

 高等教育规划教材

UML基础与建模 实践教学

王先国 主编



提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS

高等教育规划教材

UML 基础与建模实践教程

王先国 主编



机械工业出版社

本书是一本关于 UML 建模的实践教程,以大量案例为基础,重点介绍了 UML 体系结构、UML 元素语义、语法、UML 建模方法和 RUP 统一过程。全书分为 3 篇,共 17 章。第 1 篇(第 1~13 章)为 UML 语言基础,内容包括:UML 语言体系结构、UML 组成元素、UML 图的表示方法、UML 图的作用;第 2 篇(第 14~15 章)为 UML 高级技术,内容包括:Rose 双向工程、RUP 统一软件过程;第 3 篇(第 16~17 章)为 UML 建模实践,内容包括:网上书店建模和气象站数据建模,本篇重点介绍了领域建模与分析过程、用例建模与分析过程、动态建模与分析过程、RUP 分析和设计过程。

本书重点突出了 UML 语言的表示方法、系统建模方法和建模过程。书中所有的概念、技术、建模方法都通过实例来演示,内容精炼,表达简明,实例丰富,非常适合作为高等院校计算机专业及相关专业的教材,也可以作为培训机构相关专业的培训教材。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ:2966938356,电话:010-88379739)。

图书在版编目(CIP)数据

UML 基础与建模实践教程/王先国主编. -北京:机械工业出版社,2015.3
高等教育规划教材

ISBN 978-7-111-51554-8

I. ①U… II. ①王… III. ①面向对象语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 214330 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:郝建伟 责任编辑:郝建伟

责任校对:张艳霞

责任印制:李洋

北京振兴源印务有限公司印刷

2016 年 1 月第 1 版·第 1 次印刷

184mm×260mm·14.25 印张·349 千字

0001—3000 册

标准书号:ISBN 978-7-111-51554-8

定价:36.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
电话服务 网络服务

服务咨询热线:(010) 88379833

机工官网:www.cmpbook.com

读者购书热线:(010) 88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面防伪标均为盗版

金书网:www.golden-book.com

出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

前言

UML 基础与建模实践是计算机科学专业和软件工程专业学生的必修课程，也是一门非常重要的课程。尽管市面上介绍 UML 语言的图书不少，但是很少有一本书的内容在系统分析、设计过程中全面涵盖领域建模、用例建模、动态建模。也很少有教材在系统分析、设计过程中采用 RUP 统一过程开发方法。一般的教材对建模过程的分析和设计与建模方法是脱节的，因此，学生不能真正理解建模技术和建模方法，在实践中更谈不上正确地运用 UML 语言实现面向对象的分析和设计。

学生在建模实践中重点解决的问题主要集中在以下三点：第一，真正理解 UML 表示法，知道如何使用它们；第二，理解统一开发过程（RUP），知道在哪种情况下采用哪种模型来构造系统；第三，知道如何运用建模技术和建模方法。

本书在内容组织上完全针对上述的三个问题，体系结构安排合理，知识表达通俗易懂，知识讲解深入浅出，以两个完整的案例为大、中型软件系统的建模提供了开发步骤、技术提示和表示方法。具体特点如下。

1) 在体系结构的安排上强调内容的系统性、连贯性、逻辑性和实用性。对 UML 元素的语义、语法和建模方法的讲解由易到难逐层展开，便于读者学习和理解。

2) 对 UML 语言的讲解充分体现了文字描述和图形描述的结合。通过文字描述，详细地定义了 UML 元素的语义、语法；通过图形将 UML 元素可视化、规范化；对每个 UML 元素的讲解采用实例演示 UML 元素的语义和使用方法，使读者易于理解。

3) 从框架到细节表达知识。首先对知识进行概要描述，然后分解知识、简化知识，对知识进行详细描述，这样，将复杂的建模技术、建模方法简单化，抽象问题具体化。

4) 提供完整的建模实例。本书以网上书店为例，为读者提供了详细的建模过程、建模技术和建模方法。整个建模流程是可以操作的，也是可以模拟的，学生能真正做到学以致用。

5) 在写作上，本书以 UML 设计元素为主线，以系统建模为目标，运用实例系统地阐明了 UML 语言基础、建模技术和建模方法。本书技术、方法和实践结合生动，知识表达通俗易懂，既可作为高等院校计算机专业及相关专业的教材，也可以作为培训机构相关专业的培训教材。

本书由王先国主编，杨仕范、蔡木生参与了本书的编写工作。本书作者在大型软件公司从事应用系统的分析和设计工作，在开发系统过程中积累了丰富的系统建模方法，能熟练地运用 UML 语言把系统需求分析和系统设计形式化为标准的需求分析文档和设计文档。如有建议或在学习中遇到疑难问题，欢迎大家给作者发送电子邮件（作者邮箱：wangxg588@sohu.com）。本书配备了教学大纲和课件，如果需要，请与出版社联系。

本书中的分析和设计案例，虽然经过了多次修改和审核，但难免会存在疏漏和错误，恳请读者批评指正。

编者

目 录

出版说明

前言

第1章 UML 概述	1
1.1 UML 简介	1
1.1.1 UML 简史	1
1.1.2 UML 定义	2
1.1.3 UML 的特点	2
1.2 模型	3
1.2.1 模型的用途	4
1.2.2 建模目标	5
1.2.3 建模原则	5
1.2.4 为什么要建模	5
1.2.5 系统开发中的模型分类	6
1.3 UML 工具与工具选择	6
1.4 UML 语言应用	7
1.5 小结	8
1.6 习题	8
第2章 UML 语言体系	9
2.1 UML 语言组成	9
2.2 事物	10
2.2.1 结构事物	10
2.2.2 行为事物	13
2.2.3 分组事物	13
2.2.4 注释事物	13
2.3 关系	13
2.4 图和视图	16
2.4.1 UML 图	16
2.4.2 UML 视图	18
2.5 规则和公共机制	19
2.6 系统建模与视图	21
2.7 小结	21
2.8 习题	22
第3章 类图	23
3.1 类的表示	23
3.2 类图的概念	24

3.3	类图中的元素	25
3.4	类间关系	29
3.4.1	依赖关系	29
3.4.2	泛化关系	30
3.4.3	实现关系	31
3.4.4	关联关系	31
3.4.5	关联的属性	33
3.5	阅读类图	35
3.6	小结	36
3.7	习题	36
第4章	对象图	37
4.1	对象	37
4.1.1	对象的三要素	37
4.1.2	对象分类	37
4.2	对象的表示	38
4.3	对象图	39
4.4	对象间的关系	40
4.5	类图与对象图	41
4.6	阅读对象图的方法	42
4.7	小结	42
4.8	习题	42
第5章	包图	43
5.1	包	43
5.2	包的表示	43
5.2.1	包命名	44
5.2.2	包中的元素	44
5.2.3	包的构造型表示法	46
5.3	包图实例	46
5.4	包间关系	47
5.4.1	依赖关系	47
5.4.2	泛化关系	49
5.5	包的传递性	49
5.6	创建包图的方法	50
5.7	包图应用	51
5.7.1	对成组元素建模	51
5.7.2	对体系结构建模	53
5.8	小结	54
5.9	习题	54
第6章	顺序图和协作图	55
6.1	顺序图	55

6.1.1	顺序图的组成	55
6.1.2	顺序图的表示	55
6.1.3	组合区与操作符	58
6.1.4	场景建模	64
6.2	协作图	66
6.2.1	协作图的组成	66
6.2.2	循环和分支控制	67
6.2.3	协作图与顺序图的差异	68
6.3	小结	69
6.4	习题	69
第7章	活动图	70
7.1	活动图的基本概念	70
7.2	活动图的表示	71
7.3	活动图分类	72
7.3.1	简单活动图	72
7.3.2	标识泳道的活动图	73
7.3.3	标识对象流的活动图	74
7.3.4	标识参数的活动图	75
7.3.5	标识别针的活动图	75
7.3.6	标识中断的活动图	76
7.3.7	标识异常的活动图	76
7.3.8	标识扩展区的活动图	78
7.3.9	标识信号的活动图	79
7.3.10	标识嵌套的活动图	80
7.4	活动图的两中建模方法	80
7.4.1	对工作流程建模	81
7.4.2	对操作流程建模	81
7.5	小结	81
7.6	习题	82
第8章	交互概况图	83
8.1	交互概况图的基本概念	83
8.2	交互概况图的绘制	84
8.3	小结	85
8.4	习题	85
第9章	定时图	86
9.1	定时图的表示	86
9.2	定时图应用	86
9.3	小结	88
9.4	习题	88
第10章	状态机图	89

10.1	状态机的组成	89
10.2	状态机图的表示	90
10.2.1	状态的表示法	90
10.2.2	外部迁移的表示法	91
10.2.3	分支的表示法	94
10.3	迁移分类	95
10.4	状态分类	96
10.4.1	简单状态	96
10.4.2	复合状态	97
10.4.3	历史状态	99
10.4.4	子状态机间异步通信	100
10.4.5	建立状态机图的步骤	100
10.5	状态机图应用	101
10.6	小结	101
10.7	习题	101
第 11 章	构件图	102
11.1	接口、端口和构件	102
11.1.1	接口表示法	102
11.1.2	端口表示法	103
11.1.3	构件	104
11.1.4	构件类型	105
11.2	构件的表示	105
11.2.1	未标识接口的构件	106
11.2.2	标识了接口的构件	106
11.3	构件间的关系	106
11.4	构件图分类	108
11.4.1	简单构件图	108
11.4.2	嵌套构件图	109
11.5	制品	110
11.5.1	制品的表示	110
11.5.2	制品的构造型表示	110
11.5.3	制品的种类	111
11.5.4	制品与类的区别	111
11.6	构件图应用	111
11.6.1	对可执行程序建模	111
11.6.2	对源代码进行建模	112
11.7	小结	113
11.8	习题	113
第 12 章	部署图	115
12.1	部署图的基本概念	115

12.2	部署图组成	116
12.2.1	结点	116
12.2.2	连接	117
12.3	部署图应用	117
12.3.1	设计阶段的部署图	117
12.3.2	实现阶段的部署图	118
12.4	小结	119
12.5	习题	119
第13章	用例图	120
13.1	用例图的基本概念	120
13.2	参与者和用例	120
13.2.1	参与者	121
13.2.2	用例	121
13.3	参与者之间的关系	123
13.3.1	识别参与者	123
13.3.2	参与者间的关系	123
13.4	用例之间的关系	124
13.4.1	包含关系	124
13.4.2	扩展关系	125
13.4.3	泛化关系	126
13.5	参与者与用例之间的关系	126
13.6	组织用例	127
13.7	用例规格描述	128
13.7.1	事件流	128
13.7.2	用例模板	129
13.7.3	用例优先级	130
13.7.4	用例粒度	131
13.8	用例描述实例	131
13.9	用例建模要点	133
13.10	小结	134
13.11	习题	134
第14章	Rose 的双向工程	135
14.1	双向工程简介	135
14.2	正向工程	135
14.3	逆向工程	137
14.4	实例应用	138
14.5	小结	143
14.6	习题	143
第15章	统一软件过程 (RUP)	144
15.1	统一软件过程概述	144

15.1.1	RUP 的四个阶段	145
15.1.2	RUP 的迭代模型	146
15.2	RUP 中的核心 workflow	147
15.2.1	需求 workflow	148
15.2.2	分析 workflow	151
15.2.3	设计 workflow	153
15.2.4	实现 workflow	156
15.2.5	测试 workflow	158
15.3	RUP 裁剪	162
15.4	小结	162
15.5	习题	163
第 16 章	网上书店系统分析与设计	164
16.1	领域建模	164
16.1.1	领域建模方法	164
16.1.2	领域建模过程	164
16.2	用例建模	171
16.2.1	用例建模方法	172
16.2.2	用例建模过程	172
16.3	动态建模	189
16.3.1	动态建模方法	189
16.3.2	动态建模过程	189
16.4	小结	196
16.5	习题	196
第 17 章	气象监测系统分析与设计	197
17.1	初始阶段	197
17.1.1	气象监测系统需求	197
17.1.2	定义问题的边界	197
17.1.3	系统用例	203
17.2	细化阶段	204
17.2.1	气象检测系统用例	204
17.2.2	系统架构设计	210
17.3	构造阶段	211
17.3.1	帧机制	211
17.3.2	发布计划	212
17.3.3	传感器机制	213
17.3.4	显示机制	215
17.3.5	用户界面机制	215
17.4	交付阶段	217
17.5	小结	218
17.6	习题	218

第1章 UML 概述

UML 是一种图形符号系统，是一种通用的软件设计语言，可以用该种语言对任何具有静态结构和动态行为的系统进行表示、描述、模拟、可视化和文档化。

1.1 UML 简介

UML (Unified Modeling Language, 统一建模语言) 是用来对业务系统和软件系统进行可视化建模的一种语言。在面向对象的软件开发过程中，常采用本语言对系统的产品进行说明、可视化和文档编写。

1.1.1 UML 简史

公认的面向对象建模语言出现于 20 世纪 70 年代中期。从 1989 年到 1994 年，这种设计语言的数量从不到十种增加到了五十多种。

20 世纪 90 年代，出现了一批新的软件开发方法，其中，Booch、OMT 和 OOSE 等方法得到了广泛的认可。然而，采用不同方法进行建模不利于开发者之间的交流。而 UML 则统一了 Booch、OMT 和 OOSE 的表示方法，而且对其作了进一步的发展。1997 年，UML 被国际对象组织 OMG 采纳为面向对象的建模语言的国际标准，它融入了软件工程领域的新思想、新方法和新技术。UML 不仅支持面向对象的分析与设计，而且还支持从需求分析开始的软件开发的全过程。数年来，UML 凭借其简洁明晰的表达方式、超凡脱俗的表达能力，为业界所广泛认同。

Booch 是面向对象方法最早的倡导者之一，他提出了面向对象软件工程的概念。1991 年，Booch 将之前面向 Ada 的工作扩展到面向整个对象设计领域。Booch 方法较适用于系统的设计和构造。

20 世纪 90 年代，Rumbaugh 等人提出了面向对象的建模技术 (OMT, Object Modeling Technology, 一种软件开发方法)，该方法采用了面向对象的概念，并引入独立于语言的表示符，同时使用对象模型、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。该方法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，但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 OOSE (Object - Oriented Software Engineering) 方法，该方法最大的特点是面向用例 (Use - Case)，并在用例描述中引入了外部角色的概念。用例的概念是精确描述需求的“重要武器”，同时用例贯穿于整个开发过程，包括对系统的测试和验证。

此外，同一时期还有 Coad/Yourdon 方法产生，即著名的 OOA/OOD (Object - Oriented Analysis/Object - Oriented Design)，它是最早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。

面对众多的建模语言，用户首先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；其次，众多的建模语言实际上各有千秋；第三，虽然不同的建模语言大多雷同，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，并根据应用需求，取其精华，去其糟粕，求同存异，统一建模语言。

1994年10月，Grady Booch和Jim Rumbaugh开始致力于这一工作。他们首先将Booch 1993和OMT-2统一起来，并于1995年10月发布了第一个公开版本，称之为统一方法UM 0.8 (Unified Method)。1995年秋，OOSE的创始人Jacobson加盟到这一工作。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，即UML 0.9和UML 0.91，并将UM重新命名为UML (Unified Modeling Language)。

1996年，一些机构将UML作为其商业策略。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。当时的企业成员有DEC、HP、I-Logix、Itellicorp、IBM、ICON Computing、MCI Systemhouse、Microsoft、Oracle、Rational Software、TI以及Unisys。这一机构对UML 1.0 (1997年1月发布)及UML 1.1 (1997年11月17日发布)的定义和发布起了重要的促进作用。

1.1.2 UML 定义

首先，UML融合了Booch、OMT和OOSE方法中的基本概念，而且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必然成为这些方法以及其他方法的使用者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，而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，它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，但由于不同的组织和不同的应用领域，需要采取不同的开发过程。

UML的定义包括UML语义和UML表示法两个部分。

- UML语义：指UML元素符号代表的含义，UML的所有元素在语法和语义上提供了简单、一致、通用的定义和说明，使开发者能在语义上取得一致，消除了因人而异的最佳表达方法所造成的影响。此外，UML还支持元素语义的扩展。
- UML表示法：对UML每个元素符号的表示方法进行了规范。开发者或开发工具在使用这些图形符号时都遵循相应的UML符号的表示准则。

1.1.3 UML 的特点

UML统一了Booch、OMT和OOSE等方法中的基本概念，主要特点如下。

- UML是非专利的第三代建模和规约语言。在开发阶段，UML语言用于说明、可视化、构建和书写面向对象软件制品。
- UML语言应用于软件开发周期中的每一个阶段。OMG (Object Management Group, 对象管理组织) 已将该语言作为业界的标准。
- UML最适用于数据建模、业务建模、对象建模和组件建模。
- UML作为一种模型语言，它可以使开发人员专注于建立产品的模型和结构。当模型建立之后，模型可以被UML工具转化成指定的程序语言代码。

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，它支持从需求分析开始的软件开发的全过程。

1.2 模型

模型就是用图形对一个物体或系统的简化表示，如地球仪就是一个模型，它是对地球的简化表示。用户可以用模型来表示现实领域中的业务流程和工作流程，也可以用模型表示软件领域中的软件系统的组成和结构。

1. 描述系统的模型

在软件领域，将软件系统的模型分为逻辑模型和物理模型两大类。逻辑模型只描述未来系统应该做什么，并不规定采用什么技术来实现系统。逻辑模型不依赖任何技术，系统分析阶段的模型都是逻辑模型；物理模型不仅描述了未来系统做什么，还规定了实现系统所采用的技术，在设计阶段的模型都是物理模型。物理模型是逻辑模型的进一步细化。

2. 描述事物的模型

任何事物都可以用模型来简化表示，下面是生活中常常遇到的4种模型。

1) 交通模型。道路交通图、道路交通标志等图示，如图1-1的模型就是对广州地铁的表示。

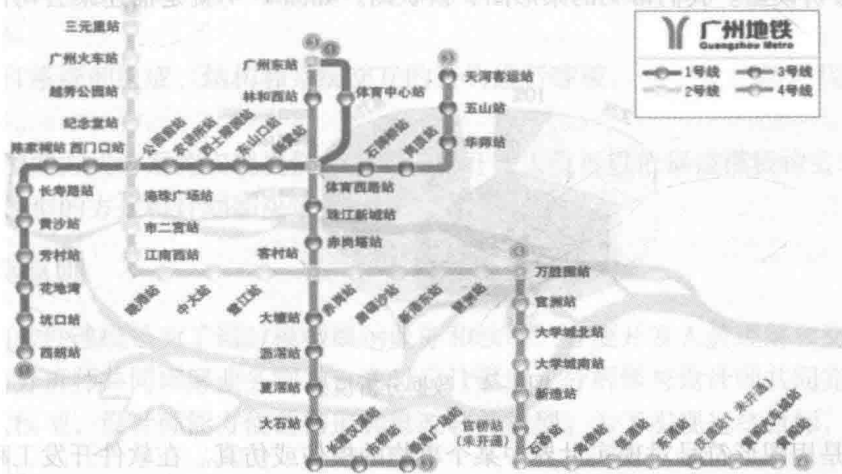


图1-1 广州地铁模型

2) 建筑模型。建筑物模型、沙盘等用来描述建筑物的图形，如图1-2就是描述某集团公司建筑物的模型。



图1-2 建筑模型

3) 设计模型。用来描述管线设计、电路板设计的图形。如图 1-3 就是描述某个局部电路的设计模型。

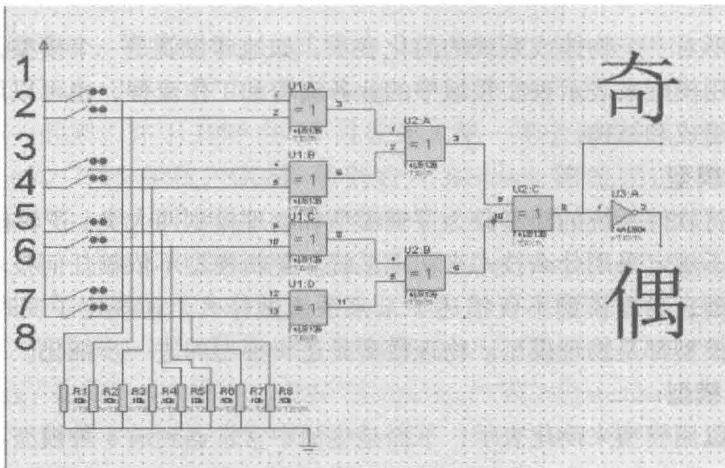


图 1-3 电路设计模型

4) 数据分析模型。我们常见的条形图、饼状图。如图 1-4 就是描述某公司四种产品年销售所占份额。

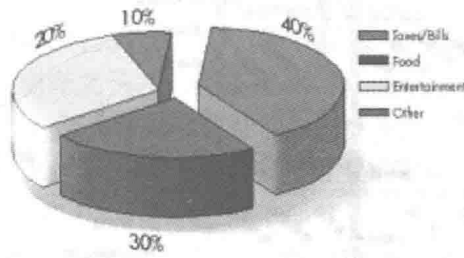


图 1-4 数据分析模型

上述模型是用图形符号对现实世界中某个事物的模仿或仿真。在软件开发工程中，模型主要用来描述问题域和软件域。问题域主要包括业务、业务规则、业务流程和工作流程等；软件域主要包括软件组成、软件结构和软件部署等。

1.2.1 模型的用途

开发软件活动包括两个方面的工作：第一是理解业务和需求（理解问题域）。即，对业务内容、业务过程和业务规则的理解和表示；第二是规划和设计软件系统（设计解决方案）。

由于业务规模和复杂度不断增加，软件的规模和复杂度也随之不断增加，因此，人们对业务的理解以及对软件的设计和构造也越来越困难。此时，在理解业务和需求时，只有借助 UML 这种建模语言来表示和理解业务；在规划和设计软件系统时，只有借助 UML 语言来设计和构造软件系统，以表示和展现系统的组成和交互。

总之，在软件开发活动中，UML 主要用于两个方面的建模：第一是，用 UML 语言对业务系统建模，便于分析师展现和理解业务；第二是，用 UML 对软件系统建模，便于设计师设计、修改软件系统。

1. 对业务系统建模

业务建模就是用 UML 符号表示业务内容和业务过程。用户对自己的业务过程建模，不仅是为了理解业务的内容中规定了要做什么，业务是如何进行的，同样也是为了识别业务的变更对业务造成的影响。对业务建模，有助于发现业务的优缺点，找出需要改进和优化的地方，在某些情况下还可以对几个可选的业务过程进行仿真。

2. 对软件系统建模

软件建模就是用 UML 符号表示软件的体系和组成，方便软件设计人员理解和修改软件方案，确保软件设计和计划能正确的实现，同时，一旦设计和计划需要修改时，修改后的软件系统同样经受得起时间的检验，例如当在一个软件系统中增加一个组件时，要保证系统不会因为增加了该组件而崩溃。

1.2.2 建模目标

对业务系统和软件系统进行建模，主要实现下面 5 个目标。

- 1) 对业务系统进行可视化，建立业务模型。以业务模型为中介，领域专家、用户和需求分析师便于对业务内容和业务过程的理解和交流。
- 2) 对软件架构进行可视化，建立软件体系模型。以体系模型为中介，设计师便于对软件系统宏观理解。
- 3) 对软件系统的组成、结构和系统交互的行为进行建模，便于设计师和代码编写人员对软件深入理解。
- 4) 用模型的方式为系统实现提供一个模板，开发人员可以依据该模板构造软件系统。
- 5) 通过模型的方式将计划和决策文档化。

1.2.3 建模原则

对业务和软件建模是为了通过模型展示业务和软件，方便开发人员理解和交流。通过需求模型，用户与分析师共同理解业务需求；通过设计模型，分析师与设计师共同完成软件设计任务；通过设计模型，设计师能方便的构造和修改软件模型。为了实现这些目标，建模时应遵循以下几个原则。

- 1) 仅当需要时才为业务或软件系统构建模型。简单的业务和软件系统不需要建立模型。
- 2) 模型应该真实地反映业务系统的需求，或者模型能反映软件系统本身的组成和结构。
- 3) 模型应该反映设计师的设计方案。
- 4) 构建模型时，最好用一组相对独立的模型从不同的侧面描述重要的业务或软件系统。

1.2.4 为什么要建模

UML 是一种公共的、可扩展的、应用广泛的设计语言，它可应用于软件开发活动中的每个阶段（分析、设计、实现、测试），而且可以表示每个阶段的产品。

作曲家会将其大脑中的旋律谱成乐曲，建筑师会将其设计的建筑物绘制成蓝图，这些乐曲、蓝图就是模型（Model），而建构这些模型的过程就称为建模（Modeling）。软件开发如同音乐谱曲及建筑设计，其过程中也必须将需求、分析、设计、实现、部署等各项工作流程的构想和设计蓝图表示出来，供分析师、设计师、程序员、测试人员沟通、理解和修改。

建立大厦和搭建狗窝的区别是搭建狗窝不需要设计。因为建立大厦规模庞大，并且设计复

杂，所以，建立大厦前必须有大厦的设计蓝图，而搭建一所狗窝很简单，不需要特别的设计。同理，要开发大规模的复杂软件系统，必须首先了解系统需求，然后对未来的软件系统设计一个蓝图，即，对软件系统进行建模。

1.2.5 系统开发中的模型分类

软件系统从分析、设计到实现的整个过程，包含多种模型的开发。下面描述常见的两种模型分类方法。

1. 按模型的用途分类

如果按模型在软件开发过程中所起的作用，将它们分为3种，它们是：用来表示业务或软件系统的组成和结构的对象模型；用来表示业务系统或软件系统功能的用例模型；用来表示业务或软件系统中组件是如何交互的动态模型。

- 用例模型：从用户的角度展示系统的功能。用例模型常由用例图表示。
- 对象模型：模型展示了软件系统的组成和结构。对象模型由类图或对象图表示。
- 动态模型：展现系统的内部行为。动态模型常由顺序图、活动图和状态图表示。

2. 按产生模型的阶段性分类

在软件开发过程中，不同阶段产生不同的模型。模型按软件开发的阶段性可分为以下五种。

- 业务模型：展示业务过程、业务内容和业务规则的模型。常用对象模型表示业务模型。
- 需求模型：展示用户要求和业务要求的模型。需求模型常由用例模型表示（用例模型由用例图和用例描述组成）。
- 设计模型：设计模型包含架构模型和详细设计模型。架构模型展示软件系统的宏观结构和组成；详细设计模型展示软件的微观组成和结构。详细设计模型常由对象模型表示。
- 实现模型（也称为物理模型）：描述了软件组件及其关系（常由构件图或部署图组成）。
- 数据库模型：描述数据组成及其关系。

1.3 UML 工具与工具选择

UML 工具是帮助软件开发人员方便使用 UML 的软件，它的主要功能包括：支持各种 UML 模型图的输入、编辑和存储；支持正向工程和逆向工程；提供和其他开发工具的接口。不同的 UML 工具提供的功能不同，各个功能实现的程度也不同。在选择 UML 工具时应主要考虑的几方面因素是：产品的价格、产品的功能以及与自己的开发环境结合是否密切。

目前主要的 UML 工具有 Rational 公司的 Rose、Together Soft 公司的 Together 和 Microsoft 公司的 Visio 等。

1. UML 工具介绍

Rational 公司推出的 Rose 是目前最好的基于 UML 的 Case 工具之一，它把 UML 有机地集成到面向对象的软件开发过程中。不论是在系统需求分析阶段，还是在对象设计、软件的实现与测试阶段，它都提供了清晰的 UML 表达方法和完善的工具，方便建立其相应的软件模型。使用 Rose 可以方便地进行软件系统的分析和设计，并能很好地与常见的程序开发环境衔接在一起。