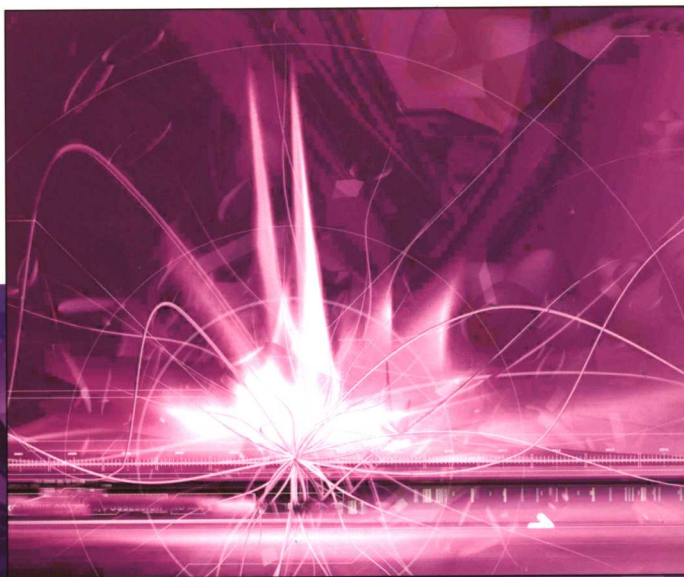


● 面向对象软件开发



# 程序设计 的模式语言 · 卷 1

Pattern Languages  
of Program Design 1



James O. Coplien 编  
Douglas C. Schmidt

谢文亮 梁金昆 译



清华大学出版社

# 程序设计的模式语言·卷 1

James O. Coplien  
Douglas C. Schmidt 编  
谢文亮 梁金昆 译

清华大学出版社  
北 京

Authorized translation from the English Language edition, entitled PATTERN LANGUAGES OF PROGRAM DESIGN 1, 1ST Edition by James, Douglas, published by Pearson Education, Inc, publishing as Addison Wesley, Copyright©1998 by Addison Wesley Longman, Inc.  
EISBN: 0-201-60734-4

All Rights Reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese edition published by Tsinghua University Press, Copyright © 2006

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Addison-Wesley 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2002-0659 号

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

#### 图书在版编目(CIP)数据

程序设计的模式语言·卷1/(美)科普林(Coplien, J.O.), (美)施密特(Schmidt, D.C.)编; 谢文亮, 梁金昆译. —北京: 清华大学出版社, 2006.5

书名原文: Pattern Languages of Program Design 1

ISBN 7-302-12794-8

I. 程… II. ①科… ②施… ③谢… ④梁… III. 程序设计语言学 IV. TP311.1

中国版本图书馆CIP数据核字(2006)第033317号

出版者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社总机: 010-62770175 客户服务: 010-62776969

责任编辑: 常晓波

印刷者: 北京密云胶印厂

装订者: 三河市李旗庄少明装订厂

发行者: 新华书店总店北京发行所

开 本: 185×230 印张: 28 字数: 621千字

版 次: 2006年5月第1版 2006年5月第1次印刷

书 号: ISBN 7-302-12794-8/TP·8144

印 数: 1~3000

定 价: 49.00元

# 简介

PLoP (Pattern Languages of Program Design) 创立的宗旨是创造新文件。这暗示着 PLoP 的创始人对现有的文化有些不满，事实的确如此。这些创始人（面向对象编程社区的几位知名人物）已经意识到：他们的才华已受到文件偏见的限制。这种偏见源于科学出版物的传统，一言以蔽之就是喜新厌旧，总是追逐最新最近的发明或发现，而忽视传统的东西，而不论这些所谓的发明或发现是否有用。PLoP 的创始人对传统的兴趣也许源自他们对软件重用的研究，或者来自他们的观察：尽管采用了最新的技术，但是由于缺少常用的解决方案，项目最终还是失败了。最重要的是：所有的创始人一致认为他们应该集中精力从事解决方案的推广。PLoP 大会就是在这种背景下召开的。

本书是“模式形式”系列书籍的第一部，这是我们心目中最佳的解决方案共享方式。我们要求作者提交“模式形式”方面的论文，但不要求实际地阐述形式的含义。Christopher Alexander 创造了“模式语言”这一术语并且在他的著作 *The Timeless Way of Building* 中很好地阐述了该形式。很多作者对这项工作都很熟悉，而更多的人则是通过 OOPSLA 研讨会和 Internet 讨论组了解到它的。即便如此，我们还是感到各位作者希望畅言如何将 Alexander 的形式应用到计算机编程领域。有一点我们始终坚持：每篇论文描述一个解决方案，可以用来解决一个问题。

读者在阅读本书时会发现，各位作者的解决方案所覆盖的问题范围相当广。这也意味着并非每一章都能引起读者同等的兴趣。我们希望随着 PLoP 社区的发展和成熟，PLoP 自身能随着兴趣方向而分化。未来的书籍不会要求它们有如此广泛的读者。在此我们希望读者能够细读每一章，即使您并不打算立即应用所读到的内容，也可以启发表达模式的新灵感，并且的确会提出一个角度来理解开发软件如此困难的原因。

虽然涉及的主题丰富多样，PLoP 的作者和参与者之间却达成了一些令人惊叹的共识。例如，多数人都已发现：解决方案（模式的本质）可以轻松超越其表达形式的准确特性。模式最终必定停留在人的头脑中，所以各种书写风格（从标准模板的标签化分节到更具风格的运行段落）对模式成功的贡献都不如一些更基本的元素。这些元素包括：设立问题及其上下文，分析施加于解决方案的影响力，以及（最重要的）提供具体的解决方案。包含这些元素的模式都成功了。

PLOP'94 的核心特色就是作者研讨会，作者听取对他们论文的小组讨论并就其中的亮点和不足展开争论，而不是把论文发给每位听众了事。这就给了作者一个机会，不仅知道如何交流，还能发现他们所研究的技术的其他替代方案。（几十年前在创作性写作社区发源而来的）作者研讨会是新作者学习前辈的经验技巧推敲他们见解的重要论坛。我们很感谢 Richard Gabriel 在 1994 年春天把我们领入作者研讨会。就我们所知，这是它第一次被用于技术社区，不过看起来效果甚佳。

这就是催生本书的理念和过程。我们对结果很满意，相信各位读者也会。因此我们要全速前进。每周我们都会发现新的证据：我们和我们的作者所促成的焦点转移对我们编写程序的方法具有深远的影响。他们也通过实例昭示了我们对软件工程未来的憧憬。

参与本书翻译和审校工作的有：谢文亮、马睿倩、梁金昆、张丽萍、王新彦、刘敏、李娟、刘颖、吴东升、郑刚、李楠、贺倩、李蓉、张颖、刘伟、王彦、王婷、李晓东、吴玲、胡静、张旭、王东、张雄、卢冰、高宏、李晓梅、汪刚、张健、刘洋、苏建忠、王欣轩、于涛、李敏等，在此表示感谢。

# 序 言

在以系统而极为易懂的方式确定和优化各种软件开发专家意见而做的大量努力方面，本书达到了前所未有的高度。本书以发表在 First Annual of Conference of Pattern Languages of Programs (PLoP) 的论文为基础（此次大会于 1994 年 8 月在伊利诺斯州的 Monticello 附近举行）。但是本书不仅仅是大会论文的大纲而已。本书的内容大量源自关于面向对象设计模式的新文献成果，它是关于正在不断发展壮大的软件模式与模式语言领域的系列文献汇编的第 1 部。

设计模式捕获了在特定上下文中开发应用程序时重复出现的解决方案的静态和动态结构。因为它们将软件系统开发中最基本的挑战暴露无遗，所以设计模式是改善软件质量的重要技术手段。设计模式所反映的关键挑战包括：在开发人员之间交流体系结构知识，适应新的设计范例或体系结构风格以及避免通常只有通过（痛苦的）经验才能领略到的开发陷阱和缺陷。

模式文献的大量积累业已存在——不是关于软件的而是关于建筑施工的。Christopher Alexander 从 15 年前就开始改善其架构模式形式，并且不再参照数百年前的架构模式，但到了最近模式才在软件中起关键作用。Peter Coad 在 1992 年发表了一篇 CACM 文章（Coad, 1992），在文中他记录了 Alexandrian 模式和软件体系结构之间的联系。直到 1993 年模式才作为研讨会、会议专题和期刊出版物的成果进入日常话题。Erich Gamma 等人的草稿 *Design Patterns* (1995) 在 1993 年和 1994 年被广泛传播。这项里程碑式的工作以书面形式提供了第 1 个软件模式的全面集合并确立了模式形式的新标准。Peter Coad 后来的工作在其著作 *Object Models: Strategies, Patterns, and Applications* (Coad, 1995) 中登峰造极。*Strategies, Patterns, and Applications* (Coad, 1995) 各种文献的雏形比这本书中收集的各种模式更有价值。

如果读者仔细阅读本书，就会发现众多不同的模式形式。有些模式吸取了 Alexander 的风格，而其他一些则得益于 Erich Gamma 及其同事的贡献，还有一些源自 Peter Coad 的模式，而有些则完全是原创的。我们尽全力保留作者的原创形式，并且尽量避免对各表达方式进行改动：我们从不强制任何统一的写作风格。尽管本书缺少单个作者的声音，但是我们也不会以其他形式引入。希望广大读者能与我们一同庆祝新文献正文的成形阶段的多样性。

本书各章的确出自同期软件文献中最热门的文献汇编。编纂工作是正在进行的、循环的工作。在大会召开前，各位作者与来自模式社区的“领导者”一道工作，目的就是使每个模式都包含纯粹的精髓：清晰的问题陈述、体现问题的解决方案以及推动解决方案的力量的清晰表述。然后每一章都在 PLoP'94 大会的作者研讨会中经历了深入的编辑：作者、评论者以

及其他研讨会参与者在一起讨论每篇论文的亮点和不足。评论者更倾向于强调积极方面并对内容、风格和表达方式提出建设性意见。

产品编辑也是集体努力的成果。在 The Hillside Group 与同事进行的讨论为早期编辑决策指引了方向。当我们把精力集中在资料的逻辑组织结构上时，我们在 Addison-Wesley 的朋友创造了一种统一设计，将被动动词和虚拟前置词都去掉，并且在后勤、版权谈判和许多细节上给予我们很大支持。尽管他们与我们的学究口味不同，但是在页面设计和格式化方面对所有文字资料的可读性贡献卓著。因此，尽管我们收到了各种格式的手稿，但我们在尽力将它们整合为单一设计的同时保留了每种格式的精髓。

如果没有 Deborah Lafferty、Penny Stratton 和 Tom Stone 的帮助我们无法完成此书，他们为这本书所倾注的细心和关注如同对待自己的作品一样。我们为有他们这样的团队成员感到骄傲和高兴，而最最重要和特别的是深深的谢意。

最后，我们需要强调：如果没有各位作者的智慧、勤劳和奉献就不会有这本书。世界上任何审稿工作都不能代替高质量的作品，能与这些出色的作者一起工作我们深感荣幸。

J.O.C.

D.C.S

# 目 录

## 第 1 部分 框架和组件

|                                        |    |
|----------------------------------------|----|
| 第 1 章 功能点选 .....                       | 4  |
| 第 2 章 一套基于工具和材料比喻的工具构造和集成模式语言 .....    | 6  |
| 背景和动机 .....                            | 6  |
| 总体方法 .....                             | 7  |
| 中心思想和设计比喻 .....                        | 7  |
| 设计比喻和设计模式之间的区别 .....                   | 8  |
| 工具和材料比喻 .....                          | 9  |
| 工具构造和集成设计模式 .....                      | 12 |
| 图形符号 .....                             | 12 |
| 设计模式线路图 .....                          | 12 |
| 工具构造设计模式 .....                         | 14 |
| 单个工具设计模式的应用 .....                      | 22 |
| 工具集成 .....                             | 23 |
| 扩展范例 .....                             | 23 |
| 工具集成设计模式 .....                         | 24 |
| 展望：系统边界 .....                          | 27 |
| 实际运转的工具和材料 .....                       | 28 |
| 工具和材料比喻 .....                          | 30 |
| 展望 .....                               | 30 |
| 致谢 .....                               | 30 |
| 参考文献 .....                             | 31 |
| 第 3 章 灵活的命令解释器：一个可扩展、语言无关解释器系统模式 ..... | 33 |
| 第 4 章 旧服务器，新客户端：一套客户/服务器框架模式语言 .....   | 39 |
| 介绍 .....                               | 39 |
| 关于具体化（REIFICATION）的注解 .....            | 40 |
| 例外（Exception） .....                    | 40 |
| 单人游戏（SOLITAIRE） .....                  | 41 |



|                                                          |    |
|----------------------------------------------------------|----|
| 记录对象 (OBJECTS FROM RECORDS) .....                        | 41 |
| 请求 (REQUEST) .....                                       | 42 |
| 实物化 (MATERIALIZATION) .....                              | 42 |
| 终结化 (FINALIZATION) .....                                 | 43 |
| 身份管理 (IDENTITY MANAGEMENT) .....                         | 43 |
| 巨滚 (MEGA-SCROLLING) .....                                | 44 |
| 搜索业务对象 (SEARCHING BUSINESS OBJECTS) .....                | 44 |
| 依赖性 (DEPENDENCY) (模型—视图、广播、观察者、MVC) .....                | 45 |
| 创建和更新业务对象 (CREATING AND UPDATING BUSINESS OBJECTS) ..... | 45 |
| 工厂方法 (Factory Method) .....                              | 46 |
| 窗口保管 (WINDOW-KEEPING) .....                              | 46 |
| 察看 (VIEWING) .....                                       | 47 |
| 一个客户/服务器框架 (A CLIENT/SERVER FRAME- WORK) .....           | 47 |
| 结论 .....                                                 | 47 |
| 致谢 .....                                                 | 48 |
| 参考文献 .....                                               | 48 |

## 第 2 部分 系统和分布式处理

|                                                               |    |
|---------------------------------------------------------------|----|
| 第 5 章 一套生成式的分布处理模式语言 .....                                    | 54 |
| 对模式和它们用途的一些思考 .....                                           | 54 |
| 朝着一套分布式计算的模式语言前进 .....                                        | 55 |
| 名称: 定义数据结构 (DEFINE THE DATA STRUCTURE) .....                  | 55 |
| 名称: 确定名词 (IDENTITY THE NOUNS) .....                           | 56 |
| 名称: 抽取公用属性 (FACTOR OUT COMMON ATTRIBUTE) .....                | 57 |
| 名称: 规范化角色 (NORMALIZE THE ROLES) .....                         | 59 |
| 名称: 确定问题领域关系 (IDENTIFY PROBLEM DOMAIN<br>RELATIONSHIPS) ..... | 61 |
| 名称: 引入虚拟属性 (INTRODUCE VIRTUAL ATTRIBUTES) .....               | 62 |
| 名称: 激活数据 (ANIMATE THE DATA) .....                             | 63 |
| 名称: 时间线索分析 (TIME THREAD ANALYSIS) .....                       | 64 |
| 命名: 决定参与者 (DETERMINE THE ACTORS) .....                        | 64 |
| 原始会议模式 .....                                                  | 65 |
| 参考文献 .....                                                    | 68 |
| 第 6 章 G++:一套计算机集成制造模式语言 .....                                 | 69 |
| 摘要 .....                                                      | 69 |

|                                                                                                            |            |
|------------------------------------------------------------------------------------------------------------|------------|
| 介绍 .....                                                                                                   | 69         |
| 应用领域 .....                                                                                                 | 70         |
| 论文组织 .....                                                                                                 | 70         |
| 模式语言 .....                                                                                                 | 70         |
| 模式 1 控制层次的层级 (A HIERARCHY OF CONTROL LAYERS) .....                                                         | 72         |
| 模式 2 控制模块之间的可见性和通信 (VISIBILITY AND<br>COMMUNICATION BETWEEN CONTROL MODULES) .....                         | 74         |
| 模式 3 按并发性给对象分类 (CATEGORIZE OBJECT FOR<br>CONCURRENCY) .....                                                | 75         |
| 模式 4 事件触发的动作 (ACTIONS TRIGGERED BY EVENT) .....                                                            | 77         |
| 模式 5 服务“等待”(SERVICES“WAITING FOR”) .....                                                                   | 77         |
| 模式 6 客户端/服务器/服务模型: 实现控制模块 (THE<br>CLIENT/SERVER/_SERVICE MODEL:IMPLEMENTATING<br>MODULES OF CONTROL) ..... | 79         |
| 模式 7 “多类型服务”控制模块的实现 (IMPLEMENTATION OF<br>“MULTIPLE KINDS OF SERVICES” CONTROL MODULES) .....              | 80         |
| 模式 8 控制模块的接口 (THE INTERFACE TO CONTROL MODULES) .....                                                      | 81         |
| 模式 9 原型和现实 (PROPERTY AND REALITY) .....                                                                    | 83         |
| 模式 10 控制模块分布 (DISTRIBUTION OF CONTROL MODULES) .....                                                       | 84         |
| 和“设计模式编目”比较 .....                                                                                          | 87         |
| 结论 .....                                                                                                   | 88         |
| 致谢 .....                                                                                                   | 88         |
| 参考文献 .....                                                                                                 | 88         |
| <b>第 7 章 分层体系结构的创建模式</b> .....                                                                             | <b>91</b>  |
| 摘要 .....                                                                                                   | 91         |
| 背景 .....                                                                                                   | 91         |
| 模式 .....                                                                                                   | 91         |
| 阶梯模式 .....                                                                                                 | 91         |
| 桥接模式 .....                                                                                                 | 94         |
| 对称重用模式 .....                                                                                               | 95         |
| 提升引用以增强可重用性 .....                                                                                          | 97         |
| 参考文献 .....                                                                                                 | 97         |
| <b>第 8 章 半对象+协议模式 (HOPP)</b> .....                                                                         | <b>98</b>  |
| <b>第 9 章 主从模式</b> .....                                                                                    | <b>101</b> |
| 摘要 .....                                                                                                   | 101        |

|            |     |
|------------|-----|
| 介绍 .....   | 101 |
| 模式 .....   | 103 |
| 理由 .....   | 103 |
| 适用性 .....  | 103 |
| 分类 .....   | 103 |
| 描述 .....   | 103 |
| 动态行为 ..... | 104 |
| 方法 .....   | 104 |
| 实现 .....   | 105 |
| 示例 .....   | 105 |
| 讨论 .....   | 106 |
| 参考文献 ..... | 107 |

### 第 3 部分 业务对象

|                                 |     |
|---------------------------------|-----|
| 第 10 章 信息完整性的 CHECKS 模式语言 ..... | 110 |
| 整值 .....                        | 110 |
| 异常值 .....                       | 111 |
| 无意义行为 .....                     | 112 |
| 回显 .....                        | 113 |
| 可见含义 .....                      | 114 |
| 延迟验证 .....                      | 115 |
| 快速预测 .....                      | 116 |
| 假定发布 .....                      | 116 |
| 预测确认 .....                      | 117 |
| 诊断式查询 .....                     | 117 |
| 第 11 章 账户号码模式 .....             | 119 |
| 模式 .....                        | 119 |
| 参考文献 .....                      | 122 |
| 致谢 .....                        | 122 |
| 第 12 章 星型结构：一种查询优化方案模式语言 .....  | 123 |
| 查询优化过的数据库 .....                 | 124 |
| 整个业务实体 .....                    | 126 |
| 关键业务活动和影响 .....                 | 127 |
| 事务处理历史 .....                    | 128 |
| 人员、地点和事物 .....                  | 130 |

|           |     |
|-----------|-----|
| 时间 .....  | 132 |
| 维上卷 ..... | 133 |

## 第 4 部分 过程与组织

|                           |     |
|---------------------------|-----|
| 第 13 章 产生式开发过程模式语言 .....  | 137 |
| 介绍 .....                  | 137 |
| 语言内容 .....                | 138 |
| 语言的驱动因素 .....             | 139 |
| 组织内无名的质 .....             | 140 |
| 语言原理 .....                | 140 |
| 备注 .....                  | 141 |
| 致谢 .....                  | 141 |
| 模式语言 .....                | 142 |
| 模式 1: 确定组织的规模 .....       | 142 |
| 模式 2: 自选团队 .....          | 144 |
| 模式 3: 独奏爱好者 .....         | 144 |
| 模式 4: 确定进度 .....          | 145 |
| 模式 5: 形式服从功能 .....        | 146 |
| 模式 6: 角色的专业知识 .....       | 146 |
| 模式 7: 逐步深入 .....          | 147 |
| 模式 8: 见习生 .....           | 147 |
| 模式 9: 组织服从位置 .....        | 148 |
| 模式 10: 组织服从市场 .....       | 149 |
| 模式 11: 开发人员控制过程 .....     | 149 |
| 模式 12: 老板 .....           | 150 |
| 模式 13: 体系结构设计师控制产品 .....  | 151 |
| 模式 14: 反向法则 .....         | 152 |
| 模式 15: 体系结构设计师也进行实现 ..... | 152 |
| 模式 16: 检查体系结构 .....       | 153 |
| 模式 17: 代码所有权 .....        | 153 |
| 模式 18: 根据测试设计划分应用设计 ..... | 154 |
| 模式 19: 进行品质保证 .....       | 155 |
| 模式 20: 客户交往 .....         | 155 |
| 模式 21: 分组验证 .....         | 157 |
| 模式 22: 场景定义问题 .....       | 157 |

|                                  |            |
|----------------------------------|------------|
| 模式 23: 雇佣分析员                     | 158        |
| 模式 24: 防火墙                       | 159        |
| 模式 25: 看门人                       | 159        |
| 模式 26: 形成循环圈                     | 160        |
| 模式 27: 转移职责                      | 161        |
| 模式 28: 水牛山                       | 161        |
| 模式 29: 内部工作流                     | 163        |
| 模式 30: 每个角色需要 3~7 个助手            | 165        |
| 模式 31: 命名的稳定基础                   | 166        |
| 模式 32: 分治                        | 167        |
| 模式 33: 分解各个阶段                    | 167        |
| 模式 34: 轮轴、钢圈和辐条                  | 168        |
| 模式 35: 美学模式                      | 169        |
| 模式 36: 耦合减少延迟                    | 169        |
| 模式 37: 原型                        | 169        |
| 模式 38: 不要打盹                      | 172        |
| 模式 39: 配对开发                      | 172        |
| 模式 40: 中止阻塞                      | 172        |
| 模式 41: 不要打断一个中断                  | 173        |
| 模式 42: 成功补偿                      | 173        |
| 参考文献                             | 175        |
| <b>第 14 章 生命周期以及支持演变和复用的重构模式</b> | <b>178</b> |
| 介绍                               | 178        |
| 背景对象演化                           | 180        |
| 模式: 原型化均衡 (First-Pass) 设计        | 180        |
| 上下文                              | 180        |
| 问题                               | 180        |
| 解决方案                             | 181        |
| 相关模式                             | 181        |
| 模式: 扩展初始原型                       | 182        |
| 上下文                              | 182        |
| 问题                               | 182        |
| 解决方案                             | 182        |
| 相关模式                             | 183        |
| 模式: 合并程序以支持演化和重用                 | 183        |

|                                                     |            |
|-----------------------------------------------------|------------|
| 上下文 .....                                           | 183        |
| 问题 .....                                            | 183        |
| 解决方案 .....                                          | 184        |
| 相关模式 .....                                          | 185        |
| 模式: 从继承层次结构中演化聚合 .....                              | 186        |
| 上下文 .....                                           | 186        |
| 问题 .....                                            | 186        |
| 解决方案 .....                                          | 187        |
| 相关模式 .....                                          | 188        |
| 模式: 创建抽象超类 .....                                    | 188        |
| 上下文 .....                                           | 188        |
| 问题 .....                                            | 188        |
| 解决方案 .....                                          | 189        |
| 相关模式 .....                                          | 190        |
| 讨论 .....                                            | 190        |
| 致谢 .....                                            | 191        |
| 参考文献 .....                                          | 191        |
| <b>第 15 章 RAPPeL: 一种用于面向对象开发的“需求分析过程”模式语言</b> ..... | <b>194</b> |
| 摘要 .....                                            | 194        |
| RAPPeL 的基本原理 .....                                  | 194        |
| 警告 .....                                            | 195        |
| 模式语言模式 1: 构建正确的东西 .....                             | 197        |
| 模式 5: 客户期望的管理与满足 .....                              | 198        |
| 模式 9: 客户和谐 .....                                    | 198        |
| 模式 14: 赞助人意图 .....                                  | 199        |
| 模式 17: 定义需求 .....                                   | 200        |
| 模式 20: 问题域分析 .....                                  | 201        |
| 模式 22: 信息需求 .....                                   | 202        |
| 模式 24: 域对象的寻找和定义 .....                              | 203        |
| 模式 25: 域对象的分类、关联与分组 .....                           | 204        |
| 模式 26: 域对象的细化 .....                                 | 205        |
| 模式 27: 对象老化 .....                                   | 206        |
| 模式 28: 对象造型 .....                                   | 207        |
| 模式 30: 行为性需求 .....                                  | 208        |
| 模式 32: 展望 .....                                     | 210        |

|                                             |            |
|---------------------------------------------|------------|
| 模式 34: 需求规范 .....                           | 210        |
| 模式 36: 业务规则 .....                           | 213        |
| 解决方案 .....                                  | 213        |
| 约束用例的规则 .....                               | 213        |
| 约束对象及其状态的规则 .....                           | 214        |
| 模式 40: 实用外部需求 .....                         | 215        |
| 模式 50: 用户界面需求 .....                         | 215        |
| 模式 97: 原型 .....                             | 216        |
| 模式 99: 需求验证 .....                           | 217        |
| 参考文献 .....                                  | 218        |
| <b>第 16 章 毛虫的命运: 一种用于分析-设计转换的模式语言 .....</b> | <b>219</b> |
| 摘要 .....                                    | 219        |
| 介绍 .....                                    | 219        |
| CF 模式语言 .....                               | 220        |
| 模式 1: 并发执行线程 .....                          | 221        |
| 模式 2: 并发线程同步 .....                          | 221        |
| 模式 3: 协作工作包 .....                           | 222        |
| 模式 4: 工作包内容 .....                           | 223        |
| 模式 5: 工作包状态报告 .....                         | 224        |
| 模式 6: 工作包完工报告 .....                         | 225        |
| 模式 7: 工作包优先级 .....                          | 225        |
| 模式 8: 工作包安全性 .....                          | 226        |
| 模式 9: 程序形态 .....                            | 228        |
| 模式 10: 系统成员的角色 .....                        | 230        |
| 模式 11: 决策制定者的角色 .....                       | 230        |
| 模式 12: 工作人员的角色 .....                        | 231        |
| 模式 13: 接口的角色 .....                          | 231        |
| 模式 14: 信息化角色 .....                          | 232        |
| 模式 15: 小家族系统 .....                          | 233        |
| 模式 16: 通过对话完成的工作 .....                      | 234        |
| 模式 17: 关键区域保护 .....                         | 234        |
| 模式 18: 事件获取 .....                           | 235        |
| 模式 19: 事件路由 .....                           | 236        |
| 模式 20: 人机界面角色是一种特殊的接口角色 .....               | 237        |
| 模式 21: 数据知其出处 .....                         | 238        |

|              |     |
|--------------|-----|
| CF 的使用 ..... | 238 |
| CF 的状态 ..... | 239 |
| 相关事宜 .....   | 239 |
| 参考文献 .....   | 240 |

## 第 5 部分 设计模式及其编目

|                               |            |
|-------------------------------|------------|
| <b>第 17 章 一种模式系统 .....</b>    | <b>244</b> |
| 摘要 .....                      | 244        |
| 介绍 .....                      | 244        |
| 一种模式系统 .....                  | 245        |
| 如何描述模式 .....                  | 246        |
| 模式的分类 .....                   | 247        |
| 粒度 .....                      | 247        |
| 功能 .....                      | 248        |
| 结构化原则 .....                   | 249        |
| 模式 .....                      | 250        |
| 体系架构级框架 .....                 | 250        |
| 设计模式 .....                    | 251        |
| 异构的体系架构 .....                 | 253        |
| 经验 .....                      | 255        |
| 未决事宜 .....                    | 255        |
| 相关研究工作 .....                  | 256        |
| 结论与展望 .....                   | 257        |
| 参考文献 .....                    | 258        |
| <b>第 18 章 设计模式之间的关系 .....</b> | <b>260</b> |
| 摘要 .....                      | 260        |
| 介绍 .....                      | 260        |
| 设计模式目录的总结构 .....              | 261        |
| 关系分类 .....                    | 261        |
| 需要解决的问题 .....                 | 261        |
| 关系类别 .....                    | 263        |
| 分类 .....                      | 263        |
| 使用分类 .....                    | 265        |
| 更改关系和设计模式 .....               | 265        |



|                                |            |
|--------------------------------|------------|
| 一个新设计模式: Objectifier .....     | 265        |
| 其他更改 .....                     | 266        |
| 设计模式层 .....                    | 267        |
| 层中的组织 .....                    | 267        |
| 基本设计模式和技术 .....                | 267        |
| 解决典型软件问题的设计模式 .....            | 269        |
| 针对特定应用域的设计模式 .....             | 269        |
| 其他组织 .....                     | 269        |
| 相关著作 .....                     | 270        |
| 结论 .....                       | 270        |
| 参考文献 .....                     | 271        |
| 附录: 对象化器模式描述 .....             | 272        |
| <b>第 19 章 发现现有应用中的模式 .....</b> | <b>276</b> |
| 介绍 .....                       | 276        |
| 模式 .....                       | 277        |
| 容器模式 .....                     | 277        |
| 高层设计模式 .....                   | 278        |
| 低层设计模式 .....                   | 280        |
| 小结 .....                       | 281        |
| 参考文献 .....                     | 281        |
| 附录 .....                       | 281        |
| <b>第 20 章 实现模式 .....</b>       | <b>302</b> |
| 介绍 .....                       | 302        |
| 抽象模式和具体模式 .....                | 302        |
| 实现过程中模式被摒弃 .....               | 303        |
| 多簇相互依赖的类 .....                 | 304        |
| 模式类 .....                      | 305        |
| 常用模式库 .....                    | 307        |
| 例证 .....                       | 311        |
| 概念问题 .....                     | 312        |
| 致谢 .....                       | 313        |
| 参考文献 .....                     | 313        |
| 附录 A: 模式作为一个类 .....            | 313        |
| 附录 B: 类组合的 C++ 实现 .....        | 314        |