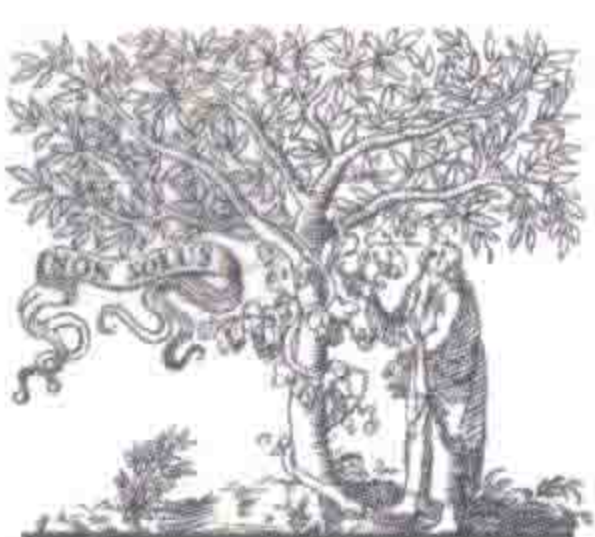




ELSEVIER
爱思唯尔

面向对象信息系统 的设计思想

(美) 劳尔·西德尼·瓦兹拉威克 著
吴 骅 王学昌 杨莉灵 译

Object-Oriented Analysis and
Design for Information Systems

清华大学出版社

面向对象信息系统的设计思想

(美) 劳尔·西德尼·瓦兹拉威克 著

吴 骅 王学昌 杨莉灵 译

清华大学出版社

北 京

内 容 简 介

本书详细阐述了与面向对象信息系统设计思想相关的基本解决方案,主要包括业务建模,高级别需求,基于用例的项目规划,扩展用例,概念建模,功能建模与 OCL 契约,域层设计,代码生成,测试,接口层设计与 IFML,数据持久性等内容。此外,本书还提供了丰富的示例以及代码,以帮助读者进一步理解相关方案的实现过程。

本书适合作为高等院校计算机及相关专业的教材和教学参考书,也可作为相关开发人员的自学教材和参考手册。

ANÁLISE E PROJETO DE SISTEMAS DE INFORMAÇÃO ORIENTADOS A OBJETOS 3ª EDIÇÃO

Raul Wazlawick

ISBN: 9788535279849

© 2013 Elsevier Editora Ltda. All rights reserved.

Object-Oriented Analysis and Design for Information Systems, 1e

Raul Sidnei Wazlawick

ISBN: 9780124186736

Copyright © 2014 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by Elsevier (Singapore) Pte Ltd Press and Tsinghua University Press.

Copyright © 2018 by Elsevier (Singapore) Pte Ltd and Tsinghua University Press. All rights reserved.

Published in China by Tsinghua University Press under special arrangement with Elsevier (Singapore) Pte Ltd..

This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授予清华大学出版社在中国大陆地区(不包括香港、澳门特别行政区以及台湾地区)出版与发行。未经许可的出口,视为违反著作权法,将受法律制裁。

本书封底贴有 Elsevier 防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号 图字: 01-2016-3551

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

面向对象信息系统的设计思想/(美)劳尔·西德尼·瓦兹拉威克著;吴骅,王学昌,杨莉灵译. —北京:清华大学出版社,2018

(书名原文: Object-Oriented Analysis and Design for Information Systems, 1st Edition)

ISBN 978-7-302-47604-7

I. ①面… II. ①劳… ②吴… ③王… ④杨… III. ①信息系统-系统分析 ②信息系统-系统设计
IV. ①G202

中国版本图书馆 CIP 数据核字(2017)第 155191 号

责任编辑: 钟志芳

封面设计: 刘 超

版式设计: 魏 远

责任校对: 马子杰

责任印制: 刘海龙

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市铭诚印务有限公司

经 销: 全国新华书店

开 本: 203mm×260mm

印 张: 22.5

字 数: 588 千字

版 次: 2018 年 6 月第 1 版

印 次: 2018 年 6 月第 1 次印刷

印 数: 1~4000

定 价: 120.00 元

产品编号: 061041-01

译者序

围绕统一建模语言（UML）和统一过程（UP），由对象管理组织（OMG）选定为国际标准的信息系统软件开发已经持续了几十年，故有必要出现一本为软件工程师提供实用且经过实践检验的方法来进行分析和设计的书籍。

与介绍本领域的其他书籍不同的是，其他书籍大多是对 UML 模型图的罗列陈述以及对其各种用法的介绍，本书则关注分析师和设计师的日常工作，并解释在开发过程中各种模型图的异同，以更好地开发和优化系统。

本书着重于分析师和设计师日常涉及的活动，包括业务建模、需求工程、分析和设计等，也涉及项目管理、实施、测试等内容。整本书使用一系列的案例来解释分析和设计方面的技术，引入了基本的原则，并扩展到一些切实可行的技术，将其用于开发和设计表现，以及使用 UML 模型等。

本书的翻译由吴骅组织完成，参与本书翻译的还有王学昌、林展宏、周娟、刘红军、王玲、郑正正、秦双夏、莫鸿强、李远明、陶日然、黄善斌、廖义奎、杨莉灵等人，感谢各位参与者的帮助。没有他们的帮助就无法完成这项工作。由于水平有限，译文中的不当之处在所难免，恳请同行及各位读者朋友不吝赐教。

译者

序 言

在软件工程项目实施初期，人们往往会找不到头绪，而要统一开发团队成员的意见有时也很困难，同时，信息和功能需求往往又不清晰，利益相关者（客户）的需求也会经常发生变化，要明确所有需求对开发者来说过于理想化，因此有必要有一本书为软件工程师提供实用的、经过实践检验的方法来进行分析和设计。Raul Wazlawick 就为我们提供了这样一本书。

在过去的半个多世纪里，也曾经出现过很多计算机软件分析和设计的方法，但在我看来，采用面向对象的方法是最直接和有效的。在软件开发过程早期的分析和设计方案都是不稳定的，经常需要修改，但“对象”往往是问题中所有元素中最容易发现的。

面向对象范式具有极好的弹性，使开发者可以在概念级别上用人们易于理解的方法来描述一个对象，然后通过迭代和细化的过程，开发人员可以在底层抽象地提炼这些对象并有效利用。在本书中，Wazlawick 教授为读者提供了完整的指南。

在本书的一开始，他和其他优秀的分析师一样，使用 UML 作为标志性的方法开展工作。通过启发、细化、表示等过程，为读者提供了理解问题所需的工具。他着重使用用例，帮助读者掌握实例中对于理解问题以及做好项目计划有重要意义的优秀机制。

他坚信优秀的分析师和设计师必须反复工作，这样才能更好地完成工作和更详细地理解要解决的问题。为了达到此目的，他为读者提供了解决问题的一些面向对象技术和 UML 工具。

本书使用一系列的案例来解释分析和设计方面的技术，引入了基本的原则，扩展到一些切实可行的技术，将其用于开发和设计表现，以及 UML 模型使用等。

在职业生涯中，我与很多公司的众多软件专家都有过接触。不同的公司和不同的人对软件开发过程中遇到的问题有不同的看法，也有各自针对这些问题的解决之道。但是很少有人会对下面这句话提出异议：“如果你不知道目的地在哪，你永远都到不了那儿（If you don't know where you're going, it is very, very difficult get to there）”。

软件专家需要一个明确的“目的地”和一个“路线图”才能更好地分析和设计问题。在本书中，Raul Wazlawick 为大家提供了用于绘制“路线图”的工具，通过本书，你可以到达你所需要去的“目的地”。

Roger S. Pressman 博士

作者简介

Raul Sidnei Wazlawick 是巴西弗洛里亚努波利斯 UFSC 大学的计算机科学副教授。他拥有计算机科学学士和硕士学位、工程学博士学位（1993 年）。Wazlawick 曾是 IFIP 工作组信息化委员会和高等教育信息通信技术委员会主席、巴西计算机协会董事、软件工程巴西研讨会（2006 年）大会主席、IFIP 计算机教育问题世界会议（2009 年）联席主席。截至目前，他已经用葡萄牙语出版了 3 本教科书并公开发表了近 100 篇论文。他从 1986 年就开始从事面向对象系统的相关工作，主要研究领域为面向对象的软件工程。他曾带领巴西公共卫生系统^①的一个软件开发团队，并经常在大学、公司和政府机构做与软件工程相关的主题报告。20 世纪 90 年代初，他开始成为一名软件工程顾问。

^① <http://dab.saude.gov.br/portaldab/esus.php>

致 谢

在这里，我要感谢通过各种方式帮助我促成本书编写的人们：早在 1987 年，Luiz Fernando Bier Melgarejo 教授就开始引导我走进面向对象系统的世界；我的同事 Marcos Eduardo Casa 教授，在 20 世纪 80 年代，当面向对象的概念还被认为是“来自外星球的东西”时，他就开始从事结对编程工作直到现在；我的同事 Antonio Carlos Mariani 教授开发的 Actors World 工具能帮助我们讲述面向对象程序设计的课程；同时还要感谢公司允许我在本书中用实际的生产环境来介绍这些技术；我的学生 Everton Luiz Vieira、Kuesley Fernandes do Nascimento 和 Iuri Cardoso 跟我一起整理了当中的一些新技术理论；圣卡塔琳娜州联邦大学（Federal University of Santa Catarina, UFSC）的信息和统计学院（Department of Informatics and Statistics, INE）为开展这项工作提供了机会；另外特别感谢软件工程师 Gilmar Purim 为本书第 1 版（2003 年出版）雏形所提供的富有成效的帮助。

我还要感谢学习了我所开设的面向对象分析与设计这门课程的一百多位学生们。他们所遇到的疑问、误区和困难促使我进一步地研究和学习；感谢我的朋友 Rogério Cid Bastos 教授为我指明了方向并给予了我前进的动力，最后感谢我的朋友和兄弟们给我带来的欢乐和友爱。

前 言

本书采用说教和深入研究相结合的方式介绍开发信息系统的过程中面向对象元素分析和设计的要素。

与介绍本领域的其他书籍不同的是，其他书籍大多是对 UML 模型图的罗列陈述以及对其各种用法的介绍，本书则关注于分析师和设计师的日常工作，并解释在开发过程中各种模型图的异同，以更好地开发和优化系统。

尽管距面向对象方法的提出已经有四十多年，但还是有许多机构觉得用它来开发高质量的软件存在一定困难，特别是在他们所开发的系统缺乏可维护性和灵活性时。本书所介绍的信息和技术主要是为了弥补这些不足。

作者已将书中所介绍的技术成功应用在了许多软件开发公司以及一些高校的科研项目当中。

本书主要面向计算机专业人员（分析师、设计师以及程序员等）、计算机科学专业的学生，包括具有一定面向对象开发或建模经验的本科生和研究生。初学者和具有一定经验的读者会发现本书的内容会对提高他们的项目质量有很大的帮助。而本书的目的就是要帮助他们更好地完成设计。

与按照事件活动的顺序进行分析和设计的瀑布模型（Royce, 1970）不同的是，类似 UP 这样的迭代模型提出这些活动（也称为事件）在任何阶段内均十分重要。

本书将更多地着重于分析师和设计师日常会涉及的活动，包括业务建模、需求工程、分析和设计等，也涉及项目管理、实施、测试等内容。对于其他内容的详细信息，可以参考 2010 年 Pressman 出版的软件工程类图书，或 2003 年 Kruchten 出版的有关统一过程的图书。

UP 初始阶段的目标是建立系统的常规视图及其上下文，并计划下一步的开发。该阶段的两个主要建模工具分别是概念建模（conceptual modeling）（第 6 章和第 7 章内容）和高级用例（high-level use case）（第 3 章内容）。第 2 章业务建模（business modeling）的重点程度是根据需要开发项目的进展情况决定的。

在初始阶段的后期，高级用例可用于作为该项目其余部分（第 4 章内容）的基础规划。通常情况下，该规划涉及用例的评级，根据它们的复杂性及其优先级分为小型、中型或者大型等。当每个用例的复杂性已知时，就可以估算开发这些用例所需的平均时间（Karnen, 1993），因此可能会计划将迭代操作作为一组活动，这些活动在可利用的时间内是切实可行的。

统一过程的细化阶段完成一系列的迭代操作。每一个迭代都会被分配给一组要开发的用例^①。开发独立的用例通常会从扩展到它的流开始（第 5 章内容）。当然，用例可能会被表示为系统时序图（system sequence diagram），以帮助发现必须被执行的系统查询和命令。

在细化阶段，随着架构体系的演变，概念模型（conceptual model）会不断地进行修改和完善（第 6 章和第 7 章内容）。

^① 迭代可能会有一些相应的计划用于处理已知的风险，或由客户、用户甚至开发团队产生的需求变更。

系统查询和系统命令可能由某种编程语言实现,但开发人员也许会为系统查询和系统命令指定一些契约 (contracts) (第 8 章内容),这对于定义测试用例可能会有所帮助 (第 11 章内容)。这些契约相当于系统的功能模型,定义了每一个查询和命令所涉及的输入内容和输出结果。

经过上述阶段,现在可能已经开始实施系统,或者已有相关工具自动生成代码。不过,此时还是追求更为简洁的代码,以利于为每个系统操作新生成一些图集,如时序图或通信图等 (第 9 章内容)。这些图指明不同的对象如何交换信息,以便明确职责并获得预期的契约中所指定的后验条件。通过这个过程,设计模式更容易应用到代码中。这些图同时也指明每个类中需要实施的方法 (除了 `getter` 函数和 `setter` 函数以外),以及如何实施等。

在细化阶段,要完成接口 (界面) 的设计 (第 12 章内容)。本书介绍了 IFML 算法^②作为系统建模的一个可选工具。IFML 是一个全新的面向信息流的建模语言标准,2013 年 3 月起被 OMG 标准化组织采用。

虽然构建阶段更强调数据库结构的生成 (第 13 章内容) 和代码的运行 (第 10 章内容),但细化阶段也同样会涉及这些内容。如果有合适的工具,也可以自动生成持久性数据。但即使其是自动化的,也应该了解该持续机制如何运行。同样,代码生成也可以表示为一些自动化的规则。

互联网上提供了很多练习供读者参考,方便读者更深入地理解各项内容。

^② <http://www.ifml.org/>。

目 录

第 1 章 概述.....	1
1.1 关于本书.....	1
1.2 面向对象系统开发.....	2
1.3 统一建模语言 (UML)	2
1.4 统一过程 (UP)	3
1.5 当前进度.....	5
1.6 思考题.....	5
第 2 章 业务建模.....	6
2.1 业务建模概述.....	6
2.2 系统概要视图.....	8
2.3 业务用例.....	10
2.3.1 业务参与者与业务工作者	11
2.3.2 自动化的机遇.....	12
2.4 业务活动图.....	14
2.4.1 基本元素.....	14
2.4.2 控制流节点.....	16
2.5 业务状态依赖.....	18
2.6 其他说明.....	21
2.7 当前进度.....	22
2.8 思考题.....	22
第 3 章 高级别需求	23
3.1 高级别需求概述.....	23
3.2 系统参与者.....	24
3.3 系统用例.....	24
3.3.1 独立会话.....	26
3.3.2 交互.....	27
3.3.3 一致性结果.....	28
3.3.4 基本用例.....	29
3.3.5 简要用例.....	29
3.3.6 系统边界.....	29
3.4 如何在业务模型中发现系统用例.....	30
3.5 需求.....	32

3.5.1	需求获取.....	33
3.5.2	需求获取不是需求设计.....	34
3.5.3	需求挑战.....	35
3.5.4	显性和隐性的功能需求.....	35
3.5.5	非功能性需求.....	36
3.5.6	非功能性需求的持久性和短暂性.....	36
3.5.7	强制性需求和希望性需求.....	37
3.5.8	补充需求.....	38
3.6	初步概念模型.....	41
3.7	当前进度.....	45
3.8	思考题.....	45
第4章	基于用例的项目规划.....	46
4.1	软件项目中的工作量估算和风险分析.....	46
4.1.1	Ad hoc 方法.....	46
4.1.2	参数法.....	47
4.1.3	风险分析.....	49
4.2	用例点分析.....	51
4.2.1	UAW (未调整参与者权重).....	52
4.2.2	UUCW (未调整用例权重).....	53
4.2.3	UUCP (未调整用例点).....	54
4.2.4	TCF (技术复杂度因子).....	54
4.2.5	EF (环境因子).....	60
4.2.6	UCP (已调整用例点).....	63
4.2.7	工作量.....	63
4.2.8	日历时间与团队平均规模.....	64
4.2.9	详细用例的计数方法.....	66
4.3	规划迭代项目.....	66
4.3.1	估算迭代周期.....	67
4.3.2	迭代数量.....	67
4.3.3	每用例点的工作量.....	68
4.3.4	团队负载能力.....	68
4.3.5	定义用例优先级.....	69
4.3.6	规划阶段和迭代.....	70
4.4	当前进度.....	72
4.5	思考题.....	72
第5章	扩展用例.....	73
5.1	扩展用例概况.....	73

5.2	主要流.....	74
5.3	替代流.....	76
5.3.1	场景.....	76
5.3.2	变体.....	77
5.3.3	异常处理.....	78
5.4	书写规范.....	82
5.4.1	基本用例及实际用例.....	83
5.4.2	明确的信息.....	84
5.4.3	身份验证与选择.....	84
5.4.4	强制性步骤.....	84
5.4.5	补充步骤.....	87
5.4.6	不恰当步骤.....	87
5.5	包含用例与片段.....	88
5.6	版型化用例的扩展.....	89
5.6.1	报表扩展.....	90
5.6.2	CRUD 扩展.....	90
5.7	扩展用例的其他内容.....	94
5.7.1	利益相关者.....	94
5.7.2	先验条件.....	94
5.7.3	成功的后验条件.....	95
5.7.4	已知问题.....	95
5.8	系统时序图.....	95
5.8.1	时序图元素.....	95
5.8.2	将用例扩展为系统时序图.....	97
5.8.3	将接口连接到 façade 控制器.....	99
5.8.4	无状态机制.....	102
5.8.5	有状态机制.....	103
5.8.6	系统时序图中的替代流.....	104
5.9	当前进度.....	108
5.10	思考题.....	108
第 6 章	概念建模：基础篇.....	109
6.1	概念建模概述.....	109
6.2	属性.....	111
6.2.1	属性类型.....	111
6.2.2	初始值.....	112
6.2.3	派生属性.....	112
6.2.4	枚举.....	113
6.2.5	基本类型.....	114

6.3	概念.....	115
6.3.1	唯一属性.....	115
6.3.2	系统控制类.....	116
6.4	关联.....	116
6.4.1	规则重数.....	118
6.4.2	关联方向.....	119
6.4.3	派生关联.....	120
6.4.4	聚合与组合.....	122
6.4.5	n 元关联.....	123
6.5	集合.....	125
6.5.1	集.....	125
6.5.2	有序集.....	126
6.5.3	Bag.....	126
6.5.4	序列.....	126
6.5.5	映射.....	127
6.5.6	分区.....	128
6.5.7	关系.....	129
6.6	概念模型组织.....	129
6.6.1	泛化、特化与继承.....	130
6.6.2	关联类.....	132
6.6.3	模态类.....	135
6.7	常量.....	138
6.8	概念模型的迭代架构.....	141
6.8.1	如何发现概念和属性.....	141
6.8.2	依赖和独立概念.....	144
6.8.3	如何发现关联.....	146
6.8.4	概念模型迭代结构的示例.....	147
6.9	当前进度.....	149
6.10	思考题.....	150
第 7 章	概念建模：模式.....	151
7.1	概念模型模式概述.....	151
7.2	高内聚.....	151
7.3	规约类.....	154
7.4	数量.....	155
7.5	测量.....	156
7.6	策略.....	156
7.7	组合.....	158
7.8	组织层级.....	159

7.9	对象连接.....	159
7.9.1	复制并替换.....	160
7.9.2	替代.....	160
7.9.3	本质/现象.....	161
7.9.4	解除连接.....	162
7.10	账户/交易.....	162
7.11	范围.....	166
7.12	时态模式.....	167
7.12.1	有效性.....	167
7.12.2	历史记录.....	167
7.12.3	时态.....	168
7.12.4	双时态.....	169
7.13	讨论.....	170
7.14	当前进度.....	170
7.15	思考题.....	171
第 8 章	功能建模与 OCL 契约.....	172
8.1	功能建模介绍.....	172
8.2	先验条件.....	175
8.2.1	参数保证.....	176
8.2.2	互补约束.....	176
8.2.3	先验条件保证.....	177
8.2.4	先验条件、异常与常量.....	178
8.3	瞬态关联.....	178
8.4	查询返回.....	179
8.5	后验条件.....	181
8.5.1	改变属性值.....	182
8.5.2	创建实例.....	183
8.5.3	添加链接.....	184
8.5.4	删除实例.....	185
8.5.5	删除链接.....	186
8.5.6	标准格式后验条件.....	187
8.5.7	后验条件组合.....	187
8.5.8	前值.....	188
8.5.9	覆盖对象集合的后验条件.....	189
8.5.10	后验条件与真实事件.....	190
8.6	异常.....	190
8.7	CRUD 的模式契约.....	192
8.7.1	创建契约.....	192

8.7.2	更新契约.....	193
8.7.3	删除契约.....	194
8.7.4	检索契约.....	197
8.8	对象列表模式契约.....	197
8.9	用例相关契约.....	198
8.10	当前进度.....	199
8.11	思考题.....	200
第 9 章	域层设计.....	201
9.1	域层设计概述.....	201
9.2	对象职责分配.....	202
9.3	可见性.....	205
9.3.1	关联可见性.....	205
9.3.2	参数可见性.....	211
9.3.3	本地声明可见性.....	214
9.3.4	全局可见性.....	215
9.4	基于后验条件的动态建模.....	215
9.4.1	实例创建.....	216
9.4.2	链接添加.....	218
9.4.3	属性值修改.....	220
9.4.4	实例销毁.....	220
9.4.5	移除与替换链接.....	221
9.4.6	条件式后验条件.....	222
9.4.7	异常.....	223
9.4.8	与集合有关的后验条件.....	224
9.5	系统查询.....	225
9.6	委托与低耦合.....	227
9.7	设计类图.....	230
9.8	当前进度.....	232
9.9	思考题.....	233
第 10 章	代码生成.....	235
10.1	代码生成简介.....	235
10.2	类与属性.....	235
10.3	单向关联.....	238
10.3.1	单向关联（一对一）.....	240
10.3.2	单向关联（一对多）.....	243
10.3.3	单向限定关联.....	243
10.3.4	有关联类的单向关联.....	246

10.4	双向关联.....	248
10.4.1	合作伙伴.....	248
10.4.2	单向实现.....	253
10.4.3	带有中间对象的实现.....	254
10.5	委托方法及系统操作.....	255
10.6	过滤查询模式.....	257
10.7	当前进度.....	259
10.8	思考题.....	260
第 11 章	测试.....	261
11.1	测试简介.....	261
11.2	功能测试.....	262
11.2.1	等价类划分.....	262
11.2.2	限值分析.....	263
11.3	存根和驱动程序.....	264
11.4	测试驱动开发.....	265
11.5	单元测试.....	265
11.6	系统操作测试.....	269
11.7	用例测试（系统、验收和业务循环测试）.....	271
11.8	当前进度.....	273
11.9	思考题.....	275
第 12 章	接口层设计与 IFML.....	276
12.1	接口层设计简介.....	276
12.2	交互流建模语言（IFML）.....	276
12.3	视图组件.....	277
12.3.1	Details.....	278
12.3.2	Multiple details.....	279
12.3.3	Simple List.....	280
12.3.4	List.....	280
12.3.5	Checkable List.....	281
12.3.6	Form.....	282
12.3.7	Hierarchy.....	283
12.4	Page.....	284
12.5	Flow.....	285
12.5.1	通用导航流.....	285
12.5.2	数据流.....	287
12.5.3	参数绑定.....	287
12.5.4	多值参数绑定.....	289

12.6	超文本组织.....	290
12.6.1	站点视图.....	291
12.6.2	区域.....	291
12.6.3	主页面、地标页面和默认页面.....	291
12.7	Web 接口模式.....	292
12.7.1	级联索引.....	292
12.7.2	过滤索引.....	293
12.7.3	导览.....	294
12.7.4	视角.....	295
12.7.5	概览加描述.....	295
12.7.6	顶层导航.....	296
12.8	接口中的建模操作.....	298
12.8.1	创建操作.....	298
12.8.2	删除操作.....	299
12.8.3	更新操作.....	300
12.8.4	链接、断开和重新链接操作.....	301
12.9	CRUD 操作的 IFML 模型.....	303
12.10	使用 IFML 的用例接口建模.....	306
12.11	当前进度.....	308
12.12	思考题.....	310
第 13 章	数据持久性.....	311
13.1	数据持久性简介.....	311
13.2	对象关系映射 (ORM).....	312
13.2.1	类与属性.....	312
13.2.2	关联.....	315
13.2.3	继承.....	324
13.3	保存及加载对象.....	326
13.3.1	虚拟代理.....	326
13.3.2	代理器与物化.....	329
13.3.3	缓存.....	330
13.4	整体过程回顾.....	332
13.5	思考题.....	334
后记		335
参考文献		336