够把 WebApp 恢复到早期的版本。
- 找出不匹配或不一致的组件接口。
- 及时识别需要重构的组件。

重构原则。当你重构 WebApp 时，你必须：

- 理解一般的重构（参见重构主页 www. refactoring. com/catalog/index. html 中的重构例子）。
- 当有重构机会时，经常是一点一点进行变化的（但不要进行不必要的变化——如果 Cunningham&Cunningham，Inc. 网站 http：//c2. com/cgi/wiki? lfltlsWorkingDontChange 上关于不变的部分还在，请参阅）。
- 确保实现和设计以明显的方式进行沟通。

测试原则。在你完成第一个组件之后，必须：

- 进行严格的走查。
- 执行单元测试，更正发现的错误。
- 选择那些最可能发现错误而不是隐藏错误的测试。

第 15 章详细介绍了 WebApp 的测试问题。

12.4 部署

部署活动包括三个动作：打包、发布和评估。因为 WebApp 的开发实质是增量的，所以部署不只发生一次，而是会在 WebApp 迈向完成的过程中发生很多次。组件的发布可以以一种非常细粒度的方式完成：在测试完单个组件后，从预发布服务器发布新组件到产品服务器。例如，发布对内容或功能的更正、演化中列表中重要新条目，或对已经部署了的 WebApp 的用户“现在”就要求的新内容或功能，可能是合适的。然而，在很多情况下，这种方法可能并不合适。

如果有很多不断发生的小变化，用户可能会变得困惑，而且集成错误的可能性和意料之外的副作用可能会增加。在大多数情况下，最好把一组变化打包发布（除非一个变化更正一个破坏性的错误）。

每一个打包发布周期都为最终用户提供了一个具有可用功能和特征的可用的 WebApp 增量。每个评估周期都为 WebApp 团队提供重要指导，并为下一个增量做出内容、功能、特征和方法的修改。

12.4.1 是否存在通用的部署任务

WebApp 很多的部署任务和传统软件以及信息系统的部署任务相似，但是其他一些则是 Web 领域所特有的。例如，传统软件系统的打包可能涉及创建安装功能，这些安装功能分布在对于那些将要“安装”应用的人来说合适的媒介上。对 WebApp 来说，打包的焦点是确定发布到产品服务器的组件集合以及何时及如何发布的准则。[⊖]

因为新的 WebApp 的功能或内容必须立即为所有相关用户使用，所以发布动作还必须考虑如下问题：

⊖ 例如，产品服务器是否会因为发布而需要临时离线？在一个使用平缓时期完成发布是否更好？

- 将如何沟通或宣告变化？
- 如何指导用户使用复杂内容或功能？
- 是否应该为错误报告或其他与新内容或功能相关的问题提供特殊的反馈机制？
- 变化了的内容是否应当纳入搜索引擎的索引中？

当部署一个大量使用的 WebApp 时，新内容和功能的影响是瞬时发生的。建立一个质量保证方法来帮助避免“坏”增量的瞬时的负面影响是非常重要的。

一组通用的部署任务集展示在下面的框注中。应当注意，和“决定发布集合”、“决定发布过程”、“支持计划”和“建立评估过程”相关的很多子任务都是作为 WebE 过程早期实行的计划活动的一部分而启动的。

任务集



部署任务集样例

部署计划：这些任务关注对部署做计划。

决定发布集合。

- 识别要发布的新组件。
- 确认组件的配置是合适的。
- 和客户确认发布集合。

决定发布过程。

- 计划发布日程表和时机。
- 计划变化的沟通。
- 建立故障应急计划。

支持计划。

- 计划内部支持机制。
- 建立支持渠道。
- 进行内部支持培训。
- 建立问题日志过程。

部署执行：这些任务包括实际的部署。

建立评估过程。

- 计划评估过程。
- 计划用户跟踪。
- 建立用户反馈机制。
- 创建问题日志。
- 创建问题记录机制。
- 分析评估数据。

发布组件。

- 确认成功的发布。

执行变化沟通任务。

进行持续不断的支持。

一个新的 WebApp 增量的发布可能伴随（很可能）业务过程的重组。这种重组元素常常是问题最大的，而且在 WebApp 开发和部署中是被理解和管理得最少的。关于业务过程重组和工作流修改的详细讨论超出了本书的范围。如果你还有兴趣，花些时间来参阅两本关于这个主题的很棒的书——[Har03] 和 [Sha01]。

12.4.2 指导 WebE 团队的部署原则

一个 WebApp 增量的部署代表任何 Web 工程项目的一个重要的里程碑。一个 WebApp 的顺利部署为最终用户提供了内容和功能以及立竿见影的收益。但是如果部署的计划拙劣、错误很多，而且执行效率很低，那么最终用户会很恼火和沮丧。为了确保顺利部署，团队在准备交付一个增量的时候，应当遵循一些关键原则：

原则 1：必须管理客户对 WebApp 增量的期望。客户通常并不希望看到团队承诺交付而又没有实现。这会导致反馈变得无用并且会挫伤团队的士气。Naomi Karten [Kar94] 在她关于管理客户期望的书中提道：“管理客户期望首先应该从考虑你应该与客户沟通什么和如何沟通。”她建议开发者必须认真地处理与客户有冲突的信息（例如，你不可能在给定的时间框架内完成的工作做出承诺；或者在某次软件增量交付时交付了多于当初承诺要交付的东西，这将使得在下

一个增量所要交付的工作比你承诺的少）。敏捷开发方法和使用预发布服务器在这里可以起到帮助。敏捷方法将便于内容和功能的增量交付，为客户提供 WebApp 的更早的反馈，因此支持对客户期望管理。使用预发布服务器使得增量在公开发布之前展示给客户，由此就更早地被精化。

原则 2：一个完整的交付包应该经过安装和测试。再一次，使用预发布服务器对于为一个特定的 WebApp 增量展示内容和功能来说是非常宝贵的。而且增量在所有可能的操作环境中（不同的硬件、不同的浏览器配置和网络带宽，以及不同的安全性设置）经过实际用户完整的 β 测试。

原则 3：必须在 WebApp 交付之前建立支持制度。最终用户期望在出现问题时得到及时的响应和精确的信息。如果支持跟不上或者根本就没有，那么客户将立即变得十分不满。应该计划支持，准备支持过程和响应，并且建立合适的记录保持机制，这样 WebE 团队就能够对支持请求的种类进行分类评估。

作为一个例子，如果 SafeHomeAssured. com WebApp 含有一个基于表单的对更多信息的查询，那么在发布之前，团队应当确保有适当的业务过程使 SafeHomeAssured. com 员工对提交的查询做出响应。

原则 4：有缺陷的 WebApp 应当首先改正再交付。迫于时间的压力，一些 WebApp 开发者会交付一些低质量的增量，还在这些增量中提醒客户“构建中”并警告客户：“这些缺陷将在下次发布时解决。”这样做是错误的。在软件业有这样一种说法：“客户在几天之后就会忘记你所交付的高质量产品，但是他们将永远不会忘记那些低质量的产品给他们带来的问题。WebApp 会时刻提醒他们问题的存在。”

已交付的 WebApp 为最终用户提供一些利益，但是同时它也为 WebE 团队提供一些有用的反馈。当一个增量投入使用后，应该鼓励最终用户对 WebApp 的特征、功能、易用性、可靠性以及其他合适的特性做出评价。开发团队应该收集和记录用户反馈，以便：1）对交付的增量及时做出修改（如果需要）；2）确定将合并到下一个增量中的变更；3）针对这些变更相应地修改设计；4）修订下一个增量的计划（包括交付时间表）以反映变更。获取 WebApp 的反馈相对简单，因为开发者可以实现复杂的记录日志和跟踪机制来分析用户的行为。

12.4.3 如何使用版本控制和 CMS

我们将在第 16 章详细讨论版本控制和 CMS。然而，这两者在构建和部署活动中都扮演着重要角色。现在有必要进行简单概述。

在本书的第 1 章中，我们提到变更常常很快并且有时影响深远，变更是所有 Web 工程项目的一个关键驱动力。接着变更管理是整体 WebApp 成功构建和部署以及每个增量发布过程中的关键。版本控制和 CMS 都是变更管理工具，提供如下通用的优点［Dar00］：

- 确保所发布的内容正确，而且和其他内容一致。
- 控制和跟踪内容的变更，包括对谁可以做出变更进行强制的实现机制。
- 检验实现了一个功能的正确版本及其和相关功能的版本的正确对应。
- 使 WebE 团队在系统失败或崩溃时，能快速重建 WebApp。
- 允许团队在最新版本遇到严重的、未预见的错误时，回滚到前一个版本。

随着 WebApp 的规模和复杂性的增加，构建和部署的每一步的全面版本控制和健壮 CMS 的要求也在增加。本质上，版本控制和 CMS 在团队试图做出很多快速变更时，扮演着“统治者”的角色。通过把工作稍微放慢一点儿，就可改进 WebApp 的质量，并有助于 WebE 团队避免高代价的错误。



内容管理

场景：SafeHomeAssuredLcom WebE 团队在会议室内召开一个紧急会议，在增量 5（监控服务）部署后的 15 分钟。

参与者：WebE 团队的负责人和其他成员。

会话：

团队负责人：你告诉我的是我们有一个问题。

团队成员 1：是的，一个大问题。我们在测试和预发布服务器上对新的增量进行了全面的测试，它运作良好，但是我们现在在产品服务器上发现了一个真正的问题，而且已经发布出去了。

团队负责人：那就是你先前在电话中说的。请解释这个问题！

团队成员 2：嗯，我们还不是很确定为什么，但是看起来在允许已有客户订购新的监控服务的新过程之间好像有某种奇怪的相互作用。结果，有时当某个人下新订单来监控一个新传感器时，会停止监控此人所有其他的传感器。我们现在还不知道原因，不过整个团队正在忙着调试，应当很快就能解决。

团队负责人：听着，这相当严重。因为我们一直在传这个新版本中有这个新功能，有很多客户想试试。你可以想象：我们本该监控他们的住宅——由于这个漏洞无法监控而有人却失窃了所导致发生的诉讼。我认为除了停止使用这个增量直到我们知道问题已经解决了之外，别无选择。

团队成员 1：好吧。我们可以停止新的增量，并快速（在接下来的 5 分钟内）回到之前的版本简单地通过触发内容管理系统中完整系统的前一版本的重新上载即可。你确认要这样做吗？客户不会对新功能的突然消失感到疑惑吗？

团队负责人：是的，它不是好的 PR，但是我已经跟管理部门说过了，他们也同意了：这样做比承担这个新增量可能造成问题的风险小。我希望你安排让 WebApp 回到先前的版本，现在！

团队成员 1：好的——5 分钟就会完成。

12.5 构建和组件的使用

组件级设计常常在总体功能和信息架构建立之后，作为构建活动的一部分进行。当核心 WebApp 功能的设计完成后，目标就成了把设计模型转化为一个可运行的 WebApp 增量。WebApp 设计模型的抽象层次还是比较高，而可运行的 WebApp 的抽象层次却比较低。这种转化具有挑战性，因为可能会在 WebE 过程后期阶段引入难以发现和改正的微小错误。Edsger Dijkstra，有助于我们理解传统软件设计技术的主要贡献者，在其著名的讲座[Dij72]中写到：

> 软件似乎不同于很多其他产品。对那些产品而言，一个规则是：更高的质量意味着更高的价格。那些想要真正可靠软件的人们将会发现，他们必须找到某种方法来避免开始时的大部分错误，结果，程序设计过程将变得更加容易……高效率的程序员不应该把他们的时间浪费在调试上——他们在开始时就不应该引入错误。

尽管这段话是在很多年以前说的，但是现在依然适用。当设计模型被转化为源代码、Web 页面、CMS 内容或链接结构的时候，我们仍然遵循一系列设计原则，以保证不仅能够完成转化任务，而且还不“在开始时就引入错误。”

12.5.1 什么是通用组件

通俗地说，一个组件是基于计算机的系统中的一个模块化的构建块。更正式地说，OMG 统一建模语言（UML）规范［OMG04］中将组件定义为“……系统的一个模块化的、可部署的和可替换的部件，该部件封装了实现并暴露了一组接口。”当这个概念应用于 WebApp 时，一个组件可能是指一个功能构建块，也可能指一个内容构建块（例如，一组 Web 页面和其他诸如样式表和显示模板这样的相关内容元素，或获取了一个通用领域模型的数据库模式）。

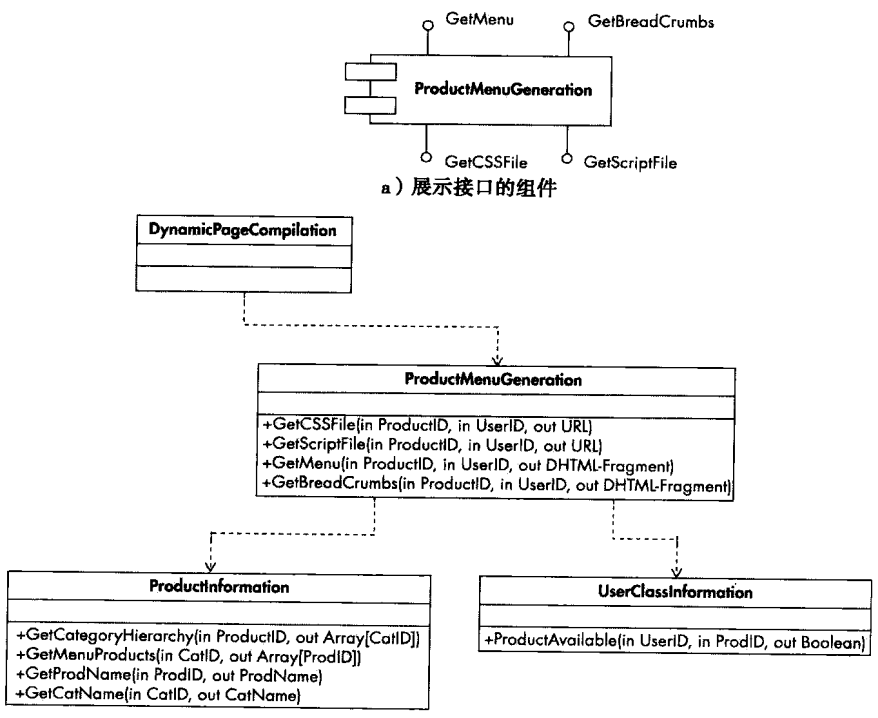
正如在第 10 章中的讨论，功能架构可以根据构建 WebApp 组件、这些组件相互连接的方式来进行描述。因为组件驻留在 WebApp 架构的内部，所以它们常常和其他组件以及存在于 WebApp 边界之外的其他实体（例如，其他系统、设备和人员）进行通信和合作。

术语组件的真正含义，不同开发者可能见解有所不同。在接下来的几节中，我们将了解两个关于“组件是什么以及在设计建模中如何使用组件”的重要观点。

12.5.2 如何定义面向对象组件

在面向对象软件工程（例如，［Let06］、［Bru04］）环境中，一个组件包括一组相互协作的类。组件内的每个类都被充分细化，包括所有属性和与其实现相关的操作。作为设计细化的一部分，所有与其他设计类相互通信和协作的接口也必须予以定义。

为了说明细化过程，考虑图 11-5 中展示的功能架构中的“产品菜单生成”组件。这个组件负责生成可用于 Web 页面的动态创建以及为这些页面提供导航元素的 DHTML 代码（例如，HTML 片段、需要的 JavaScript 模块和 CSS 样式表）。图 12-3 展示了这个组件的一个面对对象设计。



a）展示接口的组件

b）组件设计

图 12-3 SafeHomeAssured. com 中 ProductMenuGeneration 组件的基于 OO 的设计

这个组件提供确定 CSS 样式表的 URL 和 JavaScript 文件的方法，这些 URL 和 JavaScript 文件需要包含在总体的组成页面中，还包含支持导航元素的相关的样式和脚本信息。这些方法被传递给当前用户和产品 ID，这样就可以进行任何合适的定制。

这个组件还提供了两个方法，为产品目录和导航层次返回 DHTML 片段（被嵌入到产生的产品页面中）。再一次，这些功能被传递给相关的产品 ID 以及用户 ID，这样菜单和其他基于页面的表示就能够被定制，不会提供到那些对当前用户不可用产品的链接。

DynamicPageCompilation（动态页面编辑）组件会使用 ProductMenuGeneration（产品菜单生成）组件以获得必要的导航组件，然后这些组件将被嵌入到总体的 DHTML 页面，传递给客户端浏览器。

这个设计任务产生正在开发的组件的一个详细设计（细化）。它被应用于作为架构设计的一部分而定义的每个组件。组件一旦被细化，就指定了每个属性、操作和接口；确定适合每个属性的数据结构，设计算法细节以实现每个操作相关的处理逻辑。最后，设计接口实现所需要的机制。对面向对象软件来说，这可能包含为实现系统内对象间通信所需要的所有消息的描述。

传统的面向对象类图模型并不能很好地处理 Web 内容对象（页面、样式表、XML 内容等），因为内容对象并不封装操作（除了有嵌入的 JavaScript 脚本的地方）。然而，并没有理由不能以和功能类相同的方式进行建模［Con02］。

12.5.3 如何定义传统组件

在传统软件工程的环境中，一个组件就是程序的一个功能要素，程序由处理逻辑、实现处理逻辑所需的内部数据结构以及能够使得组件可以被调用和实现数据传递的接口构成。一个传统组件也称为一个模块，作为软件架构的一部分，承担如下三种重要角色之一：1）控制组件，协调问题域中所有其他组件的调用；2）问题域组件，完成部分或全部的客户要求的功能；3）基础设施组件，负责完成问题域中所需要的相关的处理功能。在 WebApp 中，我们可以扩展角色以包含：4）表示可用信息的内容组件——既可以作为静态 Web 页面和其他的单独文件，又可以是诸如 XML 或数据库这样的某种结构化形式的内容。

12.5.4 好组件有哪些特征

好的组件的四个基本设计原则已经在面对对象软件工程中被广泛采用，它们也适用于 WebApp 的组件级设计。应用这些原则的基本动机是创建更顺应变更以及当变更发生时降低副作用的传播的 WebApp 设计。在开发每个 WebApp 组件时，开发者以如下这些原则［Mar00］为指导：

原则 1：设计者应该采用一种无需对组件自身内部（内容、代码或逻辑）做修改就可以进行扩展（在它针对的信息或功能域内）的方式来详述一个 WebApp 组件。为了达到这个目的，设计者要在那些实际需要的组件特征和可能的扩展之间创建抽象缓冲区。

这似乎和不要试图预测未来的改善的敏捷原则相悖，但我们认为这是一个恰当的折中。我们不是认为可能在后面迭代中相关的功能或信息应该基于一种直觉来设计，而是应该确保设计是可扩展的。

例如，假设 SafeHomeAssured. com 基于 Web 的远程监控功能使用一个必须检测各类安全性传感器的状态的 Detector 类。随着时间的推移，安全传感器的数量和类型将会不断增长。如果内部处理逻辑实现为一个 if-then-else 的结构顺序，其中每个这样的结构都针对一个不同的传感器类型，那么对于新添加的一个传感器类型，就需要增加额外的内部处理逻辑（仍然是另一个 if-then-else 结构），而这违反了第一条原则。

图 12-4 中表明了一种遵循第一个原则的实现 Detector 类的方法。对于各种不同的传感器，接口都向 Detector 组件呈现了一致视图。如果增加一种新的传感器类型，那么 Detector 类（组件）不需要做出任何变化。这个设计遵守了第一个原则。类似地，当我们设计导航结构或页面模板时，它们应该可以处理新内容而无需变化。考虑 SafeHomeAssured. com 的产品目录浏览功能。当添加新产品时，产品动态页面编辑中使用的模板应当保持不变。

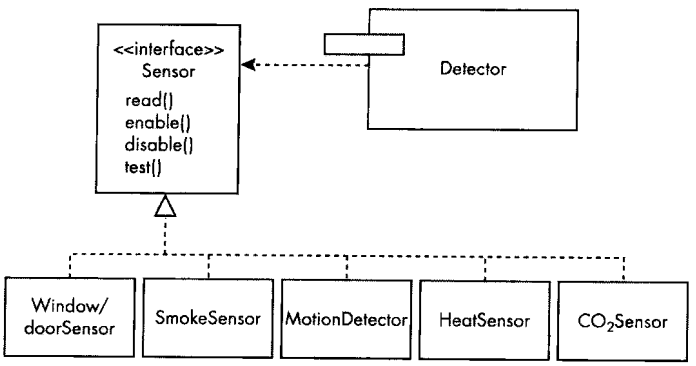


图 12-4 可扩展性设计：遵循第一个组件设计原则

原则 2：将子类传递给组件来取代基类时，使用基类的组件应该仍能够继续正确完成其功能。这就要求派生自基类[⊖]的任何子类必须遵守基类和使用该基类的组件之间的隐含约定。在这里的讨论中，“约定”既是一个前置条件——组件使用基类之前必须为真；又是一个后置条件——组件使用基类之后应该为真。当设计者创建导出子类时，这些子类必须同时符合前置条件和后置条件。在 WebApp 环境中，这一原则说明，如果内容 *X* 包含在一个特定的 Web 页面内对一个用户有意义，那么当它包含在相同导航层次内的另一个页面中时也应该有意义。

对 SafeHomeAssured. com 来说，如果产品内容生成组件能够为指定的一件产品生成一个相关页面片段，那么它也应该能够为任何包含额外细节的特殊产品做同样的事情。

原则 3：一个组件依赖于其他实体组件（不是依赖于诸如接口的抽象）越多，就越难以扩展。例如，如果我们想要创建一组相似观感的 Web 页面，那么总体美学（观感）应当捕获为一个单独的表示组件（例如，一个样式表），而不是嵌入到单独的页面中。这将使得观感的修改容易得多。

原则 4：设计者应该为每一个主要的客户类型都创建一个特有接口。只有那些与特定客户类型相关的操作才应该出现在该客户的接口说明中。如果多个客户要求相同的操作，那么这些操作应该在每一个特定的接口中都加以说明。对 WebApp 来说，这就意味着内容、功能或链接只有在用户被授权使用时才对其可用。如果用户看到了某个 WebApp 组件却发现无法访问，那么将（通常）只会使用户感到困惑。

12.6 组件级设计指导准则[⊜]

除了我们刚刚讨论的原则之外，在 WebApp 组件级设计的过程中，还可以使用一系列实用的设计指导准则。这些指导准则可以应用到组件、组件的接口以及对于最终设计有着重要影响的

⊖ 一个基类是一个通用类，其他派生类可以根据它进行创建。例如，如果 Car 是一个基类，那么 Coupe 和 Sedan 就是派生类。一般来说，派生类继承了基类的属性。

⊜ 这一节提到了 UML 的概念和专用术语。如果你对 UML 不熟悉，我们建议你回顾一个 UML 指南。可以在 www. uml. org/#Links-Tutorials 上的 Object Management Group 网站上找到一系列好的 UML 指南。

依赖和继承特征等方面。Ambler [Amb02] 给出了如下的指导准则：

组件。对那些已经被确定为架构模型的一部分的组件，应该建立命名约定，并对其做进一步的精化，使其成为组件级模型的一部分。架构组件的名字应该来自于问题域，并且应该能够被所有查看架构模型的利益相关者理解。这些名字常常出现在 WebApp 信息架构的设计过程中或底层领域类模型中。例如，对每个阅读 SafeHomeAssured.com 的类名 FloorPlan 的人来说，不论他们的技术背景如何，它都是可以理解的。

另一方面，基础组件或精化后的组件级类应该以能够反映其实现意义的名称来命名。例如，如果在 SafeHomeAssured.com 的产品搜索实现过程中，搜索结果存储在一个数组中，那么 ProductResultArray 可能是一个合适的名称。

还应该在体系设计层次使用构造型（stereotype）[○] 来帮助确定组件的性质。例如，<<Infrastructure>>（基础设施）可能用于确定一个基础组件，<<database>>（数据库）可能用于确定一个服务于一个或更多设计类或者整个系统的数据库，<<table>>（表）可以用于确定一个数据库内的一张表。图 12-5 说明取自 WAE 的一系列好的构造型，用于 WebApp 内的信息组件。

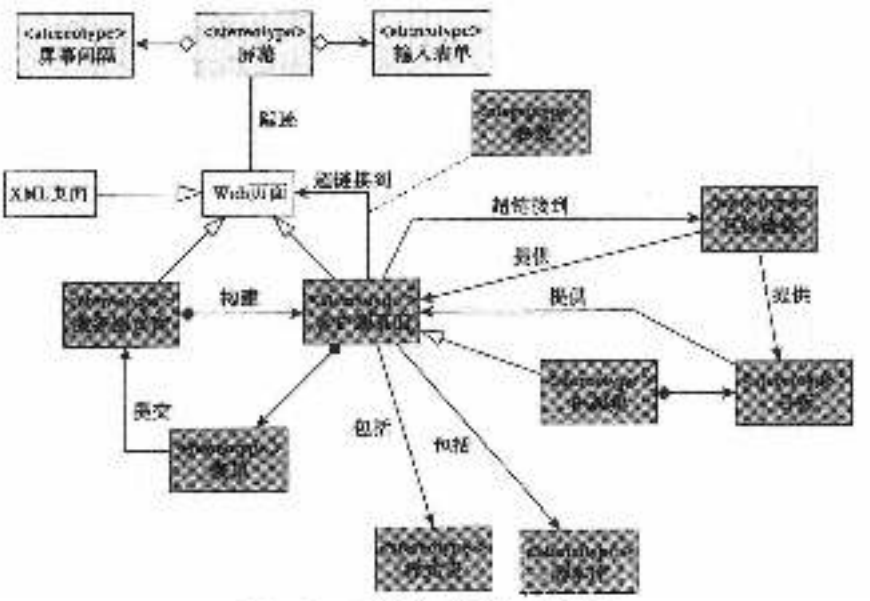


图 12-5 WAE 类构造型的示例子集

来源：源自 [Gro02]。

接口。接口提供关于通信和协作的重要信息（也可以帮助你实现第 12.5.4 节中讨论的第一个组件设计原则）。然而，接口表示的随意性会使组件图趋于复杂化。Ambler [Amb02] 建议：1）当组件图变得复杂时，在较正式的 UML 矩和曲线箭头记号方法中用接口的“棒棒糖”表示记号；2）为了保持一致性，接口应该放在组件框的左手边；3）即使其他接口也适用，也只表示那些与组件相关的接口。这些建议旨在简化 UML 组件图，使其易于查看。

依赖与继承。为了提高可读性，依赖关系是从左到右的，继承关系是自下（派生类）而上（基类）的。另外，组件之间的相互依赖关系应该通过接口来表示，而不是采用“组件到组件”的方法来表示。类似地，和信息架构一样，导航流应该表示为相关的图中从左到右或从上到下的表示。

○ 一个构造型是 UML 内的一种“可扩展性机制”[Arn02]，它使得 Web 工程师能够定义一个特殊的建模元素，其语义是工程师自己定义的。在 UML 中，构造型表示为双尖括号（比如，<<stereotype>>）。

12.7 组件的设计步骤

在本章前面，我们提到组件级设计本质上是精化的（例如，这是采取一个宽泛的抽象架构设计，然后对它进行细化来获得设计细节）。作为一个设计者，你必须将分析模型和架构模型中的信息转化为一种设计表示，为指导构建任务提供充分的细节。当应用到 WebApp 中时，下面的步骤表示出一系列用于组件级设计的典型任务：

步骤 1：标识出所有与问题域相对应的信息和功能设计类。使用信息和功能的分析模型和架构模型（第 7、10 和 11 章）来识别用于构建每个 WebApp 页面的具体内容和详细的功能组件。对信息组件来说，这是第 10 章中讨论的一种详细切片设计形式。考虑 SafeHomeAssured. com，一个对应于特定的安全性传感器的 Camera 类可能就是一个功能组件的例子。

步骤 2：确定所有基础设施域对应的信息交互和功能设计类。这些类没有在分析模型中进行描述，并且在功能架构模型中经常被忽略，但是此时必须对它们进行描述。这种类型的类和组件包括诸如列表框和菜单翻滚（常常作为可用的可重用组件）等客户端交互组件、操作系统组件、内容和数据管理组件等。

在 SafeHomeAssured. com 中，信息交互类的一个例子是每个页面顶部的主菜单翻滚的设计——为在客户端的实现进行设计（因此很可能使用 JavaScript）。功能设计类的一个例子是 SearchResultGeneration 类。

步骤 3：精化所有不能作为可重用组件的设计类。细化要求详细描述实现类需要的所有接口、属性和操作。在实现这个任务时，必须考虑采用设计试探法（例如，组件内聚和耦合）。图 12-6 中展示了一个例子，说明 SearchResultGeneration 类的细节。

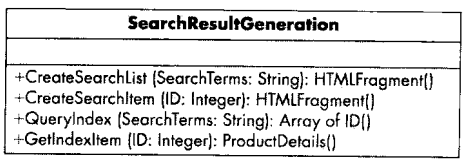


图 12-6 SafeHomeAssured. com 的 SearchResultGeneration 类

步骤 4a：在类或组件协作时说明消息的细节。分析模型中利用协作图来展示分析类之间如何进行协作。在组件级设计过程中，某些情况下通过对系统内对象间传递消息的结构进行说明来表现协作的细节是必要的。尽管这是一个可选的设计活动，但是它可以作为接口规范说明的前提，这些接口显示了系统中组件是如何通信和协作的。图 12-7 给出了一个协作图，说明了发生在生成一组搜索结果过程中的访问序列。

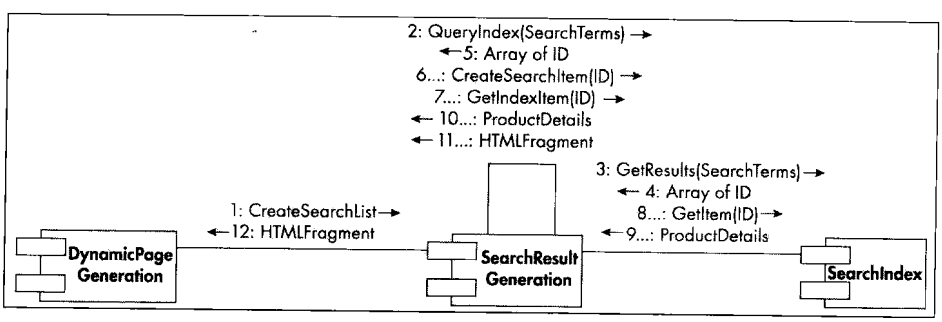


图 12-7 SafeHomeAssured. com 的 SearchResultGeneration 协作图

步骤4b：为每一个组件确定适当的接口。在组件级设计中，一个UML接口是一组外部可见（即公共的）的操作。更正式地讲，一个接口相当于一个抽象类，该抽象类提供了设计类之间的一个可控连接。实际上，暴露的内容和可用的操作是为设计类定义的。抽象类内的每个操作（接口）都应当是内聚的；也就是说，它应该展示那些关注于一个有限功能、子功能或者一个信息视图的处理。

步骤4c：精化属性并且定义相应的数据类型和实现所需的数据结构。一般来说，描述属性的数据结构和数据类型都需要在实现时所采用的程序设计语言中进行定义。在前面使用的SafeHomeAssured. com的例子中，包括确定用于管理查询搜索索引时返回的ID数组的数据结构。

步骤4d：详细描述每个功能组件中的处理流。这可能需要结合使用一种程序设计语言（例如，Java）和一种自然语言（例如，英语）来创建伪代码或使用UML活动图来完成。在SafeHomeAssured. com的CreateSearchList()方法中，WebApp组件的设计者可能定义如下的处理流：

```
call QueryIndex (SearchTerms) result = ArrayIDs
For each ID in ArrayIDs
Call CreateSearchItem(ID) result = HTMLFragment
    Append HTMLFragment to QueryFragment
Return QueryFragment
```

步骤5：描述持久数据（Web页面、数据库等）和动态构建数据。数据库和文件通常会凌驾于一个单独的组件设计描述之上。在大多数情况下，这些持久数据存储起初被指定为架构设计的一部分。在SafeHomeAssured. com中，设计者将把用户订单定义为持久的。另外，设计者很可能同时想要记录诸如他们的搜索和查询之类的事情，这样用户就能够在后面使他们的体验更个性化。

步骤6：开发并精化类或组件的行为表示。UML状态图被用作分析模型的一部分，以表示系统的外部可见的行为和单独分析类的局部行为。在组件级设计过程中，有时对设计类进行行为建模是必要的。

步骤7：精化部署图以提供额外的实现细节。UML部署图作为架构设计的一部分，采用描述符形式来表示。在这种表示形式中，主要的系统功能（常常描绘为子系统）都表示在容纳这些功能的计算环境中。在SafeHomeAssured. com中，产品目录位于单独的数据库服务器上，这个服务器运行在和Web服务器不同的服务器上。这会带来问题，诸如：数据库索引在哪里生成？前面描述的SearchResultGeneration组件存在于哪里？

步骤8：考虑每个组件级设计表示，并且时刻考虑其他替代选择。纵观全书，我们始终强调设计是一个迭代过程。创建的第一个组件级模型不可能像迭代N次之后那么全面、一致或精确。在进行设计任务时，重构是十分必要的。

另外，你的眼光不应该狭隘。设计中经常会有其他替代设计方案，在没有决定最终的设计模型之前，最好的设计者会考虑所有（或大多数）的方案。你应当开发其他设计方案，并对每一个进行仔细考虑。

12.8 本章小结

构建和部署是WebE过程中一次或可能很多次迭代的顶点。在全面沟通、周密计划和详细工程建模之后，WebE团队构建和部署一个WebApp增量。产生最终用户可用的一个可操作增量。

构建和部署是把所有事物整合在一起的WebE活动。它们不是孤立发生的。大部分WebApp的构建和部署与在构建中使用的具体技术、工具和语言深深交织在一起。而且这些WebE技术种

类繁多，演化得也非常快。

构建活动包括选择、编码、创作、集成、重构和测试动作。当这些动作（和为每个动作定义的 WebE 任务）完成时，将产生一个准备好交付给最终用户的可运行的 WebApp。

组件级设计常常和构建活动并行进行。设计和实现由一组简单原则来指导。一旦完成这些动作，产生的 WebApp 组件就要进行重构以改进交付的最终产品。

部署活动包括三个动作：打包、发布和评估。因为 Web 工程本质上是增量的，所以部署不只发生一次，而会在 WebApp 完成的过程中发生很多次。和构建一样，可以应用一组原则来指导部署。

构建是一个基于组件的活动。设计良好的组件（不管是来自第三方还是由 WebE 团队开发）将呈现出四个关键特性，这些特性将产生一个易于维护的 WebApp。一系列清晰的设计和构建步骤可以在构建活动中指导 Web 工程师。

快速构建和部署的无情的需求迫使 WebApp 开发者采取捷径。试图变得敏捷并没有任何过错，但是如果捷径将导致错误和最终用户的不满，那么“速度的需求”就适得其反。

一个 WebE 团队应该在开发开始之前建立一个支持开发的有效环境。这样的环境使用测试服务器和预发布服务器，使得在快速开发的变更增量地发布之前进行有效评估。而且一旦有错误发生，这种环境允许尽快回滚到所部署的 WebApp 的前一个版本。

在第 13 章和第 14 章，我们将了解增加 Web 工程师开发效率的方法。在第 13 章中，我们将了解设计模式以及如何通过权衡已有知识和方案，从设计过程中提取一些臆测。在第 14 章中，我们将快速了解开发者可用的各种技术以及如何能够最有效地权衡这些技术。基本上，我们将了解如何最有效地使用前沿技术而不会被它们所制约。

参考文献

[Amb02] Ambler, S., “UML Component Diagramming Guidelines,” 2002, www.modelingstyle.info/ (accessed November 18, 2006).

[Aro02] Arolow, J., and I. Neustadt, *UML and the Unified Process*, Addison-Wesley, 2002.

[Ben99] Bentley, J., Programming Pearls, Addison-Wesley, 1999.

[Boh00] Bohl, M., and M. Rynn, *Tools for Structured Design: An Introduction to Programming Logic*, 5th ed., Prentice-Hall, 2000.

[Bru04] Bruegge, B., and A. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java*, 2nd ed., Prentice-Hall, 2004.

[Con02] Conallen, J., *Building Web Applications with UML*, 2nd ed., Addison-Wesley Object Technology Series: Addison-Wesley, 2002, p. 336.

[Dar99] Dart, S., “Change Management: Containing the Web Crisis,” *Proc. Software Configuration Management Symposium*, Toulouse, France, 1999, www.perforce.com/perforce/conf99/dart.html (accessed August 13, 2007).

[Dar00] Dart, S., *Configuration Management: The Missing Link in Web Engineering*, Artech House Publishers, November 2000.

[Dij72] Dijkstra, E. W., “The Humble Programmer,” *Communications of the ACM* vol. 15, no. 10, 1972, pp. 859-866. Turing Award lecture. Also available at www.cs.utexas.edu/users/EWD/ewd03xd/EWD340.PDF (accessed August 13, 2007).

[Fir01] Firesmith, D. G., and B. Henderson-Sellers, *The OPEN Process Framework: An Introduction*,

Addison-Wesley, 2001.

[Fow99] Fowler, M., K. Beck, J. Brant, W. Opdyke, and D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison Wesley, 1999.

[Hai01] Haire, B, B. Henderson-Sellers, and D. Lowe, "Supporting Web Development in the OPEN Process: Additional Tasks," *COMPSAC'2001 : International Computer Software and Applications Conference*, IEEE Computer Society Press, Chicago, 2001.

[Hai02] Haire, B., D. Lowe, and B. Henderson-Sellers, "Supporting Web Development in the OPEN Process: Additional Roles and Techniques," in Z. Bellahsene, D, Patel, and C. Rolland (eds.), *Proc. Object-Oriented Information Systems*, Montpellier, France, 2002, vol. LNCS 2425, Springer-Verlag, Berlin, pp. 82-94.

[Har03] Harmon, P., *Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes*, Morgan Kaufmann Series in Data Management Systems, 2003, Morgan Kaufmann Publishers Inc.

[Hen01] Henderson-Sellers, B., B. Haire, and D. Lowe, "Adding Web Support to OPEN," *Journal of Object Oriented Programming* vol. 14 no. 3, 2001, pp. 34-38.

[Hen02] Henderson-Sellers, B., D. Lowe, and B. Haire, "OPEN Process Support for Web Development," *Annals of Software Engineering* vol. 13, 2002, pp. 163-202, http: //manta. cs. vt. edu/ase (accessed August 13, 2007).

[Hun99] Hunt, A., D. Thomas, and W. Cunningham, *The Pragmatic Programmer*, Addison-Wesley, 1999.

[Kar94] Karten, N., *Managing Expectations*, Dorset House, 1994.

[Ker78] Kernighan, B., and P. Plauger, *The Elements of Programming Style*, 2nd ed., McGraw-Hill, 1978.

[Knu98] Knuth, D., *The Art of Computer Programming*, 3 volumes, Addison-Wesley, 1998.

[Let06] Lethbridge, T., and R. Laganiere, *Object-Oriented Software Engineering: Practical Software Development Using UML and Java*, McGraw-Hill, 2005.

[Mar00] Martin, R., "Design Principles and Design Patterns,"
www. objectmentor. com , 2000 (accessed August 13, 2007).

[McC93] McConnell, S., *Code Complete*, Microsoft Press, 1993.

[McG01] McGovern, G., and R. Norton, *Content Critical: Gaining Competitive Advantage Through High-Quality Web Content*, 2001, Financial Times Prentice Hall.

[OMG04] Object Management Group (OMG), *UML2. 0 OCL Specification*, 2004, www. omg. org/cgi-bin/doc? ptc/2003-10-14 (accessed August 13, 2007).

[Pil05] Pilone, D., and N. Pitman, *UML2. 0 in a Nutshell*, O' Reilly, 2005.

[Sha01] Sharp, A., and P. McDermott, *Workflow Modeling: Tools for Process Improvement and Application Development*, Artech House, 2001.

第13章 设计模式

一位年轻的 WebApp 开发者在坐着喝第三杯咖啡的时候谈的全是速度。“我们要在 3 周内完成这个网站，没有多少时间去关注你说的过程。”

这位年轻人的话仅仅是部分正确的——速度是最重要的，尤其是在构建活动开始之后。但是一个敏捷的 WebE 过程可以使得 WebE 团队创建产生速度的引擎。如果你想快速地把网站开发出来，就不能拒绝过程的概念。

在面向速度的环境中，Web 工程师没有时间重新设计轮子（Wheel）。重新设计会浪费时间，更糟糕的是，一个新“轮子”可能没有先前设计的好。如果已经存在一个可用的合理解决方案，那么就应该使用。

对于 Web 工程师来说，利用那些解决过类似问题的人的工作很有意义。这可以通过使用模式（pattern）策略来实现。Brad Appleton［App98］对设计模式的定义如下：“模式是一种命名的洞察力财富，承载了对于竞争事件中某确定环境下重复出现的问题的已经证实的解决方案的精髓。”

换句话说，一种模式描述了一个具体问题及其好的设计方案。这些信息使用模板并以可预测的方式记录下来，以便重复使用。但是一种好的设计模式不仅仅是一个问题和解决方案的描述，它还为 Web 工程师们提供对这两者的洞察，并使他们能够确定这种模式适合于当前的需求。

在本章中，我们研究模式是什么及其如何最有效地使用。我们将通过一些最常见的模式来说明。

13.1 模式：理解概念

设计模式的概念最初由架构师 Christopher Alexander 提出，他对使用模式语言来表示设计知识感兴趣。他对模式的描述如下［Ale77］：“每种设计模式都描述了一个在我们所处的环境中不断重复发生的问题，然后描述该问题的解决方案的核心，这样，你就可以重复地使用该解决方案而根本无需用同样的方式重复工作两次。”

Alexander 对设计模式的定义意味着每个开发者都可以理解模式并把它应用到必须解决的具体问题上。模式保持不变，但实现是需要调整的。因为确定和描述的是一个适合正在解决的问题的已证明方案，所以节省了时间。而解决方案的确定和描述通常都要比方案的真正构建耗时得多。

13.1.1 什么是模式

模式有很多种解释，但是对大多数人来说，模式是代表某些事物概念的特定类。然而，在 Web 工程中，它有更精确的意思。

Gamma 等人［Gam95］在一本关于软件设计模式的书中介绍了一种用于描述软件设计模式的语言。书中写道：

> 你会在很多面向对象系统中看到类和相互通信的对象的重复模式。这些模式解决特定的设计问题，使面向对象设计更加灵活、优雅，最终重用性更好。它们帮助设计者将新的设计建立在以往工作的基础上，重用以往成功的设计方案。一个熟悉这些模式的设计者能够立即将它们应用于设计问题中而不需要再去发现它们。

模式的关键是它们并非仅仅展示一个特定解决方案或一类解决方案的模板，而是提供从一类问题到一类解决方案的映射。每种设计模式的目的都是提供一个描述，以使设计者可以确定：

1）这种模式是否适合当前的工作；2）这种模式是否能够重用（由此，节约设计时间）；3）这种模式是否能够用于指导开发一个类似的但功能或者结构不同的模式。一个好的模式应该使 WebApp 开发者（可能不是专家）利用其他人的专长。只要你对问题有足够理解，能够识别相关模式，那么你就可以使用这种模式来指导开发方案的某个特定部分。

模式在规模、关注点和适用性上可以有很大的差别。一种模式可能描述用于一个交互（例如，从五个或更少条目的集合中唯一地选择一个条目）的单一片段的方案的类型（例如，一系列的单选按钮）。另一种模式可能描述一个应用的整体结构（例如，一个矩阵导航结构）。一种模式可能注重解决交互、导航、功能、表示的问题或者问题的其他方面。

模式还应该提供确定其将要被应用的环境的足够细节。为了说明，考虑从五个或更少条目的集合中唯一地选择一个条目。有两种可能的方案：1）使用一组单选按钮；2）使用一个下拉列表。哪种方案更好呢？为了回答这个问题，你必须对问题有更多的理解。如果所有选项同时（一直）可用（可见），那么就选用单选按钮。如果按钮的范围动态变化，而仅仅是当前的选择需要可见，那么一个下拉列表可能就更加合适。尽管这个例子不怎么重要，但是它说明即使是简单的设计决策也可能涉及不是立即可用或者可能被忽略的知识。好的模式提供足够的信息来帮助设计者做出好的设计选择[⊖]。

13.1.2 模式怎么描述

成熟的设计准则会利用数以千计的设计模式。这些模式描述了在机械、电力、化学、民用、自动化、航空、生物和很多其他工程原则中经常遇到的问题，并提出了这些问题检验过的方案。所有的设计模式都可以用框住中展示的模板［Mai03］进行描述。

信息



设计模式模板

模式名称。以一个简短而有表现力的名字来描述模式的本质。

意图。对模式及其做什么进行描述。

别名。列出模式所有的其他名称。

动机。提供一个设计问题的例子。

适用性。说明该模式所适用的特定设计方案。

结构。描述实现该模式所需要的类。

参与者。描述实现该模式所需类的职责。

协作。描述参与者怎样协作来实现它们的职责。

效果。描述影响模式的“设计影响因素”和在模式实现时必须考虑的潜在取舍。

相关模式。交叉引用的相关设计模式。

设计模式的描述还可能会考虑一组设计影响因素。设计影响因素描述与使用模式的应用领域相关的非功能性需求（例如，易维护性、可移植性）。此外，影响因素定义了可能限制设计实现方式的一些约束。本质上，设计影响因素描述使用设计模式所必须的环境和条件（例如，一个给定的接口模式只有在用户已经有了特定领域知识的情况下才合适）。模式特征（结构、参与者和协作）应该指明设计的属性，调整这些属性可以使模式适应各种各样的问题。这些属性表示了设计的特点，可以通过搜索（例如，通过一个数据库）找到一个合适的模式。最终，与使用一个设计模式相关的指导原则指出了设计决策的结果。

应该慎重选择设计模式的名称。软件重用中的一个关键技术问题是当存在几百或几千个候选模式

⊖ 对模式更多讨论的一个好的起点是 Wikipedia 的由 http://en.wikipedia.org/wiki/Design_pattern_(computer_science)。

时，缺乏找到合适的可重用模式的能力。一个有意义的模式名称非常有助于查找“合适”的模式。

13.2 WebApp 模式：设计焦点和粒度

尽管有很多不同的模式及其分类方法，但是我们只重点研究模式的两个方面：模式的设计焦点和考虑的粒度级别。*设计焦点*确定设计模型的哪个方面是相关的（例如，信息架构、导航、交互）。*粒度*确定正在考虑的抽象级别。（例如，模式是应用到整个 WebApp、一个 Web 页面、一个子系统或者一个 WebApp 组件?）

13.2.1 如何使用设计焦点来确定模式

在前面关于 Web 工程设计的章节中，我们讨论了各种不同的设计层次。换句话说，我们越深入 WebApp 设计，设计焦点就越狭窄。设计信息架构时遇到的问题（和解决方案）不同于进行界面设计时遇到的问题（和解决方案）。因此，很自然就可以为设计焦点的不同层次开发 WebApp 设计模式，这样你就可以处理在每个层次遇到的独特问题（和相关方案）。设计模式可以以如下方式进行分类：

- **信息架构模式**和信息空间的整体结构以及用户与信息的交互方式相关。
- **导航模式**定义导航链接结构，如分层、环和浏览。
- **交互模式**辅助用户界面的设计。这种模式描述了界面如何把特定行为产生的结果通知用户，用户如何基于使用环境和用户期望扩充内容，如何最好地描述链接所暗示的目的地，如何把正在进行的交互行为以及界面相关的问题告知用户。
- **表示模式**可以辅助通过界面给用户展示内容。这种模式描述了如何组织用户界面控制功能，以达到更好的可用性；如何展示界面行为和其所影响的内容对象之间的关系；如何建立有效的内容层次。
- **功能模式**定义工作流、行为、过程、通信和 WebApp 内的其他算法元素。

在大多数情况下，当遇到交互设计问题时，探索信息架构模式集是无益的。你应该审查交互模式，因为那才是和正在进行的工作相关的设计焦点。

13.2.2 粒度为什么是模式的一个重要特征

当一个问题涉及“全局”问题的时候，我们会努力开发针对此全局的方案（并使用相关的模式）。相反，当焦点非常窄（例如，从 5 个或更少条目的小集合中唯一地选择一个条目）的时候，方案（和相应模式）的针对性就会相当窄。就粒度级别来说，模式可以描述为如下级别：

- **WebApp**。这种抽象级别通常和定义应用整体结构的*架构模式*相关，体现了不同组件或增量之间的关系，并定义了说明架构元素（页面、包、组件、子系统）之间关系的规则。
- **页面/屏幕/子系统**。这些设计模式解决设计中的特有元素，例如解决一些设计问题中的组件聚合、页面上元素之间的关系或者用于影响组件间通信的机制。一个例子可能是 WebApp 主页面布局的大报（broadsheet）模式[⊖]。
- **组件**。组件模式与 WebApp 单独的小规模元素相关。例如，单独的交互元素（例如，单选按钮、文本块）、导航条目（例如，你如何编排格式）或者功能元素（例如，特定的算法）。

⊖ 一种大报模式解决可读和可访问方式的各种不同内容的布局问题。建议的一种方案是把内容像大版式报纸（一张报纸的头版）那样进行显示，有多个专栏、可能链接到更多细节的报头标题和专题文本。参见 White Hat Web Design and SEO Optimization Services 网站 www. white-hat-web-design. co. uk/articles/newspaper. php 中的讨论和例子 UTS Union Ltd 网站 www. utsunion. uts. edu. au/。

也可能定义不同模式和不同种类的应用或领域之间的关联。例如，一组模式（在不同级别的设计焦点和粒度）可能与电子商务 WebApp 特别相关。

13.3 模式仓库

绝大多数 Web 工程师都认识到模式是一个有用的工具。然而，他们面临的主要问题是发现和所需解决的问题合适的已有模式。更具体地说，就是需要一些机制使你以一种有效的方式来找到模式。这种机制必须提供一种描述问题特征、设计焦点、所需粒度以及 WebApp 范围的方法。基于这些描述，这种机制应当提供适当的候选模式。模式库就提供这样的机制。

13.3.1 什么是模式库

模式库是已证实的、对 WebE 团队可能有用的模式的有组织的收集。本质上，它是一个数据库，可以使你完成在前面段落中讨论的机制。

一个模式库随着时间的推移而增长和演化。好的 WebE 团队将建立一个一旦新模式被识别就能被捕获的过程。模式被提出并提交给库，然后由其他人进行审查，他们对模式进行评论并就它的适用性和质量发表看法。在某些情况下，检查者将在模式的演化过程中提出使其更加有效的添加或删除建议。

13.3.2 Web 工程师可参考哪些模式资源

有很多很好的关于和 WebApp 开发相关的模式的信息源。表 13-1 包含一组有用的起始资源——尽管总会出现其他资源（而且已有资源有时会消失）。

表 13-1 WebApp 设计模式库和信息源

A	Web Design Patterns by Martijn van Welie **www.welie.com/patterns/**
B	Improving Web Information Systems with Navigational Patterns **www8.org/w8-papers/5b-hypertext-media/improving/improving.html**
C	Hypermedia Design Patterns Repository **www.designpattern.lu.unisi.ch/**
D	IAWiki **http://iawiki.net/WebsitePatterns**
E	IBM Patterns for e-Business **www-128.ibm.com/developerworks/patterns/**
F	Patterns for Personal Web Sites **www.rdrop.com/~half/Creations/Writings/Web.patterns/index.html**
G	Designing Interfaces: Patterns for Effective Interaction Design **http://designinginterfaces.com/**
H	WebPatterns **http://webpatterns.org/**
I	The Interaction Design Patterns Page **www.visi.com/~snowfall/InteractionPatterns.html**
J	InteractionPatterns by Tom Erickson **www.pliant.org/personal/Tom_Erickson/InteractionPatterns.html**
K	Improving Web Information Systems with Navigational Patterns **www8.org/w8-papers/5b-hypertext-media/improving/improving.html**
L	An HTML 2.0 Pattern Language **www.anamorph.com/docs/patterns/default.html**
M	Common Ground—A Pattern Language for HCI Design **www.mit.edu/~jtidwell/interaction_patterns.html**
N	Indexing Pattern Language **www.cs.brown.edu/~rms/InformationStructures/Indexing/Overview.html**

13.3.3 WebE 团队能够创建自己的模式集吗

一些模式专家认为很难创建团队自己的模式，因为团队受自己方案的影响。模式应当表示最佳实践，而且能够以一种支持其在其他问题环境中重用的方式进行描述。有时，对开发者来说很难做到公正评估他们自己的方案，因为他们常常觉得自己的方案最好。实际上，这些方案只是他们所能确定的最好的方案。然而，最佳实践表明，创建一个本地模式库来捕获和保留 Web 工程师的专门技能是合理的——尤其是当公司依赖于特殊的可能继续努力的个人的技巧时。

模式库的最简单形式可以是包含了本章前面的设计模式模板框中描述的设计模式模板中所列出的关键领域的电子数据表。稍微复杂的版本可能是一个有前端的支持域验证、元数据标记和其他功能的数据库。你还可能会考虑的另一种替代——而且更多地符合 WebApp 的开发——基于内联网的 Wiki，它允许对模式进行编辑和评估[⊖]。不论模式库具体采用什么形式，真正的挑战不在于设计库而在于设计捕获和使用模式的过程。

模式捕获技能并在很多 WebE 项目中重用这些技能。因此，为了使模式适应于你的环境，应该实现可持续的过程，而且必须有明显的高层管理支持。

一个模式库只有当其包含了很多问题描述及其解决方案时才有用（并被使用）。为了达到这个目的，组织必须承诺资金和支持以确保 Web 工程师有时间把他们的专长添加到不断发展的库中。换言之，应该把库和使库增长所需的过程融入到组织文化中。而且如果有人愿意拥护这个观点，那么将有帮助。



构建模式库

场景：WebE 团队会议室内的高级员工会议——SafeHomeAssured.com 项目开始的 2 年前

参与者：首席技术官（CTO）和高级项目经理

会话：

CTO（主持会议）：好的，那覆盖了本次会议的主要议题。其他人还有什么想讨论的主题吗？

项目经理 1：嗯，我确实有一件来自于刚刚这个项目的事情，我认为它可以帮助我们解决前面讨论的一些质量问题。

CTO（越来越感兴趣）：继续……

项目经理 1：我发现开发者的一个问题是，当他们遇到问题时就花大量时间来开发解决方案，后来却只发现当时时间花得不值。

项目经理 2：我不太明白——你的意思是他们开发了无用的方案？

项目经理 1：不，完全不是。大多数时候这些方案都是好的——我们肯定要去创建。我的意思是，既然我已经在这儿呆了这么久，我开始关注到有设计和正在开发的设计类似——尤其是自从我开始专攻这些已经做了更多工作的远程设备监控的 WebApp。我们很多的开发者在这里的时间都不长——所以他们并没有经验。他们要一直重新发明轮子。

CTO：是的，但这不就是我们有项目团队会议、开放式办公室等的原因吗——分享你们如何解决问题的思想——就不用花大量时间来重做那些别人已经做过了的事情吗？

项目经理 1：当然——那有时可以。但是有时前面做的人已经走了或者忘记了，或者只是没有联系起来而已。这不利于我们在不同团队之间分享知识。我认为我们真正需要的是收集普遍性问题和我们已经开发的方案。

⊖ 典型的例子参见 www.iewiki.net 的信息架构 Wiki 网站。

项目经理3：哦——你的意思是一个模式库。我有一个工作的地方有一个，而且它确实很有用。所有的开发者都必须在每个项目的末尾花一点儿时间把他们能识别的任何模式添加到集合中——从本质上来说那只是一个用一些表单和一个简单的搜索界面组成的简单的WebApp。

CTO：模式？库？解释一下。

项目经理3：[解释概念。]

CTO：但是创建库和对它进行增加不是要花费宝贵的时间吗？你把它算到谁的头上？

项目经理3：确实要花时间——但是这样做终节省的时间更多。而且我们并不把它算到任何人头上。或者相反，我们把就它记在一个已经建立的内部账户上——然后每当一个开发者使用系统中一种模式的时候，他或她就可以指出使用模式节省了多少时间。

CTO：好吧——这听起来很有趣。你下周来开会时带给我们一个如何对此进行管理的具体提议如何。

13.3.4 如何发现和使用模式

把信息存入库中只是成功了一半。模式只有在需要时可以被找到才有用。随着分析和设计的进行，Web工团队可以审查出现的WebApp模型[5]以确定正在描述的问题的各个方面能否被一个或多个模式覆盖。可以通过如下两种方式之一来确定可能的模式：

1. 基于当前问题的特征来检索库，并找到一个或多个候选模式。
2. 库中的模式以一种更全面的方式来浏览，以发现和当前问题匹配的模式。

一旦发现了一个候选模式，模式模板内包含的信息可以通过询问如下的问题来进行评估：

- 这种模式适合于WebApp的领域吗？就是说，上下文是否合适？
- 这种模式适合于设计焦点和问题的粒度吗？
- 这种模式不需要做出和当前WebApp设计不兼容的任何权衡或妥协就能使用吗？
- 这种模式不需要做出和设计的其他元素不兼容的任何权衡或妥协就能使用吗？

如果这些问题都可以得到肯定的回答，团队就应该对模式进行更详细的评估。



应用模式

场景：增量6（传感器控制）的开发过程中，SafeHomeAssured.com设计者在桌边进行的非正式讨论

参与者：WebApp设计者和SafeHomeAssured.com的首席系统架构师

会话：

系统架构师：摄像机控制的界面设计进行得如何？

WebApp设计者：很好的——我已经编写了大多数的基础代码来连接到实际的传感器，没有太多问题。我也已经开始考虑用户从远程Web页面上实际移动、平移和缩放摄像机时的界面，不过我还不确定我做的是否正确。

系统架构师：那么，你想到了什么呢？

WebApp设计者：嗯，需求是摄像机控制需要高度的交互性——当用户移动控制时，摄像机应当是可能实地移动。所以我想到了像一个普通摄像机那样设计一组按钮，然后当用户点击时就可以控制摄像机。

系统架构师：嗯，是的，那应该可以，但是我不确定它是对的。每次点击一个控件，你需要等待整个客户端—服务器之间的通信，那么将不能获得很好的快速反馈。

[5] WebApp模型在第6章～第11章进行了详尽讨论。

WebApp 设计者：那正是我所想的——以及我为什么对这种方法不是很满意的原因，可是我不知道除此之外我能怎么做。

系统架构师：那为什么不用“交互设备控制”模式呢！

WebApp 设计者：唔——那是什么？我从没听说过。

系统架构师：它本质上就是用于你描述的问题的一种模式。如果你在我们的模式数据库中找，你会发现更多的细节。它所提议的解决方案从本质上说就是创建到设备服务器的控制连接，通过它可以发出控制命令。那样你就不需要发送普通的 HTTP 请求。这种模式甚至显示了你如何通过一些简单的 AJAX 技术来实现。你有一个简单的客户端 JavaScript，直接和服务器通信，而且在用户做出任何动作时发送命令。

WebApp 设计者：好主意！我将到模式数据库中核实它。

系统架构师：嗯——但是记得审查模式的其他问题。我好像记得那儿特别需要注意安全问题。我想可能是因为你正在创建一个独立的控制信道，这可能绕开正常的 Web 安全机制。

WebApp 设计者：很有道理——我很可能会想不起这些！谢谢。

13.4 模式示例

为了说明可用于 WebApp 设计的模式的范围，并提供各种模式来源的集合，我们提供了一组可用于 Web 工程工作中的模式示例。对于每个大类，我们还考虑一个应用于 SafeHomeAssured. com 的具体的例子。在下面的表格中，源（source）列中的字母指的是表 13-1 中列出的信息源。

13.4.1 有可能定义解决业务层问题的模式吗

如前所述，模式可以在任何抽象级别开发。业务级模式与整体 WebApp 和利益相关者如何使用 WebApp 相关。一般来说，这些模式描述了在建模活动（分析和早期设计阶段）中要考虑的各种方法，参见表 13-2。

表 13-2 业务级模式示例

模式种类（模式名称）	来源	简要说明
业务模式 协作（用户到用户） 信息聚合（用户到数据） 扩展的企业（业务到业务） 自服务（用户到业务）	E	这些模式描述了不同形式的利益相关者交互：用户和其他用户的交互（如拍卖网站）、用户和业务的交互（如电子商务应用）、业务之间的动作（如有供应链管理的应用）
集成模式 访问集成 应用集成	E	这些模式描述了业务模式的组合，这些业务模式可用于集成多个应用、多种访问模式以及构建无缝应用的多个信息源
组合模式 账户访问 电子商务	E	这些模式描述了组合业务模式和集成模式来提高执行复杂业务功能的总体方案。一个典型的例子是一种电子商务模式，它链接了一种自服务模式（描述用户参与）和一种有应用集成模式的信息聚合模式（用于访问目录等）

（续）

模式种类（模式名称）	来源	简要说明
电子市场 门户 **电子商务** 预约处理 案例学习 登录 时事通讯 优质内容锁 产品顾问 产品比较 产品配置 购买过程 注册 购物车 仓库定位器 证明书 虚拟产品展示	A	这些模式描述了电子商务应用的特定例子，寻求用户与企业间能支持的不同的交互方式
网站类型 艺术网站 汽车网站 品牌宣传网站 运动网站 商业网站 社区网站 企业网站 跨国公司网站 博物馆网站 我的网站 新闻网站 门户 基于 Web 的应用	A	这些模式描述了特定领域的 WebApp，突出应用域内需要的宽泛内容、结构、导航和功能

作为一个例子，考虑 SafeHomeAssured. com 的增量 3（产品报价单和处理产品订单），要开发支持用户订购产品的功能。

13.4.2 因为交互很普遍，就必然有很多交互模式吗

绝对是这样！已经发布的很多交互模式都重点支持用户和 WebApp 的交互。这是一个很宽广的领域，包含了一般的用户体验、用户界面观感和布局的设计以及交互的支持。交互模式还解决 WebApp 交互的构建和使用方法，包括支持用户交互的详细交互支持和功能（例如，搜索或索引），参见表 13-3。

表 13-3 界面模式示例

模式种类（模式名称）	来源	简要说明
用户体验 社区建设 娱乐 信息查找 学习 购物	A	这些模式定义了 WebApp 的主要目的以及支持这些目的的过程中发生的用户交互形式。例如，一个目的是社区建设的网站应该支持用户收集和共享信息。这个问题描述了如果通过诸如论坛和推荐这样的机制来实现它

（续）

模式种类（模式名称）	来源	简要说明
搜索 高级搜索 FAQ 帮助向导 搜索区域 搜索提示 搜索结果 简单搜索 网站索引 网站地图 专题页面	A	搜索的目的和方法有很多。这些模式描述了适合不同的搜索上下文的解决方案
页面布局 活动引用 行为分组 行为期望 我在这里 索引导航 信息分解 信息交互耦合 信息交互解耦 过程反馈 可选关键字 可选搜索空间 简单搜索界面	C	界面和布局模式描述了设计每个 Web 页面上实际对象的安排方法。这是一个典型的可视化沟通问题，而且会影响用户和页面发生联系的方式。例如，这些模式解决如提供导航的当前状态信息（例如，踪迹）和为用户提供激活一个界面对象的结果的信息（例如，这个链接将把我带到哪里）的问题
组织页面 卡片栈 中央区域 可关闭面板 对角线平衡 流体布局 可移动面板 响应式展开 响应式激活 标示部分 可见框架	G	这些模式和前面的类似，描述了 Web 页面的安排以及这种安排如果解决具体问题。例如，卡片栈（例如，标签）解决当单个页面太挤时，如何把内容安排成组的问题
命令和动作 操作面板 命令历史 多级撤销 进度指示器 智能菜单项	G	这些模式和具体界面组件以及如何设计它们以最大化易用性有关。通常，这和在支持对用户的有效反馈进展以及功能的可用性有关

SafeHome



选择一种商业模式

场景：增量 3（产品报价单和处理产品订单）的设计开始时的初始讨论

参与者：WebApp 设计者和 SafeHomeAssured. com 的首席系统架构师

会话：

WebApp 设计者：我昨天开始对增量 3 做一些初始考虑，但是我认为在深入之前应该征求你的一些想法。

系统架构师：好的，你想到了什么呢？

WebApp 设计者：就像你说的，为已经解决过的问题重新设计方案是没有意义的。所以我就想知道可以发现有什么适合于我们的产品报价单和订购系统。我们已经粗略地把需求列在了卡片 5［获取一个产品报价单］和 8［订购一件产品］上，而且在我们的利益相关者可以接受的定价模型和支付类型上做了一些工作。我认为审查可能的方案最好的开始是你为我展示的 IBM 电子商务模式网站。现在，我把它显示在屏幕上。看，很明显这是一个自服务应用——但是它是什么类型呢？

［设计者继续访问下面的页面：www-128.ibm.com/developerworks/patterns/u2b/select-application-topology.html。］

系统架构师：唔，有很多不同的可能。

WebApp 设计者：是的，但是看一下顶部的这个表格。它帮助我们发现一种合适的模式。在这种情况下，关键的驱动力看起来相当清楚——客户端渴望改进组织上的高效性并保持应用尽可能简单。那看起来好像指出了定制显示是合适的模式。它本质上涉及在已有的订购系统之上提供一个显示层，而非努力从头开始重做。

［设计者和架构师继续讨论提出架构的细节，把这种模式作为探讨设计问题的基础。］

作为一个例子，SafeHomeAssured.com 的增量 2 允许一个用户搜索特定传感器上的信息。这个特定用例的详细分析表明，包含一个搜索引擎是合适的。因为搜索是一个很常见的问题，所以有很多有关搜索的模式就不足为奇了。参看表 13-3 列出的模式，我们发现了下面的模式以及它们解决的相关问题：

高级搜索。用户必须从大量的条目中寻找一个具体条目。

帮助向导。用户需要一个和网站相关的特定主题的帮助，或者当他们需要寻找网站内一个具体页面的时候。

搜索范围。用户必须找到一个页面。

搜索提示。用户需要知道如何控制搜索引擎。

搜索结果。用户必须处理搜索结果列表。

对 SafeHomeAssured.com 而言，产品的数目不是特别大，而且每种产品都有一个相对简单的分类，所以一个高级搜索或一个搜索向导可能是不必要的。类似地，搜索足够简单，不需要搜索提示。对搜索框的描述（部分）展示在下面的框注中：

模式示例

简单搜索：搜索框模式

问题：用户需要找到一个条目或具体信息。

使用场合：任何已经有主要导航的网站。用户可能想在一个种类内搜索一个条目。用户可能想进一步指定一个查询。

方案：提供一种搜索

搜索界面

提供搜索功能，它由一个搜索标签、一个关键字文本框、一个过滤器（如果合适）和一个“Go”按钮组成。按下返回键和选择 Go 按钮的功能相同。还在单独页面内提供搜索提示和例子。到那个页面的链接置于搜索功能旁边。搜索术语的编辑框足以容纳 3 种典型的用户查询（一般在 20 个字符左右）。如果有多于两个过滤器，使用一个组合框或一个单选按钮来选择过滤器。

搜索——编辑框——在/用于——过滤器——Go 按钮或只是……

——编辑框——Go 按钮

显示搜索结果

搜索结果和一个至少包含"搜索结果"或类似内容的清晰的标签显示在一个新页面。搜索功能连同输入的关键字仍然显示在页面顶部，以使用户能知道搜索的关键字是	什么。 等等 （摘自 www.wefic.com/patterns/showPattern.php?patternID=search。）

模式接着描述关于如何访问、表示、匹配搜索结果等大量细节。基于此，SafeHomeAssured.com WebE 团队可以把如图 13-1a 所示的搜索工具组成整体，所有页面都含有搜索工具。当完成搜索时，会出现如图 13-1b 所示的结果。



图 13-1 SafeHomeAssured.com 的搜索功能示例

13.4.3 有哪些可用的导航模式

一个 WebApp 的成功常常与用户在信息空间内移动的容易程度相关。当一个 WebE 团队开始设计的时候，会出现很多导航问题。这些问题可能像在穿过导航路径时订购内容那样简单，也可能像综合导航结构的布局那样复杂。导航模式处理这些和其他在 WebApp 导航结构中表达的与内容组织相关的问题。（参见表 13-4。）

表 13-4 导航模式示例

模式种类（模式名称）	来源	指南
导航模式 主动引用 界标 新闻 上下文中的结点 基于设置的导航 路径值	D	这些常见模式描述了适合不同上下文的内容安排，例如，新闻（News）模式描述了这样一种结构：最近新闻内容（可能有一个简要介绍）被直接链接到主页面或主要页面，而旧的内容需要通过一个结构化的存档才可用

（续）

模式种类（模式名称）	来源	指南
结构导航 收集中心 复杂实体 指南 混合收集 索引导航 导航上下文 导航策略 新闻 作为导航视图的结点 机会链接 可选的搜索引擎 可配置导航 结构化答案	C	这些模式和前面的模式类似，侧重于对内容进行不同的安排以满足不同的目的。例如，指南（Guided tour）模式处理用户在头脑中没有一个具体目标时，帮助他们获得一个结构概况。相反地，机会链接（Opportunistic Linking）模式描述了当用户在信息空间中移动时，通过提供链接提示来保持用户的参与这种问题的解决方案
导航 导航层级 目录 双标签 分面导航 无页眉菜单 图标菜单 图像浏览器 主导航 地图导航 元导航 扫雷 覆盖菜单 重复菜单 伸缩菜单 滚动菜单 快捷菜单 分割导航 勒索菜单 踪迹菜单	A	和描述具体的导航结构不同，这些模式描述了用于访问和理解这种结构的机制。虽然说大部分具体机制对于大多数设计者（和实际用户）来说很熟悉，正在解决的问题却不一定如此明显。作为一个例子，扫雷（Minesweeper）模式涉及只有当用户移动鼠标滑过触发点（例如，用户可以“猎取”感兴趣的内容）时才有的内容摘要（常常链接到进一步的信息）的动态显示。当用户寻找具体信息时，这种模式并不合适，但它却能提供了一种使用户参与到探索或娱乐网站（或子网站）的有效方法
游览 动画过渡 清晰的录入点 颜色编码部分 全局导航	G	这些模式描述了界面方面，包括导航结构和可用选项的理解
组织内容 需要时显示 引人注目的分支 单窗口层次深入 双面板选择器 向导	G	这些模式强调导航选项的特定实例，而不是一般的结构。例如，吸引人的分支模式描述直接从内容嵌入链接来提供上下文导航，而不是单独的结构，wiki 就是很好的一个例子

以 SafeHomeAssured. com 的增量 6 为例，说明导航模式可能有用的情况。WebE 团队必须实现支持一个注册用户控制和监控用户空间内各种传感器。在这种情况下，开发者认识到一个用户可能想要不断快速地选择和观看不同的传感器。在这样的情况下，用户必须反复向后导航到对传感器的选择会很繁琐。更简化的机制是下框所描述的双控制面板选择器模式。

模式示例

快速选择：双控制板选择器模式[①]

描述：在界面上并排放置两个面板。第一个显示用户可以任意选择的一组条目；另一个显示选择条目的内容。

使用场合：你正在展示一组对象、种类或者甚至是动作。信箱中的信息、网站的部分、图书馆中的歌曲或图像、数据库记录、文件等都是很好的选择。每个条目都有和自身相关的令人感兴趣的内容，如电子邮件的文本信息或关于文件大小或日期等细节。你想要用户看到列表的总体结构，而你也想要他们以自己的速度、选择的顺序穿过条目。

从物理上来说，你工作的显示器足够大，可以同时显示两个分离的面板。非常小的手机显示不能应付这种模式，不过一个诸如 Blackberry 或 iPhone 手机的屏幕则可以。

原因：双面板选择器（Two-Panel Selector）是一个学术的但却是非常常见和强大的惯例。人们很快就看到他们要在一个面板上选择条目，在另一个面板上就看到条目的内容。他们可能在电子邮件客户端、Windows 浏览器、网站等使用过；无论哪种方式，他们都把这些概念应用到其他类似的应用。

当两个面板同时可见时，用户可以来回地快速切换注意力，现在看看总体结构的清单（"我还有多少未读的电子邮件信息？"），现在看看一个对象的细节（"邮件的内容是什么？"）。这种紧密集成和其他物理结构，如两个分离窗口或一个窗口逐层分析（One-Window Drilldown）相比，有几个好处。

使用方式：把可选择列表放在顶部或左边的面板上，而把显示细节的控制板放在下边或右边。这利用了大多数从左到右表达的用户期望的视觉控制。（试着为从右到左表达的人颠倒过来。）

当用户选择一个条目的时候，立即在第二个控制板中显示其内容或细节。选择应该用一个简单的点击来完成。但是当你做这件事的时候，为用户提供一种从键盘改变选择的方式，尤其是用箭头键。这既减少了用于浏览的实际工作量和时间，又有利于只使用键盘时的可用性。

使被选择的条目清晰可见……

① 摘自 http://designinglaterlanes.com/Two-Panel_Selector 上的设计界面网站。

SafeHomeAssured.com WebE 团队把如图 13-2 所示的导航设计组成整体。



图 13-2 传感器检控功能的导航设计示例

13.4.4 内容和表示模式适合使用的地方

WebE 团队使用业务模式来帮助解决和业务环境内的 WebApp 使用相关的总体设计问题。团队使用交互模式和导航模式来建立与 WebApp 的通信以及在信息空间内的移动。最终，使用内容模式和表示模式来详述 WebApp 的内容并组织内容以支持 WebApp 的目标。参见表 13-5。

表 13-5 内容和显示模式示例

模式种类（模式名称）	来源	简洁描述
基本页面类型 主章节页 搜索页面 合同页面 事件日程 论坛 访客留言 主页面 输入错误信息 打印机友好页面 处理页面 产品页面	A	这些模式描述 WebApp 的具体组件和包含在支持这些组件中适合的内容。例如，一个事件日程（Event Calendar）（可能用在一个团体网站中，如一个本地冰雪俱乐部）不仅提供了事件列表，而且通常还会通过提示一个日期选择时间线查询事件，以及支持分类和过滤（对有很多事件的网站）或支持它
显示复杂数据 缩略列表 翻转书 总览加细节 行条文 可分类表格 树表	G	这些模式描述了把内容展示给用户的特定方法。例如，总览加细节（Overview Plus Detail）模式显示了一段内容，但同时也指示了它在整个上下文的抽象或总结中的位置，并告诉用户当他们向下获取细节的时候，可以进行加载整个或内容的空间展示
使它好看 选择处理 按照背景 少数颜色，多些价值	G	这是具体图像设计模式的一个小的部件。然而需要注意，一般来说，图像设计问题通常表达为设计启发方法或准则（例如，查看 www.useit.com 上 Jakob Nielsen 的可用性网站），不是连接具体问题和方案的模式

再次考虑第 13.4.3 节中的选择和传感器监控的例子。在这个增量构建完之后，SafeHomeAssured.com 团队进行了一些可用性测试，发现用户发现他们很难快速知道选择哪个传感器（使后图 13-2 展示的单选按钮的顶部面板）。再次考虑设计的这个方面，设计者发现了框注书描述的地图导航员（Map Navigator）模式。

模式示例



提供一种导航竞争：地图导航员模式[F]

问题：用户需要找到一幅地图上选项的位置。

使用场合：网站有可能去搜索一个具体的位置。例如，一个公司网站或电子商务网站可能有一个商店定位器，使用户可以找到一个物理商店。在其他情况下，例如一个允许人们找到任意目的地的网站，用户将看到他们的搜索结果显示为地图上的位置。

解决方案：展示一幅地图，它有着用户的关注点，并提供在所有角落的导航链接。

这幅地图在图像中心显示兴趣点（point of interest，POI）。使用不同的标志或颜色标记不同的 POI，并提供一个示例来解释它们。如果只有一个 POI，那么就提供那个 POI 的准确位置。在有多个 POI 的情况下，可以使

用扫雷（mincewcoping）［另一种模式］在用户把鼠标移过某个 POI 的时候显示那个 POI 的细节。用户可以通过选择角落中的任何导航链接来在地图上移动他们的“窗口”。页面将会重新载入，同时显示地图稍微不同的部分。添加缩放并标示地图的比例。很多人可能想把地图打印出来以随身携带，所以必须有一个可用的打印机友好页面。

原因：我们从现实世界中辨别地图，并且也乐于在 Web 上看到。Web 上的导航特征因需要重新载入页面而并不理想，但是这不会带来可用性问题……

① 摘 自 www.welie.com/patterns/showPattern.php?patternID=map-navigator(交互设计模式网站Welie.com)。

把这种模式应用到传感器选择和监控的一个较小重新设计的结果展示在图 13-3 中。



图 13-3 传感器监控功能的导航设计的修改版本

13.5 本章小结

在给出 Web 的快速变化性质和设计有效的 WebApp 的复杂性的情况下，对 Web 工程师来说，在正确的时间使用最好的设计方案是一个挑战。任何可以帮助你确保方案是合适的和反映最佳实践的解决方案都是被鼓励的——这也正是模式想要实现的。

模式捕获专家的知识，并为开发者提供一种记录其来之不易的经验的方法。如果例这些经验收录在库目录中，那么模式就使得这些知识对其他人可用了。在本章中，我们只讨论了对于

Web工程师可用的大量模式的很少的一部分。你可以使用表13-1或本章其他表中列出的网站开始你对合适模式的搜索，或者和你的组织一起开发内部模式库。不论你采用哪种方法，你的组织都将避免在每次遇到一个设计问题时都进行重新设计。

在第14章中，我们将看一看技术、工具和环境（它们实际上使Web工程师完成这本书的大部分所讨论的工程模型），来圆满结束对WebApp构建的考虑。

参考文献

[Ale77] Alexander, C., *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, 1977.

[Amb02] Ambler, S., "UML Component Diagramming Guidelines," 2002, www.modelingstyle.info/ (accessed November 18, 2006).

[App98] Appleton, B., "Patterns and Software: Essential Concepts and Terminology," 1998, www.cmcrossroads.com/bradapp/docs/patterns-intro.html (accessed August 13, 2007).

[Ber98] Bernstein, M., "Patterns in Hypertext," *Proc. 9 th ACM Conf. Hypertext*, ACM Press, 1998, pp. 21-29.

[Bus96] Buschmann, F., et al., *Pattern-Oriented Software Architecture*, Wiley, 1996.

[Fow03] Fowler, M., et al., *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2003.

[Gam95] Gamma, E., et al., *Design Patterns*, Addison-Wesley, 1995.

[Gar97] Garrido, A., G. Rossi, and D. Schwabe, "Patterns Systems for Hypermedia," 1997, www.inf.puc-rio.br/~schwabe/papers/PloP97.pdf (accessed August 13, 2007).

[Ger00] German, D., and D. Cowan, "Toward a Unified Catalog of Hypermedia Design Patterns," *Proc. 33 rd Hawaii Intl. Conf. On System Sciences*, IEEE, vol. 6, Maui, Hawaii, June 2000, www.turingmachine.org/~dmg/research/papers/dmg_hicss2000.pdf (accessed August 13, 2007).

[Mai03] Maioriello, J., "What Are Design Patterns and Do I Need Them?" developer.com, 2003, www.developer.com/design/article.php/1474561 (accessed August 13, 2007).

[Per99] Perzel, K., and D. Kane, "Usability Patterns for Applications on the World Wide Web," 1999, http://jerry.cs.uiuc.edu/~plop/plop99/proceedings/Kane/perzel_kane.pdf (accessed August 13, 2007).

第 14 章　技术和工具

写 Web 工程技术很像是试图站在铁路站台上辨认快速移动的火车内的人。你知道他们在上面，但是火车移动得实在太快，对某节车厢内的人来说，在他们被另一节车厢取代之前确实很难认出他们。

新的工具和 Web 技术变化得非常快。实际上，在你阅读本章时，就会有一种完全不同的技术和工具的景观。作为一名 Web 工程师，你将必须评估和使用我们在写作本章时还不存在的语言、工具、框架和环境。

例如，现在，支持 Web 2.0 的技术（例如，AJAX 和 RSS）是很多讨论的主题，但是到你阅读本章时，很可能其他技术[⊖]将成为焦点。在本章中，我们将讨论 Web 技术的现状，说明 WebE 的方法和可能性。我们将讨论如下两类主要技术：

1. **实现工具**。包括各种不同的技术，例如 Web 应用服务器、内容管理系统、文件共享系统和安全管理。

2. **开发工具**。包括设计建模、问题跟踪和应用测试。

在有些情况下，这两类之间的边界很模糊。例如，一种设计工具可能被集成到内容管理系统中。

14.1　常见问题

在讨论具体技术之前，有几个重要的常见问题值得进行考虑。第一个问题强调需要划分 WebApp 的能力及其实现工具。第二个问题处理我们使用工具的来源：是由开发者和用户的开放社区开发的还是由一个单独商业供应商提供的？第三个问题考虑 WebApp 种类对可用于开发 WebApp 的技术和工具的类型的影响。

14.1.1　分而治之如何影响工具和技术

分而治之（separation of concern）[⊜]原则认为一个问题或一种方案的概念化应该被分解为独特的组件，在对组件间交互进行最小化的时候，这些组件的关注点是特定的方面。用软件工程的行话来说，单一关注点通常被为一个组件的内聚性，组件交互称为组件间的耦合。我们总是努力增加内聚性而降低耦合。结果是一个更加易于理解和维护的系统。

对 WebApp 来说，分而治之通常意味着分离如下 WebApp 组件：

- 内容存储
- 内容自适应
- 显示
- 显示自适应
- 内容组织和导航
- 功能（例如，搜索和工作流管理）

⊖ 虽然猜测将来会出现什么技术很困难，但这并不会阻止我们的尝试。在不久的将来（可能在你读这个的时候就已经出现了）将会有非常重要的技术，包括：主流的语义 Web 技术、不断出现的 Web 服务和便于适应不同用户客户端的工具并有助于开发上下文感知应用的技术。

⊜ 关于分而治之更多信息的一个好的起点是 http://en.wikipedia.org/wiki/Separation_of_concerns（2007 年 8 月 13 日）上关于分而治之的 Wikipedia 文章。

很多应用框架采用一些和上述的分而治之类似的形式。例如，内容管理系统（第16章）常常把内容存储（存储为内容对象）和内容显示（存储为显示模板）分开。甚至基础技术本身都已经开始逐渐向增加分而治之转变。最早的Web技术（HTML），把显示和导航直接嵌入到内容之中，导致大量的维护工作。后续有技术用来分离这些元素：XML表示内容，CSS表示显示的格式化，XLink表示链接结构。通过强调这些较新的技术的分而治之，大大加强了维护（浏览、改正、适应和加强）已有WebApp的能力。

回顾在第11章中展示的MVC（模型-视图-控制器）框架，我们看到分而治之的另一个例子。MVC定义了一个把内容和导航（模型）从应用接口（视图）和模型（控制器）的功能维护中解耦的框架。很多实现工具都适应了MVC或其变种。

14.1.2 选用哪种技术——开源还是私有

在你评估技术和工具景观时，在开放源代码和私有技术之间的选择就成为了一个重要问题。Wikipedia[1]定义开源如下：

开放源代码（open source）描述驱动了对最终产品来源（源代码）访问的生产和开发中的实践。一些人把它视为一种源程，其他人把它视为一种实用方法学。在开放源代码被广为采用之前，开发者和生产商使用各种短语来描述这个概念；开放源代码这个短语随着因特网的发展逐渐流行起来，并能够提供各种生产模型、沟通途径和交互团体。最终，开源软件成为了开放源代码的标志。

和开源相比，私有工具和技术由一个单独供应商开发，而供应商通常不会把源码提供给最终用户或其他人。

开源工具（例如，Apache Web服务器和Eclipse开发环境）的优势是无限的发展、延续性以及快速演化和改进。开源工具的演化确实由市场驱动，而最终用户社区在使用私有工具的时候，常常会感到失去所有权。开源工具内的错误能够相当快地被发现，带来可称赞的产品质量。然而，当需要支持的时候，社区开发环境会带来问题。

私有工具的优势常常是改进的支持（或至少有义务支持）和一个更密集的发布周期。这样的工具常常被更有效地打包，而且对非技术专家往往更加容易在他们的组织内实施。

一般来说，对开源和私有Web应用技术和工具之间的选择应该基于你对如下问题的回答：

- 工具能否满足需要的能力和将要被部署的功能？
- 报告的质量和扩展性能否适合你的需要？
- 工具所预测的演化方向能否满足你现在和将来的需要？
- 工具是否有足够的支持资源、联机文档和帮助？
- 工具的成本是否在项目或组织预算范围内？

一旦你确定了这些问题的答案，你的团队就可以辩论开源和私有技术哪个对你适合。



开源和私有

场景：SafeHomeAssured.com的会议室，在增量2开发的初始讨论期间

参与者：WebE团队负责人和WebE技术经理

会话：

[1] http://en.wikipedia.org/wiki/Open_source。

WebE 团队负责人：好，那么我们已经同意存储内容的逻辑方法就是简单地把它放在利益相关者使用的已有数据库格式而不是试图把它映射到 XML 上。这将使他们能够继续促进已有的用于更新内容的过程的改变。下一个问题是我们如何从相关 WebApp 组件内访问这个数据库。我一直设想可能使用一个像 SQL 管理器那样的商用私有数据库。你认为呢?

WebE 技术经理：噢，是的，我们可以这样做，但是考虑到这个项目的性质，我认为一个开源方案可能更好一些，类似 MySQL 或 Ingres。

WebE 团队负责人：为什么？从功能上来说，都很简单。一个私有数据库服务器难道不是更容易快速建立吗?

WebE 技术经理：是的，但是营销部门已经指出了他们认为很可能继续进行变化和演化，而且他们不希望被束缚在一个特定平台上，而开源产品通常为我们提供更多到其他产品和环境的适应性。

WebE 团队负责人：嗯——有道理。你就把这些记录下来，然后我将在我们做出最终决定之前和营销部门一起讨论，好吧?

14.1.3 WebE 技术应用种类的影响

在第 1 章中，我们讨论了很多 WebApp 种类——信息型、下载型、可定制型、交互型、用户输入型、面向事务型、面向服务型、门户型、数据库访问型和数据仓库型。在大多数情况下，WebApp 种类帮助确定可以应用的 WebE 技术的类型。

早期的 WebApp 是信息型的。也就是说，它们以静态单一方向流的形式为最终用户提供基于超媒体的内容。当它们包含功能的时候，主要意图是改变控制流或内容或其显示的性质。

近些年，WebApp 演化成复杂得多的用户的双向参与。这些 WebApp 常常都称为 Web 2.0。这类应用的典型例子是：

- 大众分类（Folksonomy）。那些支持内容提供和分类的协同生成的 WebApp（例如，Flickr，www. flickr. com），旨在支持信息检索。
- 混搭（Mashup）。包含多个信息源混合的 WebApp，提供协同成果（参见 www. programmableweb. com 上有关可编程 Web（Programmable Web）的一长串例子）。
- 社交网络（Social networking）。致力于构建用户之间富链接的 WebApp（例如，参见 en. wikipedia. org/wiki/Social_networking_sites 上关于 Wikipedia 的 Social Networking Site 列表）。
- 维基（Wiki）。使用户能够添加、移动和编辑常见内容的 WebApp，从而支持对 WebApp 社区的构建（例如，参见 www. wikipedia. com 上的 Wikipedia）。

甚至最近，WebApp 使用客户端界面技术和服务器端功能的混合来创建应用，这些应用酷似更多地与桌面应用相关的丰富的交互性。例子包括 Writely（现在的 Google Docs），一个基于 WebApp 的字处理器；Kiko（www. kiko. com），一个日历 WebApp；和 Box（www. box. net），一个在线文件存储 WebApp。

随着 WebApp 总体复杂性、交互层次、功能范围和内容范围的增加，Web 技术必须跟上步伐。Web 技术已经演化为创建逐渐消除那些在历史上限制 WebApp 全球影响的约束（对用户和维护者）的应用。

14.2 实现工具和技术

WebApp 开发的新手（在有些情况下，甚至是一个有经验的 Web 工程师）很容易在数目庞

大的 Web 应用技术和快速变化面前迷茫。考虑如下随机选择的标准和技术代表列表——所有这些在一个 WebApp 的实现中都可能是有用的。这只是可用平台和工具的一个极小的部分[㊀]。

> ……, Ajax, ASP, ColdFusion, Communique, CSS, DOM, Eclipse, Gnutella, GRDDL, Hibernate, HTML, InkML, J2EE, JavaScript, JBoss, Jigsaw, Joomla!, JSP, Kerberos, Lasso, Mambo, MathML, Maven, OWL, PHP, PICS, Plone, PNG, RDF, Rhythmyx, Scarab, Shale, Shibboleth, SMIL, SOAP/XMLP, SPARQL, Struts, SVG, SVN, TeamSite, Tomcat, Torque, Turbine, Vignette, WebLogic, WebSphere, Xforms, XHTML, XLink, XML, XPath, XPointer, XSL, and XSLT, Zope, ……

一个 Web 工程师如何了解如此丰富的技术的最新情况呢？简单地回答是避免试图去精通每种技术，而应该去了解多类可用技术。只有在必要的时候才去查看技术和工具的具体实例。

14.2.1 什么是应用框架

一个应用框架是一组用于实现一个应用基本结构的库和（或）组件。框架提供了一个基础架构和大量支持此架构的代码。对 WebApp 来说，这常常包括一些用于管理内容、访问控制系统和数据库接口、管理用户会话以及处理显示和风格的机制。

WebApp 框架覆盖的应用范围很广。简单的框架有一个单一的主要目的，如从数据库内容中生成页面。复杂的框架则解决各种特征和需要。很多更复杂的框架基于诸如模型-视图-控制器（MVC）这样的常见模型，MVC 模型在本章的前面和第 11 章中进行了简短的讨论。一个很好的基于 MVC 的框架的例子是 Apache Struts，它用于创建 Java Web 应用，支持清晰的分而治之，并且允许不同的开发者（图形设计者、内容专家、编码者）管理应用的相关部分。

框架通常基于一种主要的实现语言。例如，Struts[㊁]基于 Java，FuseBox[㊂]基于 PHP，Zope[㊃]基于 Python，而 Catalyst[㊄]基于 Perl。

通常，因为框架会影响所采用的总体设计架构，所以在使用一个应用框架时，需要早些在设计过程中作出选择。然后框架会影响总体设计，同时随着其作出具体的实现决策而指导 WebE 团队。

关于 Web 应用框架更多信息的一个好的起点可以在 http：//en. wikipedia. org/wiki/Web_application_frameworks 上的 Wikipedia 找到。

14.2.2 如何应用内容管理系统和版本控制技术

丰富的、种类繁多的内容是几乎所有 Web 应用的核心特征。这些内容是由多种用户生成和管理的，同样的内容常常被不同类的用户以多种方式进行使用。一个内容管理系统（CMS）[㊅]支持在基于 Web 的系统中对内容进行管理、编辑、转化和更新。

内容管理系统支持的功能种类繁多（参见 www. cmsmatrix. org/上的 CMS 矩阵），而且不同的内容管理系统支持不同的功能：

㊀ 正如我们在本章引言中提到的，在你读这个列表时，所列出的一些技术已经消失，而很多新技术（没有列出）已经出现。

㊁ http：//structs. apache. org/

㊂ www. fusebox. org/

㊃ www. zope. org/

㊄ www. catalystframework. org/

㊅ 内容管理系统将在第 16 章进行更详细的讨论。

- 显示模板、主题和皮肤。
- 监控、统计和内容跟踪。
- 内容预发布和部署。
- 对验证用户和控制特定内容的编辑和查看的安全管理。
- 支持众多应用：wikis、论坛、留言本、事件日历、FAQ 等。

更复杂的 CMS 提供版本控制功能（第 16 章），使 WebE 团队可以跟踪内容的变化，还允许一个应用的状态“回滚”到内容的上一个版本。关于 CMS 更多的信息，参见 en. wikipedia. org/wiki/List_of_content_management_systems 上关于 Wikipedia 的系统列表和 www. cmswatch. com 上关于 CMS Watch 的列表。

14.2.3 WebApp 怎样提供搜索能力

很多工具支持 WebApp 内的搜索能力。其中最简单的工具提供一个产生网站内容索引的索引器和一个查询索引的搜索工具。更高级的搜索工具提供对搜索进行定制、对索引策略进行配置和支持分析搜索日志的工具。

和应用框架不同，就更小范围的内容管理系统而言，搜索引擎功能相对来说是定义良好的，对应用架构没有太多影响。因此，除非搜索功能特别复杂或少见，否则把对具体搜索工具的决策推迟到后期开发过程也是合理的。更多关于搜索工具的信息可以在 www. searchtools. com 和 www. searchenginewatch. com 上找到。

14.3 开发工具和技术

有数以百计的软件开发工具可以适应到 Web 工程工作，并且几乎同样多的工具已经适应到支持基于 Web 的系统的实现。然而，这些 Web 工程工具的大多数都针对底层的实现问题。几乎没有支持 Web 工程过程（例如，沟通、需求收集、分析和设计建模）或更广的项目方面（例如，项目管理、测试、问题跟踪）的工具。

为 Web 工程而存在很多工具确实都改编自起初为传统软件应用所开发的工具。这种适应常常有些麻烦或只是对 WebApp 特定的特征进行了有限的考虑（例如，参见第 14.3.3 节中关于问题跟踪的讨论）。在其他情况下，工具被开发为搜索原型，并没有完全适应到完全商业强度的使用，因而也就不是特别健全。

然而，确实存在很多种 Web 工程工具。我们将在下面的几节中讨论一些工具类。

14.3.1 怎样获得有助于建模活动的工具

尽管传统软件的建模和设计可用的工具有很多，但是只有很少的一些是专门为 WebApp 的建模开发的。最常见的工具是那些把应用建模的一些元素集成到实际实现工具中。例如，Microsoft 的 FrontPage 和 Adobe 的 Dreamweaver（两者都是主要的 Web 设计和发布工具）每个都提供了强大的页面级设计工具，但是由 Microsoft 和 Adobe 提供的更高层结构化建模工具都较为有限。它们都允许设计者图形化地创建和编辑一个展示了基本 WebApp 结构的网站地图视图，但是这些模型在表示诸如功能工作流这样的元素方面复杂程度都相对有限，而且几乎没有对业务过程、用例或使用场景建模的能力。其他工具（特别是针对代码生成的工具）通常都有类似的限制。

一个针对 Web 建模工具的例子是 WebRatio[⊖]，它在部分方面提供了较丰富的建模能力。虽然

⊖ 参见 www. webratio. com/上关于 WebRatio 更多的信息。

WebRatio离提供商业化丰富实现的工具相距甚远，但它确实提供了对如下事物很多的支持：应用内容建模和记录、前端界面，特别是一个应用内需要的导航结构。这个工具还在一定程度上对项目文档和来自设计的应用代码生成进行了自动化——虽然大多数工具都支持自动应用生成，但是生成的代码一般需要进行相当大的调整。

Jochen Rode［Rod05］针对建模领域中针对Web的工具的限制和优点，提供了一个关于这些工具的有趣讨论。

14.3.2 专门针对WebApp的测试工具

已经开发出来的支持WebApp测试的工具有很多。关于这些工具信息的几个好的来源可以在如下地方找到：

- Software QA/Test Resource Center：www.softwareqatest.com/qatweb1.html
- Web Test Tools Directory：www.webtesttools.com/
- Applied Testing and Technology：www.aptest.com/resources.html

在你浏览这些网站所引用的很多测试工具的时候，你会注意到这些工具中的绝大多数都是针对于测试一个已经完成的WebApp，针对于评估或测试一个WebApp规格说明或设计的工具相对较少。测试工具可以归为如下几类：

- 语言验证工具（例如，HTML检查、按照模式验证XML内容、数据库内容的完整性检查）
- 导航验证（例如，检查链接是否合法、测试页面可达性）
- 性能和负载测试（例如，服务器延时、压力测试）
- 平台和环境评估（例如，多个浏览器和版本、平台内的WebApp完整性测试）
- 功能测试（例如，对客户端JavaScript和服务器端组件的验证）
- 使用分析（例如，分析访问日志和搜索查询）

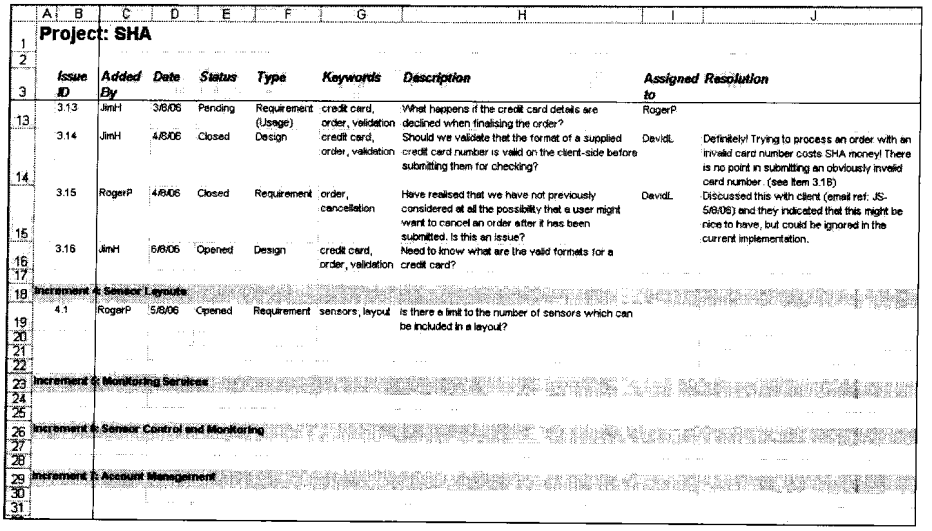
这些工具中的很多都可以把它们应用于对一个已完成WebApp在发布之前进行静态使用，或者通过对运行中WebApp不断进行的自动化分析来动态使用。例如，有很多种测试工具可以被配置为自动爬行于一个客户可见的WebApp页面，寻找诸如损坏的链接、过时的内容或者与给定模板不一致的设计等具体错误。

14.3.3 支持WebE过程管理的工具

要是存在一套支持Web项目管理、WebApp需求获取和格式化、项目范围和成本估计、问题跟踪和质量保证的工具就好了。然而，到目前为止，直接支持Web工程过程的工具还是非常少。

大多数WebApp开发者都会结合使用更普通的软件项目管理工具（例如，Microsoft的Project）和他们自己制作的定制工具。考虑如图14-1所示的例子，SafeHomeAssured.com团队使用Microsoft Excel电子数据表来跟踪开发过程中出现的问题。在这个例子中，开发团队足够小和有凝聚力，这样的一个轻量级方法足以获得出现的跟踪问题所期望的结果及其随后处理的方法。

虽然缺乏Web工程过程支持工具很不幸，但是考虑到主导大多数WebApp开发的敏捷方法，这在某种程度上来说是可以理解的。很多工具在安装和维护中都需要很大的开销，因此如果管理不慎，常常会施加与敏捷性冲突的开发开销。例如，考虑到大多数WebApp详细导航设计的快速演化，在一种设计工具中获取的那个设计的正式模型将变得相当难以维护。



Project: SHA

Issue ID	Added By	Date	Status	Type	Keywords	Description	Assigned to	Resolution
3.13	JimH	3/6/06	Pending	Requirement (Usage)	credit card, order, validation	What happens if the credit card details are declined when finalising the order?	RogerP	
3.14	JimH	4/6/06	Closed	Design	credit card, order, validation	Should we validate that the format of a supplied credit card number is valid on the client-side before submitting them for checking?	DavidL	Definitely! Trying to process an order with an invalid card number costs SHA money! There is no point in submitting an obviously invalid card number. (see item 3.1B)
3.15	RogerP	4/6/06	Closed	Requirement	order, cancellation	Have realised that we have not previously considered at all the possibility that a user might want to cancel an order after it has been submitted. Is this an issue?	DavidL	Discussed this with client (email ref: JS-5/6/06) and they indicated that this might be nice to have, but could be ignored in the current implementation.
3.16	JimH	6/6/06	Opened	Design	credit card, order, validation	Need to know what are the valid formats for a credit card?		
Increment 4: Sensor Layouts								
4.1	RogerP	5/6/06	Opened	Requirement	sensors, layout	Is there a limit to the number of sensors which can be included in a layout?		
Increment 5: Monitoring Services								
Increment 6: Sensor Control and Monitoring								
Increment 7: Account Management								

图 14-1 在 SafeHomeAssured. com 中使用一个 Excel 问题跟踪电子表格的问题跟踪示例

14.4 本章小结

在本章中，我们提供了可以应用到 WebApp 开发的工具和技术的一个简短综述。Web 开发者可用工具中的绝大多数都针对于 WebApp 的构建。到目前为止，支持 WebApp 建模或开发过程本身的工具还相对较少。

在你审查对 Web 工程工作可能有用的工具和技术时，应当考虑三个重要问题。第一个（分而治之）关注分离 WebApp 功能的需要和使你实现功能的工具。第二个问题针对你将使用的工具的来源——它们是由开发者和用户的开放社区开发的还是由一个单独商业源提供的？第三个问题考虑一个具体 WebApp 种类如何指定可以使用的工具和技术的类型。

适合所有 WebApp 类型的 WebE 技术种类繁多，包括应用框架、内容管理系统和搜索能力。开发技术包括建模和测试工具以及过程支持。

对一个 WebApp 开发者来说，真正的挑战是确定合适的建模、测试和过程支持的正式程度。我们希望看到出现轻量级的、被无缝集成到一个敏捷方法中而不会导致规格说明和设计不灵活的工具。遗憾的是，这样的工具还没有出现。

参考文献

[Rod05] Rode, J., "Web Application Development by Nonprogrammers," Virginia Polytechnic Institute, PhD dissertation, 2005, http: //scholar. lib. vt. edu/theses/available/etd-07062005-152028/unrestricted/dissertation. pdf (accessed August 13, 2007).

第 15 章　WebApp 测试

Web 工程过程通常都很紧迫。在进行沟通、计划、分析、设计以及构建时，利益相关者由于担心来自其他 WebApp 的竞争，迫于客户的要求并担心失去市场，因而迫使 WebApp 仓促上线。结果，在 WebE 过程中，技术活动常常开始较晚。例如，有时给 WebApp 测试所剩的时间很短，这可能是一个灾难性的错误。为了避免发生这种错误，你的 WebE 团队一定要确保每个 WebE 的工作产品都具有高质量。Wallace 和他的同事［WAL03］在谈到这一点时讲道：

> 不应该等到项目完成时才进行测试。在你写第一行代码之前，就开始测试。持续而有效地进行测试，你将开发出耐用得多的 Web 站点。

由于不能在传统意义上测试分析和设计模型，所以 Web 工程团队应该进行结对走查（第 5 章）和可运行的测试，目的是在 WebApp 交付给最终用户使用之前发现并改正错误。

15.1　测试概念

测试是为了发现（并最终改正）错误而运行 WebApp 的过程。事实上，由于基于 Web 的系统以及应用位于网络上，并与很多不同的操作系统、浏览器（或其他诸如机顶盒、PDA 或移动电话等接口设备）、硬件平台、通信协议及“暗中的”应用系统进行交互作用，查找错误对于 Web 工程师来说是一个重大的挑战。

为了理解 Web 工程环境中的测试目标，我们必须考虑 WebApp 质量的多种维度[⊖]。在此讨论中，我们考虑与 WebE 测试工作特别相关的质量维度，同时，我们也讨论作为测试结果所碰到的错误的特性以及为发现这些错误所采用的测试策略。

15.1.1　什么是质量维度

应该将质量纳入良好的 WebApp 分析与设计中。复杂而关键的应用组件的质量可以通过进行结对走查（第 5 章）来评估。这种评估方法在各种设计模型创建时就评估其中的不同元素，但是有些组件可能没有进行结对走查[⊜]，而是应用为发现错误和其他质量问题而设计的一系列测试步骤。评估和测试都要检查下面质量维度中的一项或多项［MIL00］：

- *内容*。在语法及语义层对内容进行评估。在语法层，对基于文本的文档进行拼写、标点以及文法方面的评估；在语义层，（所表示的信息的）正确性、（整个内容对象及相关对象的）一致性以及清晰性都要评估。
- *功能*。对功能进行测试，以发现与客户需求不一致的错误。对每一项 WebApp 功能，评估其正确性、不稳定性以及与相应实现标准（例如，Java 或 XML 语言标准）的总体符合程度。
- *结构*。对结构进行评估，以保证它正确地表示 WebApp 的内容及功能是可扩展的，并支持新内容、新功能的增加。
- *可用性*。对可用性进行测试，以保证界面支持各种类型的用户，各种用户都能够学会并

⊖ WebApp 质量已经在第 5 章中讨论过。

⊜ 在理想世界中，要对每个信息型和功能型组件进行评审。但是在现实世界中，WebE 团队几乎没有资源或时间来这样做。有些情况下，尤其是在没有使用“结对编程”方法的时候，只有关键组件在编码之前进行评审。

使用所有导航语法和语义。

- 导航性。对导航性进行测试，确保检查所有的导航语法和语义，以发现任何导航错误（例如，死链接、不合适的链接、错误链接）。
- 性能。在各种不同的操作条件、配置及负载下，对性能进行测试，以保证系统响应用户的交互并能处理极端的负载情况，而且不会出现不可接受的运行性能的降低。
- 兼容性。在客户端及服务器端，在各种不同的主机配置下通过运行 WebApp 对兼容性进行测试，目的是发现针对特定主机配置的错误。
- 互操作性。对互操作性进行测试，以保证 WebApp 与其他应用系统和数据库有正确的接口。
- 安全性。通过评定可能存在的漏洞并试图对每一个漏洞进行攻击，来对安全性进行测试。任何成功的突破尝试都被认为是一个安全漏洞。

已经开发了 WebApp 测试的策略和多种战术，用来测试这些质量维度，在第 15.1.3 节中将对这些进行讨论。

15.1.2 WebApp 环境中发生什么类型的错误

我们已经谈到，在任何软件环境下，测试的主要目的是发现错误（并改正它们）。对于成功的 WebApp 测试，所发现的错误具有许多能够对测试和错误发现有重大影响的独特特性[Ngu00]：

很多类型的 WebApp 测试发现的错误都首先表现在客户端（即通过在特定浏览器、PDA 或移动电话上实现的界面），因此，你往往看到了错误的征兆，而不是错误本身。

一个 WebApp 是在很多不同的配置以及不同的环境中实现的，因此，要在最初遇到错误的环境之外再现错误，可能是很困难的，或是不可能的。

虽然许多错误是不正确的设计或错误的编码（HTML、客户端脚本、服务器端脚本或其他程序设计语言）的结果，但是，很多错误的原因都能够追溯到 WebApp 配置。在三层架构（客户、服务器或网络本身）中跟踪和设计有关的那些错误是很困难的。某些错误应归咎于静态的操作环境（即进行测试的特定配置），而有些错误则归咎于动态的操作环境保护（即瞬间的资源负载或与时间相关的错误）。

上述这些错误的特点说明：在诊断 WebE 过程中发现的所有错误中，环境起着非常重要的作用。在某些情况（例如，内容测试）下，错误的位置是明显的；但对于很多其他类型的 WebApp 测试（例如，导航测试、性能测试、安全测试），错误的根本原因可能相当难以确定。

15.1.3 应该应用什么测试策略

WebApp 测试策略采用所有软件测试所使用的基本原理，并使用面向对象系统所常用的策略和战术。下面的步骤对此方法进行了总结：

1. 评审 WebApp 的内容模型，以发现错误。
2. 评审界面模型，保证适合所有的用例。
3. 评审 WebApp 的设计模型，找出导航错误。
4. 测试用户界面，找出表现机制和导航机制中的错误。
5. 对选择的功能组件进行单元测试。
6. 对贯穿架构的导航进行测试。
7. 在各种不同的环境配置下，实现 WebApp，并测试 WebApp 对于每一种配置的兼容性。

8. 进行安全性测试，试图攻击 WebApp 或其所处环境的弱点。

9. 进行性能测试。

10. 通过可监控的最终用户群对 WebApp 进行测试；对他们与系统的交互结果进行评估。包括内容和导航错误、可用性、兼容性、WebApp 的可靠性以及性能等方面的评估。

由于很多 WebApp 在不断演化，所以 WebApp 测试是 Web 支持人员的一项持续活动，他们使用回归测试，这些测试是从首次构建 WebApp 时所开发的测试中导出的。

15.1.4 多少测试计划是必需的

对某些 Web 开发人员来说，计划一词的使用（在任何上下文中）是一种亵渎。如我们在前面章节中讲述的那样，这些开发人员只是直接跳入构建——希望开发一个招人喜爱的 WebApp。而 Web 工程师则认识到计划工作建立了所有工作遵循的路线图。这种努力是值得的。

Splaine 和 Jaskiel [Spl01] 在他们关于 WebApp 测试的书中说道：

> 除了最简单的网站，对集成测试计划的需要很快就变得很明显。通常，在初测试中发现的最初的错误数量很大，以致并不是所有的错误都能在第一次被检测到时就能定位。这就增加了测试 Web 站点或 Web 应用的人员的负担。他们不仅要幻想新的新的测试，还必须记住以前的测试是如何进行的，用来对 Web 站点或应用系统进行可靠的重复测试，并确保已知的错误都被排除，同时没有引入新的错误。

每位 Web 工程师面临的问题是：我们如何“幻想构建的新的测试”，这些测试应该集中在什么地方？对这些问题的回答包含在测试计划中。

WebE 的测试计划确定了：1）开始测试时所使用的测试任务；2）执行每一项测试任务所生产的工作产品；3）对测试结果进行评估、记录、以及在进行回归测试时应用的方式。在某些情况下，测试计划被集成到项目计划中；而对于非常大或复杂的那些 WebApp，测试计划可以是一个独立的文档。



建立测试标准

场景：SafeHomeAssured.com 团队负责人的办公室

参与者：团队负责人和 WebE 团队的两个成员，在增量 2（详细的产品信息和下载）的交付日期临近的时候

会话：

团队成员 1：事情进展顺利。所有的产品内容都已经到位，而且每件产品都有合适的导航链接。

团队负责人：很好。我们什么时候开始测试？

团队成员 1：测试？嗯，我们还没有。我的意思是，没有太多要测试的东西。

团队成员 2：它就像第一个增量那样，几乎全是内容和显示。我们并没有为增量 1 做太多的测试。

团队负责人：但是我们确实仔细检查了所有的内容，不是吗？

[两个团队成员都点头。]

团队负责人：而且我们确保了所有的链接都能工作，甚至包括指向虚假页面的链接，对吧？

[两个团队成员都点头。]

团队负责人：所以我们做了内容和导航测试，对吧？而这个增量要复杂得多。我们有很多不同的内容，这些内容有很多不同的链接。

团队成员 1：那么准确地说，你想让我们做什么？

团队负责人：花一个小时计划这件事情，然后告诉我你们将如何“测试”这个增量。在这里，测试可能意味着检查和实际的测试，也就是测试所有的导航链接和下载链接，确保它们都能无故障地工作，测试界面以确保它对于用户来说是清晰的，你们甚至可能要做一些性能测试，确保下载不会永远无法完成，以及类似的事情。

团队成员 2：在审核的过程中，我们要在开发增量时进行了结对走查和……

团队负责人（打断）：太棒了，那将使事情顺利进行，但在这里，我认为我们应当对审核进行扩展，包含更多的检查，以防万一。

团队成员 1：好的。我们可以这样做。所以基本上来说，你希望我们确保增量 2 中所有的东西都能工作。

团队负责人：实际上并不是这样。我想让大伙做的是，在你们把增量 2 交付给用户之前，尽你们所能在增量 2 中找出尽可能多的错误。

团队成员 2（看起来有些迷茫）：但是我认为我们的工作是证明它可以工作。

团队负责人（笑）：有人曾经说过一些值得我们记住的关于测试的东西。“测试可以显示存在 bug，但是它不能确保不存在 bug。”你的工作是在用户为你发现 bug 之前找出尽可能多的 bug。

15.2 测试过程概述

Web 工程的测试，首先测试最终用户能够看到的内容和界面功能，随着测试的进行，考测试信息和功能架构以及导航的各个方面。用户可能知道这些 WebApp 元素，也可能不知道。最后，测试的焦点转到对于最终用户并不总是可见的测试技术能力——WebApp 基础设施及安装和实现问题。

图 15-1 将 WebApp 的测试过程与描述关键设计动作的设计金字塔相并列。需要注意的是，当测试流从左到右、从上到下移动时，首先测试 WebApp 设计中的用户可见元素（金字塔的顶端元素），之后对基础设施的设计元素进行测试。

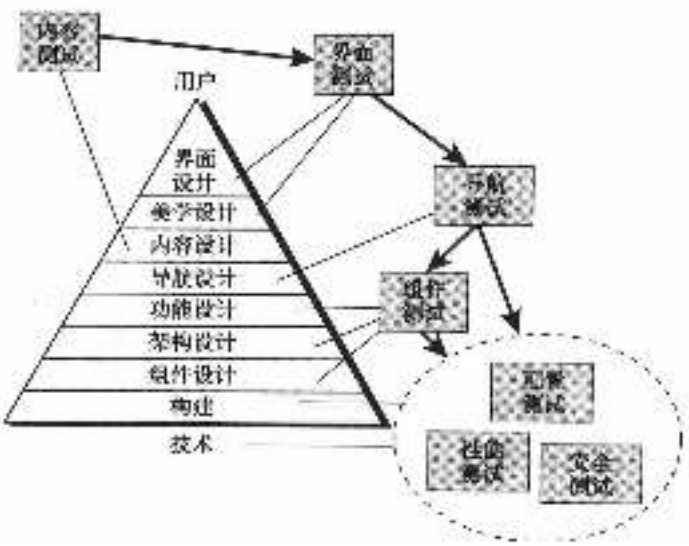


图 15-1 测试过程

内容测试（和评审）试图发现内容方面的错误，这些内容实际上表示每个 WebApp 的一部

分。这项测试活动在很多方面类似于对已写文档的审查。事实上，大型 WebApp 会有（或征召）专业审查人员来发现排字错误、语法错误、内容一致性错误、图形表示错误和交叉引用错误。除了检查静态内容方面的错误，这项测试步骤还要考虑从数据库系统（数据库系统已经被集成到 WebApp 中）所维护的数据（这些数据是数据库系统的一部分）中导出的动态内容。在很多情况下，这种形式的测试（和许多其他的类似）也许还需要在 WebApp 的生命周期中继续不断地增加新的内容。例如，对于 SafeHomeAssured. com 而言，会期望新产品不断地加入 CPI 的产品目录中——实现这些的内容需要检查。

*界面测试*验证用户界面的交互机制以及美学方面，目的是发现由于实现了糟糕的交互机制而导致的错误，或者由于不小心而导致的遗漏、不一致或歧义性。

*导航测试*在测试用例的设计中使用从分析活动中导出的用例，这些测试用例对照导航设计检查每一个使用场景。对照用例及导航路径，对在界面布局中实现的导航机制（例如，菜单条）进行测试，以保证识别并改正阻止用例完成的任何错误。

*组件测试*检查 WebApp 中的内容和功能单元。在内容架构中选择的“单元”是 Web 页面。每一个 Web 页面包括内容、导航链接以及处理元素（表单、脚本、Java 小应用程序）。在 WebApp 功能架构中的“单元”可能是定义的直接向最终用户提供服务的功能组件或使 WebApp 达到其所有能力的基础结构组件。每一个组件的测试方法与传统软件中单个模块的测试方法一样，在大多数情况下，使用面向黑盒的测试㊀。然而，如果处理程序复杂，也可能使用白盒测试㊁。除了功能测试，也对数据库能力进行测试。

随着 WebApp 架构的建立，导航测试和组件测试被用于集成测试。集成测试的策略依赖于所选择的内容架构和 WebApp 架构（第 10 章）。如果把内容架构设计成线性、网格或简单层次结构，那么，可以用一种自底向上或自顶向下的方式对 Web 页面集成测试。然而，如果使用了混合层次或网络（Web）架构，集成测试类似于在面向对象系统中所使用的方法。*基于线程的测试*㊂可用来集成响应一个用户事件所需要的 Web 页面集合（可以使用导航路径来定义合适的集合）。每一个线程都要集成，并且对其实施单独测试。应用回归测试确保没有产生副作用。*聚类测试*（cluster testing）集成一系列协作的页面（通过检查用例及导航路径来决定）。导出测试用例以发现协作中的错误。

应该在可能的范围内对 WebApp 架构的每一个元素进行单元测试。例如，在 MVC 架构（第 10 章）中，对模型、视图及控制器组件进行单独测试。在集成测试的基础上，对穿过这些元素的控制流及数据进行详细评估。

*配置测试*试图发现特定的客户或服务器环境中的错误。生成一个交叉引用矩阵，定义所有可能的㊃操作系统、浏览器㊄、硬件平台以及通信协议，然后进行测试，找出每一种可能与配置相关的错误。

*安全测试*将设计的一系列测试合并起来，攻击 WebApp 及其环境中的弱点，目的是发现可能的安全性破坏。

*性能测试*包括一系列测试，设计这些测试用来评估：1）WebApp 的响应时间及可靠性如何受增长的用户通信量和功能复杂性的影响？2）哪些 WebApp 组件与性能降级有关？哪些使用特

㊀ 黑盒测试的设计并不需要组件内部的知识，只需要传给组件接口的信息来设计测试。
㊁ 白盒测试技术利用组件内部结构，对于功能组件来说，是指组件内部的处理逻辑。
㊂ 基于线程的测试集成一系列用于响应一个用例或导航路径的内容或功能组件（或 Web 页面）。
㊃ WebApp 的关键任务和环境将面临的特定方面的可能性一起表示这个方面将被测试与否。
㊄ 对于 HTML 和 JavaScript 的翻译，浏览器在实现其自身微妙的不同“标准”方面很糟糕。

点发生了降级的发生？3）性能降级是如何影响整个 WebApp 的目标及需求的？

任务



WebApp 测试

1. 评审利益相关者的需求。
   - 标识关键用户目标和动机。
   - 对每类用户的用例进行评审。
2. 建立优先级，以确保每一个用户目标都将被充分地测试。
3. 通过清楚要实施的测试类型定义 WebApp 测试策略。
4. 开发一个测试计划。
   - 规定测试进度，并对每个测试分配职责。
   - 指定自动化测试工具。
   - 规定每一类测试的验收标准。
   - 详细说明缺陷跟踪机制。
   - 定义问题报告机制。
5. 执行单元测试。
   - 评审内容的语法和语义错误。
   - 评审内容的许可性权限。
   - 测试接口机制的正确操作。
   - 测试每一个组件（例如，脚本），确保正确的功能。
6. 进行集成测试。
   - 利用用例来测试界面的语义。
   - 实施导航测试。
7. 进行配置测试。
   - 评估客户端的配置兼容性。
   - 评估服务器端的配置。
8. 进行性能测试。
9. 进行安全性测试。

15.3 内容测试

WebApp 内容中的错误可以小到简单的印刷错误，或者大到不正确的信息、不合适的组织，或者违背知识产权法。内容测试试图在用户碰到这些问题及很多其他问题之前就发现它们。

内容测试结合了评审和可运行的测试用例的生成。采用评审来发现内容中的语义错误。可运行的测试用于发现内容错误，这些错误可被跟踪到动态导出的内容，这些内容来从一个或多个数据库中获取的数据驱动。

15.3.1 内容测试的目标是什么

内容测试有三个重要的目标：1）找出基于文本的文档、图形表示和其他媒体中的语法错误（例如，打字错误、文法错误）；2）找出当导航发生时所展现的任何内容对象中的语义错误（即信息的精确性和完备性方面的错误）；3）找出展示给最终用户的内容的组织或结构方面的错误。

为了达到第一个目标，可以使用自动拼写和语法检查工具。然而，很多语法上的错误可以逃脱这种工具的检查，必须由审查人员（测试人员）人为发现。如我们在前面章节中所讲的那样，文本审查是发现语法错误的唯一最好的方法。

语义测试关注于在每一个内容对象内所显示的信息。评审人员（测试人员）必须回答以下问题：

- 信息确实是最新且准确的吗？
- 信息简洁扼要吗？
- 内容对象的布局对于用户来说容易理解吗？
- 嵌入在内容对象中的信息易于发现吗？

- 对于所有从其他地方导出的信息，是否提供了合适的引用？
- 显示的信息是否是内部一致的？与其他内容对象中所显示的信息是否一致？
- 内容是否具有冒犯性？是否容易误解？或者是否会引起诉讼？
- 内容是否侵犯了现有的版权或商标？
- 内容是否包括补充现有内容的内部链接？链接正确吗？
- 内容的美学风格是否与界面的美学风格相矛盾？

对于大型的 WebApp（包含成百上千个内容对象）来说，要获得所有这些问题的答案可能是一项令人畏惧的任务。然而，未发现的语义错误将动摇用户对 WebApp 的信任，并且会导致基于 Web 的应用的失败。

在内容测试期间，要对内容架构的结构和组织进行测试，以确保将所需要的内容以合适的顺序和关系展现给最终用户。例如，SafeHomeAssured. com WebApp 显示了关于传感器的多种信息，其中传感器是安全和监视产品的一部分。内容对象提供描述信息、技术规格说明、照片和相关的信息。SafeHomeAssured. com 内容架构的测试试图发现这种信息的表示方面的错误（例如，用传感器 Y 的照片来描述传感器 X）。

15.3.2 数据库测试如何用于验证内容

现代的 Web 应用系统不只是展示静态的内容对象。在很多应用领域中，WebApp 要与复杂的数据库管理系统连接，并构建动态的内容对象，这种对象是使用从数据库中获取的数据实时创建的。

例如，用于金融服务的 WebApp 能够产生某种特殊权益（例如，股票、保证金或共有基金）的复杂的文本信息、表格信息和图形信息。当用户已经申请了某种特殊的权益信息之后，就会动态创建表示这种信息的复合内容对象。为了完成此任务，需要以下步骤：1）查询大型权益数据库；2）从数据库中抽取相关的数据；3）抽取的数据必须被组织为一个内容对象；4）将这个内容对象（代表由某个最终用户请求的定制信息）传送到客户环境显示。每一个步骤的结果都可能发生错误，并且一定会发生。数据库测试的目标是发现这些错误。然而，WebApp 的数据库测试会由于以下多种原因而变得复杂：

1. 原始客户端请求的信息很少被表示成能输入到数据库管理系统（DBMS）的形式（例如，结构化查询语言 SQL）。因此，应该设计测试，找出在将用户的请求翻译成能够被 DBMS 处理的格式的过程中所产生的错误。

2. 数据库可能离加载 WebApp 的服务器很远。因此，应该设计测试，找出 WebApp 和远程数据库之间通信中所存在的错误[⊖]。

3. 从数据库中获取的原始数据一定要传递给 WebApp 服务器，并且这些原始数据要被正确地格式化，以便随后传递给客户端。因此，应该设计测试，证明 WebApp 服务器接收到的原始数据的有效性，并且还要生成另外的测试，证明转换的有效性，将这种转换应用于原始数据，能够生成有效的内容对象。

4. 动态内容对象一定要以能够显示给最终用户的形式传递给客户端。因此，应该设计一系列的测试，1）找出内容对象格式方面的错误；2）测试与不同的客户端环境配置的兼容性。

考虑这 4 种因素，对图 15-2 中记录的每一个“交互层”[Ngu01] 都应该应用测试用例的设计方法。测试应该保证：1）有效信息通过界面层在客户与服务器之间传递；2）WebApp 正确地

⊖ 当遇到分布式数据库或者需要访问数据仓库时，这些测试可能变得很复杂。

处理脚本，并且正确地抽取或格式化用户数据；3）用户数据被正确地传递给服务器端的数据转换功能，此功能将合适的查询格式化（例如，SQL）；4）查询被传递到数据管理层[⊖]，该层与数据库访问程序（很可能位于另一台机器）通信。

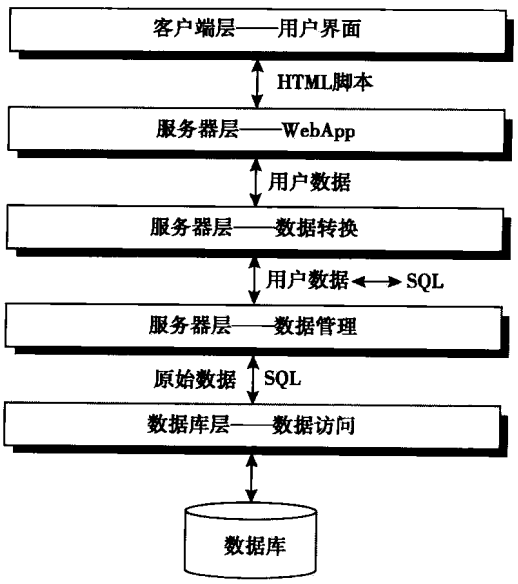


图 15-2 交互层次

通常使用可复用的组件来构造图 15-2 所示的数据转换层、数据管理层和数据库访问层，这些可复用的组件都分别进行了验证，并且被打成一个包。如果是这种情况，WebApp 的测试则集中在图 15-2 所示的客户端层与前两个服务器层（WebApp 和数据转换）之间交互的测试用例的设计。

应该对用户界面层进行测试，确保对每一个用户查询都正确地构造了 HTML 脚本，并且正确地传输给服务器端。还应该对服务器端的 WebApp 层进行测试，确保能够从 HTML 脚本中正确地抽取出用户数据，并且正确地传输给服务器端的数据转换层。应该对数据转换功能进行测试，确保创建正确的 SQL，并且传给合适的数据管理组件。

对合理地设计这些数据库测试时需要了解的主要技术进行详细讨论超出了本书的范围，感兴趣和需要这一领域的读者可以参见［Sce02］、［Ngu01］和［Bro01］。

15.4 用户界面测试

在 WebE 过程中，需要在三个不同阶段点对 WebApp 的用户界面进行验证与确认。在沟通（第 4 章）和建模阶段（第 7 章），对界面模型进行评审，确保与用户需求及分析模型的其他元素相一致；在设计阶段（第 9 章），对界面设计模型进行评审，确保已经达到了为所有用户界面建立的通用质量标准，并且正确描述了特定于应用系统的界面设计问题。由于用户交互是通过界面的语法和语义来表示的，所以，在测试阶段，重点转移到特定于应用系统的用户交互方面的执行。另外，测试提供了对可用性的最终评估。

15.4.1 界面测试策略

界面测试的总体测试策略是：1）找出与特定的界面机制相关的错误（例如，未能正确执行

⊖ 数据管理层通常合并了 SQL 调用层接口（SQL-CLI），如 Microsoft OLE/ADO 或 Java 数据库连接（JDBC）。

菜单链接的错误或者输入数据格式的错误）；2）找出界面实现导航语义方式的错误、WebApp 的功能性错误或内容显示错误。为了实现此策略，必须完成以下一些战术步骤：

- **对界面要素进行测试，确保设计原则、美学和相关的可视化内容对用户有效，并且没有错误**。要素包括字体、颜色、框架、图片、边界、表以及在 WebApp 运行过程中所产生的相关界面元素。
- **采用与单元测试类似的方式测试单个界面机制**。例如，设计测试用例对所有的表单、客户端脚本、动态 HTML、脚本、流内容以及应用特定的界面机制（例如，电子商务应用系统中的购物车）进行测试。在很多情况下，测试可以专门集中在这些机制中的一种（“单元”），而不包括其他界面要素和功能。
- **对于特殊的用户类别，在用例或导航路径的环境中测试每一种界面机制**。这种测试方法与集成测试类似，因为当界面机制被集成到一起使用例或导航路径可执行时，才能够进行测试。
- **与选择用例及导航路径有所不同，此方法要对全部的界面进行测试，找出界面的语义错误**。这种测试方法类似于确认测试，因为其目的是证明与特定的用例或导航路径语义相一致。正是在这个阶段，进行一系列的可用性测试。
- **在多种环境（例如，操作系统、浏览器）中对界面进行测试，确保其兼容性**。实际上，可以将这一系列测试看成是配置测试的一部分。

15.4.2 如何测试界面机制

当用户与 WebApp 交互时，通过一种或多种界面机制发生交互，这些机制包括［Spl01］：

- **链接**。把用户链接到一些其他内容对象或功能的导航机制。
- **表单**。包含由用户填充的空白区域的结构化文档。区域中的数据作为一个或多个 WebApp 功能的输入。
- **客户端脚本**。脚本语言命令编写的，通过表单或其他用户交互处理的程序指令集。
- **动态 HTML**。提供对由客户端通过脚本或层叠样式表（CSS）控制的内容对象访问。
- **客户端弹出窗口**。无需用户交互而弹出的小窗口[⊖]。这些窗口可以是面向内容的，也可能需要某些形式的用户交互。
- **服务器端脚本**。服务器端脚本实现允许一个 Web 服务器与用户动态交互的方法（例如，一个包含一些表单的 WebApp 可能使用 CGI 脚本、Servlet 或 ASP 脚本的 WebApp，用来在用户提交表单时处理其中包含的数据）。
- **流内容和推送内容**。流内容（streaming content）是指当素材（通常是音频或视频）以一种仍然在下载过程中就可以播放的方式下载（不必等到全部内容下载完）。推送内容（push content）是指内容对象自动从服务器端下载而不是等待客户端的请求。
- **Cookie**。由服务器发送而由客户端浏览器存储的作为特定用户交互结果的一块数据。数据的内容是 WebApp 特定的（例如，用户识别数据或一个用户已经选择的购物项的列表）。
- **应用特定的界面机制**。包括一个或多个“宏”界面机制，如购物车、信用卡处理或运费计算器。

在下面的段落中，对每一种界面机制，我们简要介绍一下测试时需要考虑的内容：

⊖ 弹出窗口的使用已经变得很普遍，并且很多用户都对它们很敏感。应该明智地使用它们。

链接。对每一个导航链接进行测试，确保获得正确的内容对象或功能[⊖]。构建与界面布局（例如，菜单条、索引项）相联系的所有链接列表，然后分别运行每一个链接。另外，一定要检查每一个内容对象内的链接，以找出错误的 URL 或者指向不正确的内容对象或功能的链接。最后，应该对指向外部 WebApp 的链接进行精确性测试，并且对其进行估计，确定随着时间的推移这些链接将变得无效的风险有多大。

表单。在宏观层次上进行测试，以确保：1）对表单中的标识域给出正确标记，并且为用户可视化地标识出强制域；2）服务器接收到了表单中包括的所有信息，并且在客户端与服务器之间的传输过程中没有数据丢失；3）当用户没有从下拉菜单或按钮组中进行选择时，使用合适的默认项；4）浏览器功能（例如，“回退”箭头）没有破坏输入到表单中的数据；5）对输入数据执行错误检查的脚本工作正常，并且提供了有意义的错误标志信息。

在更具体的层次上，测试应该确保：1）表单域有合适的宽度和数据类型；2）表单建立了合适的安全措施，防止用户输入的文本字符串长度大于某个预先定义的最大值；3）对下拉菜单中的所有合适的选项进行详细说明，并按照对最终用户有意义的方式排序；4）浏览器“自动填充”特性不会导致数据输入错误；5）Tab 键（或一些其他键）能够使输入焦点在表单域之间正确移动。

客户端脚本。当脚本运行时，使用黑盒测试找出处理中的一些错误。由于脚本输入通常来自作为表单处理的组成部分而提供的数据，这些测试通常与表单测试联合进行。应该进行兼容性测试，确保所选择的脚本语言在支持 WebApp 的环境配置中工作正常。另外，还要测试脚本本身。Splaine 和 Jaskiel［Spl01］建议：“应该保证公司的［WebApp］标准明确了客户端（和服务器端）脚本使用的首选语言和脚本语言的版本。”

动态 HTML。运行包含动态 HTML 的每个 Web 页面，确保动态显示正确。另外，应该进行兼容性测试，确保动态 HTML 在支持 WebApp 的环境配置中工作正常。

弹出窗口。进行一系列测试，以确保：1）弹出窗口具有合适的大小和位置；2）弹出窗口没有覆盖原始的 WebApp 窗口；3）弹出窗口的美学设计与界面的美学设计相一致；4）附加到弹出窗口上的滚动条和其他控制机制被正确定位，并具有所需的功能。

服务器端脚本。一旦接收到经过验证的数据，黑盒测试的侧重点就集中在数据的完整性（当数据被传递给服务器端脚本时）和脚本处理。此外，进行性能测试，确保服务器端的配置符合脚本多重调用的处理要求［Spl01］。

推送内容。测试应该展现用户能够在必要时请求和限制推送内容，同时展示内容交付是适时的。

流内容。测试应该证明流动数据是最新的，并且显示正确，能够无错误地暂停，而且很容易重新启动。

Cookie。服务器端的测试和客户端的测试都需要。在服务器端，测试应该确保一个 cookie 被正确构造（包含正确的数据），并且当请求特定的内容和功能时，此 cookie 能够正确地传输到客户端。此外，测试此 cookie 是否具有合适的持续性，确保有效日期正确。在客户端，通过测试来确定 WebApp 是否将已有的 cookie 正确地附到了特定的请求上（发送给服务器）。

应用特定的界面机制。测试是否与界面机制定义的功能和特性清单相符合。例如，Splaine 和 Jaskiel［Spl01］为电子商务应用中所定义的购物车功能提出了下面的检查单：

- 对能够放置到购物车中的物品的最小数量和最大数量进行边界测试，还要测试刚好大于

⊖ 这些测试可以作为界面测试或者导航测试的一部分。

最大和小于最小的数量。

- 对一个空的购物车的“结账”请求进行测试。
- 测试从购物车中正确地删除一件物品。
- 测试一次购买操作是否清空了购物车中的内容。
- 测试购物车内容的持久性（这应该作为客户需求的一部分进行详细说明）。
- 如果用户请求显示已经保存的内容，测试 WebApp 将来是否能够记起购物车的内容（假设没有购买活动发生）。

一个 WebE 团队可以为这些指南开发一个界面测试检查单，适应到要测试的特定功能。这个检查单用于指导团队设计和执行测试。

15.4.3 如何测试界面语义

一旦对每一个界面功能都已经进行了“单元”测试，就可以将界面测试的焦点转移到界面的语义。界面的语义测试“评价设计在考虑用户因素、提供清楚的指导、传递反馈并保持语言与方法的一致性方面做得如何”[Ngu01]。

全面回顾一下界面设计模型，我们能够得到前面的段落中所包括的问题的部分答案。然而，一旦实现了 WebApp，就应该对每一个用例场景（针对每一类用户）进行测试。本质上，用例就变成了设计测试序列的输入。测试序列的目的是找出那些妨碍用户获得与用例相关的目标的错误。

在对每一个用例进行测试的过程中，Web 开发团队需要维护一份检查单，确保每一个菜单项都至少被运行一次，并且内容对象中的每一个嵌入的链接都已经被使用。此外，测试序列应该包括不适当的菜单选择和链接使用，目的是确定 WebApp 是否提供了有效的错误处理和恢复。

15.5 可用性测试

可用性测试评价用户能够在多大程度上与 WebApp 进行有效交互，WebApp 在多大程度上指导用户行为，提供有意义的反馈，并坚持一致的交互方法。从这个意义上说，可用性测试[⊖]与界面语义测试（在第 15.4 节中讨论过）类似。可用性检查和测试不是集中在某个交互目标的语义上，而是要确定 WebApp 界面在多大程度上使用户的生活变得轻松[⊜]。

可用性测试可以由 Web 开发团队设计，但是测试本身由最终用户进行。在测试时，可以采用下面的步骤 [Spl01]：

- 定义一组可用性测试类别，并确定每类测试的目标。
- 设计测试，使其能够评估每个目标。
- 选择将执行测试的参与者。
- 当进行测试时，指导参与者与 WebApp 的交互。
- 开发一种机制来评估 WebApp 的可用性。

可用性测试可能发生在多种不同的抽象级别：1）评估特定的界面机制（例如，表单）的可用性；2）评估所有 Web 页面（包括界面机制、数据对象以及相关的功能）的可用性；3）考虑整个 WebApp 的可用性。

⊖ 可用性测试的更多有价值的指导可以参考 www. ahref. com/guides/design/199806/0615jef. html。

⊜ 这种问题常用术语“用户友好性”来表示。当然，问题在于不同的用户对于“友好”界面的感觉可能是完全不同的。

可用性测试的第一步是确定一组可用性类别，并对每一个类别建立测试目标。下面的测试类别和目标（以问题的形式进行描述）举例说明了这种方法[⊖]：

交互性。交互机制（例如，下拉菜单、按钮、指针）容易理解和使用吗？

布局。导航机制、内容和功能放置的方式是否能让用户很快地找到它们？

可读性。文本是否很好地进行了编写，并且是可理解的[⊜]？图形表示是否直观和容易理解？

美学。布局、颜色、字体和相关的特性是否使 WebApp 易于使用？用户对 WebApp 的观感是否“感觉舒适”？

显示特性。WebApp 是否使屏幕的大小和分辨率得到了最佳使用？

时间敏感性。是否能够及时使用或获取重要的要素、功能和内容？

个性化。WebApp 是否能适应多种用户或个别用户的特殊要求？

可访问性。残疾人是否可以使用该 WebApp？

对于上面的每一种可用性，都需要设计一系列测试。在某些情况下，“测试”可以是对 Web 页面的可视化审查；而在有些情况下可以重新执行界面的语义测试，但在下面的实例中，可用性是极为重要的。

作为一个例子，我们考虑对交互和界面机制进行可用性评估。Constantine 和 Lockwood [Con03] 建议应该对下列界面要素进行可用性评审和测试，它们是：动画、按钮、颜色、控件、对话、域、表单、框架、图形、标签、链接、菜单、消息、导航、页面、选择器、文本和工具条。当评估每个要素时，可以由执行测试的用户对其进行定性分级。图 15-3 描述了用户可能选择的一系列评估“级别”。这些级别可以应用于每个单独的要素、所有的 Web 页面或者整个 WebApp。

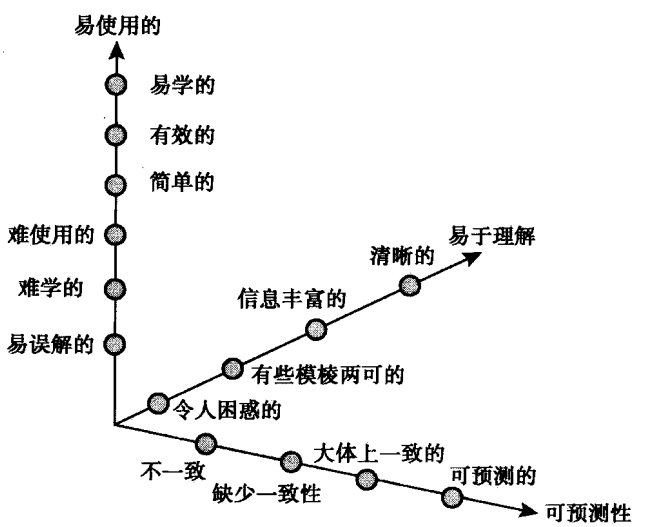


图 15-3 可用性的定性评估

⊖ 对于其他的可用性问题，参见“可用性”框注。

⊜ FOG 可读性指数和其他工具可以用来定量地评估可读性。更多细节参见 http://en.wikipedia.org/wiki/Readability_test。

信息

可用性

Constantine［Con95］指出，可用性并非产生于美学、交互机制的发展水平，或者内置的界面智能等方面；而是，当界面的架构适合于将要使用这些界面的用户的需求时，才获得可用性。

确定你所构建的系统是否可用的唯一办法就是进行可用性评估或测试。观察用户与系统的交互，同时回答下列问题［Con95］：

- 在没有连续的帮助或用法说明的情况下，系统是否便于使用？
- 交互规则是否能够帮助一个有经验的用户工作得更加有效率？
- 交互机制是否能在用户知识变得更加有经验时变得更灵活？
- 系统是否已经过调试，使之适应其运行的物理环境和社会环境？
- 用户是否意识到系统的状态？用户是否一直知道他们所处的位置？
- 界面是否是按照一种合理并且一致的方式来构建？
- 交互机制、图标和程序是否与界面相一致？
- 交互是否能够提前发现错误并帮助用户修正它们？
- 界面是否能够容错？
- 交互过程是否简单？

如果上述每个问题的回答都是肯定的，那么我们就可以认为这个系统是可用的。

可用性系统带来的诸多好处在于［Dan99］：提高销售量和用户满意度，具有竞争优势，在媒体中获得良好的评论，获得良好的口碑，降低支持的费用，减少文档开销，减少来自不满意用户的投诉。



界面和可用性测试

场景：CPI公司的Web工程区域

参与者：SafeHomeAssured.com团队的年为结对工作的两个成员，他们正在评估SafeHomeAssured.com的增量4（空间布局和安全系统设计）的界面和可用性测试的结果。

会话：

团队成员2（看起来闷闷不乐）：伙计们，界面测试很顺利，但我们在可用性测试中出问题了。

团队成员1（看起来同样闷闷不乐）：是的，看看支持增量4的用户评价吧。我们平均只有拿到5分中的2分。我已经花了一些时间和参与的用户进行交谈，看起来效果并不太好。

团队成员1：你知道，我担心生们在理解如何构建布局的时候会有麻烦［看看一个和图9-4等价的实际界面］，确实是这样。

团队成员2：对我来说，使用它看起来很容易。

团队成员1：哥们儿，那不重要；对他们来说，它并不是明显的。看看这些评论：

"无法明白如何开始。"

"这太难了。"

"我在对空间进行布局的时候，很容易出现错误。而且老是引起很多错误。"

"我试着保存我的布局，但是不知道稍后如何获取它。"

"这太糟了。"

团队成员2（有些畏缩）：哦！至少所有的链接、功能和基础架构能够很好地工作。

团队成员1（苦笑）：是这样的，但是用户讨厌它，所以我们就被夹在中间了。

团队成员2：也许不是这样的。看起来大部分问题都已经十复杂性，我怀疑很多是因为用户不习惯像这样做这些事情，所以它对于他们来说根本就不直观。因为用户要做的事情，功能性界面有复杂的，所以我们不可能

通过使它变得更简单来解决问题。如果我们做一个动画教程来展示如何进行所有的关键绘制和布局管理功能，以帮助用户理解，会怎么样呢？我们同时还能有 10 或 12 个事先定义的建筑平面图（你知道，对普通的房屋设计）这样用户就不必做布局了。

团队成员 1：这些想法可能有帮助，但是在我们做工作之前，我们必须首先让用户来运行它。同时我们将不得不拖延这个增量的部署日期。

团队成员 2：伙计，我对这不满意。

团队成员 1：如果我们把它发布出去，然后获得数以千计的类似评论［挥舞着记录着可用性测试评论表的纸］，那么你将更加不高兴。所以这些评论还是少一些好。

团队成员 2（叹息）：我想是这样。我们开始吧……又有事情做了。

15.6 兼容性测试

一定要在不同的环境中运行 WebApp。不同的计算机、显示设备、操作系统、浏览器和网络连接速度都会对 WebApp 的运行造成很大的影响。每一种计算配置都可能使客户端的处理速度、显示分辨率和连接速度有所不同。操作系统反复无常的行为可能导致 WebApp 处理上的问题。不管 WebApp 中 HTML、CSS 和 JavaScript 标准化程度如何，不同的浏览器有时会产生稍微不同的结果。对特殊的配置所需要的插件可能是已有的，也可能还没有。

在某些情况下，小的兼容性问题显得不是很严重；而在有些情况下，就可能遇到严重的错误。兼容性问题导致的最常见的问题是可用性差。这可能来自如下方面：下载速度可能变得让人无法接受，缺少所需要的插件可能使内容难以获取，浏览器的不同可能会较大地改变页面的布局，字型可能会被改变且变得难以辨认，或者表单可能被错误地组织。*兼容性测试*试图在 WebApp 上线前发现这些问题。

兼容性测试的第一步是定义一组“通常遇到”的客户端计算配置和它们的变型。实际做法是创建一种树结构，并在上面标识每一种计算平台、典型的显示设备、此平台支持的操作系统、可用的浏览器、可靠的 Internet 连接速度以及类似信息。下一步，WebE 团队导出一系列的兼容性确认测试，可以从现有的界面测试、导航测试、性能测试和安全性测试中导出。这些测试的目的是找出配置的差异所导致的错误和运行问题。

15.7 组件级测试

组件级测试（component-level testing）也称为*功能测试*（function testing），它集中于一系列的测试，试图找出 WebApp 功能方面的错误。每一个 WebApp 功能都是一个软件模块（用多种程序设计语言或脚本语言中的一种实现的），并且可以用黑盒（以及在某些情况下的白盒）技术对其进行测试。

组件级测试用例通常由表单级的输入而驱动。一旦定义了表单数据，用户就可以选择按钮或其他控制机制来启动运行。下面是典型的测试用例设计方法［Pre05］：

- **等价类划分**。将功能的输入域划分为输入（种）类，可以从这些输入类中导出测试用例。通过对输入表单进行评估，可以决定哪些数据类与功能有关。对于每个输入类，都导出它的测试用例并运行，而其他的输入类保持不变。例如，一个电子商务应用系统可能实现一个计算运输费用的功能。在通过表单提供的多种运输信息中，有用户的邮政编码。可以设计测试用例，通过给定多种邮政编码的值，这些值可能有不同种类的错误，（例

如，不完整的邮政编码、不正确的邮政编码、不存在的邮政编码和错误的邮政编码格式）尝试找出邮政编码处理中的错误。

- **边界值分析**。对表单数据的边界进行测试。例如，前面提到的运费计算功能需要指出产品运输所需要的最大天数，在表单中记录的最少天数是2天，最大天数是14天。然而，边界值测试可能输入值0、1、2、13、14和15，用来确定功能如何对有效输入边界之内、之外以及边界点的数据做出反应[⊖]。
- **路径测试**。如果功能的逻辑复杂性较高[⊜]，可以使用路径测试来确保程序中的每条独立路径都已经被执行。

除了这些测试用例设计方法，还可以使用称为强制错误测试（forced error testing）[Ngu01]的技术导出测试用例。这些测试用例故意使WebApp组件进入错误条件，目的是找出在错误处理过程中发生的错误（例如，不正确或不存在的错误提示信息、由于错误的发生导致WebApp失效、错误的输入导致错误的输出和与组件处理有关的副作用）。

每个组件级测试用例详细说明了所有的输入值和由组件提供的预期的输出。可以将测试过程中产生的实际输出数据记录下来，供将来的支持和维护阶段参考。

在很多情况下，WebApp功能的正确运行依赖于与数据库的正确接口，其中数据库可能位于WebApp的外部。因此，数据库测试是组件测试中不可分割的一部分。Hower [How97] 对此进行讨论时这样写道：

> 数据库驱动的网站可能包括Web浏览器、操作系统、插件应用、通信协议、Web服务器、数据库、[脚本语言]程序……强化安全以及防火墙之间的复杂交互。这种复杂性使得不可能测试每种可能的依赖和使站点发生错误的每件事情。同时，一般Web站点开发项目的时间进度安排都很紧张，所以最好的测试方法是进行风险分析，以确定测试工作该集中在哪里？风险分析应该包括考虑测试环境与实际产品环境匹配的紧密程度。测试场景完全模仿真实的用户、Internet连接、调制解调器、通信、硬件、客户端、负载、数据和数据库表大小等吗？这些不同很重要吗？理想情况下，测试和产品环境将是一样的，但是预算约束常常阻止这一相同性，并且风险是这些不同可能会扭曲测试结果，而且使得一些测试类型变得无关联。风险分析中其他方面的考虑一般还包括：
>
> - Web站点中的哪种功能对于其目的是最关键的？
> - 网站的哪些区域与数据库的交互量最大？
> - 站点的CGI、Applet和ActiveX等组件的哪些方面最复杂？
> - 哪些类型的问题会导致最多的抱怨或最坏的影响？
> - 网站的哪些部分将最受欢迎？
> - 网站的哪些方面的安全性风险最高？

当为WebApp组件以及相关的数据库功能设计测试用例时，应该考虑Hower所讨论的与风险相关的所有问题。

15.8 导航测试

用户在WebApp中的游历与访问者在商店或博物馆中漫步很相似。有很多路径可走，有很多

⊖ 在这种情况下，一个较好的输入设计会排除潜在的错误。最大天数可以从下拉菜单中选择，从而排除用户指定禁止的输入。

⊜ 逻辑复杂性可以通过计算算法的环路复杂性来确定。详细内容参见[Pre05]。

站可以停下来，有很多事情去学习和观看，可以启动很多活动，并且可以做决策。如我们所讨论的那样，每个访问者到来时都有一系列的目标，在这个意义上，这种导航过程是可预测的。同时，导航过程又可能是无法预测的，因为访问者受到他所看到的或学到的某事的影响，可能选择一条路径或启动一个动作，而这对于最初的目标来说并不是典型的路径或动作。导航测试的工作是：1）确保允许 WebApp 用户经由 WebApp 游历的机制都是起作用的；2）确认每个导航目标都能够被合适类型的用户获得。

15.8.1 如何测试导航语法

实际上，导航测试的第一个阶段在界面测试期间就开始了。应该对各导航功能进行测试，以确保每个导航都完成了预计的功能。应该对下面的每个导航功能进行测试［Spl01］：

- **导航链接**。这些链接包括 WebApp 的内部链接，指向其他 WebApp 的外部链接以及特定 Web 页面中的锚。每个连接都应该被测试，以确保选择链接时能够获得正确的内容和功能。
- **重定向**。当用户请求一个不存在的 URL，选择一个目标地址已经被移走或者名字已经被改变的链接时，就会用到这些重定向的链接。应该给用户显示一条提示信息，并且将导航重定向到另一页（例如，主页）。通过请求不正确的内部链接或外部 URL，并评价 WebApp 如何处理这些请求，对重定向进行测试。
- **书签**。虽然书签是浏览器功能，但是还是应该对 WebApp 进行测试，确保当创建一个书签时，能够抽取出有意义的页标题。
- **框架和框架集**。每个框架包含特定的 Web 页面内容。一个框架集包含多个框架，并且可以使多个 Web 页面同时显示。由于框架和框架集彼此之间可以嵌套，应该对这些导航和显示机制进行内容的正确性、布局及大小的适合性、下载性能和浏览器性能方面的测试。
- **站点地图**。站点地图给所有 Web 页面提供一个完整的内容表格。应该对站点地图的每个入口进行测试，以确保链接引导用户到达合适的内容和功能。
- **内部搜索引擎**。复杂的 WebApp 通常包括成百上千的内容对象。内部（本地）搜索引擎允许用户在 WebApp 中搜索关键字，发现所需要的内容。搜索引擎测试确认搜索的精确性和完备性、搜索引擎的错误处理特性以及高级的搜索特性（例如，在搜索域中使用布尔操作符）。

前面已经提到的某些测试可以由自动工具执行（例如，链接检查），而有些要手工设计和执行。导航测试的目的始终是确保在 WebApp 上线之前发现导航功能方面的错误。

15.8.2 如何测试导航语义

随着导航设计的进行，你创建“一组信息和相关的导航结构，这些导航结构相互协作，以完成相关的用户需求的子集时”［Cac02］。这些有时被称作*导航语义单元*（navigation semantic unit，NSU），并且定义了一系列连接导航结点（例如，Web 页面、内容对象或功能）的导航路径（称为“导航路”）。作为一个整体，每个 NSU 允许用户获得特殊的需求，这种特殊的需求针对某类用户，由一个或多个用例定义。导航测试应检查每个 NSU，以确保能够实现这些需求。在测试每个 NSU 时，WebE 团队一定要回答下面的问题：

- 此 NSU 是否没有错误地全部完成了？
- 在为此 NSU 定义的导航路径的上下文中，（为一个 NSU 定义的）每一个*导航结点*（navigation node）是否都是可达的？
- 如果使用多条导航路径都能完成此 NSU，每一条相关的路径是否都已经被测试？

- 如果使用用户界面提供的指南来帮助导航，当进行导航时，方向正确并可理解吗？
- 是否具有返回到前一个导航结点以及导航路径开始位置的机制（不同于浏览器的“回退”箭头）？
- 大型导航结点（例如，一个长的 Web 页面中的锚点链接）中的导航机制工作正常吗？
- 如果一个功能在一个结点上运行，并且用户选择不提供输入，NSU 的剩余部分能完成吗？
- 如果一个功能在一个结点上运行，并且在功能处理时发生了一个错误，NSU 能完成吗？
- 在到达所有结点之前，是否有办法终止导航？然后又能返回到导航被终止的地方，并从那里继续？
- 从站点地图可以到达每一个结点吗？结点的名字对最终用户有意义吗？
- 如果可以从某个外部信息源到达 NSU 中的一个结点，可能推移到导航路径的下一个结点吗？可能返回到导航路径的前一个结点吗？
- 当运行 NSU 时，用户知道他在内容架构中所处的位置吗？

类似界面测试和可用性测试，导航测试应该由尽可能多的不同的支持者进行。测试的早期阶段由 Web 工程师进行，但后来的测试应该由其他的项目利益相关者、独立的测试团队进行，最后应该由非技术用户进行，目的是彻底检查 WebApp 导航。



导航测试

场景：CPI 公司的 Web 工程区域

参与者：作为结对工作的 SafeHomeAssured.com 团队的两个成员，他们正在计划 SafeHomeAssured.com 的增量 4（空间布局和安全系统设计）的导航测试

会话：

团队成员 1：对这个增量来说，确实没有太多的导航。本质上，它大部分都是和创建空间布局并获得监控设备位置的推荐相关的功能。

团队成员 2：一直都有导航。听着，我们现在就要去测试与布局相关的所有链接（看着一个和图 9-4 等价的实际界面，以及到动画指南和事先定义的建筑平面图的新链接[作为界面和可用性测试的反馈的后果]）。

团队成员 1：那我们需要做什么呢？

团队成员 2：我们必须为导航语义和语法设计测试。每个链接都意味着一个 NSU——一个导航语义单元。例如，链接“编辑布局”（参见图 9-4）包含有导航路径和结点的一个特定集合。

团队成员 1：好，你的意思是，如果要编辑一个布局，我们必须从一个数据存储中取回它，而要完成取回，我们需要输入它的名字，然后重新生成它。

团队成员 2：我们还需要测试与取回相关的错误处理。例如，如果一个用户指定了错误的名字怎么办？

团队成员 1：而且导航路径部分可能会考虑如果用户忘记了名字会发生什么？我忘记了我们设计的东西。我们要列出这个用户下的所有文件名吗？无论如何，我们必须测试错误处理。

团队成员 2（想了一会儿）：这种情况怎么样：我们在创建一个布局的过程中，用户决定启动另一个 NSU。

团队成员 1：你的意思是另一个 NSU？

团队成员 2：嗯。假设我想要得到一个监控设备的产品规格说明，这在增量 4 中推荐的，同时我使用导航链接“产品规格说明”。

团队成员 1：但是那个 NSU 已经作为增量 2 的部分被测试过了（详细产品信息和下载）。

团队成员 2：我知道，但是这是一个不同的上下文。我们已经确保了当用户导航到产品说明的时候，我们不会丢失已经创建的布局。我们还必须保证有一种方法可以返回到布局，而且当我们返回的时候不会有错误。

团队成员 1：但是如果我们测试每一个 NSU，那么它将没完没了。我真的不认为……

团队成员 2（打断）：我们只是测试那些在这个上下文中非常有可能涉及的 NSU 以及那些和相同的内容对象有某些交互的 NSU，并不是每一个可能的 NSU。

团队成员 1（笑）：我感觉好点了，但并不太多。

15.9 配置测试

配置的可变性和不稳定性是使 Web 工程面临挑战的重要因素。硬件、操作系统、浏览器、存储容量、网络通信速度和多种其他客户端因素对每个用户都是难以预料的。另外，某个用户的配置可能会有规律地改变（例如，操作系统升级、新的 ISP 和连接速度），其结果可能是客户端环境容易出错，这些错误既微妙又重要。如果两个用户不是在相同的客户端配置中工作，一个用户对 WebApp 的印象以及与 WebApp 的交互方式可能与另一个用户的体验有很大不同。

配置测试的工作不是去检查每一个可能的客户端配置，而是测试一组很可能的客户端和服务器端配置，以确保用户在所有配置中的体验都是一样的，并且将特定于特殊配置的错误分离出来。

15.9.1 如何测试服务器端

在服务器端，设计配置测试用例以验证所计划的服务器配置（即 WebApp 服务器、数据库服务器、操作系统、防火墙软件、并发应用系统）能够支持 WebApp，而且不会发生错误。实质上，WebApp 被安装在服务器端环境并进行测试，目的是找出与配置有关的错误。

当设计服务器端的配置测试时，你应该考虑服务器配置的每个组件。在服务器端的配置测试期间，需要询问并回答以下问题：

- WebApp 与服务器操作系统完全兼容吗？
- 当 WebApp 运行时，系统文件、目录和相关的系统数据是否被正确创建？
- 系统安全措施（例如，防火墙或加密）能在允许 WebApp 运行，并对用户提供服务的同时，不会发生冲突或性能下降吗？
- 是否已经对所选择的具有分布式服务器配置[⊖]（假如存在一种配置）的 WebApp 进行了测试？
- 此 WebApp 是否与数据库软件进行了适当的集成？是否对数据库的不同版本敏感？
- 服务器端的 WebApp 脚本运行正常吗？
- 系统管理员的错误对 WebApp 运行的影响是否已经被检查？
- 如果使用了代理服务器，在站点测试时，是否已经明确这些代理服务器在配置方面的差异？

15.9.2 如何测试客户端

在客户端，配置测试更多地集中在 WebApp 与配置的兼容性，这些配置包括下面组件的一种

⊖ 例如，可能使用单独的应用服务器和数据库服务器，两台机器之间通过连接进行通信。如果连接失败会怎么样？

或多种替换［Ngu01］：

- **硬件**。CPU、内存、存储器和打印设备。
- **操作系统**。Linux、Macintosh 操作系统、Microsoft Windows、基于移动的操作系统。
- **浏览器软件**。FireFox、Internet Explorer、Safari、Mozilla/Netscape、Opera 及其他浏览器。
- **用户界面组件**。ActiveX、Java Applet 等。
- **插件**。QuickTime、RealPlayer 等。
- **连接性**。电缆、DSL、常规的调制解调器、工业级连接（例如，T1 线）。

除了这些组件，其他配置变量包括网络软件、难以预测变化的 ISP 服务以及客户机上并发运行的应用系统。

为了设计客户端配置测试，Web 工程团队必须将配置变量的数量减少到可管理的数目[⊖]。为了实现这一点，要对每一类用户进行评估，以确定此类用户可能遇到的配置。此外，行业市场上的共享数据可以用来预测最多的可靠组件组合，然后，再在这些环境中测试 WebApp。

15.10 安全性和性能测试

因为安全性测试和性能测试处理 WebApp 基础结构的三个不同的元素——提供给 Internet 用户入口的服务器端环境、提供给最终用户直接访问 WebApp 界面的客户端环境以及在客户机和服务器之间的通信连接通路。但是，相似性仅限于此。安全性测试关注于未经授权地访问 WebApp 的内容和功能，以及在服务器端和此 WebApp 互操作的其他系统。性能测试关注于 WebApp 的操作特性以及这些操作特性是否满足最终用户的需求。

15.10.1 如何确定 WebApp 是安全的

WebApp 的安全性测试是一个复杂的主题，在有效地完成安全性测试之前，必须要对该主题有充分的了解[⊜]。WeApp 及其所处的客户端和服务器端环境对于外部的电脑黑客、对单位不满的员工、不诚实的竞争者以及其他（想偷窃敏感信息、恶意修改内容、降低性能、破坏功能或者给个人、组织或业务制造麻烦的任何）人都是一个有吸引力的攻击目标。

应该设计安全性测试用例去探查在某些方面存在的弱点，如客户端环境、当数据从客户端传到服务器并从服务器再传回客户端时所发生的网络通信以及服务器端环境。这些领域中的每一个都可能会受到攻击。发现可能会被怀有恶意的人利用的弱点，这是安全性测试人员的任务。

在客户端，弱点通常可以追溯到早已存在于浏览器、电子邮件程序或通信软件中的缺陷。Nguyen［Ngu01］描述了一个典型的安全漏洞：

> 经常提到的缺陷之一是缓冲区溢出，这种缺陷使得恶意代码能够在客户端机器上运行。例如，向浏览器中输入的 URL 长度远远大于为 URL 分配的缓冲区容量，如果浏览器没有错误探测代码来确认输入的 URL 的长度，则会导致内存覆写（缓冲区溢出）错误。经验丰富的电脑黑客能够聪明地利用这种缺陷，通过写一个带有可运行代码的很长的 URL，使浏览器毁坏或改变安全性设置（从高到低），在最坏的情况下甚至会破坏用户数据。

对客户端的另一个可能的攻击是对放置在浏览器中的 cookie 的未被授权的访问。怀有恶意创

⊖ 在每种可能的配置构件的组合中运行测试是非常耗费时间的。

⊜ Andrews 和 Whittaker［And06］、Galbraith 等人［Gn03］、McClure 和他的同事［McC03］以及 Garfinkel 和 Spafford［Gar02］提供了关于此主题的有用信息。

建的站点能够获取包含在合法的 cookie 中的信息，并且用此信息危害用户的隐私，或者更精确的是为偷窃行为设置舞台。

客户端和服务器之间通信的数据易受电子欺骗行为的攻击。当通信路径的一端被怀有恶意的实体暗中破坏时，电子欺骗行为就发生了。例如，用户会被恶意的网站所欺骗，它看起来好像是合法的 WebApp 服务器（与合法的 WebApp 服务器具有相同的观感），其目的是窃取密码、私有信息或信用数据。

在服务器端，攻击包括拒绝服务攻击和恶意脚本，这些恶意脚本可以传到客户端，或者用来使服务器操作丧失功能。另外，服务器端数据库能够在没有授权的情况下被访问（数据窃取）。

为了防止这些（和很多其他）攻击，可以实现以下一种或多种安全机制 [Ngu01]：

防火墙（firewall）。它是硬件和软件相结合的过滤机制，它检查每一个进来的信息包，以确保信息包来自合法的信息源，阻止任何可疑的数据。

认证（authentication）。它是确认所有客户端和服务器身份的一种验证机制，只有当两端都通过了检验才允许通信。

加密（encryption）。它是保护敏感数据的一种编码机制，通过对敏感数据进行某种方式的修改，使得怀有恶意的人不能读懂。通过使用数字证书（digital certificate），加密得到了增强，因为数字证书允许客户端对数据传输的目标地址进行检验。

授权（authorization）。它是一种过滤机制，只有那些具有合适的授权码（例如，用户 ID 和密码）的人才能访问客户端或服务器环境。

安全性测试的目的是探测这些安全机制中的漏洞，这些漏洞能够被怀有恶意的人所利用。

在实际设计安全性测试时，需要深入了解每一种安全机制的内部工作情况，并充分理解所有的网络技术。如果 WebApp 是业务关键的，维护敏感数据或者可能是黑客的目标，应该将安全性测试外包给擅长这些技术的公司。



准备安全性测试

场景：CPI 公司的 Web 工程会议室

参与者：团队负责人、来自营销部门的两个代表、SafeHomeAssured.com 团队的一些成员，在沟通活动开始之后讨论安全性测试的需求

会话：

团队负责人：基于需求获取的结果，很明显我们已经获得了一些必须为 SafeHomeAssured.com 解决的隐私和安全性问题。

营销代表 1：我真的喜欢有它视频监控功能，但是你能想象如果有人侵入我们的系统并窃取客户的隐私会发生什么。它将会是一场灾难和一场很大的诉讼。

团队负责人（点头）：视频不是唯一的方面。我们需要保护客户，防止任何非授权用户访问系统的任何功能，无论窃贼侵入系统并使一个客户的安全系统失效，或者访问他们的账户信息……

团队成员 1：你们说得对。这就是我认为我们应该做的事情，对系统进行设计，那么……

团队负责人（打断）：咔，我们正在超越自己。在我们设计任何东西之前，我们要理解 SafeHomeAssured.com 存在的安全问题。

营销代表 2：我们预算了一个安全顾问的费用。

团队负责人：很好，我想让顾问立刻加入进来，而且我想要的不仅是一个顾问，一旦构建 WebApp 完毕，我将做所有的安全性测试。

团队成员2：如果告诉我们他们想要进行的是什么测试，我们就能够确保以针对测试的方式来设计WebApp。有点儿像考试作弊条。

营销代表1：是这样的，但是我认为你还应该获得他们关于我们需要如何设计这个WebApp来使它抗攻击的建议。

团队成员1：没有“抗攻击”的WebApp，只有非常安全的WebApp。

团队负责人（用手指列举要点）：是这样的……我们需要保护从服务器移到客户端的所有数据，我们需要保护网站发送的所有内容，我们需要对密码和其他账户信息加密，我们需要保护客户不受CPI的不诚实的员工的侵害，我们需要……

营销代表2（打断）：我们需要把所有这些事情和更多事情的列表交给安全顾问。

团队成员2：然后我们需要知道他们将要执行的测试的种类……详细的！

团队负责人（看着在座的每个人）：你们每个人写一个列表，然后在明天上午9点之前通过E-mail发给我。我将对它们进行组合，然后我们将联系一些安全顾问和测试公司，并从那里开始。

团队成员1：我们要做任何安全性测试吗？

团队负责人：我确保你们要做，但是繁重的任务最好还是留给专家。

15.10.2 如何测试WebApp的性能

你的WebApp要花好几分钟下载内容，而竞争者的站点下载相似的内容只需几秒钟，没有什么比这更让人灰心的了；你正设法登录到一个WebApp，却收到“服务器忙”的信息，建议你过一会儿再试，没有什么比这更让人烦恼的了；WebApp对某些情形能够立即做出反应，而对有些情形却似乎进入了一种无限等待状态，没有什么比这更让人感到惊慌的了。所有这些事件每天都在Web上发生，并且所有这些都是与性能相关的。

性能测试（performance testing）用来发现性能问题，这些问题可能是由以下原因产生的：服务器端缺乏资源，网络带宽太低，数据库容量太小，操作系统功能不完善或不牢固，WebApp功能设计糟糕，以及可能导致客户端-服务器性能下降的其他硬件或软件问题。性能测试的目的是双重的：1）了解系统如何对负载（即用户的数量、事务的数量或总的数据量）增加做出反应；2）收集将促使修改设计的度量数据，从而使性能得到改善。

15.10.3 性能测试的目标是什么

设计性能测试来模拟现实世界的负载情形。随着同时访问WebApp的用户数量的增加，在线事务数量或数据量（下载或上传）也随之增加，性能测试将帮助回答下面的问题：

- 服务器响应时间是否降到了值得注意的或不可接受的程度？
- 在什么情况下（就用户、事务或数据负载来说）性能变得不可接受？
- 哪些系统组件应该对性能下降负责？
- 在多种负载条件下，对用户的平均响应时间是多少？
- 性能下降是否影响系统的安全性？
- 当系统的负载增加时，WebApp的可靠性和精确性是否会受影响？
- 当负载大于服务器容量的最大值时，会发生什么情况？
- 低性能对公司收入的影响是什么？

为了得到这些问题的答案，要进行两种不同的性能测试：1）负载测试——在多种负载级别

和多种组合下，对真实世界的负载进行测试；2）压力测试——将负载增加到强度极限，以此来确定 WebApp 环境能够处理的容量。下面的两节将分别考虑每一种测试策略。

15.10.4 负载测试如何评估性能

负载测试的目的是确定 WebApp 和其服务器环境如何响应不同的负载条件。当进行测试时，下面变量的排列定义了一组测试条件：

N，并发用户的数量；

T，每单位时间的在线事务数量；

D，事务服务器每次处理的数据负载。

每一种情况下，在系统正常的操作范围内定义这些变量。当每一种测试条件运行时，收集下面的一种或多种测量数据：平均用户响应时间、下载标准数据单元的平均时间或者处理一个事务的平均时间。WebE 团队对这些测量进行检查，以确定性能的急剧下降是否与 N、T 和 D 的特殊组合有关。

负载测试也可以用于为 WebApp 用户估计建议的连接速度。以下面的方式计算总的吞吐量 P：

$$P = N \times T \times D$$

作为一个例子，考虑一个大众体育新闻站点。在某一给定的时刻，4000 个并发用户平均每三十秒提交一次请求（事务 T）。每一次事务需要 WebApp 下载一篇平均长度为 12KB 的新文章，因此：

N = 4000 用户

T = 0.033 事务/秒

D = 12KB/事务

可如下计算吞吐量：

$$P = [4000 \times 0.033 \times 12\text{KB}] / 60\text{s} \approx 1600\text{KB/s}$$

因此，服务器的网络连接将不得不支持这种数据传输速度，应对其进行测试，确保它能够达到所需要的数据传输速度。

15.10.5 压力测试如何评估性能

压力测试（stress testing）是负载测试的继续，但是，在压力测试中，我们强迫变量 N、T 和 D 满足操作极限，然后超过操作极限。这些测试的目的是回答下面的问题：

- 系统“逐渐”降级吗？或者当容量超出时，服务器停机吗？
- 服务器软件会给出“服务器不可用”的提示信息吗？更一般地说，用户知道他们不能访问服务器吗？
- 服务器队列请求增加资源吗？一旦容量要求减少，会释放队列所占用的资源吗？
- 当容量超出时，事务会丢失吗？
- 当容量超出时，数据完整性会受到影响吗？
- N、T 和 D 的哪些值迫使服务器环境失效？如何来证明失效了？自动通知会被发送到位于服务器站点的技术支持人员那里吗？
- 如果系统失效，需要多长时间才能回到在线状态？
- 当容量达到 80% 或 90% 时，某些 WebApp 功能（例如，计算密集的功能、数据流动能力）会停止吗？

有时将压力测试的变种称为脉冲/击弹测试（spike/bounce testing）[Spi01]。在这种测试中，增加负载达到最大容量，然后迅速回落到正常的操作条件，然后再增加。通过回弹系统负载，测试者能够确定服务器如何调度资源来满足非常高的需求，然后当一般条件再现时释放资源（以便为下一次脉冲做好准备）。



准备性能测试

场景：CPI公司的Web工程会议室

参与者：团队负责人和SafeHomeAssured.com团队的两个成员，讨论增量6（监控设备的在线控制）的建模活动之前的性能测试

会话：

团队负责人：营销部门确实非常关注我们构建在SafeHomeAssured.com的这个增量中的视频监控功能。

团队成员1（叹气）：它有很多设计问题和测试问题。

团队成员2：在过去的工作中，我做过一个有在线视频流的项目。我们在客户网站建立了一个环境，它允许我们从监控摄像机中导出视频流，这就意味着有一个盒子，它基于我们每秒传输的帧数建立正确的编码选项，将视频流导入我们的服务器（可能到一个客户专有的Web页面）。这样就可以对访问进行控制，然后……

团队成员1（打断）：有一些专利技术可能对我们有所帮助。我想到的一种技术在视频流的前面发送一个Java Applet，视频流将在客户端的能使用Java的浏览器中进行解码——没有插件或其他的配置问题。我认为我们应该……

团队成员2（打断）：好了——我确定我看过一些可以绕过服务器在客户端浏览器和实际摄像机之间建立点对点连接的东西，这样就没有任何真正的瞬时负载，除了页面第一次被载入的时候。

团队负责人（打断）：对此我们要做很多设计工作，但我不想在这里做。在休息之前，我想要讨论一下这个增量的性能测试。

团队成员1：如果没有一个可靠的设计概念，那么很难做。

团队成员2：嗯，性能测试有一些必须要解决的问题：随着访问视频的用户的数目的增加，服务器的响应如何降低？我们将使用一个经典的*S-T-D*分析［参见第15.10.4节］来获得对请求的吞吐量的处理等

团队成员1：我们还需要确定任何性能的降低是否都将对系统安全性产生影响。在给定它的隐私方面的情况下，这是很关键的。

团队负责人：我们需要做一系列的压力测试，含义就是我们需要有一个完全的测试环境，配有来自很多位置的流视频，从多种PC机进行访问等。

团队成员1：嗯，你是对的。我认为我们没有把任何这些考虑到我们为这个增量估计的时间中。

团队负责人（摇头）：遗憾的是我们没有。我将和营销部门谈谈，协商一下这个增量的部署日期。

团队成员2：他们将不会接受的。

团队负责人：是的，它们不会。如果他们不给我们更多的时间，我们就无法测试性能。我们将不知道在最终用户看来系统是什么样的。如果我们不知道它如何运行，那么我们将无法保证用户会对它满意。如果用户不满意，我们将失去业务，获得不好的口碑，并且将在长期的经营收中失去收益。

团队成员2：所有这些都是因为我们没有执行测试吗？

团队负责人（笑）：这只是我的看法，我坚持这样认为。

信息



WebApp 测试工具分类

Lam [Lam01] 在电子商务测试方面的论文中介绍了一种有用的自动化工具的分类法，这可以直接适用于 Web 工程环境中的测试。我们对每一个种类都提供了代表性工具[1]。

配置和内容管理工具：对 WebApp 的内容对象和功能组件进行版本管理和变更控制。

代表性工具：

在 www.davceston.com/scm/CMTools.html 上有全面的列表。

数据库性能工具：测量数据库的性能，诸如执行所选择的数据库查询的时间，这些工具帮助进行数据库优化。

代表性工具：

BMC 软件（www.bmc.com）。

调试器：调试器是典型的程序设计工具，可发现和解决代码中的软件缺陷。它们是大多数现代应用系统开发环境的一部分。

代表性工具：

Accelerated Technology（www.acceleratedtechnology.com）；

IBM VisualAge Environment（www.ibm.com）；

JdebugTool（www.debugtools.com）。

缺陷管理系统：记录缺陷，跟踪它们的状态以及解决方案。有些缺陷管理系统还包括报告工具，提供缺陷传播以及缺陷解决率方面的管理信息。

代表性工具：

EXCEL Quickbugs（www.excelsoftware.com）；

McCabe TRUETrack（www.mccabe.com）；

Rational ClearQuest（www.rational.com）。

网络监测工具：监视网络拥塞的级别。它们对识别网络瓶颈以及测试前端系统和后端系统之间的连接很有用。

代表性工具：

在 www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html 上有全面的列表。

回归测试工具：存储测试用例和测试数据，并且在连续的软件变更之后，可以重复使用这些测试用例。

代表性工具：

Compuware QARun（www.compuware.com/products/qacenter/qarun）；

Rational VisualTest（www.rational.com）；

Seque Software（www.seque.com）。

站点监测工具：通常从用户的角度监测站点的性能。使用这些工具编辑统计表，诸如端到端的响应时间和吞吐量，并周期性地检查某个站点的有效性。

代表性工具：

Keynote Systems（www.keynote.com）。

压力工具：在高级别的运行使用状态下，帮助开发者探测系统的行为，并找出系统的极限。

代表性工具：

Mercury Interactive（www.merc-int.com）；

Scapa Technologies（www.scap-atech.com）。

系统资源监视器：系统资源监视器是大多数 OS 服务器和 Web 服务器软件的一部分，它们监视资源，如磁盘空间、CPU 使用和内存。

代表性工具：

Successful Hosting.com（www.successfulhosting.com）；

Quest Software Foglight（www.quest.com）。

测试数据产生工具：帮助用户产生测试数据。

代表性工具：

在 www.softwareqatest.com/qatweb1.html 上有全面的列表。

---

[1] 这里试录的工具并不代表本书支持这些工具，只是这些工具的一些样例。另外，工具的名字被作研究的根据公司的商标。

测试结果比较器：帮助将一个测试集合的结果与另一个测试集合的结果进行比较。用这些比较器来检查代码的改变没有对系统行为造成不利影响。
代表性工具：
在 www. aptest. com/resources. html 上可找到有用工具的列表。

事务监视器：测量大量事务处理系统的性能。
代表性工具：
QuotiumPro（www. quotium. com）；
Software Research eValid（www. soft. com/eValid/index. html）。

网站安全性工具：帮助探测潜在的安全性问题。你可能经常安装安全性探查和监视工具，按计划安排运行这些工具。
代表性工具：
在 www. timberlinetechnologies. com/products/www. html 上有全面的列表。

15.11 本章小结

WebApp 测试的目标是对每一个 WebApp 质量维度进行检查，找出可能导致质量失效的错误或问题。应该对内容、功能、结构、可用性、导航性、性能、兼容性、互操作性、容量和安全性方面进行重点测试，在设计 WebApp 时进行的评审也属于测试的范围。

WebApp 测试策略检查每一个质量维度，从考察内容、功能或导航“单元”开始。一旦单独的单元都已经被确认，重点就转到测试整个 WebApp。为了完成这项工作，很多测试来自于用户的视角，并由用例所包含的信息所驱动。应该编写 Web 工程测试计划，并确定测试步骤、工作产品（例如，测试用例）以及测试结果的评价机制。测试过程包括 7 个不同的测试类型。

内容测试（和评审）主要对内容的多种分类进行测试，目的是发现语义或语法上的错误，这些错误会影响内容的精确性，或者对展示给最终用户的方式有影响。界面测试检查用户与 WebApp 之间通信的交互机制，并对界面的美学方面进行确认，目的是发现由于实现糟糕的交互机制、遗漏、不一致或界面语义不明确所导致的错误。

导航测试使用从分析活动中得出的用例，在测试用例的设计中，测试用例对照导航设计检查每一个使用场景。对导航机制进行测试，确保识别并改正妨碍用例完成的任何错误。组件测试检查 WebApp 中的内容和功能单元。每一个 Web 页面所封装的内容、导航链接和处理元素，它们形成了 WebApp 架构中的“单元”。因此，必须对这些单元进行测试。

配置测试试图发现针对特殊的客户端或服务器环境的错误和兼容性问题，然后进行测试，发现与每一种可能配置有关的错误。安全性测试包括一系列测试，探测 WebApp 及其环境中存在的弱点，目的是发现安全漏洞。性能测试包括一系列测试，当对服务器端资源容量的要求增加时，评估 WebApp 的响应时间和可靠性。

在第 16 章，我们将研究另一种质量保证机制——变更和内容管理。正如在第 1 章中所讨论的，变更在 WebApp 构建和维护中是永恒存在的。因此，必须开发特定的技术来管理 WebApp 的内容和功能的变更。

参考文献

[And06] Andrews, M., and J. Whittaker, *How to Break Web Software: Functional and Security Testing of Web Applications and Web Services*, Addison-Wesley, 2006.

[Bro01] Brown, B., *Oracle9 i Web Development*, 2nd ed., McGraw-Hill, 2001.

[Cac02] Cachero, C., et al., “Conceptual Navigation Analysis: A Device and Platform Independ-

ent Navigation Specification," *Proc. 2 nd Intl. Workshop on Web- Oriented Technology*, June 2002, www. dsic. upv. es/ ~ west/iwwost02/papers/cachero. pdf. (accessed August 13, 2007).

[Con95] Constantine, L., "What Do Users Want? Engineering Usability in Software," *Windows Tech Journal*, December 1995, www. foruse. com. (accessed August 13, 2007).

[Con99] Constantine, L., and L. Lockwood, *Software for Use*, Addison-Wesley, 1999, See also www. foruse. com/ (accessed August 13, 2007).

[Don99] Donahue, G., S. Weinschenck, and J. Nowicki, "Usability Is Good Business," *Compuware Corp.*, July 1999, www. compuware. com. (accessed August 13, 2007).

[Gal03] Gallbraith, B. et al, *Professional Web Services Security*, Wrox Press, 2003.

[Gar02] Garfinkel, S., and G. Spafford, *Web Security, Privacy and Commerce*, O'Reilly, 2002.

[How97] Hower, R., "Beyond Broken Links," Internet Systems, 1997, http: //portal. acm. org/citation. cfm? id = 265138 (accessed August 13, 2007).

[Lam01] Lam, W., "Testing E-Commerce Systems: A Practical Guide," *IEEE IT Pro*, March/April 2001, pp. 19-28.

[McC03] McClure, S., S. Shah, and S. Shah, *Hacking Exposed: Attacks and Defense*, Addison-Wesley, 2003.

[Mil00] Miller, E., "WebSite Testing," 2000, www. soft. com/eValid/Technology/White. Papers/website. testing. html (accessed August 13, 2007).

[Ngu00] Nguyen, H., "Testing Web-based Applications," *Software Testing and Quality Engineering* May/June 2000, www. stqemagazine. com (accessed August 13, 2007).

[Ngu01] Nguyen, H., *Testing Applications on the Web*, Wiley, 2001.

[Pre05] Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 6th ed., McGraw-Hill, 2005.

[Sce02] Sceppa, D., *Microsoft ADO. NET*, Microsoft Press, 2002.

[Spl01] Splaine, S., and S. Jaskiel, *The Web Testing Handbook*, STQE Publishing, 2001.

[Wal03] Wallace, D., I. Raggett, and J. Aufgang, *Extreme Programming for Web Projects*, Addison Wesley Professional, 2003.

第 16 章　变更管理和内容管理

变更管理过程和内容管理系统相互协作，可以确保：1）Web 应用所有内容和功能上的变更可以被有效管理，而不会扰乱 Web 工程过程或降低 Web 应用自身的质量；2）所有的 WebApp 内容被合理收集、组织和展表示给请求该内容的最终用户。

为了使 WebE 过程持续顺利进行，必须管理变更，使得在建模、构建和部署一个重要 WebApp 增量时不会造成资源的浪费。为了保证质量，必须管理每个变更的 WebApp 对象，以确保变更是正确执行的、被正确记录用于将来引用，并且不会与已经完成的其他变更发生冲突。

内容管理收集、管理和发布所有已经被最终用户检查过的内容，包括已经被更改的内容（和功能）。内容管理系统将自动化工具和过程进行结合，用于保证内容基于最终用户的需求被正确地构建和表示。

16.1　变更

尽管我们都倾向于相信有关“一切都在变化中”的宣传，但事实上很多事情是不变的。当今每个业务的核心元素依然和一个世纪之前相同，我们需要创造、交易和销售商品；为客户服务；对雇员进行授权；投资公司来帮助公司成长。但与此同时，我们看到支持业务的核心元素技术、我们所开发产品的属性以及以客户为基础的数量统计的深远变革。我们从事业务活动的方式正在改变。基于 Web 的系统演化成了支持核心业务元素的技术中，所以我们必须预期基于 Web 的系统被构建和部署以后，Web 应用的需求会发生变更。

16.1.1　变更的属性有哪些

在 Web 工程的环境中，一个“变更”拥有很多重要属性：

- **描述**——从受变更影响的利益相关者角度解释变更的本质。
- **影响**——描述变更如何客观地表明自身（最终用户将看到的内容）和它将如何影响 WebApp 的内部内容和功能。
- **目标**——定义将被改变的特定 WebApp 对象（内容和功能两方面）。
- **实现途径**——描述做出该变更的技术方面。
- **历史**——记录变更被请求、评估和实现的时间，以及哪些 WebApp 内容和功能受到影响。

变更和描述这些变更的属性对于较小的工程来说比较容易管理。但是，当 Web 工程工作量规模增加的时候，就变得复杂了。内容对象和功能的数量显著增长，这些 WebApp 元素之间的相互关系变得复杂。工程的工作人数增加，而且两个人实现变更并产生冲突（缺乏其他人的知识）的可能性也在增加。然而，用于管理变更的工作量与正在变更的 WebApp 的规模和重要程度成正比。

16.1.2　为什么需要变更

第 5 章讨论了发生变更的一些原因。扼要重述为：1）非技术性利益相关者也许会对 WebApp 或者正在开发的增量产生一些事后的想法[⊖]；2）用户可能要求不同的交互模式或者需要不同的功能

⊖ 事后的想法可能是因为新的业务需求（例如，对项目市场的人口统计状况与之前定义的不同）、新的或者改变的内容或功能（例如，之前要表示的内容并非目前需要表示的内容）、管理中的变更（例如，一个新的执行经理“想以另一种新方式进行”）等而出现的。

或内容；3）Web 工程师发现需要完成一些期望之外的变更来满足 WebApp 需求或者需要提高 WebApp 的基础设施。除此之外，业务环境可能会变更，从而导致新的业务规则和需求的产生。

16.1.3 WebApp 的哪些元素变更了

“为什么”需要变更为我们提供了对变更动机的解释，可是并没有告诉我们实际上变更“什么”。就最一般的意义上说，是 WebApp 的内容改变。在这种上下文中，内容的含义是用于构建一个 WebApp 的任何信息，包括：1）结构化和非结构化格式的文本；2）图像，包括照片、图像表示、线性图和图示；3）音频，如语音记录、音乐、自然的和合成的声音；4）动作，比如视频和动画；5）导航机制，如按钮、链接和菜单；6）输入机制，如下拉菜单、表格和选择按钮；7）数据，如服务于 Web 应用的文件或数据库；8）基于客户端的功能，如在客户端执行数据处理的脚本；9）实现服务器端处理过程的基于服务器的功能。

每个请求的变更最终都会导致一个或多个内容类型的改变。例如，我们假设在 SafeHomeAssured. com WebApp 的第六个增量（安全监控设备的在线控制）实现之后，最终用户反馈指出需要更容易地操作音频和视频信息。用于监控的用户界面需要进行改变。同时，新一代廉价的微型无线高清晰度（HD）视频摄像机使得现在有可能在每个房间安装摄像机并获得高清的视频信号。然而这些变更需要控制客户站点的软件、WebApp 的功能以及用户界面自身。大量的内容对象和 WebApp 功能都需要变更。WebE 团队面临的挑战是确保：1）只有需要变更的内容发生变更；2）变更一个内容对象将不会对其他对象产生负面影响；3）旧版本的 WebApp（如支持更旧相机的技术）将可以和已经被变更的任何基于服务器的功能一起工作。

16.2 Web 工程的变更管理

在全书中我们已经讨论过 Web 应用的特殊属性和构建 Web 应用的 Web 工程过程。Web 工程一般采用一种迭代的、增量的过程模型，该模型使用敏捷软件开发中的很多原则。采用这种方法，通过客户驱动，团队通常可以在很短的时间内开发出一个 WebApp 增量。后续开发的增量添加其他的内容和功能，而且每一个增量都很可能要继续变更，这样可以使内容更丰富、可用性更好、界面更美观、导航更好、性能更好和安全性更强。因此，在 Web 工程的敏捷开发中，变更看起来是有所不同的。

回顾 Ivar Jacobson［Jac02］的观点：

> 一个敏捷团队是一个灵活的团队，可以恰当地响应变化。变化就是软件开发自身很大的一部分。构建中的软件本身的变化、团队成员的变化、使用新技术带来的变化等都会对开发的软件产品以及创建软件产品的项目本身产生影响。我们拥抱“支持变化”，它应当存在于软件开发中的任何事物中，因为它是软件的心脏与灵魂。

Jacobson 提到，Web 工程师必须拥抱变化，而且一个典型的敏捷团队避免了笨重过程、官僚主义和形式化引起的所有问题。当然，这也引出了我们在以下几节中提到的一些基本问题。

16.2.1 为什么需要管理变更

随着 WebApp 对企业的生存和发展变得越来越重要，对变更管理的需要也在增长。为什么呢？因为，如果不能有效地控制对一个 WebApp 实施的不恰当的变更（考虑到及时性和持续演化是很多 WebApp 的重要属性），那么会导致未经许可就发布新产品信息；错误的或者缺乏测试的功能可能会阻碍用户对 Web 站点的访问；安全漏洞可能会危害企业内部系统；还有可能引起其

他经济上的不如意甚至是损失惨重的后果。

应该关注哪些方面？当 WebE 准备设计变更管理策略时，有如下四个方面［Dar99］需要考虑：

内容。一个典型的 WebApp 包含了很大范围的内容——文本、图形、Applet（Java 小应用程序）、脚本、音频和视频文件、表单、动态网页元素、表格、流数据、功能和其他的很多内容。挑战在于如何将这些海量内容组织成一组合理的*配置对象*[⊖]。一旦确定了配置对象，我们就需要为这些对象建立恰当的*配置控制机制*。一种方式是通过对 WebApp 内容建模来展示对象间的关系以及每个对象的特殊属性。每个对象的本质（静态或者动态）和生存期（如临时的、固定存在的或者持久对象）就是在建立一个有效的变更管理方法时需要考虑的属性的一些例子[⊜]。例如，如果一个内容项每小时都在变，那么它就有临时的生存期。对这个内容项的控制机制就会不同于（不太正式）作为永久对象的表格构件的控制机制。

人员。因为绝大多数 WebApp 仍然使用特定的方式来开发，所以任何一个与 WebApp 相关的人员都可以（而且通常可以）创建内容。而多数内容创建者并没有技术基础，而且根本就不知道还需要变更管理。最终导致 WebApp 的变更将会（如果没有有效管理）无法控制[⊜]。所以，WebE 团队必须建立变更管理机制，使其不仅可以被技术人员接受，而且可以被开发内容的非技术人员接受。

可伸缩性。应用于小型 WebApp 的技术和控制并不能随规模很好地扩展。当将现有的信息系统、数据库、数据仓库和门户网站互相联系起来时，显然一个简单 WebApp 将会显著增长。随着规模和复杂性的增长，小的变更也会引起较深远的、无意识的影响，这种影响是有问题的。因此，变更控制机制的严格程度应该与应用的规模成正比。

法规。谁“拥有” WebApp？不管是大公司还是小公司都在争论这个问题，而这个问题的答案对 Web 工程相关的管理和控制活动具有重大影响。在某些场合，Web 开发者不属于 IT 组织，从而造成了潜在的沟通困难。Dart［Dar99］提出建议通过以下问题来帮助理解与 Web 工程相关的法规。

- 谁负责保证 Web 站点上有关信息的正确性？
- 在信息发布到站点之前，谁保证其遵循了质量控制过程？
- 由谁负责完成变更？
- 由谁承担变更的成本？

上述这些问题的解答有助于确定组织中必须采用 WebApp 变更管理过程的人员（关键的利益相关者）。

Web 工程的变更管理还不成熟，但在过去的若干年中已经有新一代专门为 Web 工程设计的工具出现。这些工具提供的功能可以帮助解决下列问题［Dar99］：

- 如何创建一个足够敏捷的变更管理过程，使其可以适应 WebApp 快速、持续的变更过程。
- 如何将变更管理的概念和工具更好地介绍给完全不熟悉相关概念的开发者。

⊖ *术语配置指* WebApp 的所有元素。一个*配置对象*是 WebApp 的命名了的部分（例如，SpaceLayout），有一系列属性用来描述配置对象，也可以与其他对象进行互相连接（例如，SpaceLayout 与 Walls、Doors 和 Windows 这些对象相连）。

⊜ Web 工程方法可以用于特别复杂的 WebApp。这些方法包括数据建模（例如，［Car01］和［Sim05］）和 UML 模型。

⊜ 有理由相信特定类型的 WebApp 可以也应该允许有机的增长。例如，一个用于展示由贡献者提供的各种艺术体裁（例如，短篇小说、电影和图画等）内容的站点，应该允许在不控制的情况下发表和修改这些内容。站点的动态属性可能是它的优势之一。然而，交付提供特定服务或关键信息的内容和功能的 WebApp，肯定不能承担由有机增长和无法控制的变更带来的过高开支。

- 如何对分布式 WebApp 开发团队提供支持。
- 在准发布环境中，如何对几乎是持续变更的内容提供控制。
- 在大量的配置对象中，如何获得对其的控制粒度。
- 如何合并配置管理功能和现有的 WebE 工具。
- 对于包含到其他对象的链接的对象，如何管理变更。

WebE 团队应该选择一个工具集来提供变更管理支持，还要提供上述问题的解决方法。

什么是基本的变更管理活动？其核心是，变更管理活动是简单的，但在 Web 工程快速变更的环境中，过程的执行会带来问题。关键是以规范并保证敏捷的方式来管理变更。但说起来容易，做起来难！

变更管理是一种“保护活动”，对它的使用贯穿于整个 WebE 过程。它包括五种活动：识别变更、变更控制、版本控制、审核和状态报告。我们将在下面各小节中简要介绍每种活动。

16.2.2 应该如何识别将要发生变更的对象

WebApp 的配置元素必须要被识别和命名。这意味着一个将被管理的内容项必须使用某种唯一的方式来标识。一些内容项的名字是显而易见的。例如，SafeHomeAssured. com WebApp 包括了所有安全性硬件（例如，传感器、摄像机、控制面板）的产品规格说明书。每个产品规格说明书都由以下一个或多个内容项组成：一个文本描述、一幅照片、一张示意图或其他表示连通关系或接口的图形、一个提供安装向导的视频片段。我们定义的一般意义上的内容对象（如产品说明书），可以用一组被命名的条目来表示：

$$\text{产品说明书（ProductSpecification）} = \text{文本（Text）} + \text{照片（Photo）} + \text{示意图（Schematic）} + \text{视频（Video）}$$

然而，为了管理 WebApp 内容的变更，我们必须识别特定的内容。目前 CPI 公司提供了一种新的微型 HD 摄像机作为其家庭监控系列产品的一部分。通过 SafeHomeAssured. com，用户可以通过网络远程控制这些无线摄像机，并传送 HD 视频影像。

摄影机型号为 HDV485。因此，使用 PS-HDV485- <类型> 这一格式来命名该摄像机的产品规格说明书中所有相关内容是恰当的，其中 <类型> 可以是 TXT、PHOTO、SCH 或者 VID。例如，摄像机的说明性文本和照片可以被分别定义为 PS-HDV485TXT. html 和 PS-HDV485PHOTO. jpg。内容名称提供了有意义的识别信息。

16.2.3 如何控制将要发生的变更

变更控制的基本方法已经在第 5 章中讨论过。回顾一下，每个被要求的变更都可以归为以下 4 种类型中的一种：

第 1 类。修正错误或改进局部的内容或功能的内容或功能变更。

第 2 类。对增量中的其他内容对象或功能组件有影响的内容或功能变更。

第 3 类。对整个 WebApp 有重大影响的内容或功能变更（例如，主要功能的扩充、重要内容的增加或减少、导航中必须的重要变更）。

第 4 类。立即会使一类或多类最终用户注意到的大的设计变更（例如，界面设计或导航方法方面的变更）。

一旦请求的变更分类完毕，就可以按照图 5-7 所示的算法来进行评估。图 5-7 被复制在这里，即图 16-1。

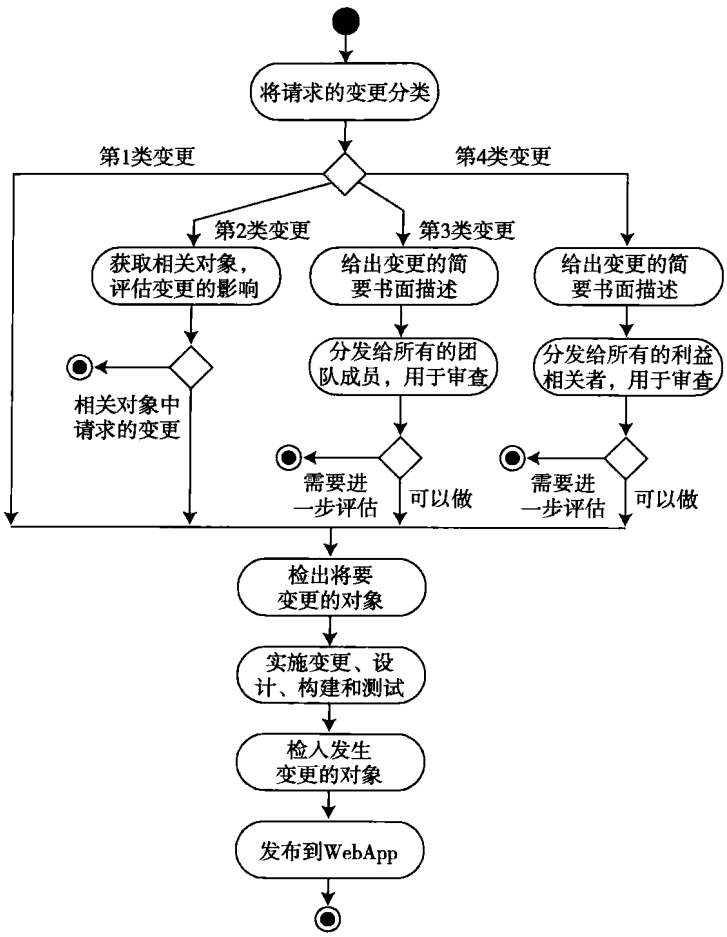


图 16-1 管理 WebApp 的变更

该图中，第 1 类和第 2 类变更可以看作是非正式的，并且可以采用一种敏捷的方式进行处理。对于第 1 类变更，你应该评估变更对于它会影响的配置对象的影响，但不需要任何外部的评审或文档。在实施变更时，需要被修改的内容对象可以先从工程数据库[⊖]中“检出”；变更完成后，将应用适当的质量保证活动。然后，对象将再被“检入”到数据库，并应用适当的版本控制机制创建软件的下一个版本。

以上这些版本控制机制与变更控制过程集成在一起，实现了变更管理中的两个主要元素——访问控制和同步控制。*访问控制*负责控制 Web 工程师（或者其他利益相关者）访问和修改特定内容对象的权限；*同步控制*协助确保两个人完成的平行变更不会相互覆盖。

对于第 2 类变更，由你负责评估变更对相关对象的影响（或者要求其他开发者来负责变更对这些对象的影响）。如果该变更不需要对其他对象做出大量变更就可以实现，那么修改时就不需要其他的评审或文档。如果需要大量的变更，就必须进一步评估和计划。

第 3 类和第 4 类变更也可以使用敏捷方式进行处理，但是需要一些描述文档和较正式的评审过程。*变更描述*（描述变更并提供对其影响的一个简要评估）适用于第 3 类变更。变更描述文档会被分发给 WebE 团队的所有成员（包括其他利益相关者），由他们对其进行评审以便更好地

⊖ 项目数据库也称为中心存储库，包含所有命名的内容对象，并且通常作为内容管理系统的一部分。请参看第 16.3 节。

评估其影响。变更描述同样适用于第4类变更，但在这种情况下，文档是由所有利益相关者一起进行评审的。



管理变更

场景：SafeHomeAssured.com团队负责人的办公室

参与者：WebE团队负责人和团队的两名成员，增量6（监控设备在线控制）的提交日期临近

会话：

团队负责人：我刚刚接到从销售部打来的一个电话，他们希望在发布这一增量之前做出一个变更。

团队成员1（轻声咒骂）：你在开玩笑吧？由于设计和安全性测试问题我们已经耽搁了，我认为不能按时提交了。

团队成员2（无奈地耸耸肩膀）：什么变更？

团队负责人：基本上，他们带来了新的问题，他们发现人们很欣赏远程控制他们住宅的想法，可是他们也害怕CPI中的一些人会有能力监视他们……

团队成员1（打断）：但我们已经在设计中解决了，通过……

团队负责人（抬起手）：我知道，但是他们担心对用户来说是如何可见。他们想要一些对于最终用户来说很显然并且安全无比的东西。[他解释了一个新选择，比如提供一个基于生物识别的登录方式使该系统只针对特定的最终用户，同时修改接口设计使其适应这种登录方式。]

团队成员2：这是硬件和软件的问题。显然是第4类变更。

团队成员1（皱眉）：如果属于第4类变更，那我们就需要提供一个全面的变更描述，讨论影响，接受所有利益相关者的评审……

团队负责人：这样，推迟所有变更到这个增量发布之后。我们将这个生物测试功能作为一个名为增量6a的新增量来发布或者作为增量7的一部分。

团队成员1：这就意味着不管我们采用何种策略，都需要修改增量7的维护计划、评估和提交日期。

团队成员2：没有人说变更是不需代价的。销售部很快会意识到这一点。

团队负责人：我想他们应该已经明白了，我们将会看到。

团队成员1（考虑了一下）：你知道我们正在进行变更，那么就有很多与摄像机控制相关的功能需要重构。这属于第1类变更，我想我们可以在做更大改进时将这些加进去。

团队负责人（笑）：你一定在开玩笑。

16.2.4 如何管理一个WebApp或其组件的不同版本

尽管对最终用户来说通常只有一个可用WebApp版本，但是也可能存在其他版本。一个老版本可能出于历史原因被存档；一个包含了不同的美学设计、内容和功能（和带一些新的导航路径）的新版本也许正在开发中。

一个版本控制系统实现或者直接集成了四种主要能力：1）项目资料库（中心存储库），存储所有关联的内容对象；2）版本管理功能，存储一个内容对象的所有版本（或者可以通过与先前版本的差异重新构建任何一个版本）；3）make工具，使得你能够收集所有相关的内容对象，并构建WebApp的特定版本。除此之外，版本控制和变更控制系统经常实现4）一个问题跟踪（也叫Bug跟踪）功能，使团队能够记录和跟踪与每个配置对象相关的所有重要问题的状态。

很多不同的自动化版本控制方法已经被提出来了。这些方法的主要区别在于用于构建一个系统指定版本的属性的完备程度和构建过程的机制。

16.2.5 WebE 团队如何确保一个变更恰当地实现了

答案分两个方面：1）进行结对走查；2）执行变更管理审核。

结对走查（在第5章中描述过）关注的是内容对象在修改后的技术正确性。走查者要评估对象，以确定它与其他内容的一致性，是否有遗漏，或者是否有潜在的负作用。结对走查适用于所有细小的变更。

一个*审核*（audit）对在走查中未考虑到的对象特征进行评估，从而对结对走查进行补充。审核提出并回答下列问题：

1. 对于所有与请求实施的变更相关的修改，引起任何额外的修改了吗？
2. 结对走查用于评估技术正确性了吗？
3. 是否遵循了 WebE 过程？是否正确地应用了局部 Web 工程标准？
4. 源代码中的变更是否被“高亮显示”？是否说明了变更日期和变更者？内容对象的属性是否反映出了该变更？
5. 是否都遵循了注变更、记录变更、报告变更等变更管理过程标？
6. 是否正确地更新了所有相关对象？

在某些情况下，审核的问题可以作为结对走查的一部分进行询问。然而，当变更管理是一个正式活动时，审核将由质量保证小组单独执行。这种正式审核还能够保证将正确的对象集成到特定增量中，并且能够保证所有文档都是最新的，而且与所构建的版本一致。

16.2.6 如何让利益相关者知道实施的变更

状态报告（status reporting）是一个变更管理任务，它解答如下问题：1）发生了什么事？2）谁做的？3）什么时候发生的？4）会影响其他对象吗？

状态报告在大型 WebApp 项目的成功中扮演着极其重要的角色。因为很多人都参与其中，这种情况就像是“左手也许并不知道右手在做什么。”两个开发者可能会修改同一个内容对象，而其目的不同，可能还有冲突。一个 WebE 团队可能会花数周的时间来构建一个 WebApp 增量，以实现一个已经过时的业务需求。一个认识到被提议的变更会带来严重负面影响的人却没有意识到该变更正在进行。状态报告可以通过增强团队中所有人之间的沟通来帮助消除以上这些问题。

在前面小节提到的每一个核心变更管理任务都必须进行精简，使得它可以尽可能地“精益”。同时，必须对变更进行管理以避免引起混乱，必须控制版本使得可以为最终用户提供一个高质量的 WebApp。

16.3 内容管理

Bob Boiko 在他［Boi05］对这一主题的广博的研究中提到，“内容管理就是获得对消息和功能的创建和发布的控制权。”内容管理与变更管理是相关的，因为内容管理系统确定了如何（从大范围的 WebApp 对象中）获取已有信息、如何按照能够提交给最终用户的方式构建现有内容，然后在客户端环境下显示这些内容的过程（有适当工具的支持）。

16.3.1 如何使用内容管理系统

当一个 WebApp 有重要内容要生成和发布[1]时，就可以使用内容管理系统（content management system，CMS）。CMS 在构建一个动态 WebApp 时最常用。动态 WebApp 能够"动态地"创建 Web 页面，即用户通常向 WebApp 请求特定的信息，WebApp 查询数据库，对信息进行相应的处理和格式化，然后展示给用户。

我们现在考虑 SafeHomeAssured.com 这一典型实例，它的特征是允许用户设计楼层的规划布局（SpaceLayout），确定哪些空间是需要保护的。一旦指定墙、门和窗户，最终用户就可以指定对于特殊监控硬件的需求（如摄像机、麦克风等）。然后，SafeHomeAssured.com 执行内部功能来选择相应的安全传感器（例如，针对 A 类型窗户的传感器），查询数据库中关于该传感器的特定信息（例如，安装需求、价格），并将其配置到一个内容模板中。最终的 Web 页面是在服务器端创建的，然后传送到客户端浏览器中由最终用户来进行验证。图 16-2 表示了这个过程。

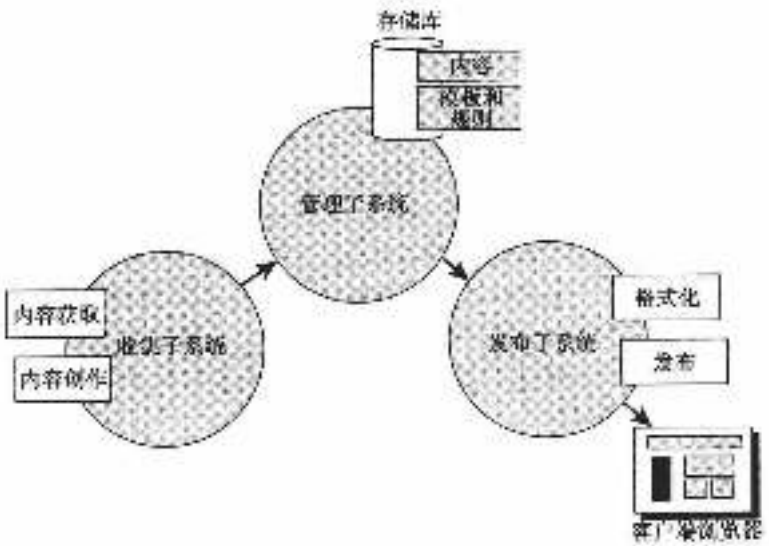


图 16-2 内容管理系统

16.3.2 CMS 的主要元素有哪些

一般来说，CMS 通过调用 3 个集成的子系统来为最终用户"配置"内容：一个收集子系统，一个管理子系统和一个发布子系统 [Bu05]。

收集子系统。内容是指内容开发者必须创建或获取的数据和其他非数值的信息。收集子系统包含了用于创建和获取内容的所有相关动作和一些必要的技术功能，这些技术功能能够将内容转变成某种标记语言（例如，HTML、XML）可以表示的形式，并将内容组织成可以在客户端有效显示的信息包。

内容的创建和查询（经常被称作创作）通常与其他 Web 工程活动并行发生，并且由非技术内容开发者实施。这一活动结合了创造性和研究的元素，并被特定的和通用的工具所支持，能够使内容作者在 WebApp 中以一种可以被标准化使用的方式来对内容进行特征化。

例如，CPI 公司中书写 SafeHome 产品规格说明书和开发概要的雇员不需要知道或者仅仅知

[1] 用于这个上下文时，"发布"一词表示内容的获取、格式化以及展现给最终用户。

遗少量的关于 Web 工程和 SafeHomeAssured. com WebApp 的内容。在很多情况下，在 WebE 项目开始之前，产品规格说明书是电子形式的。然而，产品规格说明书可以在任何时间添加新内容，而且可以相对频繁地对现有规格说明书进行变更。SafeHome 产品规格说明书应该放到数据库中（或者简单的文件管理系统中），以便每个人都可以在需要的时候访问它。

管理子系统。内容一旦被创建、查询和转换，就必须被保存到存储库中并分类，以备随后的获取和使用，而且还要进行标注以确定：1）当前状态（例如，内容对象已经实现还是正在开发中）；2）内容对象的正确版本；3）相关的内容对象。因此，管理子系统实现了一个包含以下元素的存储库：

- **内容数据库**。为存储所有内容对象而创建的信息结构。
- **数据库功能**。使得 CMS 能够查找特定内容对象（或者对象种类）、存储和检索内容对象，以及管理为此内容所创建的文件结构。
- **变更管理功能**。支持对象标识、版本控制、变更控制、变更审核和变更报告的功能元素以及相应的工作流。

除了上述这些元素，管理子系统还实现了管理功能，包括对控制内容整体结构及支持方式的元数据和规则的管理。

SafeHomeAssured. com 将要实现的管理子系统涵盖的复杂性差异，需要创建一个简单的平面文件系统。文件应该包括关于所有 SafeHome 传感器、控制面板和监控设备的目录，有指针指向所有相关产品和内容。例如，一个关于 HDV485 摄像机的文件应该有指针指向文本规格说明、照片、示意图、安装指导、操作指南和相关的 SafeHome 产品。对于复杂性范围的另一方面，WebE 团队需要构建一个精密的 SafeHomeAssured. com 数据存储库[○]，用于存储所有 SafeHome 产品的相关信息、由最终用户存储的信息（例如，订单，账户信息，一个命名的空间布局）和为 SafeHomeAssured. com 最终用户组织和发布内容而构建的“控制和配置文件”[Bol05]。图 16-3 表示了一个 CMS 数据存储库的示意图。

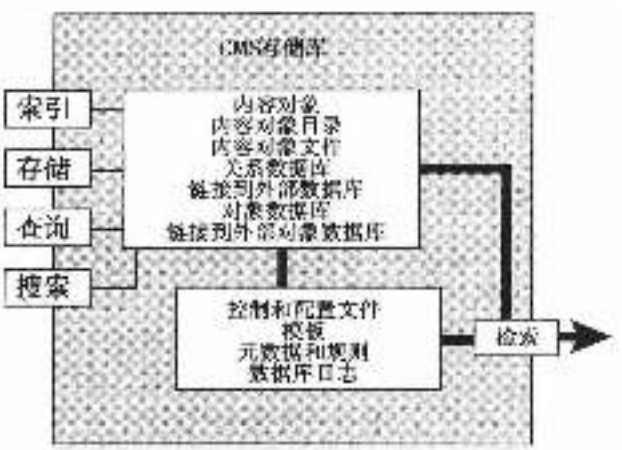


图 16-3 一个 CMS 存储库

控制和配置文件表达了一系列“规则”，这些规则由 WebE 团队进行定义，用于处理用户输入以及格式化并发布 WebApp 内容。例如，SafeHomeAssured. com 的 WebE 团队也许定义一个过早

○ 一个数据存储库包含了一个或多个链接的数据库、文件系统和相关的数据结构，该数据结构包含了 WebApp 的内容对象和内容对象被组织的层次元数据。

录入模板，它包括一种标准方法（例如，标准化的表单）来识别客户、定义需要购买的产品、指定支付模式、选择运送方式（如果需要）等。另一个模板可能是格式化一张用于支付所有传感器、控制面板和 WebApp 推荐购买的监控设备等材料的账单。

发布子系统。内容必须从存储库中提取，转换成适合于发布的形式，然后对其进行格式化以便传送到客户端浏览器中。发布子系统通常使用一系列模板来完成这些任务。每个模板对应一个功能，每个功能都可以使用以下 3 种不同元素中的一种来创建一次发布［Boi05］：

- **静态元素**。不需要进一步处理的文本、图形、媒体和脚本，可以直接传送到客户端。
- **发布服务**。调用特殊的检索服务和格式化服务功能来定制所需内容（使用预先定义的规则），完成数据转换，并构建适当的导航链接。
- **外部服务**。提供对外部共同信息（例如，企业数据）或内部应用的访问。

包含以上每一个子系统的内容管理系统适用于重要的 Web 工程项目。然而，和 CMS 相关的基本原则和功能适用于所有的动态 WebApp。可以在 CMS Matrix 站点 www. cmsmatrix. org/和 Wikipedia 站点 http：//en. wikipedia. org/ wiki/ List_of_content_management_systems 上获得大量免费、开源和商用的 CMS 软件。

16.4 CMS 的实现标准

问题不在于你是否需要管理内容——这是肯定需要的。不管项目的规模、复杂度或关注点如何，每个 WebApp 的内容都必须进行管理。真正的问题在于，需要确定内容管理的正式程度以及是否真的需要一个实际的 CMS[⊖]。

尽管我们仅提供了典型 CMS 的概要，但是很明显需要的内容管理支持软件的相当复杂。而且，创建一个适当的数据存储库、收集和识别内容对象、管理所有内容以及对最终用户的有效展示等都需要 WebE 团队和其他利益相关者做大量工作。

并非每个 WebApp 团队都需要正式的内容管理，而且很少的 WebApp 会使用过度复杂的 CMS。在 Web 工程项目早期的沟通和计划活动中，有必要搜集信息（需求）以决定内容管理需要的正式程度。当捕获需求以后（第 4 章），确定如下一系列标准来提供关于正式的内容管理和 CMS［Boi05］的指标是很重要的。

- **内容量**。内容对象类别（类）数以及每类中内容对象的大概数量。
- **贡献者人数**。那些创建将在 WebApp 中使用的内容的贡献者以及用于控制其更改内容的许可结构的复杂性。
- **变更量**。每个类型的内容和内容对象本身的变更量和频率。
- **发布量**。在 WebApp 中管理的内容产生的“发布”数量。

以下几小节我们逐个研究这些标准，了解它们如何影响内容管理的方式。

16.4.1 数量如何影响内容管理

假设你负责构建和维护一个小型商务站点，管理的内容对象（例如，描述信息、产品照片、简单 FAQ 部分和图形图像）不超过 100 个。每个内容对象都存储为一个单独的命名文件。

在这种情况下，内容管理是简单的。你清楚地知道每个内容对象在哪里，每个文件都包含了什么。这就无需正式的内容管理方法。一个 CMS 会是多余的。

⊖ 决定哪个 CMS（如果有的话）适合于你的项目会是一个挑战。在做出决定之前，你可以考虑访问开源 CMS 站点 www. opensourcecms. com，你可以在这里测试使用一些“基于 PHP/MySQL 的最佳免费且开源的” CMS 系统。

现在我们来考虑一个更大的企业构建电子商务 WebApp 的情况。企业销售 800 种商品，分为 10 个种类，每个产品都有自己的描述、照片和相关信息。除此之外，还有几百个一般内容对象充实整个站点（例如，创始人的传记和她的照片）。必须管理超过 1000 个的内容对象。

你管理这种内容量的正式程度必须提高。至少，WebE 团队必须定义文件命名约定，建立特定内容类型的简单规则（例如，一个产品描述通常被限定在 500 个字以内），以及建立变更管理的准则。

但是在这种情况下是否必须有一个 CMS 呢？为了回答这个问题，只能通过分析创建内容的人、内容变更的频率和内容展示给最终用户的方式等来获得更多的信息。

16.4.2 内容创作者的人数会影响 CMS 吗

随着内容创作者人数的增加，出现了一系列问题：1）保持特定内容类的一致性（在结构、风格和内容方面）变得更困难；2）因为变更可能是由很多不协调的资源引起的，所以变更管理变得更具挑战性；3）因为每个创作者都想拥有和控制自己创建的内容，所以集中式的内容管理会出现问题。这些问题都可以得到缓解，但是需要提高内容管理的正式程度。

为了说明这一点，考虑场景：CPI 公司市场部决定在 SafeHomeAssured. com WebApp 创建一个“Do-it-Yourself-Advice”（DIY 建议）区，目的是为现有 SafeHome 用户提供安装和使用建议。

公司向 50 位“专家”客户征求意见，要求每个客户对其选择的任何一个主题（关于 SafeHome）提出建议。如果没有为这些专家客户提供更进一步的指导，WebE 团队会立即面临一个内容管理问题。每份意见都有自己的格式和风格，长度会有很大的不同，而且关注的主题也会不同。

SafeHomeAssured. com WebE 团队决定通过提高变更管理的正式程度来缓解这些问题。首先，团队定义了一类称为 Advice 的内容对象。所有对“Do-it-Yourself-Advice”栏目的建议都归类为 Advice。更重要的是，所有 Advice 类内的内容对象都必须遵守以下规则：

- 所有 Advice 对象都必须是一个问题跟答案的形式。
- 所有问题必须是 20 字以内的一句话。
- 所有答案必须在 500 字之内，但是可以包括有关图表、照片或到其他 Advice 对象的链接。
- 所有 Advice 对象必须有时间和日期戳。
- 所有 Advice 对象必须由一个使用预先定义的 ID 和密码的专家客户提交。
- 其他专家客户可以使用预先定义的 ID 和密码对 Advice 对象增加附录。附录不能超过 250 个字。

可以把这些规则交给每个专家客户，并由 WebE 团队中负责管理这项内容的人来执行。如果需要更正式的内容管理，那么每个 Advice 对象可以通过一个基于表单的自动化工具来提交，由工具自动执行这些规则。而且还会命名这些对象并存储到一个 Advice 目录中便于使用。通过这种方式管理 Advice 内容，内容对象的一致性得到显著提高，变更管理也更加容易。

16.4.3 变更量如何影响变更管理的正式性

随着内容对象和内容创建者数量的增长，变更的数量也极有可能随之增加。简而言之，就是会有更多的东西要变更，也会有更多人找理由来变更！

在内容管理正式性的上下文中，变更量由两个不同的成分组成［Boi05］：内容对象吞吐量和变更频率。内容对象吞吐量通过考虑有多少内容对象以任何原因传入（出）数据存储库（或其他内容存储机制）来衡量。使用数学形式表达，内容对象吞吐量 T 被定义为：

$$T = \mathrm{Obj}_{\mathrm{new}} + \mathrm{Obj}_{\mathrm{old}} + \mathrm{Obj}_{\mathrm{edit}}$$

其中，

Obj_{new} = 在一个预定的时间段内（例如，一个月）添加的新内容对象的数量

Obj_{del} = 在一个预定的时间段内已经废弃并从 WebApp 中移除的内容对象的数量

Obj_{upd} = 在一个预定的时间段内通过某种方式变更（或者编辑）的内容对象的数量

随着内容对象吞吐量的增加，对内容管理的正式性要求也就增加。

变更频率指 WebApp 自身发生变更的频率。即在一个预定的时间段内，WebApp 的设计方面多久会修改一次？美学设计、内容设计、架构设计、导航设计和构件设计等的实质性变更一定会对 WebApp 内容有不同程度的影响。结果，会可能需要更正式的内容管理。

16.4.4 发布量如何影响内容管理的正式性

发布量通过两个因素来衡量：发布类型的数量和发布的个性化程度。发布类型指表现 WebApp 内容的不同结构和格式。随着发布类型的增长，通过基于模板方式来创建越来越多用户内容的需求就变得更明显了。发布的定制指最终用户的输入影响作为定制部分所展示的最终内容的程度。重量级发布定制的典型 WebApp 例子是通过用户选择来动态生成页面。

例如，考虑以下情况，CPI 公司决定扩展 SafeHomeAssured.com 的 WebApp，使其不仅服务于零售客户，也可以服务于原设备制造商（OEM）、批发销售商、零售商店、其他的安全公司和政府机构等。每一类最终用户都需要有关 SafeHome 产品的不同结构和格式的信息。用于零售的内容对象保持不变，会添加许多新的内容对象以满足不同的用户类型。但是，所有的内容对象（新的和旧的）都必须针对每类用户定制发布。

随着发布量和定制程度的增长，管理每种要求的发布物的创建变得更加困难。因此，就需要一些自动化工具（CMS）来为不同用户类型创建各种不同发布请求。



评估对 CMS 的需要程度

场景：SafeHomeAssured.com 团队负责人的办公室

参与者：在增量 2（详细的产品信息和下载文件）的建模活动开始之前，团队负责人和 WebE 团队的成员

会话：

团队负责人：我们已经有一个基本的 SafeHomeAssured.com 信息站点投入运行。现在需要做出一个架构上的决定：如何管理 SafeHome 的内容？

团队成员 1：基于目前获取的需求来看，我们大概会有 40 种单独的 SafeHome 组件，每一种都有产品规格说明书、一张照片、安装指南，有些还有视频。我们还有一些用户产生的内容。

团队成员 2：显然，一些组件与其他组件紧密相关，因此我们就链接它们。问题是我们是把这些链接“固定”在内容中还是存储为数据存储库的一部分，根据用户的需要动态产生链接？来自用户的内容也是相互关联的。例如，布局与所选择的监控组件有关。

团队负责人（沉思）：这是一个关键的架构问题，对吧？我们要实现一个储库和 CMS 吗？

团队成员 3：无论答案如何，都提出了这个增量，而且肯定会影响整个 WebApp 中创建的每一个增量。

团队负责人：什么意思？

团队成员 1：我们可以确定一件事：如果 SafeHome 成功了，WebApp 将增加，内容也会增加。例如，我和一位销售人员谈过，他希望在产品信息中有一个“建议栏”。另外一个利益相关者希望可以存档所有在线控制和监控中收集的视频；我想这应该是增量 6 中的内容。无论如何，这些特征没有框架，也许会在以后实现。所以我认为我们需要一个可靠的和可扩展的信息架构。一个好的 CMS 可以满足这个需要。

团队成员2：我认为应该保持简单。一旦陷入一个CMS中，我们将受限于它的架构和约束，如果自己实现数据库，将会很灵活。

团队成员1（皱眉）：我们不需要重新发明一个车轮。

团队成员3：如果我们可以找到一个结构满足我们要求的数据存储库，我将赞同这种方法。仅仅构建SafeHome的特征就已经有很多工作了，我们不必沉溺于实现大量数据结构、数据管理功能，尤其是已经存在的一些功能。

团队负责人：这是一个重大的决定，无休止的争论对我们没有用处。我提议［看了一下成员1］，你在接下来的几天中研究一下我们的选项。我刚刚读到一个有趣的文章："如何衡量一个CMS系统。"我会发给你URL［通过www. steptwo. com. ou/papers/kmc_evalue/查看站点Step Two Design］。如果有可能，提出3～5个满足我们需要的候选CMS。我们可以测试运行一下。

团队成员1（看起来小心翼翼）：但是……我们到底需求什么？

团队成员3：我们可以把搜集到的需求罗列出来，也许还可以推导出一些未来的需求。我们将管理的内容类型、内容对象之间关系的复杂性、静态和动态内容对象的数量、整个项目中内容对象的数量、撰写内容的人数、我们可能遇见的内容变更量之类的事情。

团队成员1（看着成员3）：你会给我一个列表，对吗？

团队成员3：当然可以。

团队成员2（对成员3说）：你刚刚提到每一个项目都只是表示了一个相对较小的系统，少于40个构件，内容层次并不是很复杂，动态对象相对很少。只有少数人产生CPI内容……

团队成员1（打断）：但是潜在的数以千计的用户都将产生他们自己的内容——报价、布局、材料清单，更不用说有关监控特性的动态内容了。

团队负责人（抬起手）：好了，停。作为这个项目的架构师，我的意见是需要一个CMS。也许我们可以使用一个非常简单的实现，也可能不使用。我们在星期五之前再做一些分析和决定。

16.5 本章小结

由于基于Web的系统支持业务的核心构件，所以这些系统在构建过程中和投入使用以后，其需求会变更。当请求变更时，明显的变更包括内容对象——文本、图像、音频、视频、导航机制、输入机制、数据以及基于客户端和基于服务器端的功能。这些变更和内容对象本身都必须进行管理。

变更管理过程包含5个主要的活动：识别变更、变更控制、版本控制、审核和状态报告。在识别变更期间会为内容对象制定一系列命名规则。变更控制定义了变更的重要程度和影响，并建立了一个能够使变更在没有混淆和错误的情况下进行的过程。版本控制使WebE团队能够维护一个内容对象的多个版本。审核是一个质量保证活动，有助于保证变更的正确进行，不会造成意想不到的后果。最后，状态报告开发的机制用于报告发生了什么变更、何时发生的、谁实施了该变更以及哪些内容对象受到了影响。

内容管理允许WebE团队开发一个过程和一套工具，用于收集、管理和发布所有与WebApp有关的内容对象。收集子系统建立了创建和获取内容以及把内容转换和组织成可以提交给最终用户的格式所必须的技术功能等基础。管理子系统实现了内容数据库和相关的变更管理功能。发布子系统把内容组织构建成最终的形式。变更管理过程的正式程度和支持该过程的系统随着

将要部署的 WebApp 的规模和复杂性的不同而不同。

参考文献

[Boi05] Boiko, B., *Content Management Bible*, 2nd ed., Wiley, 2005.

[Car01] Carlis, J., and J. Maguire, *Mastering Data Modeling*, Addison-Wesley, 2001.

[Dar99] Dart, S., "Change Management: Containing the Web Crisis," *Proc. Software Configuration Management Symposium*, Toulouse, France, 1999, www.perforce.com/perforce/conf99/dart.html (accessed August 14, 2007).

[Dar91] Dart, S., "Concepts in Configuration Management System," *Proc. Third International Workshop on Software Configuration Management*, ACM SIGSOFT'1991, www.sei.cmu.edu/legacy/scm/abstracts/abscm_concepts.html (accessed August 14, 2007).

[Jac02] Jacobson, I., "A Resounding 'Yes' to Agile Processes——But Also More," *Cutter IT Journal* vol. 15, no. 1, January 2002, pp. 18-24.

[Sim05] Simsion, G., and G. Witts, *Data Modeling Essentials*, 3rd ed., Elsevier, 2005.

第17章 未来趋势

在本书的第1章，我们提到“Web已经成为贸易、商业、通信、教育、工程、娱乐、金融、政府、工业、媒体、医药、政治、科学和交通等方面不可或缺的技术，而这些只是影响你生活中的一小部分而已。”我们提到，因为Web已经变得很重要，世界人口中每天都在使用WebApp的比例不断增长，所以只有以各种丰富的、各种各样的并且重要的方式构建WebApp才是唯一合理的。

但是，未来的情况会怎么样？Web会走向哪里？用户将怎样和一个“新的”Web交互？这种交互将如何改变我们的生活、工作和世界观？Web工程师应该如何应对Web未来的发展趋势？哪些工具和技术可以帮助他们？

说实话，我们不能确定地回答以上任何一个问题。如果说过去的十几年中Web的发展告诉了我们什么，那就是变化速度确实惊人，造成的影响很大，出现的每一个重要WebApp的特征都出乎人们的意料（至少直到它真的出现时，我们都意识到它是多么明显）。尽管如此，在最后一章中，我们还是尝试简要地探讨我们刚刚提到的这些问题。

17.1 Web和WebApp具有不断变化的特征

10年前，Web是一个传播信息的工具。少数被神化的Web专家（Webmaster）创造了WebApp，并且在很大程度上控制了信息空间。那个时代已经过去了。今天的Web给人们创造了平等的环境。BBC［BBC06］提到，“博客、Wiki、社交网络站点、文件共享服务等已经改变了Web……”Web内容爆炸的结果已经给最终用户带来了巨大的利益，但同时也产生了很多问题，这些问题包括内容的准确性、合法性、可访问性以及其总的价值。BBC继续提到：

> 但是在这个（内容）共享的世界里也存在危险。随着Web越来越深入我们的生活，它将成为一个新的战场，而且安全问题的威胁侵蚀着网络公共信任度。
>
> Web（和WebApp）可能用于窃取你的身份，哄骗你购买并不存在的商品，或者更令人担忧的是可能破坏我们社会的基础。

所以像所有卓越的技术一样，Web在提供巨大收益的同时也提供了损害用户的平台。作为用户和Web工程师，我们的工作就是利用前者，同时限制那些可能会造成后者的活动。

17.1.1 基于Web的内容和功能的交付会如何变化

CNET［CNE05］直接处理了这个问题，它说：“商业需要像一个软件公司一样运作，并鼓励程序员在他们的平台上构建服务，而不是仅仅将Web当做一个便捷的发布信息的工具……”

Web内容和功能可能会包括越来越多的混合WebApp。简单地说，*混搭应用*（mush-up application）使用由多个商业WebApp提供的信息来提供一个新的信息和功能的混合物。一个简单的例子是，一个WebApp开发者可能开发了一个WebApp（例如，www.mywikimap.com），该应用链接了一个商用站点（例如，Google Maps），该商用站点包含一个存储了全美国的汽油价格数据库。其结果就是一种混搭应用，它根据邮政编码就可以提供所在地区所有低价的加油站。

除了可以从多个WebApp结合内容和功能，从而创建比这些内容和功能的简单相加更有价值的内容和功能这一独特能力之外，Web内容还应该可以跨平台交付以实现持续访问。今天，用

户甚至可以通过笔记本电脑、手机、PDA 等无线计算设备访问 WebApp。访问将变得无处不在，这不断开放实时交付有价值内容的机会。例如，你站在一个从来没有去过的餐馆前面，想知道这个餐馆有多好。你通过手机访问“餐馆评论” WebApp，输入餐馆的名字，在进入该餐馆之前查看对它的评论。如果你不喜欢这家餐馆，你可以查询附近别的推荐餐馆。

17.1.2 WebApp 将如何变化

随着更好、更健壮的内容与更复杂功能的耦合，传统的 WebApp（Web 1.0）将继续演化。结果就是提供给我们诸如 Google Maps 这样的 WebApp，它耦合了大量通过复杂操作功能得到的卫星图片。随着混搭应用变得越来越先进有力，Web 2.0 的能力开始显现。WebApp 将连通多个内部关联的数据库和服务，使得数据对于不同类型的用户都是有用的。通过对表面看来各自独立的内容和功能建立协作，“关联数据源和服务的聚合网”将把 WebApp 的能力带上另外一个层次。

为了说明这一点，再次考虑前面提到的餐馆的例子。“餐馆评论” WebApp 的进一步发展是它将自动获知你的地理位置（通过内嵌的全球定位系统 GPS 接收器提供的数据），透明地将这一信息关联到地图数据和商业注册信息，使得它可以知道你正站在一个特定的餐馆前面，不需要你进行明确查询就可以自动向你提供相关的信息（例如，评论、菜单、定价和其他选择）。它甚至可以知道你的口味并查询有类似口味的人对这个餐馆的评论。

17.1.3 为了适应这些变化，Web 工程师必须做些什么

伴随着新一代 WebApp 的发展，网站可能都会链接“一个关联数据源和服务的聚合网”[Coa06]。为了构建这样新一代的 WebApp，Tom Coates [Coa06] 向 Web 开发者建议了一些高层指导准则。Jeremy Zawodny [Zaw06] 通过下列这些注释扩展了 Coates 的指导准则（以下使用黑体字表示）：

希望向聚合的 Web 数据中添加价值。当一个公司通过基础设施可以对全世界所有公共在线信息中的很大一部分进行浏览、获取和分析，想一下你可以如何使用这种能力来改善这个世界。对于有些人来说，只能查看很小的一个数据集合，你可以做什么？可以发现哪种模式？可以创建哪种连接？你可以为人们简化什么？

为普通用户、开发者和机器进行创建。使你所创建的任何东西易于使用、易于访问（在开源的情况下），使用结构化格式发布有用的数据。这就意味着你需要一个可使用性 geek（极客、奇客）、一个 API geek、很可能还有一个 XML/RSS/JSON geek。

从设计数据而不是页面开始。理解哪些是重要的数据，它将如何存储、表示和转移。考虑可以在这个中心存储库上创建的主要服务，你应该仅仅使得框架 geek 和图像处理 geek 参与……

确定你的首要目标对象并保证它们是可处理的。理解你服务的最根本目的是什么。如果是一个商业的购物应用，你也许需要处理人、货物和货物清单。在进一步开展构建之前彻底掌握这些，同时确保对于每个对象类都有某种方式可以从外界对其进行访问。意味着存在一个 URL，可以用来取得一项货物、一个清单等的信息。这些将是你在后面处理更复杂的事情要用到的构建块。但愿其他的也是这样。

使用可读的、可靠的并且可访问的 URL。如果 URL 很难通过电话读出或者通过电子邮件发送，那么你就还没有达到这一要求。这条准则是关于简单性和可预测性的。考

虑类似 http：//socialshopping. com/ item/12345 的 URL。你是否可以通过这个 URL 猜测它是做什么的？也许你没有意识到这件事的重要性，但不要因此阻止你关注这件事。这种事情的确很重要。考虑在 del. icio. us 中大多数 URL 是如何易于猜测和简单的，模仿它。

与外部标识符模式建立关联。如果已经存在一种已建立起的机制可以使用，就不要发明一种全新的方式来表示和构建事物。这种努力不仅是无用的，而且它大大降低了其他人适应并帮助加强你所构建的平台的机会。不管你信不信，你是在构建一个平台。

创建列表视图和批操作接口。使得易于查看给定的一种类型的所有条目，并且使成组编辑成为可能。Flickr 做到了成批上传照片。多种形式的搜索就是一个“列表视图”的典型例子。

使用标准来创建并行数据服务。开发者（和他们写的代码）将要使用你的数据。不要在事后才考虑这件事。让你的工程师们考虑他们将要如何使用数据，并确保他们设计的产品支持这些设想。另一方面，总是使用一个现有标准或者在必要的时候对一个现有标准进行扩展。考虑 RSS 和 Atom 是多么的灵活。不要再重新发明轮子。

使你的数据尽可能易于发现。你使用的名称和属性对于用户和开发者来说应该是具有描述性的，而不只是把它当作基于内部系统所开发的副产品。意思就是作为一个旁观者来考虑并做一些额外工作。

这些指导准则可以提供一个针对 Web 2.0 的 WebApp 开发的简单原则性基础。

17.1.4 Web 可以作为应用软件的平台吗

当然可以！事实上，基于 Web 的应用已经开始出现并有可能最终成为传统包装的软件的有力竞争者。正如在第 14 章提到的，WebApp 正在将客户端界面技术和服务器端功能进行混合来创建应用，这种应用模仿了桌面应用的丰富交互。然而，通过将这些应用变成基于 Web 的，我们可以添加一些在传统桌面应用中很难实现的方面：改进的共享和协作、分布式存储和访问数据以及更快的应用演化。

作为一个例子，考虑 Writely（www. writely. com），一个基于 WebApp 的文字处理器，最近已经被 Google 收购。在我们写该章的时候正在进行 Beta 测试，Writely 提供了大量有意义的文字处理功能，包括一个在线编辑、拼写检查和相关功能。它允许上传和存储现有文字处理器类型的文档，然后能够以流行的格式下载文档。它提供了一个内置的修订历史，并可以使文档“回滚”到先前的任何一个版本。

并不能确定基于 Web 的应用（如 Writely）是否会最终取代诸如 Microsoft Word 或 Corel Wordperfect 等的桌面应用，但是基于 Web 的应用软件的时代已经开始了。

17.1.5 未来的 Web 会是一个操作系统吗

随着 Web 成为某些类型应用软件的平台，有理由思考一个“WebOS”是否会在未来逐渐出现，以及这样的一个“操作系统”可能会是什么样子。Jason Kottke［Kot05］提到一个 WebOS 将包括如下三种组件：

- 一个 Web 浏览器和加载项。“变成主要的应用界面，用户通过它来查看内容、执行服务、管理本地机器的和网络上的数据，通常甚至不会知道它们之间的区别。”

- WebApp。已经被加强以提供一个更好的用户界面和其他使得它们更接近桌面应用的功能。
- 一个本地 Web 服务器（补充了本地的操作系统）。“处理从本地机器到浏览器的数据交付和内容展示。本地服务器会对任务进行高度优化，并且有能力运行本地安装的 WebApp（例如，一份 Gmail 的本地副本和所有相关数据）。”

浏览器和 Web 服务器可能仍然是针对特定操作系统的（例如，为 Windows、Mac OS 或 Linux 所写的），但是所有的 WebApp 都会使用一种与机器无关的方式针对 WebOS 进行开发。

Kottke［Kot05］描述了 WebOS 的一系列固有的优点：

> 与“单机的”WebApp 和桌面应用相比，为这种假定的平台开发的应用有一些强有力的优势。因为这些应用运行在 Web 浏览器中，所以它们就是跨平台的（假设无论谁开发了这样一个系统，他为 Windows、OS X、Linux 或手机等开发了本地 Web 服务器的部分），就像 Gmail、Basecamp 和 Salesforce. com 这些 WebApp 一样。你不需要在一个特定机器上使用特定的操作系统，只需要一个浏览器和一个本地 Web 服务器就可以访问你喜欢的数据和应用。
>
> 对于应用开发者，主要的优势在于他们可以使用同一种代码只开发一个应用，通过 WebOS 来使其运行在任意机器上，不用为多个平台（一个针对 Web、一个针对 Windows 等）编写两个或多个程序……
>
> 在本地运行应用也会给你带来了好处。你可以在没有连接网络的时候使用它们。在你喜欢的浏览器上会有一个按钮来启动 Gmail。当应用程序会用到较大的文件如图片、视频和音频时，这些文件应该在本地进行存储和操作，而不是等待通过网络传输过来。

WebOS 将以何种方式发展还是有些模糊。Nicholas Carr［Car05］提出了其他的相关趋势：“成本、安全和可靠性等顾虑将最终导致公司要求甚至比 Kottke 想象的更少的本地数据和应用。用户设备将变得更瘦，而应用、数据甚至是用户操作系统都将存在于中心服务器群中。”

17.1.6 语义 Web 将如何改变事物

有几十亿的内容对象存在于几亿的 Web 页面中。为了找到需要的内容，你可能精心选择出来一串关键字，并提供给一个或多个搜索引擎，可是结果有好几万，而很多与你想要的东西完全没有关系。

Web 工程师（和用户）面对的挑战是：更好地理解有什么内容驻留于 Web 中，并且能够很容易找到；自动且无缝地建立内容对象之间的关系；以及从表面上不同的内容和数据源之间抽取知识。为了给存在于 Web 中的内容赋予更多的意义，提出了语义 Web 机制。

语义 Web 应用一系列技术，包括资源描述框架（Resource Description Framework，RDF）、Web 本体语言（Web Ontology Language，OWL）和可扩展标记语言（Extensible Markup Language，XML）。Wikipedia［Wik06］以如下方式来描述这些技术之间的交互：

> 这些技术结合起来，以提供对 Web 文档内容的补充或替换。因此，内容可能会表示为存储在 Web 可访问的数据库中的描述性数据或者在文档内被标记（特别地，在由 XML 标记的 XHTML 中或者更多情况下完全在 XML 中，布局和表示是分开存储的）。机器可读的描述能够使内容管理器为内容添加含义，因此促进了通过计算机进行信息的自动收集和研究。

从本质上来说，XML 建立了文档语法规则；RDF 提供了用于精化描述性信息的机制；OWL 使得内容提供者可以创建与领域相关的词汇，从而使用户可以提取内容的语义。

在 1999 年，Tim Berners-Lee［Ber99］尝试探寻语义 Web 的未来时提到："我有一个梦想，Web（其中的计算机）变得有能力分析在 Web 上的所有数据——内容、链接以及人与计算机之间的事务。一个'语义 Web'应该使这一切成为可能。当语义 Web 做到这一点时，日常的交易机制、行政工作和我们的日常生活都将通过机器之间的交互来处理。人们已经吹捧了很多年的'智能代理'将最终被实现。"我们还没有做到这一点，但是 Berners-Lee 的观点可能将会在下一个十年被实现。

17.2 演化中的 Web 技术和 Web 2.0

在这一点上，我们已经讨论过了 Web 的变化特征。现在我们研究这些变化中的一些特定的技术元素及其如何与当前和未来方向相关联。我们从 Web 2.0 开始。

17.2.1 什么是 Web 2.0

Web 2.0 并不是指某种特定的技术或者一组技术，而是指 WebApp 设计的趋势。在详细讨论这些之前，也许值得暂时离题一下来探讨一些最近的历史。

当 Tim Berners-Lee 在 90 年代早期创造了第一个 Web 浏览器时，它其实并不只是一个浏览器，而是一个浏览器编辑器。只从最早的 Web 开发来看，开发的意图是让 Web 用户也可以积极主动地加入到 Web 内容的创建之中。当 Web 成为主流以后，这个最初的概念就被丢弃了（至少一部分）。这么多年以来，Web 就只是被动地发布信息的渠道，创作者通过这个渠道创造内容并使其可用，用户访问和使用这些内容。

Web 2.0 回到了 Berners-Lee 最初的原理，使得 Web 变得更加具有参与性。术语 Web 2.0 是由 Tim O'Reilly 在 2004 年提出的。在 2005 年的一篇论文中，O'Reilly 写道［Ore05］：

> 2001 年秋，网络泡沫的破灭标志着 Web 的一个转折点。许多人断定互联网被过分地炒作了，事实上网络泡沫和随之而来的股价暴跌看起来更像是所有技术变革的共同特征。股价暴跌通常标志着一个优势技术已经开始占领中央舞台。假冒者被驱逐，真正成功的故事显示出它们的优势，同时人们也开始明白是什么使它们区别开来。
>
> "Web 2.0"的概念开始于一个会议的头脑风暴时间……会不会是 dot-com 的垮台标志着 Web 的一种转折，从而使得对于"Web 2.0"行动的呼吁有了意义呢？我们同意这种观点，Web 2.0 会议也由此诞生。
>
> 之后的一年半时间中，"Web 2.0"一词已经深入人心，通过 Google 可以搜索到 950 万以上的记录。但是仅就 Web 2.0 到底是什么，仍然有巨大的分歧，有些人将 Web 2.0 贬低为没有任何意义的一个市场炒作口号，而另一些人则认为它是一种新的常规学问……
>
> 在我们最初的头脑风暴中，我们已经通过一些例子明确地表示了我们对于 Web 2.0 的理解：

Web1.0	->	Web 2.0
DoubleClick	->	Google AdSense
Ofoto	->	Flickr
Akamai	->	BitTorrent

mp3. com	- >	Napster
Britannica Online（大英百科全书在线）	- >	Wikipedia
个人网站	- >	博客（blogging）
evite	- >	upcoming. org 和 EVDB
域名投机	- >	搜索引擎优化
页面浏览数	- >	每次点击成本
屏幕抓取（screen scraping）	- >	Web 服务（Web services）
发布	- >	参与
内容管理系统	- >	wikis
目录（分类系统）	- >	标签（大众分类）
粘性	- >	聚合

O'Reilly 在文章中继续解释，Web 2.0 不是一系列特定的技术，而是一系列广泛概念和实践，反映了 Web 的发展方式，他总结如下：

> 先到此为止，总结一下，我们认为 Web 2.0 公司核心竞争力的一些方面如下所示：
> - 服务而不是打包的软件，具有高成本效益的可伸缩性。
> - 控制唯一的、难以再造的数据源，人们使用得越多，数据就应该越丰富。
> - 像信任共同开发者一样信任用户。
> - 利用集体智慧。
> - 通过客户的自服务来发挥长尾的力量。
> - 软件在单一设备级之上。
> - 轻量型用户界面、开发模型和商业模型。

为 Web 2.0 提供一个完全的定义总是有争议的。然而我们会主张 Web 2.0 是关于创造一个更有用的、互动的和集体性的 Web 体验，在其中，所有用户都能够主动地参与。

17.2.2 哪些技术支持 Web 2.0

Web 2.0 已经随着大量构建良好且体现其特性的 WebApp（例如，Wikipedia、Blogger、Flickr、BitTorrent、YouTube——仅仅列举一些比较熟知的应用）出现了。我们已经讨论过（第 14 章）一些以这些应用为中心的技术。这些技术使得 WebApp 达到了更大程度的主动参与。在本节中，我们将从 Web 的变化特征的角度来关注这些技术。下面罗列出来的技术仅仅用于说明而不是全面叙述：

博客和 Wiki。博客（blog，Web log 的简写）是一种评论或者日记，使 Web 用户能够共享个人信息、评论或者经验。内容通常是持续变化的，并且对于很多站点来说有助于提供一定程度的新鲜和活力。

博客支持工具使得不熟练的用户做很少的工作就可以创建自己的博客，构建和编辑他们的博客条目，创建一个月内或一年内的条目存档，并且通过使用户可以回复或评价内容来鼓励用户之间的交互。从设计的角度讲，博客可以用于在一个静态应用中增加沟通渠道，并且在创建社交网络中扮演了重要角色。从商业的角度讲，博客可以成为一种有力的营销工具。

Wiki 是一个支持用户编辑站点内容的 Web 站点。由此产生的合作式创作过程引起了一种社区式开发站点的出现。由此可以反映出一系列丰富的经验和观点。Wiki 常常用于支持特定团队的一些活动中（例如，作为一种项目文档工具），或者作为一种支持社交网络的方式。最著名的

Wiki 可能是 www. wikipedia. org[㊀]。

混搭应用（Mash-up）。混搭应用是一种混合 Web 应用，集成了多种来源（通常是第三方）的内容，目的是提供一个新颖的协作的结果。混搭应用通常使用简单的公共接口访问主要提供方（如 Google、Amazon 和 eBay）提供的丰富内容的环境。甚至在没有公共接口时，也经常有可能通过对站点可用数据进行简单“抓屏”来使用第三方数据。很多混搭应用的丰富交互活动代表着 Web2. 0[㊁]的发展方向。

AJAX。另外一种提供了丰富交互活动的技术就是 AJAX（异步 JavaScript 和 XML）。AJAX 通常是一组互补的技术，当结合使用时就可以创建高度交互的 WebApp，这种应用更像是一个桌面应用而不是 WebApp。这种方式使用 JavaScript 脚本来持续下载 XML 数据，为潜在使用这些数据做准备，结果是应用反应更迅速。最著名的 AJAX 应用之一是 maps. google. com。已经出现了很多 AJAX 工具集[㊂]，它们使得构建 AJAX 应用变得更容易。

聚合内容（Syndication）。聚合内容使得 WebApp 的部分可以在别处被包含。例如，你可能选择在一个新的 WebApp 中嵌入短小的在线“最新时讯”条目、小的“天气预报”功能或者最近的论坛公告。聚合内容的使用对于提供者（通过更多的曝光）和接收者（通过访问更丰富的内容）来说都能获益。存在很多聚合技术，毫无疑问最著名的就是 RSS（Really Simple Syndication）。从本质上来说，RSS 是一种 XML 数据格式，用来封装可以被聚合的内容。RSS 种子（feed）可以用于在其他网页中直接嵌入内容或者通过聚合器收集聚合内容供用户浏览。这两种情况的最终结果将是内容会更频繁地更新而与特定的用户更有关联性[㊃]。

Web 服务。Web 服务是一种软件组件（很可能是远程的），它使用特定的接口来提供定义好的功能。Web 服务支持的可能只是简单地验证一个电话号码是否符合标准格式，也可能是处理一个信用卡支付这样复杂的功能。Web 服务架构允许 Web 服务注册到一个服务代理上，然后允许客户程序定位和使用相关服务。事实上，复杂的 WebApp 可以通过组合很多服务来构建。

图 17-1 展示了构成 Web 服务架构的标准和技术集合。Web 服务描述语言（WSDL）用于描述 Web 服务接口。SOAP 是与 Web 服务进行通信的协议。其他的一系列相关标准是建立在该标准之上的。WS-BPEL（Web Services-Business Process Execution Language）描述了发生在业务过程中的状态转换交互以及如何将这些内部交互映射到 Web 服务操作上。

严格地讲，Web 服务并不关注于特定的 Web 应用类型，而是定义了一种支持创建应用的架构。然而，Web 服务的确简化了复杂应用的实现。例如，很多混搭应用的提供者就是通过 Web 服务使其内容可用的[㊄]。

元数据、本体和大众分类（folksonomy）。可以通过元数据（关于内容的信息）来使内容得到有效的使用。元数据最简单的形式是直接嵌入到 Web 页面中的信息。一个更丰富的表达是使用 RDF———一种产生有关资源声明的 XML 格式。本体（对领域的一种描述，可以作为推理的基础）可以建立在 RDF 之上，也可以产生语言，例如 OWL（Web Ontology Language）用于发布和共享描述特定领域的数据。

㊀ 参看 http：//en. wikipedia. org/wiki/BLOG、www. ojr. org/ojr/images/blog_software_comparison. cfm、http：//en. wikipedia. org/wiki/Wiki 和 www. wikimatrix. org 的 WikiMatrix 站点中相关的博客条目（在 2007 年 8 月 14 日访问）。

㊁ 参看 http：//en. wikipedia. org/wiki/Mashup_（web_application_hybrid）和 Programmable Web 站点 www. programmableweb. com/mashups 中关于 mash－up 的条目（在 2007 年 8 月 14 日访问）。

㊂ 参看 http：//en. wikipedia. org/wiki/Ajax_（programming）上的 AJAX 条目，http：//dmoz. org/Computers/ Programming/ Language/JavaScript/AJAX 上 dmoz Open Directory Project 站点中的 AJAX 文档（在 2007 年 8 月 14 日访问）。

㊃ 参看 http：//en. wikipedia. org/wiki/Web_syndication 和 http：//en. wikipedia. org/wiki/RSS_（file_format）。

㊄ 参看 Wikipedia 站点 http：//en. wikipedia. org/wiki/Web_services 和 W3C 站点 www. w3. org/2002/ws 上对 Web 服务的描述（在 2007 年 8 月 14 日访问）。

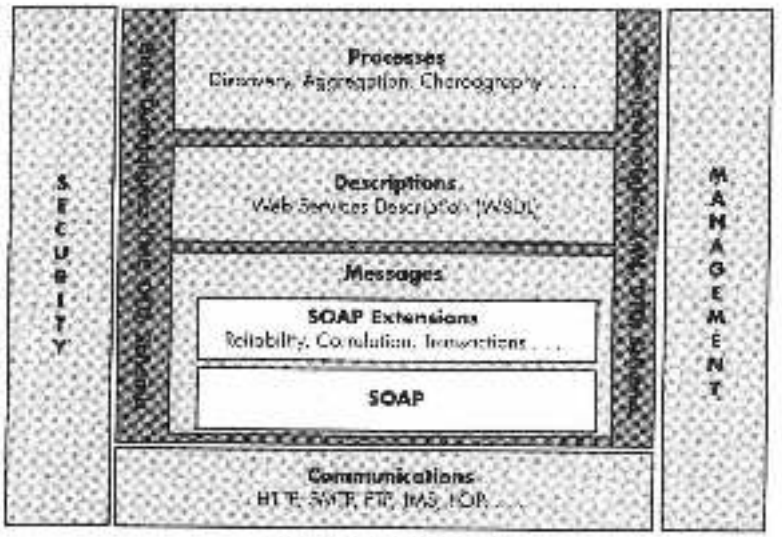


图 17-1 Web 服务架构栈

来源：www.w3.org/TR/2004/NOTE-ws-arch-20040211/上的 World Wide Web Consortium 版权。版权©2002 World Wide Web Consortium（Massachusetts Institute of Technology，European Research Consortium for Informatics，Keio University）。版权所有。转载获得了许可。

那么这些技术可能产生什么影响呢？它们可以用于创建高度结构化的信息，而这些信息可以通过不同方式进行分析和处理。例如，FOAF（Friend of a Friend）用于描述人们之间的关系。关于每个朋友的信息都可以用于构建适应性应用，这些应用会根据你的朋友是谁来定制信息。

大众分类是一种协作产生的分类系统，使用用户生成的标签和元数据来描述内容（加标签的过程通常称作标记）。其意思不是一个单独的分类，而是一组丰富且多面性的元数据集合。因为元数据中有一部分包含作者，用户可以识别出已经被分类的信息的子集（通过选择个体创建），使得信息更有意义[⊖]。

VoIP。IP 网络传输语音（VoIP）使得声音数据可以通过网络进行传输（而不是通过公共电话网络）。今天，VoIP 通过一些诸如 Vonage 和 Skype 等这样的应用对电信产生了重大影响，而且随着时间的推移，它将对 WebApp 的设计造成显著影响。例如，VoIP 使得 WebApp 可以支持播客（podcasting）和丰富用户服务体验。除此之外，它也会减少一些较为明显的差异，如在基于传输便件在不同媒体形式之间自动切换时的差异（例如，当使用手机或 PDA 等屏幕空间受限的设备时，在大屏幕上显示的文本会自动转换为 VoIP 声音流）。

位置感知（Location-aware）应用。随着 GPS 接收器变得更加普及，对于 Web 客户端来说将更有可能获得与客户端所在位置相关的信息。只要知道了地理位置，就可以开发应用以便可以根据用户当前所在的位置适当地定制用户体验。目前，这种方法的一个有限版本用于 WebApp 改变其展示的广告，其根据就是通过 IP 地址来推断用户位置。当 Web 访问变得更加普及，用户开始通过更多样的位置（和设备）来访问 WebApp，从本地化广告（"如果你有心情吃快餐，请访问 Dom 的面包店……我会在一分钟之内到达"）到交通警告（"在前方 0.25 英里处发生交通事故"），再到通用搜索（"最近的医院是……"）等任何信息。

⊖ 参看 W3C 站点中 www.w3.org/RDF/ 和 www.w3.org/2004/OWL 以及 Wikipedia 站点中 http://en.wikipedia.org/wiki/Ontology_(computer_science) 和 http://en.wikipedia.org/wiki/Folksonomy 上有关 RDF 和本体的信息（在 2007 年 8 月 14 日访问）。

这些仅仅是将在未来 Web 中扮演部分角色的少量技术[⊖]。有一些会成为大型技术中的一小部分，包含在 Web 发展的前景中。另外一些有可能会在基础的 Web 重构中变得更加关键。

17.2.3 在技术演化中应该考虑哪些关键问题[⊜]

随着 Web 2.0 的发展，有许多重要问题需要考虑。关键问题是：

可扩展性。Web 2.0 技术应该有能力进行扩展来满足高容量、任务关键的企业应用级要求。尽管可扩展性依赖于 WebApp 的特性和考虑的具体技术，但是大多数的 Web 2.0 技术已经被专门设计为分布式和可扩展的——从很多早期 Web 2.0 应用中的一些迹象中看出，可扩展性一般并不是一个重要的议题。

互操作性。Web 2.0 技术必须有能力有效地集成现存的遗留系统和数据库。目前，很多遗留应用被包装成 Web 服务。很多现存的混搭应用已经专门使用这种方式实现了。

适用性。并不是所有的商业都有机会来开发一个像 Google Map 这种创新的且令人兴奋的新产品或新系统。随之而来的问题是，哪些种类的 Web 2.0 技术可以成为 2000 强的公司为客户考虑使用的主流技术？Web 2.0 确实有若干对于很多公司来说是必需的元素。第一个就是增进的交互，通过采用诸如 AJAX 这样的技术来实现。第二，采用博客和 Wiki 的策略可以有效地吸引用户并鼓励他们参与。第三，被定制的与用户更有关联的更健壮的内容也是一种重要的业务能力。RSS 种子也许可以为此提供一种重要机制。最后应该考虑的是通过哪种形式可以使内容在别处也可使用——可能通过公共地访问相关的 Web 服务。

除了可扩展性、互操作性和适用性，另外的两个重要的问题也需要回答：

- **对于公开访问（通过应用编程接口 API）一个公司现今拥有的数据和功能而言，有哪些利益、风险和值得推荐的实践？**利益包括增加公司商品和服务的曝光率。例如，在线零售商可能提供访问商品目录和价格的功能，这会鼓励该零售商被包含到支持商品比较功能的集合站点上，这么做有可能会带来销售量的增加。而风险是这些比较也可能是有破坏性的。
- **对于允许公司客户对现存的基于 Web 的市场材料、商品描述和解决纠纷的 FAQ 提供“内容”而言，这么做有哪些利益、风险和值得推荐的实践？**利益在于出色的（至少是有帮助的）用户反馈、正面的表扬、有用的内容资源和更忠诚的客户。其不利的方面包括对公司产品和服务的负面观点，以及支持用户参与的相关系统的维护成本。

17.2.4 Web 2.0 将会如何发展

为了理解 Web 2.0 在未来会如何发展，应该先来了解早期 Web（Web 1.0）的发展。正如我们在本书前面提到的，Web 开始于非常简单的内容驱动的站点，发展到包含更复杂信息源的 WebApp，然后通过增加对更先进功能（比如电子商务应用）的支持来适应用户和团体要求变化的需求。Web 2.0 技术现在可以将用户的主动参与添加到内容的使用中。

如图 17-2 所示，下一步发展就是将交互和用户参与相同程度地添加到 WebApp 功能方面中作为目前存在的主动内容。想象可以在搜索引擎中借用他人的过滤算法，可以把你的账单支付安排上传到在线银行，或者可以参与到一个新的定制 WebApp 的协同创建中？所有的这些（甚至

⊖ 对于一个关于全面 Web2.0 前景的出色讨论，请访问 Web Design from Scratch 站点，www.webdesignfromscratch.com/future-social-web-experience.cfm（在 2007 年 8 月 14 日访问）。

⊜ 有关一系列有趣的问题和议题，一篇关于 Web2.0 的征稿中对此提出的问题参看 www.cutter.com/content-and-analysis/journals-and-reports/cutter-it-journal/callforpaper01.html（在 2007 年 8 月 14 日访问）。

更多）都会随着 Web 2.0 的发展而出现，你将有能力随时随地来进行这些活动。

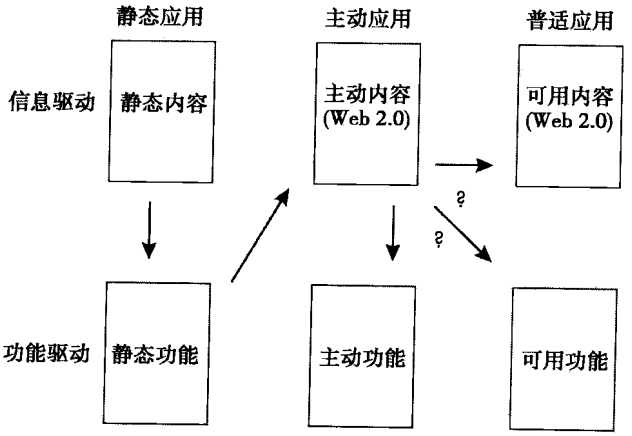


图 17-2 Web 2.0 的发展

17.3 对于未来的观点

预知绝对不容易，预测 Web 的方向及其社会影响的工作至少也是值得怀疑的。虽然这么说，但是尝试想象 Web 2.0 的“未来”依然很诱人。

在2004 年，Robin Sloan［Slo06］制作了一个虚构的电影片段，提供了一个展示关于 Web、媒介和我们每个人可能的未来的窗口。下面是原始电影的副本，网址 http：//epic. makingithappen. co. uk/new-master1. html[⊖]上有更新的版本。

2014 年，人类对信息的访问已经达到了在较早时期难以想象的广度和深度。

每个人都以某些方式来做出贡献。

每个人都参与其中，创建了一种鲜活逼真的媒体风景。无论如何，如你所知的新闻媒体已经不复存在。新闻界（Fourth Estate）的财富已经衰减。20 世纪的新闻机构成为古董，它们已经是遥远过去的孤独残余。

通向 2014 的道路开始于 20 世纪中期。

1989 年，瑞士欧洲粒子物理研究所（CERN）旗下的量子物理实验室的一位计算机专家 Tim Berners-Lee，发明了 WWW（万维网）。

1994 年，Amazon. com 成立。其年轻的创始人梦想有一家无所不售的商店。Amazon 的模型设定了网络销售标准，即建立一套自动化的个性化评价体系——一种能够提出建议的商店。

1998 年，两位斯坦福（Stanford）的程序员创建了 Google。他们的算法模仿了 Amazon 语言，Google 把链接看作是评价，在这个基础开始，这家公司创建了世界上最有效的搜索引擎。

1999 年，TiVo 颠覆了电视业，它将人们从时间（和商业）限制中解放出来。体验过 TiVo 的人，几乎没有想让它变回原来样子。

这一年，一家名为 Pyra Labs 的网络公司开始推出一种个人信息发布工具 Blogger。

Friendster 在 2002 年投放市场，成千上万的年轻人涌向其中，它的走红勾勒了异常

⊖ 该材料版权归 Robin Sloan 和 Matt Thomson 所有，这里的复制是经过许可的。

详细的年轻人的生活、兴趣爱好和社交网络等的地图。同样是在2002年，Google推出了一个新闻门户GoogleNews。新闻机构声称其违反规则。GoogleNews全部是由计算机来进行编辑的。

2003年，Google收购了Blogger。Google的意图成为一个秘密，但是他们对Blogger的兴趣并非不切实际。2003年是博客年。

2004年将作为每件事开始起步的年份而被人们记住。

Reason Magazine给订阅者发行的期刊中，将订阅者房屋的卫星照片放到了封面上，并针对每个订阅者对内容信息进行了裁剪。

Sony和Phillps揭开了世界上第一个个批量生产电子出版物的面纱。

Google推出了Gmail，为每个用户提供了1G大小的免费空间。

微软推出了一种社会新闻过滤器Newshot。

Amazon推出了一种构建在Google技术上的搜索引擎A9，它又与Amazon自己的商品推荐相结合。

随后，Google上市。

在补充了新的资金之后，Google做出了一个重大的收购——收购了TiVo。

2005年，作为对Google近期动作的回击，微软收购了Friendster。

2006年，Google将其所有的服务——TiVo、Blogger、Gmail、GoogleNews及其全部的搜索整合进了Google Grid综合平台中，这个平台提供了不受存储空间和带宽限制的功能，用于存储和共享各种不同媒体，永远在线，可以随处访问。每个用户可以选择他们自己级别的隐私。他可以在Google Grid中安全地存储内容或者公诸于众。它前所未有的方便，每个人都可以创建媒体，同时也可以消费媒体。

2007年，Microsoft开始用Newsbotster对Google的挑战做出回击，Newsbotster是一个社会新闻网络，一个新闻分享平台。它根据每个用户的朋友和同事正在阅读和浏览的内容对新闻进行分类和排序，同时允许每个人评论正在查看的内容。这一年，Sony的ePaper已经比实际的纸张更便宜。它成为Newsbotster的一种媒介。

2008年看到了挑战微软野心的联盟。Google和Amazon联合组建了Googlezon。Google提供Google Grid和独一无二的搜索技术，Amazon提供了社会评价引擎和它巨大的商业架构。两家公司合而为一后，利用了各自有关每个用户的社交网络、地理人口统计、消费习惯与兴趣的详细信息，提供完全定制化的内容以及广告服务。

2010年的新闻大战没有真正的新闻机构参与其中。

Googlezon凭借其无法匹敌的巨大软件特性最终打败了微软。通过使用全新的算法，Googlezon的计算机能够动态地生成新闻故事，从所有的内容来源和评价体系中剥离句子并分析事实，然后进行重新组合。计算机为每一个用户编写新闻故事。

2011年，沉睡的新闻界觉醒了，他们开始了首次也是最后一击。纽约时报（New York Times）公司起诉Googlezon，控告该公司的自动事实分析机制违反了版权法。这一诉讼最后由最高法院受理，2011年8月4日，最高法院最终判决支持Googlezon。

2014年3月9日，星期日。Googlezon推出了EPIC。

欢迎来到我们的世界。

“演化的个性化信息结构（Evolving Personalized Information Construct，EPIC）”是一种让混乱无序的媒体被过滤、条理化并发行的系统。现在，每个人都能够投稿——从博客文章到照相手机相片，再到视频、长篇调查报告。很多人为此得到了回报——根据投

稿的受欢迎程度从 Googlezon 巨大广告收入中抽出少许即可。

EPIC 为每个用户提供了一个定制化的内容包，依据使用者的选择、消费习惯、兴趣爱好、地理位置以及社交网络来定制他的产品。

新一代的自由编辑已经兴起，他们出售自己的能力用于 EPIC 的连接、过滤和优先级排序内容。

EPIC 允许我们混合和匹配自己想要的选择来订阅很多的编辑。更好的是 EPIC 超越过去来为内行的读者更深入、更广泛地一览这世界更微妙事务。

在未来，这些都将会实现吗？如果实现了，距离 Sloan 的设想将会有多远呢？只有时间可以告诉我们。不过，我们认为这种设想至少没有提到关于 Web 未来的方面。未来的 Web 不仅仅依赖于我们所有人参与对丰富复杂内容的创造和使用，也将依赖于我们所有人加入对丰富复杂功能的构建和使用。同样，Flickr 允许用户共享照片；YouTube 允许用户共享视频；我们可以想象未来的一个站点允许我们创造、参与和共享复杂的任务——例如，共享 SafeHomeAssured. com 中的安全传感器的监控。

17.4 Web 工程的变化特性

技术会变化，工具会演化，更好的方法会出现，旧方法会逐渐消失，但是人始终保持不变。如果我们想通过对过去半个世纪的观察来获得一些知识，那就是——大多数的技术问题都可以解决，但是与技术的发展和使用相关的人的问题将会有不断的挑战。

在 Web 工程下一个十年的发展中，我们面临的最大挑战依然保持不变：

- 认识到利益相关者的需求是极为重要的，但与此同时要明白我们交付高质量且健壮的解决方案的快速程度是有限制的。
- 精化 Web 工程的过程，使其是敏捷的、可适应的并且有效的，认识到过程不应该妨碍得到一个好的解决方案，然而根本没有过程总是会导致一个差的解决方案。
- 认识到一个期望的 WebApp 解决方案的特征经常会在开发过程中有所演化，并且认识到创建一个正确的解决方案经常会包含存在于新系统和现有过程及行为的变化之间复杂的相互影响——而且很多变化直到系统投入运行以后才会完全明了。
- 开发方法和技术让我们对问题有更好的认识，对基于 Web 的解决方案有更好的理解。也要认识到方法和技术不能替代人员沟通和对问题有意义的理解。
- 为了适应 21 世纪业务环境的动态特性而对变更进行管理，同时避免不能控制的变更造成的混乱。

作为 Web 工程师，我们将继续在现今世界上更动态、更让人兴奋的技术领域里面工作。事实上，现代世界中的每件事以及每个人都通过某种方式连接到 Web 之上。我们的工作就是确保 Web 服务于世界，使生活更加美好。

参考文献

[BBC06] BBC News, "Building on the Future of the Web," May 22, 2006, http://news.bbc.co.uk/2/hi/technology/4994570.stm (accessed August 14, 2007).

[Ber99] Berners-Lee, T., *Weaving the Web*, Harper San Francisco, 1999.

[Car05] Carr, N., "Where's the OS?" August 24, 2005, www.roughtype.com/archives/2005/08/jason_kottke_of.phpm (accessed August 14, 2007).

[CNE05] CNET News. com, "From Web Page to Web Platform," August 16, 2005,
http: //www. news. com/From + Web + page + to + Web + platform/2100-7345_3-5833940. html?tag = nl (accessed August 14, 2007)

[Coa06] Coates, T., "My' Future of Web Apps' Slides…" February 13, 2006,
www. plasticbag. org/archives/2006/02/my_future_of_web_apps_slides (accessed August 14, 2007).

[Kot05] Kotke, J., "GoogleOS? YahooOS? MozillaOS? WebOS?" August 23, 2005, www. kottke. org/05/08/googleos-webos (accessed August 14, 2007).

[ORe05] O'Reilly, T., "What Is Web2. 0: Design Patterns and Business Models for the Next Generation of Software," September 30, 2005,
www. oreillynet. com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20. html (accessed August 14, 2007).

[Slo06] Sloan, R., and M. Thomson, A video produced by the fictional "Museum of Media History," 2006, http: //epic. makingithappen. co. uk/new-master1. html. (accessed August 14, 2007).

[Wik06] Wikipedia, "Semantic Web," 2006, http: //en. wikipedia. org/wiki/Semantic_Web (accessed August 14, 2007).

[Zaw06] Zawodny, J., "Tom' s Future of Web Apps, Translated for Product Managers," 2006, http: //jeremy. zawodny. com/blog/archives/006323. html (accessed August 14, 2007).