

Mc
Graw
Hill Education

国外经典初学者入门教程系列



UML

基础教程

Jason T. Roff 著
张瑜 杨继萍 等译

UML A Beginner's Guide

Mc
Graw
Hill

清华大学出版社

UML 基础教程

Jason T. Roff 著
张瑜 杨继萍 等译

清华大学出版社
北京

Jason T. Roff

UML: A Beginner's Guide

EISBN: 0-07-222460-6

Copyright © 2003 by The McGraw-Hill Companies, Inc.

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

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education (Asia) Co., within the territory of the People's Republic of China only, excluding Hong Kong, Macao SARs 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.

本书中文简体字翻译版由美国麦格劳-希尔教育出版(亚洲)公司授权清华大学出版在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾)独家出版发行。未经许可之出口,视为违反著作权法,将受法律之制裁。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。

北京市版权局著作权合同登记号 图字 01-2002-5768 号

版权所有,翻印必究。

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

UML 基础教程 / (美)罗夫著;张瑜等译. —北京:清华大学出版社, 2003. 10

书名原文: UML A Beginner's Guide

ISBN 7-302-07122-5

I. U... II. ①罗... ②张... III. 面向对象语言, UML - 程序设计 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 073200 号

出版者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-62770175

地址: 北京清华大学学研大厦

邮编: 100084

客户服务: 010-62776969

责任编辑: 冯志强

封面设计: 蒋源源

印刷者: 世界知识印刷厂

发行者: 新华书店总店北京发行所\清华大学出版社出版发行

开本: 185 × 230 印张: 17.25 字数: 375 千字

版次: 2003 年 10 月第 1 版 2003 年 10 月第 1 次印刷

书号: ISBN 7-302-07122-5/TP · 5207

印数: 1~4000

定价: 32.00 元

作者简介

Jason T. Roff 拥有并运营着 First Factory 公司，它是一家软件开发咨询公司，擅长于 Microsoft 软件和 Internet 开发、文档制作服务和 Web 宿主(www.firstfactory.com)。

Jason 编写过多种有关数据库开发和电子商务方面的图书，目前正致力于使用 SQL Server 进行 .NET 开发和项目管理。他在 Albany University 获得了计算机科学与应用数学专业的学士学位，目前正在攻读硕士学位。

Jason 与他的妻子 Kimberly 和儿子 Zachary 幸福地生活在纽约长岛。你可以通过以下电子邮件与他联系：jason@firstfactoryinc.com。

致 谢

为了本书的圆满完成和它未来的成功，我想向许多人致谢。首先，我要感谢 McGraw-Hill/Osborne 全体人员对我的不懈支持。所有作者和可能成为作者的朋友们应该注意——这里是作者的乐土。McGraw-Hill/Osborne 全体人员可以真正做到团结协作，而这一点意义重大。

对于所有作者和准作者们，如果你想与最好的出版商合作，为他们写书的话，我建议你与 Waterside Productions 有限公司 (www.waterside.com) 的 David Fugate 联系。David 是我的代理人。多数人想到“代理人”这个词的时候会把它与 Jerry Maguire 或好莱坞联系起来。尽管 David 也住在加利福尼亚，但是他并不是你常见的那种代理人。他是一个很实际的人，是只和你探讨想法和讨论问题的人。我非常感谢 David 给我这项工作，并帮助我完成了它。

在本书编撰的整个过程中，编辑们的表现也颇为卓越。我要对许多编辑表示感谢。Ann Sellers 对我十分信任，并在许多问题上支持我，对此我向她表示感谢。Lisa McClain 在我需要时给予了我不懈的支持，她也应当得到同样的感谢。Timothy Madrid、Janet Walden 和 Bill McManus 与我紧密合作，目睹了我编撰本书的点点滴滴，而且似乎总是知道我所提出问题的答案。

我的技术编辑 Alexander Vendrov 则是一个令人惊奇的人。他比我认识的任何人都更了解 UML 规范和其应用。我坚信这种了解也体现在了本书当中，而且你会和我一样受益匪浅。我希望将来能有机会与 Alex 再次合作。

当然，还有另外一位 Alex，Alexander Petrov，他是我多年的朋友。有时是出于技术，有时是出于兴致，但无论何时、何故，他总是同我一起渡过。这么多年了，谢谢你，Alex。

我的妻子 Kimberly 忍受了许多本不应忍受的东西。在我谈论技术时，她总是倾听我说话。她容忍我对许多事情的忽略，才使我能够完成自己的工作。我爱我的妻子，我为自己得到的一切而感谢她。

我的儿子 Zachary 已经快两岁了，他是我不断写作、工作和学习的动力。因为我不得不供养他。我爱你，儿子。

Jason

前 言

统一建模语言对任何初学者而言都是一个挑战。如果他们已经了解 UML，需要学习如何应用它；如果他们想更多地了解自己已经打下的基础是为了什么，那么有许多可供开发人员使用的资源。而本书的编写是为了填补这一空白。本书会使真正的初学者对 UML 有一定的理解，并打下一些基础以使初学者高效学习 UML，并且将其应用到需要创建的系统的建模当中。

本书合适的读者

本书是为开发人员、分析者、质量保证人和项目经理编写的。不要被本书的题目误导。尽管本书的标题涉及初学阶段的人员，但本书不只是为初级开发人员或入门级分析者准备的。事实上，本书假定读者对软件以及软件是如何建立的已经有了一个基本的理解。

分析者从本书可以了解到如何从系统的用户或准用户获取业务需求。他们会学习并理解开发人员需要什么来编写系统？开发人员将会了解到如何使用分析者提供的信息来开始建立系统的具体模型，并最终编写其代码。质量保证人会了解到如何阅读要进行测试的应用程序的设计图以使他们的工作效率更高。当然，项目经理可以了解如何管理整个过程以使工程顺利进行。

如果你读到了这里，可能就已经分享了对开发软件之前，建立软件模型的理解。你进行任何主要工程之前了解了这一点，或者因为你目前正在参与一项工程，所以正在阅读本书的话，你就太幸运了。无论如何，学习正确的东西永远也不会太晚，本书将为你显示这一点。

本书内容

本书以渐进的顺序来介绍 UML，从分析开始，然后是构建和部署。

第 1 章“UML 基本原理”。在你学习为何在进行软件开发之前应该建立软件模型的同时，为你介绍了统一建模语言的概念。通过讨论 UML 的起源，包括其创立者是谁，创立它是出于何种原因，本书会向你介绍 UML 的不同构件和 Unified Process，以及如何使用 UML 的方法建立系统模型。

第 2 章“用例图”。会教你如何询问行业专家(他可能是你将来最终开发的系统的超级用户)来确定进行建模的系统的参数。这一章会向你展示如何获取这些信息，并将其表示为任何系统的构造块的用例。

第3章“面向对象设计介绍”。为你提供了与建模过程相关的面向对象概念的简介。UML 是设计用来为面向对象设计(OOD)工作的，因此需要深入理解面向对象设计。

第4章“用活动图建立 workflow 模型”。说明活动图是如何在用例图之后成为下一个逻辑步骤的。在建立系统模型时，活动图描述了用户为完成用例而经过的路径。

第5章“使用顺序图建模行为”。基于从前一章的知识和使用另一种图——顺序图的活动图建模。本章解释了这两种图的不同，并在使用时分别给出了示例。

第6章“使用类图定义域模型”。介绍如何使用类图定义设计的基本构造块，对于它们将会在后面继续修饰。

第7章“协作图”。显示了如何将类图与顺序图结合起来解释消息流中的对象如何交互，消息流在顺序图中已经介绍过了。

第8章“深入学习类图”。介绍了类图更多的功能，这些功能是在设计系统时使用的，而不是在分析时使用的。正如你要学到的，在迭代使用时 UML 最好，而类图则是可以在建模的不同级别中使用的一种图。

第9章“深入学习顺序图”。会在你越来越多地进行系统设计，而不是系统分析时，更为深入地介绍顺序图。

第10章“使用状态图建立行为模型”。本章介绍了状态图，以及它是如何应用系统当前状态建模的。在这一章中，你将了解到动作、事件和组合状态。

第11章“构建实现方式图”。展示了两种实现方式图，即部署图和组件图的用法。这两种图在许多情况下同时使用，以说明哪些地方正在分布软件构件。

第12章“使用对象约束语言”。这是对 UML 的指定约束的语言的简介。这一章解释了如何将 OCL 加到图中以指定商务约束，这些约束可以限制流程、触发事件和控制点动作。

附录 A“思考与练习答案”给出了所有 12 章中的思考与练习的答案。

最后，术语表为本书用到的 UML 和 OCL 术语提供了一个简单的参考。

如何阅读本书

本书的编写是要读者从头到尾来阅读的，同时参考书后的术语表和思考与练习答案。如果你只对使用 UML 建模的分析方面感兴趣，可以只关注第 1~4 章的内容。然而，与构建(开发)建模任务相关的人就应该阅读整部书。

即使你只关心分析问题，了解开发人员如何从你的模型来设计应用程序也是十分重要的。理解他人如何解释你的工作，以及如何在下一步实施它，将为你如何准备自己的工作起到决定作用。

本书特点

几乎每章都包括提示和注意事项来帮助你理解 UML。书中的每一章都包含许多精

心创建的 UML 图来帮助你学习相关的主题。某些章包含专家问答，以一问一答的形式来提供更多的附加信息，而这些信息可能无法从直接阅读本书得来。

每一章都包含让你一步步完成的与本章主题知识学习同步的练习。多数章介绍的新模型图包含至少一个练习以帮助你在学习完其标记符后，阅读 UML 模型图示例。多数章在本章讨论完图的建模之后包含有一个练习，帮助你对给出的 UML 图进行建模。为帮助你完成练习，每一步的答案都可以从 www.osborne.com 和 www.firstfactoryinc.com 上找到。

在编写本书的同时，我不断发现 UML 规范不仅枯燥，而且非常难以理解。所以我尽全力用轻松现实的示例把 UML 规范以一种最独特的方式叙述给你。

我编写本书所得到的享受，希望你也同样能得到。

目 录

第 1 章 UML 的基本原理	1
1.1 了解软件建模的原因	2
1.1.1 分析、设计与实现	3
1.1.2 软件生命周期模型	4
1.2 理解统一建模过程	5
1.2.1 初始阶段	6
1.2.2 筹划阶段	6
1.2.3 构建阶段	7
1.2.4 转换阶段	7
1.3 什么是 UML	7
1.3.1 UML 的历史	8
1.3.2 OMG	9
1.4 认识 UML 的构件	10
1.4.1 结构图	10
1.4.2 行为图	10
1.5 寻找有效的 UML 建模工具	12
1.5.1 Rational Rose	12
1.5.2 Visio	12
1.5.3 写字板或笔和纸	13
1.6 思考与练习	13
第 2 章 用例图	14
2.1 定义用例图	15
2.2 找到用例图建模的原因	15
2.3 认识用例图的标记符组件	16
2.3.1 系统	17
2.3.2 参与者	17
2.3.3 用例	18
2.3.4 关系	20

2.3.5 子系统	21
2.4 理解泛化技术	22
2.4.1 用例与用例的关系	22
2.4.2 参与者与参与者的关系	25
2.5 理解如何使用包含关系和扩展关系	28
2.5.1 包含关系	28
2.5.2 扩展关系	29
2.6 学习如何描述用例	31
2.7 学习如何建立用例图模型	33
2.7.1 找出参与者和用例	33
2.7.2 区分用例优先次序	34
2.7.3 细化每个用例	35
2.7.4 构建用例模型	36
2.8 思考与练习	38
第3章 面向对象设计介绍	40
3.1 认识面向对象设计的特征	41
3.2 理解 OOD 中类的用法	42
3.2.1 抽象	42
3.2.2 封装	43
3.2.3 接口	44
3.3 理解 OOD 中继承的用法	45
3.3.1 继承层次结构	47
3.3.2 多重继承	49
3.4 理解 OOD 中多态性的用法	50
3.5 思考与练习	53
第4章 活动图的工作流建模	54
4.1 定义活动图	55
4.2 了解为什么建模活动图	55
4.3 确定活动图的标记符组件	55
4.3.1 活动:动作状态	56
4.3.2 状态	56
4.3.3 转移	57
4.3.4 综合运用	58

4.4	学习如何使用条件	59
4.4.1	控制点	59
4.4.2	决策点	59
4.5	学习活动图中的其他标记符	62
4.5.1	事件和触发器	62
4.5.2	游泳道	63
4.5.3	分岔和联结	64
4.6	学习如何建模活动图	66
4.6.1	标识用例	66
4.6.2	建模主路径	66
4.6.3	建模从路径	67
4.6.4	添加游泳道	68
4.6.5	改进高层活动	70
4.7	思考与练习	73
第5章	使用顺序图建模行为	74
5.1	定义顺序图	75
5.2	为什么要建模顺序图	75
5.3	了解顺序图的标记符组件	76
5.3.1	活动对象	77
5.3.2	消息	77
5.4	理解如何使用消息进行通信	80
5.4.1	同步消息	82
5.4.2	异步消息	83
5.4.3	简单消息	83
5.5	学习顺序图使用的其他技术	85
5.5.1	创建对象	85
5.5.2	使用状态	86
5.5.3	分支和从属流	88
5.6	学习如何建模顺序图	90
5.6.1	确定工作流	91
5.6.2	布置对象	92
5.6.3	添加消息和条件	92
5.6.4	绘制总图	94
5.7	思考与练习	96

第 6 章 使用类图定义域模型	98
6.1 定义类图	99
6.2 为什么要建模类图	99
6.3 学习类图的标记符组件	99
6.3.1 类	100
6.3.2 关系	101
6.3.3 多重性	102
6.3.4 角色	103
6.4 学习如何建模类	104
6.4.1 特性和操作	105
6.4.2 对象：类实例	107
6.4.3 包	108
6.5 学习如何为特性提供信息	109
6.5.1 数据类型	109
6.5.2 初始值	111
6.5.3 多重性	111
6.5.4 派生的特性	112
6.6 学习如何为操作提供参数	113
6.7 学习如何建模类图	116
6.7.1 确定类和关联	117
6.7.2 确定特性和操作	120
6.8 思考与练习	124
第 7 章 协作图	125
7.1 定义协作图	126
7.2 为什么要建模协作图	126
7.3 了解协作图的标记符组件	126
7.3.1 对象和角色	126
7.3.2 关联角色	129
7.3.3 链接	130
7.3.4 消息	131
7.4 学习不同的消息类型	132
7.4.1 同步消息	132
7.4.2 异步消息	133
7.4.3 简单消息	133

7.5	深入研究消息	134
7.5.1	序列化	134
7.5.2	控制点条件	135
7.5.3	创建实例	136
7.5.4	迭代	137
7.6	学习如何建模协作图	139
7.6.1	确定协作图的元素	140
7.6.2	建模结构化关系	140
7.6.3	建模实例层协作图	140
7.7	思考与练习	143
第8章	深入学习类图	145
8.1	学习聚集和组成	146
8.1.1	聚集	146
8.1.2	组成	147
8.1.3	使用带有泛化的聚集和组成	150
8.2	学习关联注释	152
8.2.1	约束	152
8.2.2	鉴别器	155
8.3	学习端点标记符	156
8.3.1	有序和排序	156
8.3.2	导航	156
8.4	建模多个关联	157
8.5	思考与练习	158
第9章	深入学习顺序图	160
9.1	学习控制矩形	161
9.1.1	使用控制矩形	161
9.1.2	指定活动期间	162
9.2	学习如何建模时间	165
9.2.1	指示时间上的变化	166
9.2.2	建模时间约束	166
9.2.3	建模持续时间	168
9.3	学习如何建模迭代	170
9.3.1	建模循环	171

9.3.2 建模递归	172
9.4 学习如何建模高级消息	174
9.4.1 消息参数和返回值	174
9.4.2 指定消息序列	175
9.5 学习如何向顺序图中添加注释	176
9.6 思考与练习	179
第 10 章 使用状态图建模行为	181
10.1 定义状态图	182
10.2 为什么要建模状态图	182
10.3 学习状态图的标记符组件	182
10.3.1 状态	183
10.3.2 转移	184
10.3.3 决策点	185
10.3.4 同步	186
10.4 学习如何指定状态的动作和事件	187
10.4.1 事件	187
10.4.2 动作	188
10.5 学习如何使用合成状态	190
10.5.1 子状态	190
10.5.2 合成状态	191
10.6 学习如何建模状态图	194
10.6.1 标识出需要进一步建模的实体	194
10.6.2 标识出每一个实体的开始状态和结束状态	194
10.6.3 确定与每一个实体相关的事件	194
10.6.4 从开始状态开始创建状态图	195
10.6.5 如果必要则指定合成状态	195
10.7 思考与练习	198
第 11 章 构造实现方式图	199
11.1 定义实现方式图	200
11.2 了解为什么要建模实现方式图	200
11.3 学习组件图	200
11.3.1 组件	200
11.3.2 依赖关系	201

11.3.3 固化依赖关系	201
11.3.4 包含组件	203
11.4 学习部署图	205
11.4.1 节点	205
11.4.2 通信关联	206
11.4.3 组合组件图和部署图	208
11.5 学习如何建模实现方式图	210
11.5.1 添加节点	210
11.5.2 添加通信关联	211
11.5.3 添加组件、类和对象	211
11.5.4 添加依赖关系	212
11.6 思考与练习	215
第 12 章 使用对象约束语言	217
12.1 定义对象约束语言	218
12.2 为什么使用对象约束语言	218
12.3 学习 OCL 语法	219
12.3.1 上下文和固化类型	219
12.3.2 类型和操作	220
12.4 学习集合的用法	223
12.4.1 集	223
12.4.2 序列	224
12.4.3 袋子	224
12.4.4 其他集合常用的操作	225
12.5 思考与练习	227
附录 A 思考与练习答案	229
附录 B 术语表	248

第 1 章

UML 的基本原理

重要技能

- 1.1 了解软件建模的原因
- 1.2 理解统一建模过程
- 1.3 什么是 UML
- 1.4 认识 UML 的构件
- 1.5 寻找有效的 UML 建模工具
- 1.6 思考与练习

结构与对产品质量的影响是其他活动无法比拟的。

——James O. Coplien, AT&T 贝尔实验室

初级程序开发人员问的最多的问题之一就是，为什么要建立软件模型？这个问题之所以总是被提出，是因为很少有程序开发人员在学习编程之前学习过建模，对于自学者更是如此。事实上，多数人是从模仿他人的代码起步而成为程序开发人员的。他们对自己取得的成就的惊异很快成长为一种技能。对于多数自学成材的程序开发人员而言，这种技能成为了他们构建系统的惟一途径。本书，特别是这一章，将帮助你意识到只需一点点的建模就可以使你的系统变得强大无比！

1.1 了解软件建模的原因

我们进行软件建模的最重要的原因是为了使我们的最终产品在质量上达到一个较高水平。或许你听说过结构完备的产品最终才会赢利，但是，你却不了解高品质如何或者为何不会产生；在不同概念阶段都有一些事物发生，而最终产品是建模的直接结果，而不是开发的直接结果。高品质是指产品开发简单，开发周期短，有更好的用户文档，经过更好测试从而减少故障。

事实上，良好的结构可以持续使用，拙劣的结构只会被淘汰。建立于良好基础之上，使用达到目标的一致的方法、包含大量的重用，没有故障的代码的产品修复起来无需吹灰之力，对于建造它的人而言更是如此。

例如玩积木(Legos)，我肯定你这辈子至少玩过一次。如果你还记得(如果你有孩子，或许你正在玩这个玩具)，对于如何把积木拼装在一起并没有什么限制。你可以把砖放在小人的头顶，就和你可以把屋顶放在四面墙上一样简单。软件也是如此。多数情况下，无需太多限制你就可以编写代码。

你可能还记得，所有积木的拼装都是为了建造一个具体的建筑物，无论它是房子、空间站、医院还是城堡。你难道不做准备就开始建造这些东西吗？你难道说完“我要用这些玩具建造一个城堡”，然后就开始动工吗？许多软件开发新手就是如此。他们叫嚷到：“嗨！我知道怎么写程序，知道怎么把积木拼起来，我可以做系统！”他们的目的是好的，然而他们的技术还有待锤炼。

你还记得积木盒中有一份长达20页的手册来精确地告诉你如何一步步地建造城堡吗？这时城堡的建造是否简单了许多？可能按照指导来建造最终的成品只会花费你很少的时间。甚至我还向你保证，你这样花费的时间要远少于某些人用原始的方法进行建造所花费的时间。我还向你保证制订城堡建造说明，然后按照说明去实施所花费的时间也远少于拿起来就做，比起没有任何目的性的建造方法去实施快许多。

事实上，建立软件模型要比开发软件耗费的时间更多，而通过合理的软件建模和归