

软件工程和过程工程丛书

可视化面向对象建模技术

——标准建模语言 UML 教程——

刘超 张莉 编著

可视化面向对象建模技术

刘超 张莉 编著

北

773201/02
版社
BH

北京航空航天大学出版社

软件工程和过程工程丛书

可视化面向对象建模技术

—标准建模语言UML教程—

刘超 张莉 编著



C0465160

北京航空航天大学出版社

内 容 简 介

UML 是 OMG 于 1997 年 11 月批准的标准建模语言, 是 20 多年来面向对象技术发展史上的重大里程碑。本书首先扼要介绍 UML 的发展历史, 然后详细介绍 UML 的用例图、类图、对象图、交互图(顺序图和合作图)、状态图、活动图、构件图和配置图等九种模型的构成要素、符号表示、语法和语义, 并介绍基于 UML 的软件开发过程, 最后给出用 UML 进行程序设计的示例。在每章之后还附有习题, 以帮助读者复习与掌握该章的主要内容。本书可作为各类专业本科高年级学生和硕士研究生学习面向对象技术的教材, 也可作为工程技术人员自学面向对象技术的参考书。

图书在版编目 (CIP) 数据

可视化面向对象建模技术: 标准建模语言 UML 教程/刘超等编著. —北京: 北京航空航天大学出版社, 1999.7

ISBN 7-81012-910-4

I. 可… II. 刘… III. 面向对象语言, UML-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字 (1999) 第 24446 号

JS67061

可视化面向对象建模技术 —标准建模语言UML教程—

刘超 张莉 编著

责任编辑 杨昌竹

*

北京航空航天大学出版社出版发行
北京市学院路 37 号 (100083) 发行部电话: 82317024
<http://www.buaapress.cn.net>

E-mail: pressell@publica.bj.cninfo.net
河北省涿州市新华印刷厂印装 各地书店经销
北京赛柏科技有限责任公司录入排版

*

开本: 787×1092 1/16 印张: 9 字数: 226 千字
1999 年 7 月第 1 版 1999 年 7 月第 1 次印刷 印数: 8 000 册
ISBN 7-81012-910-4/TP·357 定价: 17.00 元

软件工程和过程工程丛书

编委会

主编 周伯生

编委 (按汉语拼音音节排序)

董士海 顾毓清 金茂忠

施伯乐 尤晋元 郑人杰

软件工程和过程工程丛书新书预告

《过程工程原理和过程工程环境基础》一书,周伯生等著,2000年3月出版。在知识经济时代,企事业正面临着科技、信息、资金、人才、市场等方面的严峻考验,企事业要在激烈的竞争环境下生存和发展,不能仅仅依靠缩短周期、降低成本和提高质量等传统手段,还必须根据企事业内外条件的变化,对变化迅速的竞争环境,动态地、准确地、快速地作出反应,并通过决策、规划、管理、调度和运作等各个层次付诸实施,实现过程系统技术和管理技术的综合集成和整体优化。本书将总结作者10年来研究过程工程原理和开发过程工程环境的经验和教训,详细描述可视化过程建模语言VPML的设计原则、符号表示以及语法和语义,给出描述企事业过程的多维模型,提出研制过程工程环境的研制原则,详细介绍作者所研制的企事业过程建模系统及其建模实例,并指出集成化过程工程环境既可在柔性信息系统的研制阶段起确定需求和优化结构的先导作用,又可在柔性信息系统的运作阶段起监视运行和动态优化的支持作用。《过程工程原理和过程工程环境基础》一书可作为计算机专业和管理专业本科高年级学生和硕士研究生学习过程工程原理和研究过程工程环境基础的教材,也可供各类企事业主管和有关工程技术人员进行企事业过程重组(BPR)和建造企事业资源规划系统(ERP)时参考。

序

社会生产力的基本结构表现为人类利用生产工具来加工劳动对象。生产力的进步取决于生产工具的变革与创新，而生产工具的进步又取决于人类利用资源的能力，呈现出一个由资源、工具、生产力、时代四个元素组成的连锁反应过程。物质、能量和信息是制造生产工具的三项基本资源。一个不以人的意志为转移的历史规律是，人类对资源的认识和利用必然从物质资源开始，进而到能量资源，再深入到信息资源。目前我们所处的时代是信息化的时代，是人类进入能够综合利用物质、能量和信息三种资源的时代。千百年来以传统的物质产品的生产、流通、消费为基本特征的物质型经济，将逐步进入以信息产品的生产、流通、利用和消费为基本特征的现代化的知识型经济。

知识经济不是以资金、设备、厂房等有形资产为主的经济，而是以知识、信息、技术等无形资产为基础，以知识增长率、知识利用率和知识回报率为核心，以信息数字化为代表的经济。知识经济是当代发展的主旋律，也是未来几十年发展的基本潮流。这种趋势是对我们的严峻挑战，也给我们带来千载难逢的良机。

我们所从事的软件工程和过程工程领域，是信息技术领域的重要组成部分。国际上，这一领域在前几年内取得了前所未有的进展，其成果超过过去 10 至 15 年来的成就总和。其中以英特网作为全球通信工具最为引人注目。把各种信息和知识有机地联系在一起，正在改变人们的生产方式、工作方式、学习方式、社交方式、生活方式以至思维方式。

计算机软件是计算机应用的灵魂。没有先进的软件产业，不可能有先进的信息产业。关于软件产业，在美国学术界与工业界一致认为将经历三个不同的发展阶段。第一个阶段是软件结构化生产阶段（70 年代中期至 90 年代中期），以结构化分析与设计、结构化评审、结构化程序设计以及结构化测试为特征。从 80 年代中期开始，软件生产开始进入以过程为中心的第二个阶段，以提出过程成熟度模型 CMM、个体软件过程 PSP 和群组软件过程 TSP 为标志，这个阶段在美国预计于 2005 年前后结束。从 1995 年开始，以软件过程、面向对象和构件重用等三项技术为基础，正在逐步进入以软件工业化生产为特征的第三个时代。

我们在十多年来的研究和开发工作实践中，一直在努力寻找适合我国特点的发展软件产业的技术途径，积累了一些经验，也有不少教训。其基本观点是：为了适应 21 世纪对信息技术的要求，软件产业必须走软件工业化生产的道路。一方面需要营造软件工程文化，培养大量既懂信息技术又懂企业管理的高级人才，建立必要的信息产业的基础设施；另一方面还需要建立过程工程、系统工程、面向对象技术、软件过程以及软件质量工程等五个支持环境，以全面支持和促进软件产业的建立和发展。下面简要论述我们对这五类环境的基本观点。

在知识经济时代，企业正面临着科技、信息、资金、人才、市场等方面的严峻考验，企业要在激烈的竞争环境下生存和发展，不能仅仅依靠缩短周期、降低成本和提高质量等传统手段，还必须根据企业内外条件的变化，对变化迅速的竞争环境，动态地、准确地、快速地作出反应，并通过决策、规划、管理、调度和运作等各个层次付诸实施，实现过程系统技术和管理技术的综合集成和整体优化。我们建议建造集成化过程工程环境来为这个目的服务，使其在柔性信息系统的研制阶段起确定需求和优化结构的先导作用，在柔性信息系统的运作阶段起监视运行和动态优化的支持作用。

集成化系统工程环境宜采用快速原型与程序变换相结合的技术，由系统需求规格说明直接生成该系统的各个功能部件及其可执行系统，然后通过系统模拟进行系统功能与性能的优化，并将该系统中的各个部件分别用硬件、软件或非计算机部件来实现，使整个系统能够合理地满足原先提出的系统需求。我们建议将集成化系统工程环境用于各类计算机应用系统的建模、模拟、分析、评估和优化，以大幅度提高软件的生产效率，同时为各类非计算机专业人员设计他所需要的软件创造条件。

面向对象技术是软件工程和过程工程领域中的重要技术，出现于 70 年代末期。它不仅是一种程序设计方法，更重要的是一种对真实世界的抽象思维方式。由于它比较自然地模拟了人类认识客观世界的方式，一经诞生就具有强大的生命力。标准建模语言 UML 不仅支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程，是近年来最重要的、具有划时代意义的成果之一。我们认为，面向对象技术是软件产业的主要支柱技术之一，是每一个合格的软件开发人员必须掌握的基本技术；采用标准建模语言 UML，至少是近 10 年内值得提倡的途径。

50 多年来计算事业的发展使人们认识到要高效率、高质量和低成本地开发软件，必须改善软件生产的过程。软件生产转向以改善软件过程为中心，是世界各国软件产业或迟或早都要走的道路。当代世界上关于软件过程有三个流派：美国的 CMM/PSP/TSP、ISO/9000 系列以及 ISO/SPICE 标准。目前，以美国国防部支持的 CMM/PSP/TSP 流派研究得最为深入，试用得也最为广泛。这里过程成熟度模型 CMM 侧重于开发组织内外的各项管理，个体软件过程 PSP 侧重于训练开发人员的技能，群组软件过程 TSP 侧重于开发项目的组织。我们主张建造面向 CMM/PSP/TSP 的软件过程支持环境。

软件系统的质量往往是决定复杂巨系统成败的关键。软件系统质量难于保证的主要原因首先是由于软件系统固有的复杂性，次之是由于软件系统及其管理工作固有的不可见性。我们认为，采用定量软件工程有利于提高软件产品的质量，改善管理工作的可见性，但是还不可能做到完全透明，因此建造软件质量工程环境势在必行。建造软件质量工程环境，除了必须有软件过程支持环境作为基础设施之外，这个环境还应该包含相互联系但可分别独立使用直接针对软件系统质量的四个子环境：软件测试系统、软件质量度量系统、软件产品评测系统以及产品质量监控系统。

我受北京航空航天大学出版社的委托，邀请了数位软件界的资深学者组成了编委会，组织编写有关过程工程、系统工程、面向对象技术、软件过程以及软件质量工程等五个支持环境方面的丛书，重点介绍有关这方面的理论基础和工程技术，以便将国内外学术界和工业界已经积累的知识供大家学习、借鉴和使用。我希望这套正在编写的丛书能对营造学术气氛、提高学术水平和技术水平起推波助澜的作用，以适应知识经济时代的要求，促进我国建立以过程工程为中心的第三代软件产业，并尽早向以软件工业化生产为特征的第三个时代过渡。但我深深知道，这一领域的知识浩如烟海，本丛书从选题至内容难免存在不足之处，诚请专家、读者对我们提出建议、批评和指正。

北京航空航天大学软件工程研究所

周伯生

1999 年 6 月 6 日

前 言

作为本书的署名作者，我们必须郑重声明，这本书事实上是在周伯生教授的主持和指导下编写完成的。同时，也有葛科、王云、孙自安和杨顺祥等研究生的辛勤工作和智慧的贡献。

早在 1996 年初，周伯生教授就已经开始关注 Booch, Rumbaugh 和 Jacobson 三人致力于统一面向对象方法的努力。1996 年 9 月，他在参加美国华盛顿召开的东部软件开发技术国际会议时，获悉 OMG 正在审批 UML 作为面向对象建模语言标准的消息，立即返回中国，与北京航空航天大学软件工程研究所商讨如何开展学习 UML 和研制 UML 支持环境的问题。1997 年 5 月，在周伯生教授的推动下，组织了由张莉副教授负责的 UML 课题组，有计划地开展了一系列的研讨、培训和开发工作。至 1998 年 9 月，张莉副教授、王雷博士、葛科、王云和杨顺祥等研究生先后完成了下列工作：

1. UML 的学习与研究。收集的资料包括 UML 1.0 和 UML 1.1 的正式文本，Martin Fowler 率先出版的 *UML Distilled – Applying The Standard Object Modeling Language* (1997) 以及 Hans-Erik Eriksson 和 Magnus Penker 合著的 *UML Toolkit*。

2. 用了大约 10 个月的时间，设计与开发了支持 UML 九种模型的标准建模语言 UML 设计系统。这项工作得益于这个课题组于 1994 年 11 月开始的与美国 FunSoft 公司合作研制与开发过程工程环境的经验。这个项目还为快速开发 UML Editor 提供了可重用的核心构件库。目前，这两个项目都在继续，将不断推出功能更完善、使用更方便的工具平台。

3. 1998 年在《计算机世界》上发表了介绍 UML 及其支持工具的 5 篇文章。

4. 1998 年组织了一次 UML 培训班，积极在国内介绍和推广这一先进的软件开发方法和技术。

刘超副教授曾于 1997 年 2 月为计算机科学与工程系的研究生开设了面向对象建模技术课程。当时选用了 Rumbaugh 的 *Object-Oriented Modeling and Design* 一书，选修的学生超过百人。1998 年春季，刘超副教授与张莉、葛科和王云合作编写了介绍 UML 的内部教材《面向对象的可视化建模技术——统一的建模语言 UML 及其应用》。这本教材成为这门研究生课程和另外两个软件工程师高级培训班的主要参考书。授课人数超过二百人。我们所研制和开发的标准建模语言 UML 设计系统则为学生上机实践提供了必要的支持平台。

在我们的教学和科研实践中，我们深切地感到，如果有一本阐述标准建模语言 UML 的教材，对培养掌握面向对象技术的高级人才和在我国加速软件产业的建设将会起积极的推动作用。因此，我们从 1998 年初开始着手收集资料和编辑本书。经多次讨论，我们认为本书应该着重介绍可视化面向对象技术和 UML 的经典内容。本书正是在这些构思和工作的基础上，由作者经过多次反复加工和修改后编写完成的。

我们真诚地感谢周伯生教授对这项工作的积极推动和具体指导，由衷地感谢葛科、王云、孙自安、杨顺祥和课题组的其他同志对这本书的贡献和工作。当然，也要借此机会感谢我们的家人对我们的理解和支持。

北京大学董士海教授、清华大学郑人杰教授和中国科学院软件技术研究所顾毓清研究员在百忙之中认真地审阅了本书，提出了许多宝贵的意见。在此，我们由衷地对这三位教授的认真指导表示感谢。

我们还要真诚地感谢北京航空航天大学出版社社长许传安教授和编审杨昌竹教授对本书编辑出版工作的精心指导和热情支持，感谢在北京航空航天大学软件工程研究所工作的李燕红和周柳芳两位小姐的细心绘图和认真排版。最后我们还要感谢北京航空航天大学软件工程研究所和北京赛柏科技公司对本书的编辑和出版提供的资助。

最后必须声明的是，限于作者水平的局限性，书中不可避免地存在一些问题和不足。因此，真诚地欢迎专家和广大读者批评指正。同时还要说明的是，UML 中用到了一些比较新的概念和术语，而目前这些词汇在国内尚无标准的中文译法。作者在参考国内已有译法的基础上，根据我们的理解尽量选用在语义上最为贴切的译法，并且在书的附录中给出中英文词汇的对照表，以便于读者参阅有关的英文资料。显然，本书中这些词汇的中文译法也难免有欠妥之处，愿与大家一起磋商。

由于 UML 是一门新兴技术，我们对它的理解仍欠全面，且 UML 本身也仍处在不断完善的过程中。因此，我们希望在本科生和研究生的教学过程中，进一步理解 UML 的基本概念，在一年半左右的时间内对其进行再版。例如，在每一章后增加评述和习题的份量，在附录中增加术语解释，以帮助读者理解 UML 的概念。同时我们也将标准建模语言设计系统的研制和应用过程中，总结经验，在教程中增加一章关于用 UML 建模的实例，以加快 UML 在我国的推广和应用的进程。

北京航空航天大学软件工程研究所

刘超 张莉

1999年6月1日

目 录

第一章 引 言	1
1.1 什么是 UML?	1
1.2 标准建模语言 UML 的发展历史.....	2
1.3 标准建模语言 UML 的主要内容.....	3
1.4 标准建模语言 UML 的主要特点.....	6
1.5 使用 UML 的目的是什么?.....	7
1.6 标准建模语言 UML 的应用领域.....	9
习题 1	10
第二章 UML 概述	11
2.1 建模技术	11
2.2 标准建模语言 UML	12
习题 2	17
第三章 软件开发过程概述	19
3.1 过程概述	19
3.2 初始阶段	20
3.3 细化阶段	20
3.4 程序重组技术	27
3.5 构造阶段	30
3.6 移交阶段	31
3.7 何时使用迭代式开发方法	32
习题 3	32
第四章 用 例	33
4.1 用户目标和系统交互功能	33
4.2 用例图	34
4.3 用例模型的获取	37
4.4 小结：用例图的图符表	39
习题 4	39
第五章 类 图	41
5.1 类和对象	41
5.2 三个层次的观点	42
5.3 关 联	43
5.4 属 性	47
5.5 操 作	48

5.6 泛化	50
5.7 授权	51
5.8 约束规则	52
5.9 什么时候使用类图	54
5.10 类图元语小结	54
习题 5	55
第六章 类图的高级概念	57
6.1 构造型	57
6.2 多值分类和动态分类	58
6.3 聚集和组成	60
6.4 派生关联和派生属性	61
6.5 接口和抽象类	62
6.6 引用对象和值对象	63
6.7 多值角色	64
6.8 不变性	64
6.9 分类和泛化	65
6.10 限定关联	65
6.11 关联类	66
6.12 参数化类	68
6.13 可见性	69
习题 6	70
第七章 包图	73
7.1 包图的基本要素	73
7.2 何时使用包图	76
7.3 包图小结	76
习题 7	76
第八章 交互图	77
8.1 顺序图	77
8.2 合作图	80
8.3 顺序图和合作图的比较	81
8.4 何时使用交互图	81
8.5 交互图小结	81
习题 8	83
第九章 状态图	85
9.1 状态图的基本要素	85
9.2 并发状态图	87

9.3 何时使用状态图	88
9.4 状态图小结	89
习题 9	89
第十章 活动图	91
10.1 活动图的基本要素	91
10.2 用例的活动图	93
10.3 泳道技术	95
10.4 分解活动	96
10.5 何时使用活动图	97
10.6 活动图小结	98
习题 10	99
第十一章 构件图和配置图	101
11.1 构件图基本要素	101
11.2 配置图的基本要素	101
11.3 结点和连接	102
11.4 构件和接口	103
11.5 对 象	103
11.6 何时使用构件图和配置图	103
11.7 构件图和配置图小结	104
习题 11	105
第十二章 UML 与程序设计	107
12.1 病人档案：领域模型	107
12.2 病人观察：说明层模型	110
12.3 进行编码	112
习题 12	117
第十三章 标准建模语言 UML 支持环境	119
13.1 柔性软件开发过程及其支持环境	119
13.2 UML 集成化支持环境	121
13.3 小 结	124
索 引	125
参考资料	131

第一章 引言

1.1 什么是 UML?

众所周知，面向对象技术是软件工程领域中的重要技术。这种全新的软件开发思想出现于 70 年代末期，由于它比较自然地模拟了人类认识客观世界的方式，一经诞生就具有强大的生命力。根据美国 1995 年底的报导，在美国其应用情况和前景预测如下表所示（其中“%”表示美国在总的软件开发工作中采用面向对象技术的比重）。

1991 年夏	3%~4%
1993 年末	9%~12%
1995 年夏	15%~20%
到 1996 年为止	~40%
到 1998 年为止	~60%
到 2000 年为止	~80%

特别需要强调的是，面向对象技术不仅是一种程序设计方法，更重要的是，它是一种对真实世界的抽象思维方式。随着计算机应用的飞速发展，软件的复杂程度不断提高，源代码的规模越来越大，项目失败的可能性也相应增加。在长期的研究与实践中，人们愈来愈深刻地认识到，建立简明准确的表示模型是把握复杂系统的关键。模型可以使人们从全局上把握系统的全貌及其相关部件之间的联系，可以防止人们过早地陷入各个模块的细节。因此，面向对象的分析和设计应该从建模开始。建模语言一直是面向对象技术的研究重点。

国际上，软件工程领域在最近三年内（1995 年 ~ 1997 年）取得了前所未有的进展，其成果超过软件工程领域过去 10 至 15 年来的成就总和。其中最重要的、具有划时代重大意义的成果之一，就是标准建模语言 UML（Unified Modeling Language）的出现。

UML 是由世界著名的面向对象技术专家 Grady Booch, Jim Rumbaugh 和 Ivar Jacobson 发起，在著名的 Booch 方法、OMT 方法和 OOSE 方法的基础上，广泛征求意见，集众家长，几经修改而完成的。设计者们为 UML 设定的目标是：

- (1) 运用面向对象概念来构造系统模型（不仅仅是针对软件）。
- (2) 建立起从概念模型直至可执行体之间明显的对应关系。
- (3) 着眼于那些有重大影响的问题。
- (4) 创建一种对人和机器都适用的建模语言。

UML 采用了一整套成熟的建模技术，广泛适用于各种应用领域。它得到了工业界的广泛支持，人们普遍认为 UML 定将成为工业界广泛接受的一种标准建模语言。在美国，截至 1996 年 10 月，UML 已经获得工业界和科技界的广泛支持，有 700 多家公司表示将采用 UML 语言作为建模语言。UML 的 1.1 版已经于 1997 年 11 月 17 日被 OMG (Object Management

Group) 批准作为标准, 这标志着近 15 年来面向对象技术中关于建模语言的争论暂时告一段落。在下面的行文中, 为了强调 UML 的地位和作用, 我们将 UML 叫做标准建模语言, 这是非常合适的。

我们认为, 在世界范围内, 至少在近 10 年内, 如同 80 年代的 C 语言和 90 年代的 C++ 语言那样, UML 将成为面向对象技术领域内占主导地位的标准建模语言。我们还认为, 采用 UML 作为我国的标准建模语言是完全必要的: 首先是因为过去数十种面向对象的建模语言都是相互独立地发展的, UML 可以消除一些潜在的不必要的差异, 以免用户混淆; 其次, 通过统一语义和符号表示, 能够稳定我国的面向对象技术市场, 使项目植根于一个成熟的标准建模语言; 同时也可大大拓宽相应的支持工具和开发环境的应用领域, 从而可以加强领域分析和软件设计开发的能力。

需要说明的是, UML 是一种建模语言, 而不是一种方法。在原理上, 任何方法都应由建模语言和建模过程两部分所构成。其中建模语言提供的这种方法中用于表示设计的符号(通常是图形符号); 建模过程则描述进行设计所需要遵循的步骤。标准建模语言 UML 统一了面向对象建模的基本概念、术语及其图形符号, 为人们建立了便于交流的共同语言。然而, 人们可以根据所开发软件的类型、环境和条件, 选用不同的建模过程。

总之, 面向对象技术是软件产业的主要技术支柱之一, 至少在近 10 年内, 应该积极提倡采用标准建模语言 UML 进行建模。本书力图在有限的篇幅内, 从实用的角度出发, 向读者准确、扼要地介绍 UML 的基本思想、模型的主要表示方法以及一些典型的应用。同时, 为了引导读者正确应用建模技术, 对过程也作了简短的讨论。由于我们学识有限, 书中难免有欠妥之处, 诚请读者不惜指正。

1.2 标准建模语言 UML 的发展历史

公认的面向对象建模语言出现于 70 年代中期。从 1989 年到 1994 年, 其数量从不到 10 种增加到了 50 多种。在众多的建模语言中, 语言的创造者各自努力推崇自己的产品, 并在实践中不断完善。但是, 希望采用面向对象方法的用户并不深知这些不同的建模语言各自的优缺点及相互之间的差异, 因而很难根据应用特点选择合适的建模语言。于是爆发了“面向对象技术的方法大战”。90 年代中期, 一批比较成熟的方法受到了工业界与学术界的推崇, 其中最引人注目的是 Booch 1993、OMT-2 和 OOSE 等。

Booch 是面向对象方法最早的倡导者之一。他在 1984 年《Ada 软件工程》(Software Engineering with Ada) 一书中就描述了面向对象软件开发的基本问题。1991 年, 他在《面向对象的设计》(Object-Oriented Design) 一书中, 将以前针对 Ada 的工作扩展到整个面向对象设计领域。他对继承和类的阐述特别值得借鉴。Booch 1993 比较适合于系统的设计和构造。

Rumbaugh 等人提出了面向对象的建模技术 (OMT), 采用了面向对象的概念并引入各种独立于程序设计语言的表示符号。这种方法用对象模型、动态模型、功能模型和用例模型共同完成对整个系统的建模, 所定义的概念和符号可用于软件开发的分析、设计和实现的全过程, 软件开发人员不必在开发过程的不同阶段进行概念和符号的转换。OMT-2 特别适用于分析和描述以数据为中心的信息系统。

Jacobson 于 1994 年提出了面向对象软件工程 (OOSE) 的方法。其最大特点是面向用

例，并在用例的描述中引入了外部角色的概念。用例的概念贯穿于整个开发过程（包括对系统的测试和验证），是精确描述需求的重要武器。目前在学术界和工业界已普遍接受用例的概念，并认为是面向对象技术走向第二代的标志。OOSE 比较适合支持商务工程和需求分析。

除了以 Booch, Rumbaugh 和 Jacobson 为代表的三种主要的面向对象技术之外，还有 Coad/Yourdon 方法，即著名的面向对象分析和面向对象设计方法 OOA/OOD，它是最早提出的面向对象的分析和设计方法之一。该方法简单、易学，适合于面向对象技术的初学者使用。但由于该方法在处理能力方面的局限，目前已很少使用。另外，在美国俄勒冈州波特兰市的 Smalltalk 组织还提出了责任驱动的设计方法和类 - 责任 - 协作卡（即 CRC 卡片），常常可以用来作为描述任务的辅助手段。

概括起来，首先，面对众多的建模语言，用户没有能力区别不同语言之间的差别，因此很难找到一种比较适合其应用特点的语言。次之，在众多的建模语言中，实际上各有千秋，需要取长补短，日臻完善。第三，虽然不同的建模语言大多相同，但仍存在某些细微的差别，这极大地妨碍了用户之间的交流。因此在客观上，极有必要在精心比较不同建模语言的优缺点及总结面向对象技术应用实践的基础上，组织联合设计小组，根据应用需求，取其精华，去其糟粕，求同存异，努力统一这些建模语言。

1994 年 10 月，Grady Booch 和 Jim Rumbaugh 开始致力于这项工作。他们首先将 Booch 93 和 OMT-2 统一起来，并于 1995 年 10 月发布了第一个公开版本称之为标准方法 UM0.8（Unified Method）。1995 年秋，OOSE 的创始人 Ivar Jacobson 加盟到这项工作中。经过 Booch、Rumbaugh 和 Jacobson 三人的共同努力，于 1996 年 6 月和 10 月分别发布了两个新的版本（UML0.9 和 UML0.91），并将 UM 重新命名为 UML。

自 1996 年开始，一些机构将采用 UML 作为其商业策略，宣布将支持并采用这种建模语言。UML 的开发者得到了来自公众的正面反应，并倡议成立了 UML 成员协会，以完善、加强和促进 UML 的定义工作。当时的成员有：DEC, HP, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI 以及 Unisys。这一机构对 UML1.0（1997 年 1 月）及 UML1.1（1997 年 11 月）的定义和发布起了重要的促进作用。

UML 是一种定义良好，易于表达，功能强大，且普遍适用的建模语言。它融入了软件工程领域的新思想、新方法和新技术。它不仅支持面向对象的分析与设计，更重要的是能够有力地支持从需求分析开始的软件开发的全过程。

面向对象技术和 UML 的发展过程可用图 1.1 来表示。在美国，如上所述，截止 1996 年 10 月，UML 获得了工业界和学术界的广泛支持，已有 700 多家公司表示支持采用 UML 作为建模语言。1996 年底，UML 已稳定地占领了面向对象技术市场的 85%，成为可视化建模语言事实上的工业标准。1997 年 11 月 17 日，OMG 采纳 UML 1.1 作为基于面向对象技术的标准建模语言。总之，UML 代表了面向对象方法的软件开发技术的发展方向，具有巨大的市场前景，也具有重大的经济价值和国防价值。

1.3 标准建模语言 UML 的主要内容

首先，UML 融合了 Booch, OMT 和 OOSE 方法中的基本概念。由于这些基本概念与其他面向对象技术中的基本概念大多相同，因而，UML 必然地成为这些方法以及其他方法的使用者们乐于采用的一种简单一致的建模语言。其次，UML 不仅是上述方法的简单汇合，

也是在这些方法的基础上广泛征求意见，集众家之长，几经修改而完成的。UML 扩展了现有方法的应用范围。第三，UML 是标准的建模语言，而不是标准的开发过程。尽管 UML 的应用必然以系统的开发过程为背景，但是对于不同的组织和不同的应用领域，UML 的使用者可以采取不同的开发过程。

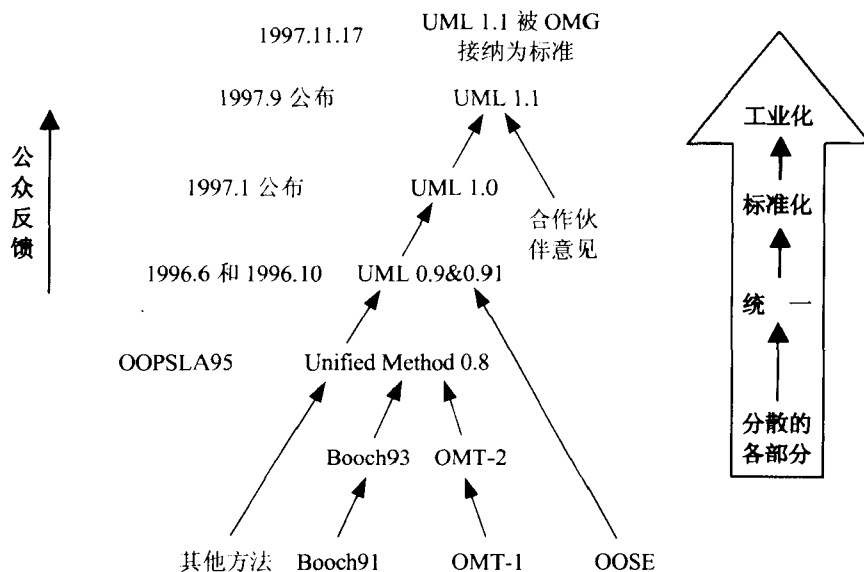


图 1.1 UML 的发展历程

表示法和元模型

标准建模语言 UML 的定义包括 UML 语义和 UML 表示法两个部分。UML 语义通过其元模型来严格地定义。元模型为 UML 的所有元素在语法和语义上提供了简单、一致、通用的定义性说明，使开发者在语义上取得了一致，消除了各种因人而异的表达方法所造成的不良影响。此外，UML 语义还支持对元模型的扩展定义。

UML 表示法定义了 UML 的表示符号，为建模者和建模支持工具的开发者提供了标准的图形符号和正文语法。这些图形符号和文字所表达的是应用级的模型，在语义上它是 UML 元模型的实例。使用这些图形符号和正文语法为系统建模就可建造标准的系统模型。

UML 采用的是一种图形表示法，是一种可视化的图形建模语言。UML 定义了建模语言的文法，例如，类图中定义了诸如类、关联、多重性等概念在模型中是如何表示的。在传统上，人们只是对这些概念进行了非形式化的定义，特别是在不同的方法中，许多概念、术语和表示符号十分相似，但不尽相同甚至相悖。人们期待更严格的定义。UML 运用元模型对语言中的基本概念、术语和表示法给出了统一且比较严格的定义和说明，从而给出了这些概念的准确含义。

图 1.2 是 UML1.1 元模型的一个片段，它表示了关联和泛化之间的关系。这个元模型规定关联和泛化分别是两类不同的关系。但一个关联关系还可联系着两个或多个关联角色，组成一个有序的二元或多元组。

对于 UML 的一般使用者而言,简单地讲,元模型是用来定义符合语法(即语法正确)的模型。因此,侧重于方法学研究的人应该理解元模型,而对于大多数用户则无需过多地探究它,而应当把重点放在逐步深刻地理解那些基本概念,掌握 UML 表示法,学会在对实际系统的分析、设计与实现的过程中,有效地运用标准建模语言 UML 建立系统模型。

鉴于本书旨在向广大读者介绍 UML 的基本概念和表示法,因此书中没有过多地强调严密性,而是沿用了传统的方法,力求直观性。至于建模语言应达到怎样的准确度才算合理,主要取决于建模的目的。如果采用能生成代码的 CASE 工具,则应严格遵循支持该建模语言的 CASE 工具的定义,以便得到可用的代码。如果使用这些模型图仅仅是为了理解需求和促进开发人员与用户之间的交流,则可以有一定的自由度。

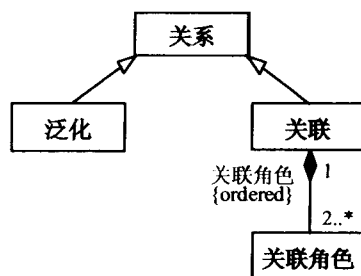


图 1.2 UML 1.1 元模型

需要说明的是,如果没有完全采用某种标准表示法,其他开发人员可能无法完全了解你的意思。但是另一方面,有时某种标准的表示法并不能满足所有的需要,这时也许有必要对它做某些调整或扩充。UML 专门为这类需要提供了一些扩展机制,如构造型等。一般而言,语言的目的是为了交流。如果语言的规则影响了交流,则不应受限于这些约束。

主要内容

客观世界是一个复杂巨系统,需要从不同的角度来考察,才能真正理解这个系统。为了能支持从不同角度来考察系统,标准建模语言 UML 定义了下列 5 类、共 10 种模型图:

第一类是用例图,它从用户角度描述系统的功能,并指出各功能的操作者。

第二类是静态图,包括类图、对象图和包图。其中类图用于定义系统中的类,包括描述类之间的联系(如关联、依赖、聚合等)以及类的内部结构,即类的属性和操作。因此类图是描述系统中类的静态结构,即它所描述的是一种静态关系,在系统的整个生命周期都是有效的。其中对象图所使用的表示符号与类图几乎完全相同。它们的不同点在于对象图只是显示类的对象实例,而不是实际的类。一个对象图是类图的一个实例。由于对象存在生命周期,因此对象图只能在系统某一时间段存在。包图由包或类组成,主要表示包与包、或包与类之间的关系。包图用于描述系统的分层结构。

第三类是行为图,描述系统的动态模型和组成对象间的交互关系。一种是状态图,它描述一类对象的所有可能的状态以及事件发生时状态的转移条件。通常状态图是对类图的补充。实际上我们并不需要为所有的类绘制状态图,而只需要为那些有多个状态、并且其行为受外界环境的影响而会发生改变的类绘制状态图。另一种称作活动图,它描述为满足用例要求所要进行的活动以及活动间的约束关系。使用活动图可以很方便地表示并行活动。

第四类是交互图,描述对象间的交互关系。一种称之为顺序图,用以显示对象之间的动态合作关系。它强调对象之间消息发送的顺序,同时也显示对象之间的交互过程。另一种是合作图,它着重描述对象间的协作关系。合作图跟顺序图相似,显示对象间的动态合作关系。除了显示信息交换外,合作图还显示对象以及它们之间的关系。如果强调时间和顺序,应当使用顺序图;如果强调通信关系,则可以选择合作图。这两种图合称为交互图。

第五类是实现图，包括构件图和配置图。构件图描述代码部件的物理结构以及各部件之间的依赖关系。一个部件可能是一个资源代码部件、一个二进制部件或一个可执行部件。它包含逻辑类或实现类的有关信息。构件图有助于分析和理解部件之间的相互影响程度。配置图定义系统中软硬件的物理体系结构。它可以显示实际的计算机和设备（用节点表示）以及它们之间的连接关系，也可显示连接的类型及部件之间的依赖性。在节点内部，放置可执行部件和对象，以显示节点跟可执行软件单元之间的对应关系。

这些图为系统的分析、开发提供了多种图形表示。它们的有机结合就有可能分析与构造一个一致的系统。从应用的角度看，当采用面向对象技术设计系统时，首先是描述需求，次之根据需求建立系统的静态模型，以构造系统的结构，第三步是描述系统的行为。其中在第一步与第二步中所建立的模型都是静态的，包括用例图、类图（包括包图）、对象图、构件图和配置图等六种图。这些图构成了标准建模语言 UML 的静态建模机制。第三步中所建立的模型或者可以执行或者表示执行时的时序状态或交互关系，它包括状态图、活动图、顺序图和合作图等四种图。这些图构成了标准建模语言 UML 的动态建模机制。因此，标准建模语言 UML 的主要内容也可以归纳为静态建模机制和动态建模机制两大类。

1.4 标准建模语言 UML 的主要特点

如上所述，UML 是在 Booch, OMT 和 OOSE 等三种方法的基础上发展起来的，是这些方法的延续和发展。因此，如果你是 Booch、OMT 和 OOSE 方法的使用者，受过这方面的培训，则你所积累的经验及使用的工具，绝大部分都可以保存下来。但与 Booch, OMT, OOSE 等其他方法相比，UML 具有表达力更强、更清晰和一致的优点。它不仅可以应用在更广泛的应用领域，而且也消除了不同方法在表示法和术语上的差异，避免了符号表示和理解上不必要的混乱。

此外，UML 不仅是从 Booch, OMT, OOSE 演变而来，而且也溶入了其他面向对象方法的可取之处。其中也包括非 OO 方法的影响。UML 符号表示考虑了各种方法的图形表示法，删掉了大量易引起混乱的、多余的和极少使用的符号，也添加了一些新符号。因此，在 UML 中汇入了面向对象领域中很多人的思想。这些思想并不都是 UML 的开发者发明的，而是这些开发者依据最优秀的面向对象方法和丰富的计算机科学实践经验综合提炼而成的。当然，UML 的开发是为了简化建模方法，因此，这些专家也经常在简单（所有元素一律用方框和文字表达）和烦琐（为每个元素设计单独的符号）之间进行权衡。为了避免增加不必要的复杂性，UML 的开发者对增加新概念采取了非常慎重的态度。在 UML 标准中新加了构造型、职责、扩展机制、线程、过程、分布式、并发、模式、合作、活动图等新概念；并清晰地地区分类型、类和实例、细化、接口、构件等概念。

事实上，符号表示和语义的族谱图相当复杂，在这里不可能对其历史做详细的叙述，在此简述它与其他方法的关系。其中用例图与 OOSE 中的描述相似。类图融合了 OMT, Booch 以及其他 OO 方法中对类图的描述。为了支持其他建模风格，增加了扩展机制，如构造型和图标等。而状态图实质上就是 David Harel 的状态图，只做了较小的改动。活动图与很多方法（包括一些非 OO 方法）中的工作流图极为相似。顺序图在许多面向对象方法中也都能看到，只是使用了不同的名字，一些方法将它称作交互作用图、信息追踪图或事件追踪图等。合作图是通过 Booch 的对象图、Fusion 的对象交互作用图以及其他方法中相应的内容修