TURING 图灵计算机科学丛书

Addison Wesley

UML用户指南

（第2版）

The Unified Modeling Language User Guide

Second Edition

[美] Grady Booch James Rumbaugh Ivar Jacobson 著

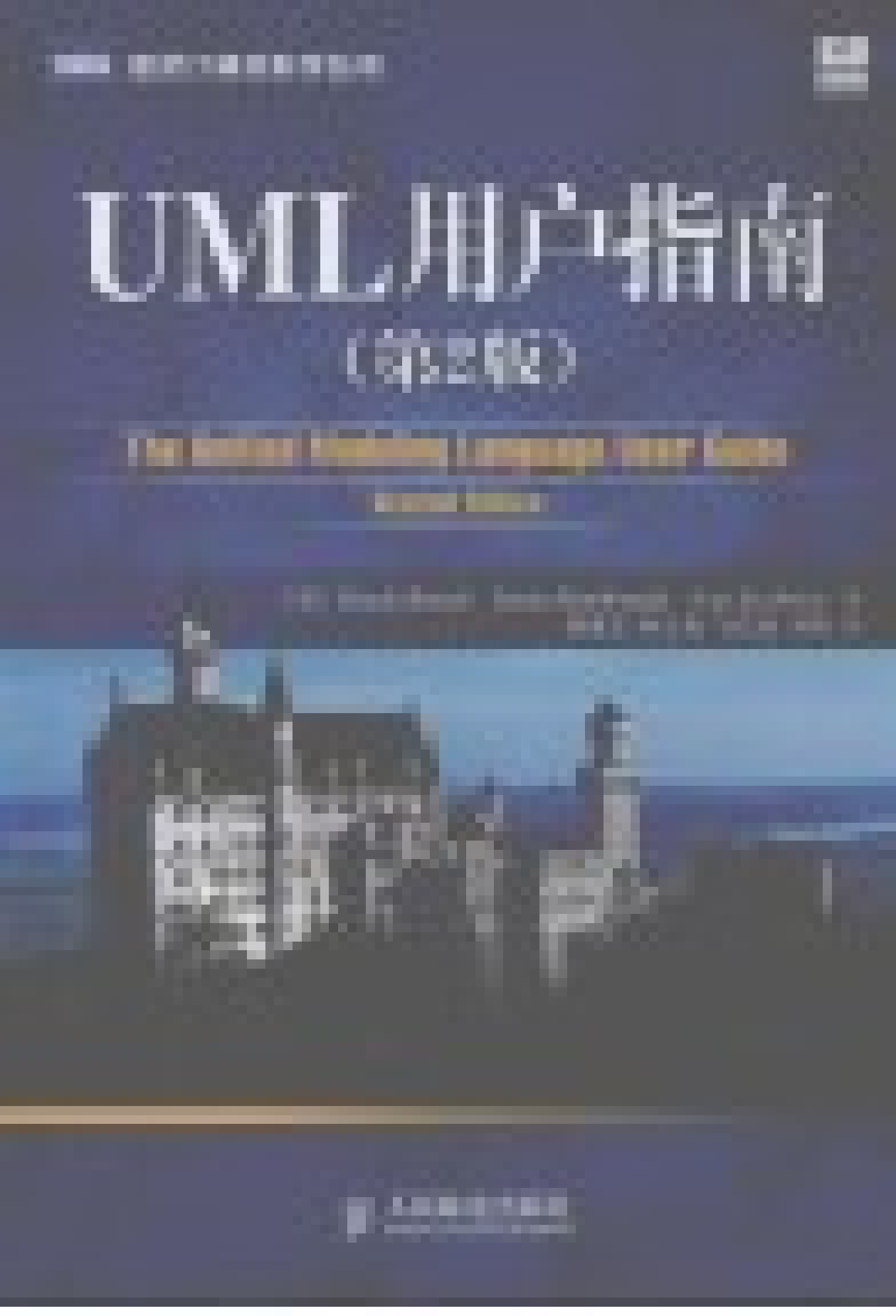
邵维忠 麻志毅 马浩海 刘辉 译

人民邮电出版社

POSTS & TELECOM PRESS

UML用户指南

（第2版）

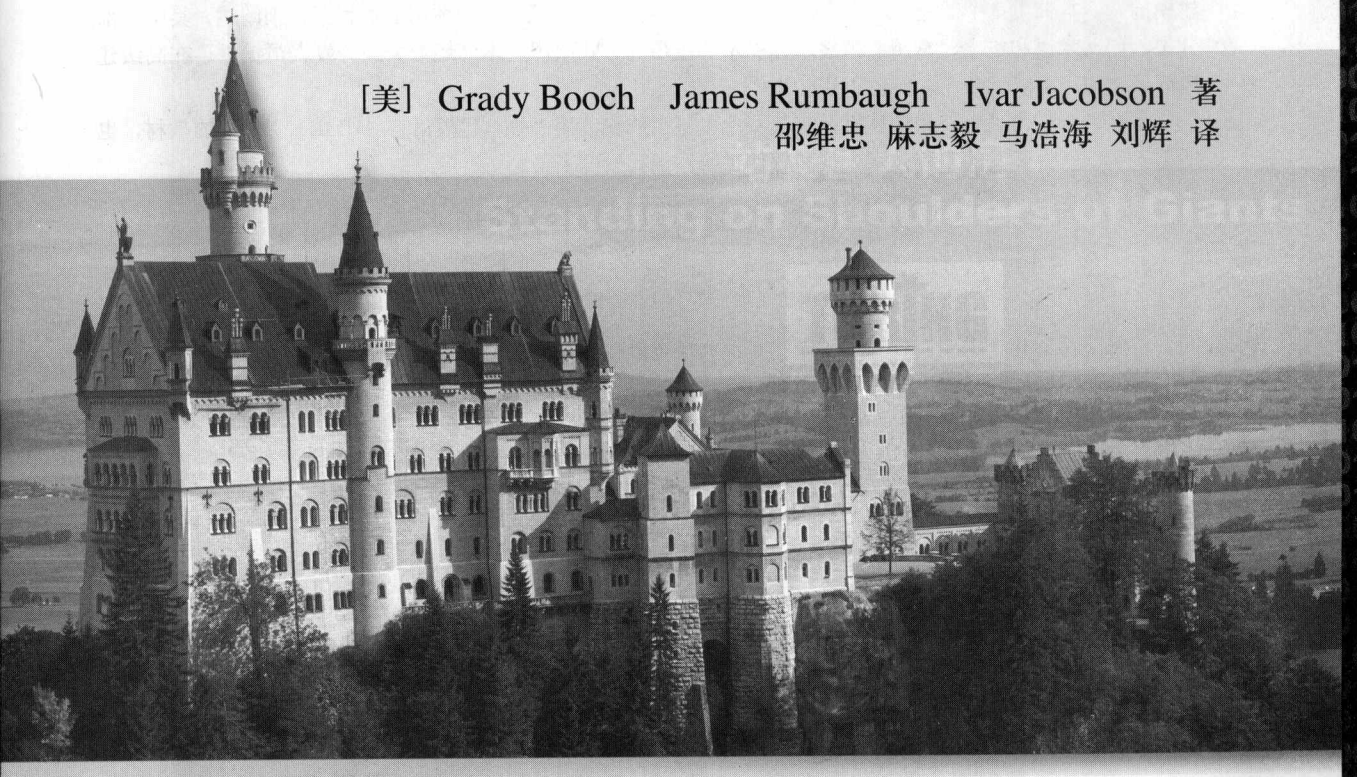


TURING 图灵计算机科学丛书

UML用户指南
（第2版）

The Unified Modeling Language User Guide
Second Edition

[美] Grady Booch James Rumbaugh Ivar Jacobson 著
邵维忠 麻志毅 马浩海 刘辉 译



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目（CIP）数据

UML 用户指南：第 2 版/（美）布奇，（美）兰宝，（美）雅各布著；邵维忠，麻志毅等译.
—北京：人民邮电出版社，2006.6（2007.7 重印）
（图灵计算机科学丛书）

ISBN 978-7-115-14833-9

内 容 提 要

本书是 UML 方面的一部权威著作，三位作者是面向对象方法最早的倡导者，UML 的创始人。本版涵盖了 UML 2.0。书中为 UML 具体特征的使用提供了指南，描述了使用 UML 进行开发的过程，旨在让读者掌握 UML 的术语、规则和惯用法，以及如何有效地使用这种语言，知道如何应用 UML 去解决一些常见的建模问题。本书由 7 个部分 33 章组成，每章都对一组 UML 特征及其具体用法进行了详细阐述，其中大部分按入门、术语和概念、常用建模技术、提示和技巧的方式组织。本书还为高级开发人员提供了在高级建模问题中应用 UML 的一条非常实用的线索。

本书适合作为高等院校计算机及相关专业本科生或研究生“统一建模语言（UML）”课程的教材，也适用于从事软件开发的工程技术人员和软件工程领域的研究人员。

图灵计算机科学丛书

UML 用户指南（第 2 版）

◆ 著　[美]Grady Booch　James Rumbaugh　Ivar Jacobson
译　邵维忠　麻志毅　马浩海　刘　辉
责任编辑　杨海玲

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：23.75
字数：578 千字　2006 年 6 月第 1 版
印数：6 001 – 7 500 册　2007 年 7 月北京第 2 次印刷

著作权合同登记号　图字：01-2005-5453 号

ISBN 978-7-115-14833-9/TP

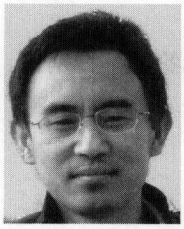
定价：49.00 元

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

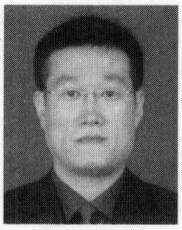
版 权 声 明

译 者 简 介

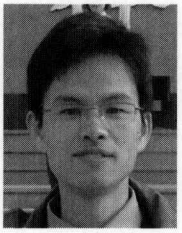
邵维忠 北京大学信息科学技术学院教授、博士生导师，中国计算机学会理事。1970 年毕业于北京大学数学力学系，1979~1983 年在计算机科学技术系任教并攻读硕士学位。早期主要从事操作系统和软件工程领域的研究。1987~1989 年在新加坡国立大学参加科技合作。回国后在导师杨芙清院士主持的国家“八五”、“九五”重点科技攻关课题“大型软件开发环境青鸟系统”和“软件工程环境工业化生产技术及系统”中担任主要技术负责人。自 1991 年起注重于面向对象建模方法的研究，并开设了研究生课程和本科生课程。写作和翻译了多部关于面向对象方法的学术著作。最近几年在面向对象技术、软件复用与构件技术、中间件技术等领域承担了多项国家高技术研究发展计划（863）项目、国家重大基础研究（973）项目和国家自然科学基金项目。曾获国家科技进步二等奖及国家部委级奖励多项。



麻志毅 北京大学信息科学技术学院副教授。1999 年在东北大学获博士学位，同年到北京大学从事博士后研究，出站后留校工作。近年来发表学术论文 40 余篇，出版学术专著 2 部，译著 4 部。主要研究方向为软件工程与软件工程环境、面向对象技术、构件技术和面向服务技术。



马浩海 内蒙古大学计算机学院副教授。2006 年在北京大学获计算机软件与理论专业博士学位。已发表学术论文 30 余篇。主要研究领域为软件工程、面向对象技术、软件建模语言和模型驱动的软件开发技术。



刘辉 北京大学信息科学与技术学院博士生。主要研究领域为面向对象建模、软件重构、元建模以及形式化软件工程方法。

译 者 序

开发一个复杂的软件系统和编写一个简单的程序大不一样，其间的差别，借用 Grady Booch 的比喻，如同建造一座大厦和搭一个狗窝。大型的、复杂的软件系统开发是一项系统工程，必须按工程学的方法来组织软件生产，需要经过一系列的软件生命周期阶段。这是人们从软件危机中获得的最重要的教益。这一认识促使了软件工程学的诞生。虽然编程仍然是重要的，但是更具有决定意义的是系统建模。只有在分析和设计阶段建立了良好的系统模型，才有可能保证工程的正确实施。正是出于这一原因，在编程领域出现的许多新方法和新技术，总是很快地拓展到软件生命周期的分析与设计阶段。

面向对象方法正是经历了这样的发展过程，它首先在编程领域兴起，作为一种崭新的程序设计范型引起世人瞩目。继 Smalltalk-80 之后，20 世纪 80 年代有一大批面向对象编程语言问世，标志着面向对象方法走向成熟和实用。此时，面向对象方法开始向系统设计阶段延伸，出现了一批早期的面向对象的设计（OOD）方法。到 20 世纪 80 年代末期，面向对象方法的研究重点转向面向对象的分析（OOA），并将 OOA 与 OOD 密切地联系在一起，出现了一批面向对象的分析与设计（OOA&D）方法。至 1994 年，公开发表并具有一定影响的 OOA&D 方法已达 50 余种。这种繁荣的局面表明面向对象方法已经深入到分析与设计领域。目前，大多数比较成熟的软件开发组织已经从分析、设计到编程、测试全面地采用面向对象方法，使面向对象毋庸置疑地成为当前软件领域的主流技术。

各种 OOA&D 方法都为面向对象理论与技术的发展做出了贡献。这些方法的主导思想以及所采用的主要概念与原则大体上是一致的，但是也存在不少差异。这些差异所带来的问题是：不利于 OO 方法向一致的方向发展，妨碍了技术交流，也给用户的选择带来困惑。在这种形势下，统一建模语言（Unified Modeling Language，UML）应运而生。

UML 是在多种面向对象分析与设计方法相互融合的基础上形成的，其发展历史可以大致概括为四个阶段。最初的阶段是面向对象方法学家的联合行动，由 G. Booch、J. Rumbaugh 和 I. Jacobson 将他们各自的方法结合起来，形成 UML 0.9。第二阶段是公司的联合行动，由十多家公司组成 UML 伙伴组织，共同提出 UML 1.0 和 1.1，于 1997 年被对象管理组织（OMG）正式采纳作为建模语言规范。第三阶段是在 OMG 控制下对 UML 规范进行修订和改进，产生了 UML 1.2、1.3、1.4 和 1.5 等版本。目前所处的阶段是本世纪开始以来的重大的修订，推出了全新的版本 UML 2.0。

UML 用于对软件密集型系统进行可视化、详述、构造和文档化，也可以用于业务建模以及其他非软件系统的建模。UML 定义了系统建模所需的概念并给出其可视化表示法，但是它并不涉及如何进行系统建模。因此它只是一种建模语言，而不是一种建模方法。UML 是独立于过程的，就是说，它可以适应不同的建模过程。UML 的出现使面向对象建模概念和表示法趋于统一和标准化。目前，UML 已成为被广泛公认的工业标准，拥有越来越多的

用户。现今的大部分面向对象系统的建模均采用 UML。

G. Booch、J. Rumbaugh 和 I. Jacobson 是 UML 的三位主要奠基人，被称为“三友”（three amigos），他们为 UML 的形成和发展做出了卓越贡献。在广大读者的殷切期待中，“三友”联名撰写的三本介绍 UML 以及 Rational 统一软件开发过程的著作（*The Unified Modeling Language User Guide*、*The Unified Modeling Language Reference Manual* 和 *The Unified Software Development Process*）于 1999 年由 Addison-Wesley 出版，深受广大读者的欢迎，被视为 UML 方面的权威性著作。在此之后，UML 经历了一次重大的版本修订，UML 2.0 问世。与 UML 1.x 系列相比，UML 2.0 从结构到内容都有许多重要的变化。在这种背景下，“三友”对他们的上述三本著作进行了再创作，以适应 UML2.0 的新内容，作为第 2 版，于 2005 年陆续出版。

现在我们翻译的《UML 用户指南（第 2 版）》（*The Unified Modeling Language User Guide, Second Edition*）是“三友”的上述三本著作中的一本，是阅读另外两本著作的基础。书中为如何使用 UML 提供了指南，旨在让读者掌握 UML 的术语、规则和惯用法，学会如何有效地使用 UML 进行开发，如何应用 UML 去解决常见的建模问题。实际上，这不仅仅是一部深入介绍 UML 的技术文献，而且处处闪烁着作者在方法学方面的真知灼见，凝结了作者在软件工程、面向对象方法、构件技术等诸多领域的经验和智慧。该书语言生动、深入浅出、实例丰富、图文并茂。对于想学习和使用 UML 的广大读者，这是一本难得的好书。该书的宗旨并不是全面地介绍 UML，也不是完整地介绍软件开发过程，这些内容属于“三友”的另外两本著作。

承担这样一本好书的翻译工作是一项愉快而又严肃的任务。尽管我们对 UML 进行过多年的研究，并且翻译过该书的第 1 版，但是在新版的翻译中仍不敢有驾轻就熟的心理。对翻译中遇到的一些疑难问题，往往要经过反复讨论，并通过对 UML 的进一步研究，才能获得比较准确的译法。这些研究工作得到国家自然科学基金项目 60473064 和国家重点基础研究发展计划（973 计划）课题 2005CB321805 的支持。忠实于原文是我们始终遵循的宗旨！但是原著中存在着个别前后不一致或者与 UML 规范不一致的现象，译文中采用了两种处理方式：对比较明显的错误在译文中做了订正，并通过译者注加以说明；对不太明显的错误按原文翻译，并在译者注中指出疑点。

本书的技术术语译法，以国家标准 GB/T11457—1995 的修订版本（已报批，预计 2006 年颁布）《信息技术 软件工程术语》和正在制定中的《统一建模语言术语标准》作为主要依据。书中绝大部分术语都遵照这两个标准化文件的译法。只有个别的几处例外，其原因在译者注中做了说明。

本书的翻译得到北京大学信息科学技术学院软件研究所许多同事的支持，其中许多问题曾与这些同事进行过深入的研究与探讨。软件研究所的面向对象建模工具课题组的教师和学生也多次参加讨论，贡献了宝贵意见。在此，向他们表示衷心感谢。同时，我们诚恳地希望广大读者对书中可能存在的疏漏和错误给予批评和指正。

译者

2006 年 5 月于北京

前　言

统一建模语言（Unified Modeling Language，UML）是一种用于对软件密集型系统的制品进行可视化、详述、构造和文档化的图形语言。UML 给出了一种描绘系统蓝图的标准方法，其中既包括概念性的事物（如业务过程和系统功能），也包括具体的事物（如用特定的编程语言编写的类、数据库模式和可复用的软件构件）。

本书旨在教会读者如何有效地使用 UML。

本书涵盖了 UML 2.0。

目标

在本书中，读者将获益于以下几点：

- 明白 UML 是什么，不是什么，以及为什么 UML 对于开发软件密集型系统的过程非常重要。
- 掌握 UML 的术语、规则和惯用法，一般说来，还将学会如何有效地使用这种语言。
- 知道如何应用 UML 去解决许多常见的建模问题。

本书为UML具体特征的使用提供了参考资料，但它不是一本全面的UML参考手册；全面的参考请参阅我们编写的*The Unified Modeling Language Reference Manual*第2版（Rumbaugh、Jacobson、Booch合著，Addison-Wesley出版公司2005年出版）[1]。

本书描述了使用UML进行开发的过程，但并没有提供对于开发过程的完整参考资料。开发过程是*The Unified Software Development Process*（Jacobson、Booch、Rumbaugh合著，Addison-Wesley出版公司1999年出版）[2]一书的重点。

最后，本书提供了如何运用 UML 去解决许多一般的建模问题的提示和技巧，但没有讲述如何去建模。本书类似于一本编程语言的用户指南，它教用户如何使用语言，而不教用户如何编程。

读者对象

进行软件开发、部署和维护的人员均可使用 UML。本书主要针对用 UML 进行建模的开发组成员，但它也适用于为了理解、建造、测试和发布一个软件密集型系统而一起工作的人员，他们要阅读这些模型。虽然这几乎包含了软件开发组织中的所有角色，但本书特别适

1. 中文版已由机械工业出版社出版，中文书名《UML 参考手册》。——编者注
2. 中文版已由机械工业出版社出版，中文书名《统一软件开发过程》。——编者注

合下述人员阅读：分析员和最终客户（他们要详细说明系统应该具有的结构和行为）、体系结构设计人员（他们设计满足上述需求的系统）、开发人员（他们把体系结构转换为可执行的代码）、质量保证人员（他们检验并确认系统的结构和行为）、库管理人员（他们创建构件并对构件进行编目）、项目及程序管理者（他们一般是把握方向的领导者，要进行有序的管理，并合理地分配资源，以保证系统的成功交付。

使用本书的人员应该具有面向对象概念的基本知识。如果读者具有面向对象编程的经验或懂得面向对象的方法，就能更容易掌握本书内容，但这并不是必需的。

怎样使用本书

初次接触 UML 的开发人员最好按顺序阅读本书。第 2 章提出了 UML 的概念模型，读者应特别予以注意。所有的章节都是这样组织的——每一章建立在前面各章的内容之上，循序渐进。

至于正在寻求用 UML 解决常见的建模问题的有经验的开发人员，可以按任意顺序阅读本书。读者应该特别注意在各章中提到的常见建模问题。

本书的组织及特点

本书主要由 7 个部分组成：

- 第一部分　入门
- 第二部分　对基本结构建模
- 第三部分　对高级结构建模
- 第四部分　对基本行为建模
- 第五部分　对高级行为建模
- 第六部分　对体系结构建模
- 第七部分　结束语

本书还包含两个附录：UML 表示法的概要和 Rational 统一过程的概要。在附录后，提供了一个常见术语表和一个索引。

每章都描述了针对 UML 具体特征的用法，其中的大部分按下述 4 节的方式组织：

（1）入门

（2）术语和概念

（3）常用建模技术

（4）提示和技巧

第 3 节“常用建模技术”提出一组常见建模问题并予以解决。为了便于读者浏览本书找到这些 UML 的应用场合，每一个问题都标有一个明显的标题，如下例所示。

对体系结构模式建模

每一章都从它所涵盖的特征概要开始，如下例所示。

本章内容

- 主动对象、进程和线程
- 对多控制流建模
- 对进程间通信建模
- 建立线程安全的抽象

类似地，把附加的解释和一般性的指导分离出来作为注解，如下例所示。

注解　UML 中的抽象操作对应于 C++中的纯虚操作；叶子操作对应于 C++的非虚操作。

UML 的语义是非常丰富的，因此对一个特征的描述自然会涉及另一个特征。在这种情况下，在自然段的最后部分标注交叉引用，正如本段这样。【第 15 章讨论构件。】

在图中使用灰色字[1]是为了表明这些文字不是模型本身的一部分，只是用于解释模型。程序代码用 Courier 字体表示以示区别，如 `this example`。

致谢

作者向 Bruce Douglass、Per Krol 和 Joaquin Miller 表示感谢，谢谢他们帮助审阅了第 2 版的书稿。

UML 简史

通常公认的第一个面向对象语言是 1967 年由 Dahl 和 Nygaard 在挪威开发的 Simula-67。虽然该语言从来没有得到大量拥护者，但是它的概念给后来的语言以很大启发。Smalltalk 在 20 世纪 80 年代早期得到了广泛的使用，到 20 世纪 80 年代晚期跟着出现了其他的面向对象语言，如 Objective C、C++和 Eiffel 等。方法学家面对新类型的面向对象编程语言的涌现和不断增长的应用系统复杂性，开始试验用不同的方法来进行分析和设计，由此在 20 世纪 80 年代出现了面向对象建模语言。在 1989 年到 1994 年之间，面向对象的方法从不足 10 种增加到 50 种以上。面对这么多的方法，很多用户很难找到一种完全满足他们要求的建模语言，于是就加剧了所谓的“方法战”。一些明显突出的方法脱颖而出，其中包括 Booch 方法、Jacobson 的 OOSE（面向对象的软件工程）和 Rumbaugh 的 OMT（对象建模技术）。其他的重要方法还有 Fusion 方法、Shlaer-Mellor 方法和 Coad-Yourdon 方法。这些方法中的每一种方法都是完整的，但是每一种方法又都被认为各有优点和缺点。简单来说，Booch 方法在项目的设计和构造阶段的表达力特别强，OOSE 对以用况作为一种途径来驱动需求获取、分析和高层设计提供了极好的支持，而 OMT 对于分析和数据密集型信息系统最为有用。

到 20 世纪 90 年代中期方法之争到了转折点，当时 Grady Booch（Rational 软件公司）、

1. 本书中用灰色字代替原版中的蓝字表示这种区别。——编者注

James Rumbaugh（通用电气公司）、Ivar Jacobson（Objectory 公司）和其他一些人开始从彼此的方法中取长补短，他们的共同成果，开始在全球范围内被公认为是领导性的面向对象方法。作为 Booch 方法、OOSE 方法和 OMT 方法的主要作者，促进我们创建统一建模语言的原因有三个。首先，我们的方法已经在朝着相互独立的方向演化，而我们希望它朝着一个方向演化，这样可以消除任何不必要的和不合理的潜在差别，因为这样的差别会加重用户的疑惑。第二，通过统一我们的方法，能够给面向对象的市场带来一定的稳定，能够让人们使用一种成熟的建模语言去设计项目，使工具开发人员把焦点集中于最有用的特征。第三，希望我们的合作能够改进三种早期的方法，帮助我们吸取教训，解决我们以前的方法不能妥善处理的问题。

统一工作之始，我们确立了 3 个工作目标：

（1）运用面向对象技术对系统进行从概念到可执行制品的建模。

（2）解决复杂系统和关键任务系统中固有的规模问题。

（3）创造一种人和机器都可以使用的建模语言。

设计一种用于面向对象分析和设计的语言与设计一种编程语言不同。首先，必须缩小问题范围：这个语言是否应包含需求描述？这个语言是否应支持可视化编程？其次，必须在表达能力和表达的简洁性之间做好平衡。太简单的语言将会限制能够解决的问题的范围，而太复杂的语言将会使开发人员无所适从。在统一现有方法的情况下，也必须小心从事。若对语言进行太多的改进，将会给已有用户造成混乱；若不对语言进行改进，则将会失去赢得更广大的用户群和使语言得到简化的时机。UML 的定义力争在这些方面做出最好的选择。

1994 年 10 月，Rumbaugh 加入 Booch 所在的 Rational 公司，自此正式开始了 UML 的统一工作。我们的计划最初注重于联合 Booch 方法和 OMT 方法。“统一方法”（当时的名称）0.8 版本（草案）在 1995 年 10 月发布。差不多就在那时，Jacobson 也加入了 Rational 公司，于是 UML 项目的范围又做了扩充，把 OOSE 也结合进来。经过我们的努力，在 1996 年 6 月发布了 UML 0.9 版本。在 1996 年全年，我们在软件工程界征求和收集反馈意见。在此期间，明显地有很多软件组织把 UML 作为商业战略来考虑。我们与几个愿意致力于定义一个强大而完善的 UML 的组织一起成立了一个 UML 伙伴组织。对 UML 1.0 版本做出贡献的合作伙伴有 DEC、HP、I-Logix、Intellicorp、IBM、ICON Computing、MCI Systemhouse、Microsoft、Oracle、Rational、TI 和 Unisys。这些合作伙伴协作产生的 UML 1.0 版本是一个定义明确、富有表现力、强大、可应用于广泛问题域的建模语言。Mary Loomis 帮助说服 OMG（对象管理组织）发布了一个标准建模语言的提案需求（RFP）。在 1997 年 1 月，作为对该提案的响应，UML 1.0 作为标准化的建模语言提交给 OMG。

在 1997 年 1 月至 7 月之间，合作伙伴的队伍不断扩大，实际上包括了所有对最初 OMG 的提议做出贡献的公司，它们是 Andersen Consulting、Ericsson、ObjecTime Limited、Platinum Technology、PTech、Reich Technologies、Softeam、Sterling Software 和 Taskon。为了制定 UML 规范，并把 UML 与其他的标准化成果结合起来，成立了一支由 MCI Systemhouse 公司的 Cris Kobryn 领导并由 Rational 公司的 Ed Eykholt 管理的语义任务组。在 1997 年 7 月，把 UML 的修改版（1.1 版本）提交给 OMG，申请进行标准化审查。1997 年 9 月，OMG 的分

析与设计任务组（Analysis and Design Task Force，ADTF）和 OMG 的体系结构部接受了该版本，并把它提交给 OMG 的全体成员进行表决。1997 年 11 月 14 日，UML 1.1 版本被 OMG 采纳。

几年来，UML一直由OMG的修订任务组（Revision Task Force，RTF）维护，陆续研制了UML的1.3、1.4和1.5版本。从2000年到2003年，一个经过扩充了的新的伙伴组织制定了一个升级的UML规范，即UML 2.0。由IBM的Bran Selic领导的定案任务组（Finalization Task Force，FTF）对这个版本进行了为期一年的评审，UML 2.0的正式版本于2005年初被OMG采纳。UML 2.0是对UML 1的重大修订，包括了大量的新增特性。此外，基于先前版本的经验，UML 2.0对先前版本的构造物做了很多的修改。可以在OMG的网站www.omg.org上获得当前的UML规范文档。

UML 是很多人的工作成果，它的思想来自于大量的先前工作。重新构造一个贡献者的完整列表将是一项很大的历史性研究工程，根据对 UML 影响大小来识别那么多的先驱者就更为困难。同所有的科学研究和工程实践一样，UML 只是站在巨人肩上而已。

目　录

第一部分　入门

第二部分　对基本结构建模

第三部分 对高级结构建模

第五部分 对高级行为建模

第六部分 对体系结构建模