



从新手到高手

夏丽华 / 编著

UML 建模、 设计与分析

从新手到高手

紧扣UML专业知识，详解UML 建模、设计与分析。

- 由高校专业教师联合编写，具备专业品质。
- 内容具有全面性、递进性和实用性
- 实例众多、图例丰富、实用性强。
- 附赠高品质素材和案例。



清华大学出版社



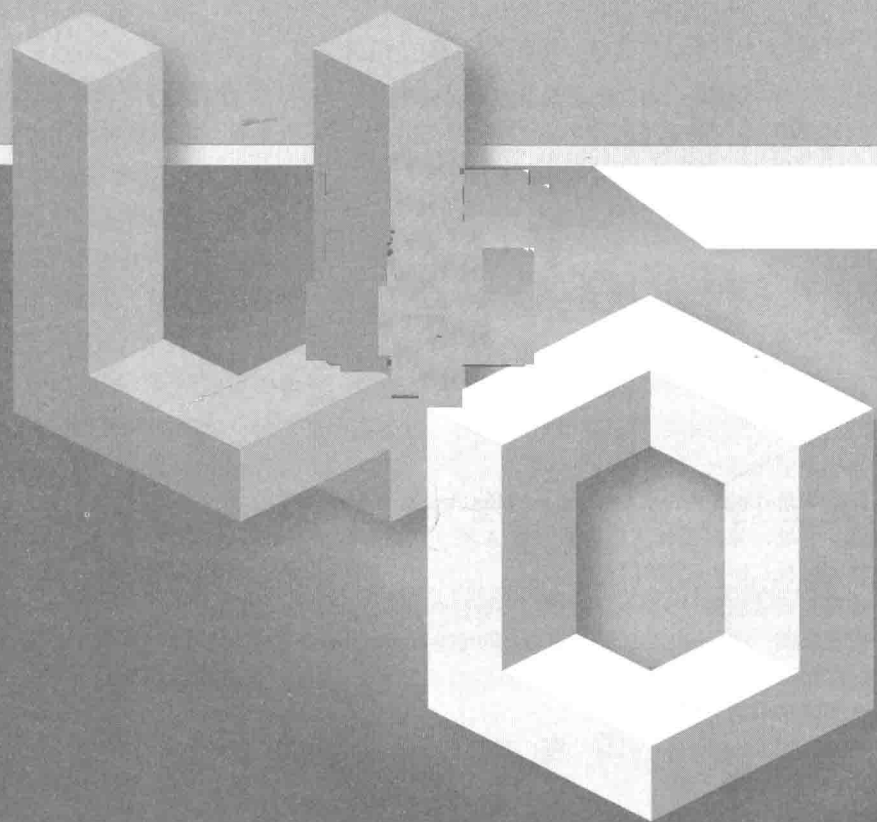


从新手到高手

夏丽华 / 编著

UML 建模、 设计与分析

从新手到高手



清华大学出版社
北京

内 容 简 介

UML 是支持模型化和软件系统开发的图形化语言, 为软件开发的所有阶段提供模型化和可视化支持, 是一种重要的建模、设计与分析工具。全书分 3 篇 19 章, 介绍了 UML 概述、UML 建模工具概述、用例和用例图、类图、对象图和包图、活动图、顺序图、通信图和时序图、状态机图、组件图和部署图、组合结构图和交互概览图、UML 与 RUP、对象约束语言、UML 扩展机制、UML 与数据库设计、基于 C++ 的 UML 模型实现、UML 与建模、Web 应用程序设计、嵌入式系统设计等内容。

本书图文并茂, 秉承了基础知识与实例相结合的特点, 其内容简单易懂、结构清晰、实用性强、案例经典, 适合 UML 建模初学者、大中专院校师生及计算机培训人员使用, 同时也是 UML 爱好者的必备参考书。

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。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目 (CIP) 数据

UML 建模、设计与分析从新手到高手/夏丽华编著. —北京: 清华大学出版社, 2019
(从新手到高手)

ISBN 978-7-302-49199-6

I. ①U… II. ①夏… III. ①面向对象语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 330884 号

责任编辑: 陈绿春 常建丽

封面设计: 潘国文

责任校对: 胡伟民

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者: 三河市铭诚印务有限公司

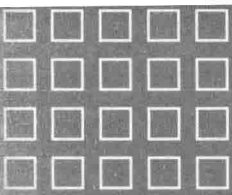
经 销: 全国新华书店

开 本: 190mm×260mm 印 张: 23 字 数: 680 千字

版 次: 2019 年 4 月第 1 版 印 次: 2019 年 4 月第 1 次印刷

定 价: 69.00 元

产品编号: 063669-01



前言

软件的发展至今已经有近 70 年的历史,面向对象技术开始有深入的研究,并广泛应用也近 50 年了,已经成为软件开发中分析、设计、实现的主流方法和技术。UML 始于 1997 年的一个 OMG (对象管理组织) 标准,是一种支持模型化和软件系统开发的图形化语言,可为软件开发的所有阶段提供模型化和可视化支持。它不仅统一了 Booch、Rumbaugh 和 Jacobson 的表示方法,而且做了进一步的发展,并最终统一为大众所接受的标准建模语言。

UML 适用于系统开发过程中从需求分析到完成测试的各个阶段:在需求分析阶段,可以用用户模型视图来捕获用户需求;在分析和设计阶段,可以用静态结构和行为模型视图来描述系统的静态结构和动态行为;在实现阶段,可以将 UML 模型自动转换为用面向对象程序设计语言实现代码。

1. 本书内容介绍

全书系统全面地介绍 UML 建模、设计与分析的应用知识,每章都提供了丰富的实用案例,用来巩固所学知识。本书共分为 19 章,内容概括如下:

第 1 章:为 UML 概述,包括认识 UML、UML 的组成、UML 的视图和通用机制、Rational 统一过程、面向对象开发等内容。

第 2 章:为 UML 建模工具概述,包括常用 UML 建模工具、使用 Rational Rose 建模、Rose 建模的基本操作、逆向工程、正向工程等内容。

第 3 章:为用例和用例图,包括用例图的构成、用例关系和描述、绘制用例图等内容。

第 4 章:为类图,包括类图的概念、泛化关系、依赖关系和实现关系、关联关系、绘制类图等内容。

第 5 章:为对象图和包图,包括对象和类、对象和链、对象图概述、包图概述、包之间的关系、对象图和包图建模、绘制对象图等内容。

第 6 章:为活动图,包括活动图概述、活动图的组成元素、分支与合并、分叉与汇合、绘制活动图等内容。

第 7 章:为顺序图,包括顺序图概述、顺序图的构成元素、建模和执行、绘制顺序图等内容。

第 8 章:为通信图和时序图,包括通信图概述、操作消息元素、时序图概述、时间约束和替代、绘制通信图等内容。

第 9 章:为状态机图,包括状态机概述、事件、动作、转移的类型、组合状态、绘制状态机图等内容。

第 10 章:为组件图和部署图,包括组件图概述、部署图概述、组合组件图和部署图、绘制部署图、绘制组件图等内容。

第 11 章:为组合结构图和交互概览图,包括内部结构、端口、协作、组成部分、使用交互、组合交互等内容。

第 12 章:为 UML 与 RUP,包括 RUP 概述、RUP 的二维空间、核心工作流程、Rose 在 RUP 模型中的应用等内容。

第 13 章:为对象约束语言,包括对象约束语言概述、数据类型、创建集合、操作集合、对象级约束、消息级约束、约束和泛化等内容。

第 14 章：为 UML 扩展机制，包括 UML 的体系结构、UML 核心语义、构造型、标记值、约束等内容。

第 15 章：为 UML 与数据库设计，包括数据库设计概述、类图到数据库的转换、完整性与约束验证、数据库实现与转换技术等内容。

第 16 章：为基于 C++ 的 UML 模型实现，包括模型元素的简单实现、实现关联、受限关联的实现、UML 关系的实现、特殊类的实现等内容。

第 17 章：为 UML 与建模，包括数据建模，业务建模和 Web 建模等内容。

第 18~19 章：通过 Web 应用程序设计和嵌入式系统设计 2 个综合案例，详细介绍了 UML 在建模、设计和分析方面的实际应用。

2. 本书主要特色

- **系统全面，超值实用** 全书提供了 15 个练习案例和 2 个综合案例，通过示例分析、设计过程讲解 UML 建模、设计与分析的应用知识。每章穿插大量提示、分析、注意和技巧等栏目，构筑了面向实际的知识体系。本书采用了紧凑的体例和版式，相同的内容下，篇幅缩减了 30% 以上，实例数量增加了 50%。
- **串珠逻辑，收放自如** 统一采用三级标题灵活安排全书内容，摆脱了普通培训教程按部就班讲解的窠臼。每章都配有扩展知识点，便于用户查阅相应的基础知识。本书内容安排收放自如，方便读者学习。
- **全程图解，快速上手** 各章内容分为基础知识和实例演示两部分，全部采用图解方式，图像均做了大量的裁切、拼合、加工，信息丰富，效果精美，阅读体验轻松，上手容易。
- **新手进阶，加深印象** 全书提供了 77 个基础实用案例，通过示例分析、设计应用，全面加深 UML 建模、设计与分析的基础知识应用方法的讲解。新手进阶部分，每个案例都提供了操作简图与操作说明。

3. 本书使用对象

本书从 UML 的基础知识入手，全面介绍了 UML 建模、设计与分析面向应用的知识体系。本书可作为高职高专院校学生学习用书，也可作为计算机办公应用用户深入学习 UML 建模的培训和参考资料。

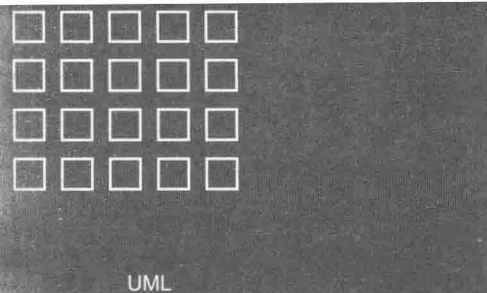
参与本书编写的人员除了封面署名人员之外，还有于伟伟、王翠敏、冉洪艳、刘红娟、谢华、张振、卢旭、吕咏、扈亚臣、程博文、方芳、房红、孙佳星、张彬、马海霞等。

由于编者水平有限，疏漏之处在所难免，欢迎读者朋友登录清华大学出版社的网站 www.tup.com.cn 与我们联系，帮助我们改进提高。

本书相关素材请扫描封底的二维码进行下载。如果在下载过程中碰到问题，请联系陈老师，联系邮箱：chenlch@tup.tsinghua.edu.cn。

编者

2019 年 1 月



目 录

准备篇

第 1 章 UML 概述	2
1.1 认识 UML	3
1.1.1 UML 的发展历程	3
1.1.2 UML 统一的作用	3
1.1.3 UML 体系结构	4
1.1.4 UML 建模流程	5
1.2 UML 的组成	5
1.2.1 事物	5
1.2.2 关系	7
1.2.3 图	7
1.3 UML 的视图和通用机制	10
1.3.1 UML 的视图	10
1.3.2 通用机制	11
1.4 Rational 统一过程	12
1.4.1 过程的特点	12
1.4.2 阶段和迭代	12
1.4.3 任务和制品	13
1.4.4 制品	14
1.5 面向对象开发	15
1.5.1 面向对象的概念	15
1.5.2 面向对象开发的概述	15
1.5.3 面向对象的主要特征	17

1.5.4 面向对象的层	18
1.5.5 面向对象的模型	19

第 2 章 UML 建模工具概述	20
2.1 常用 UML 建模工具	21
2.1.1 Visio	21
2.1.2 PowerDesigner	22
2.1.3 StarUML	22
2.2 使用 Rational Rose 建模	23
2.2.1 Rational Rose 概述	23
2.2.2 Rational Rose 工作环境	25
2.2.3 Rational Rose 中的视图	26
2.3 Rational Rose 建模的基本操作	27
2.3.1 新建 Rational Rose 模型	27
2.3.2 创建 Rational Rose 框图	28
2.3.3 保存与发布模型	30
2.3.4 导入与导出模型	31
2.3.5 设置全局选项	31
2.4 双向工程	32
2.4.1 正向工程	32
2.4.2 逆向工程	34

基础篇

第 3 章 用例和用例图	36
3.1 用例图的构成	37
3.1.1 什么是用例图	37
3.1.2 系统	37
3.1.3 参与者	38
3.1.4 用例	39
3.1.5 关系	41
3.2 用例关系和描述	41

3.2.1 泛化关系	41
3.2.2 包含关系	43
3.2.3 扩展关系	44
3.2.4 用例描述	45
3.3 绘制用例图	47
3.3.1 新建用例图	47
3.3.2 创建内容	48
3.4 建模实例：创建 BBS 论坛用例图	50

3.4.1 确定系统信息	50	5.1.5 使用对象图测试类图	87
3.4.2 前台功能概述	51	5.2 包图	88
3.4.3 构造用例模型	52	5.2.1 包	88
3.5 新手训练营	54	5.2.2 导入包	90
第4章 类图	57	5.2.3 包图概述	90
4.1 类图的概念	58	5.2.4 包之间的关系	92
4.1.1 类图概述	58	5.2.5 包图和类图的区别	93
4.1.2 类	59	5.3 对象图和包图建模	93
4.1.3 定义类	62	5.3.1 使用对象图建模	93
4.1.4 接口	63	5.3.2 使用包图建模	93
4.2 泛化关系	64	5.4 绘制对象图	94
4.2.1 泛化的含义和用途	64	5.4.1 绘制方法	94
4.2.2 泛化的层次与多重继承	64	5.4.2 绘制实例	94
4.2.3 泛化约束	65	5.5 绘制包图	95
4.3 依赖关系和实现关系	66	5.5.1 绘制包图	95
4.3.1 依赖关系	66	5.5.2 规范使用的共性	96
4.3.2 实现关系	67	5.5.3 绘制嵌套包图	97
4.4 关联关系	68	5.6 建模实例：创建机房系统对象图	98
4.4.1 二元关联	68	5.6.1 创建对象元素	98
4.4.2 关联类	72	5.6.2 创建对象关系	98
4.4.3 或关联与反身关联	72	5.7 新手训练营	99
4.4.4 聚合关系	73	第6章 活动图	100
4.4.5 组合关系	73	6.1 活动图概述	101
4.5 绘制类图	74	6.1.1 定义活动图	101
4.5.1 创建类图	74	6.1.2 活动图的主要元素	102
4.5.2 操作类图	75	6.1.3 了解活动和动作	102
4.5.3 类图的规范	76	6.2 活动图的组成元素	103
4.6 建模实例：创建 BBS 论坛类图	78	6.2.1 基本组成元素	103
4.6.1 创建实体类	78	6.2.2 其他组成元素	105
4.6.2 创建类与类之间的关系图	80	6.3 控制节点	110
4.7 新手训练营	81	6.3.1 分支与合并	110
第5章 对象图和包图	83	6.3.2 分叉与汇合	111
5.1 对象图	84	6.4 绘制活动图	112
5.1.1 对象和类	84	6.4.1 创建活动图	112
5.1.2 对象和链	84	6.4.2 操作活动图	112
5.1.3 对象图概述	85	6.5 建模实例：创建 BBS 论坛	
5.1.4 对象图和类图的区别	86	活动图	116
		6.5.1 建模步骤	116



6.5.2 创建活动图	117	8.3.4 时间	143
6.6 新手训练营	118	8.3.5 状态线	144
第7章 顺序图	120	8.3.6 事件与消息	144
7.1 顺序图概述	121	8.4 时间约束和替代	145
7.1.1 什么是顺序图	121	8.4.1 时间约束	145
7.1.2 顺序图的元素	121	8.4.2 时序图的替代表示法	146
7.2 顺序图的构成元素	122	8.5 绘制通信图	147
7.2.1 对象	122	8.5.1 创建通信图	147
7.2.2 生命线	123	8.5.2 操作通信图	147
7.2.3 消息	123	8.6 建模实例：创建 BBS 论坛通信图	149
7.2.4 激活	128	8.6.1 会员用户功能通信图	149
7.3 建模和执行	128	8.6.2 普通用户功能通信图	150
7.3.1 建模时间	128	8.7 新手训练营	151
7.3.2 执行规范	129	第9章 状态机图	153
7.3.3 建模迭代	130	9.1 状态机概述	154
7.4 绘制顺序图	130	9.1.1 状态机及其构成	154
7.4.1 创建顺序图	130	9.1.2 状态机图标记符	154
7.4.2 操作元素	131	9.2 转移	155
7.4.3 限制因素和图形项配置	132	9.2.1 转移的定义	155
7.5 建模实例：创建 BBS 论坛顺序图	133	9.2.2 事件	156
7.5.1 会员用户功能顺序图	133	9.2.3 动作	158
7.5.2 普通用户功能顺序图	134	9.2.4 转移的类型	159
7.6 新手训练营	135	9.3 组合状态	160
第8章 通信图和时序图	137	9.3.1 顺序状态	160
8.1 通信图概述	138	9.3.2 并发子状态	160
8.1.1 什么是通信图	138	9.3.3 同步状态	161
8.1.2 对象与类角色	138	9.3.4 历史状态	161
8.1.3 关联角色与链接	139	9.3.5 子状态机引用状态	162
8.1.4 消息	140	9.4 绘制状态机图	162
8.2 操作消息元素	140	9.4.1 创建状态机图	162
8.2.1 消息序列号与控制点	140	9.4.2 绘制各类元素	163
8.2.2 创建对象	141	9.5 建模实例：创建自动取款机	
8.2.3 消息迭代	141	状态机图	165
8.3 时序图概述	142	9.5.1 分析状态机图	165
8.3.1 什么是时序图	142	9.5.2 创建状态机图	166
8.3.2 时序图中的对象	143	9.6 新手训练营	167
8.3.3 状态	143	第10章 组件图和部署图	169

10.1 构造实现方式图概述	170	10.5.2 添加组件元素	181
10.1.1 组件图概述	170	10.6 建模实例：创建 BBS 论坛组 件图和部署图	182
10.1.2 部署图概述	170	10.6.1 实现 BBS 论坛组件图	182
10.1.3 组合组件图和部署图	171	10.6.2 实现 BBS 论坛部署图	183
10.2 组件图	171	10.7 新手训练营	183
10.2.1 组件	171		
10.2.2 接口	173		
10.2.3 组件间的关系与 组件嵌套	174		
10.2.4 组件图的建模应用	174		
10.2.5 组件图的适用情况	175		
10.3 部署图	176		
10.3.1 节点和连接	176		
10.3.2 部署间的关系	177		
10.3.3 部署图的适用情况及 绘制	177		
10.3.4 部署图的建模应用	178		
10.4 绘制部署图	179		
10.4.1 初识部署图	179		
10.4.2 添加元素	179		
10.5 绘制组件图	180		
10.5.1 创建组件图	180		
		第 11 章 组合结构图和交互概览图	186
		11.1 组合结构图	187
		11.1.1 内部结构	187
		11.1.2 端口	188
		11.1.3 协作	189
		11.2 交互概览图	190
		11.2.1 组成部分	190
		11.2.2 使用交互	191
		11.2.3 组合交互	192
		11.3 建模实例：创建网上购物 系统用例图	193
		11.3.1 系统概述	193
		11.3.2 创建用例图	195
		11.4 新手训练营	197

进阶篇

第 12 章 UML 与 RUP	200	12.4 Rational Rose 在 RUP 模型中的 应用	214
12.1 RUP 概述	201	12.4.1 可视化建模	214
12.1.1 理解软件开发过程	201	12.4.2 Rational Rose 建模与 RUP	214
12.1.2 什么是 RUP	201	12.4.3 Rational Rose 建模与 RUP 应用实例	215
12.1.3 RUP 的作用	202		
12.1.4 RUP 的特点	203	12.5 建模实例：创建网上购物系统 静态模型	217
12.2 RUP 的二维空间	203	12.5.1 定义系统的类	217
12.2.1 时间维	204	12.5.2 创建类关系	218
12.2.2 RUP 的静态结构	205	12.6 新手训练营	218
12.3 核心工作流程	207		
12.3.1 需求获取工作流	207	第 13 章 对象约束语言	221
12.3.2 分析工作流	209	13.1 对象约束语言概述	222
12.3.3 设计工作流	210		
12.3.4 实现工作流	211		
12.3.5 测试工作流	212		



13.1.1 对象约束语言简介	222	14.3.2 UML 标准构造型	249
13.1.2 语言结构	222	14.3.3 使用 UML 扩展机制 进行建模	252
13.1.3 语言语法	223	14.4 标记值	253
13.1.4 表达式	225	14.4.1 表示标记值	254
13.2 数据类型	225	14.4.2 UML 标准标记值	254
13.2.1 基本数据类型	226	14.4.3 自定义标记值	254
13.2.2 集合类型	227	14.4.4 标记值应用元素	254
13.2.3 OclMessage 类型	227	14.5 约束	255
13.2.4 OclVoid 和 OclAny 类型	227	14.5.1 表示约束	255
13.2.5 模型元素类型	228	14.5.2 UML 标准约束	256
13.3 集合	229	14.5.3 自定义约束	258
13.3.1 创建集合	229	14.6 建模实例：完成网上购物系统的 创建	258
13.3.2 操作集合	229	14.6.1 创建状态机图	258
13.3.3 Collection 类型	230	14.6.2 创建实现方式图	260
13.3.4 Set 类型	231	14.7 新手训练营	261
13.3.5 Bag 类型	232	第 15 章 UML 与数据库设计	263
13.3.6 Sequence 类型	233	15.1 数据库设计概述	264
13.4 语言约束	234	15.1.1 数据库设计与 UML 模型	264
13.4.1 使用约束	234	15.1.2 数据库接口	264
13.4.2 对象级约束	236	15.2 类图到数据库的转换	265
13.4.3 消息级约束	237	15.2.1 基本映射转换	265
13.4.4 约束和泛化	238	15.2.2 类到表的转换	266
13.5 建模实例：创建网上购物系统的交 互模型	239	15.2.3 关联关系的转换	268
13.5.1 顺序图	239	15.2.4 需要避免的映射情况	268
13.5.2 通信图	241	15.3 完整性与约束验证	269
13.6 新手训练营	241	15.3.1 父表的约束	269
第 14 章 UML 扩展机制	243	15.3.2 子表的约束	270
14.1 UML 的体系结构	244	15.4 数据库实现与转换技术	271
14.1.1 UML 扩展机制概述	244	15.4.1 类映射到数据库技术	271
14.1.2 4 层元模型体系结构	244	15.4.2 UML 模型转换为 数据库	272
14.1.3 元元模型层	245	15.4.3 SQL 语句实现 数据库功能	273
14.1.4 元模型层	246	15.5 建模实例：创建图书管理 系统用例图	274
14.2 UML 核心语义	247	15.5.1 需求分析	274
14.2.1 模型元素	247		
14.2.2 视图元素	248		
14.3 构造型	249		
14.3.1 表示构造型	249		

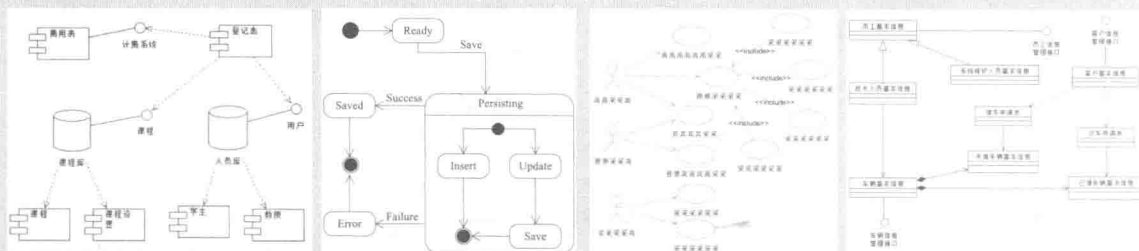
15.5.2 识别参与者和用例	275	16.6.2 定义用户界面类	296
15.5.3 用例描述	276	16.6.3 类之间的关系	298
15.6 新手训练营	278	16.7 新手训练营	299
第 16 章 基于 C++ 的 UML 模型实现	280	第 17 章 UML 与建模	303
16.1 模型元素的简单实现	281	17.1 数据建模	304
16.1.1 类	281	17.1.1 数据库设计概述	304
16.1.2 实现原理	281	17.1.2 数据库设计的步骤	305
16.2 实现关联	283	17.1.3 对象模型和数据模型的 互相转换	308
16.2.1 基本关联	283	17.2 业务建模	309
16.2.2 强制对可选或者 强制关联	284	17.2.1 业务建模概述	309
16.2.3 可选对可选关联	284	17.2.2 UML 业务建模扩展	310
16.2.4 可选对多关联	284	17.2.3 业务体系结构	312
16.2.5 强制对多关联	285	17.3 Web 建模	312
16.2.6 多对多关联	285	17.3.1 Web 建模概述	312
16.2.7 有序关联的实现	286	17.3.2 Web 建模扩展 WAE	313
16.2.8 关联类的实现	286	17.3.3 Rational Rose 中的 Web 建模	315
16.3 受限关联的实现	287	17.4 建模实例：创建图书管理 系统动态行为模型	317
16.3.1 受限关联概述	287	17.4.1 建立顺序图	317
16.3.2 强制或者可选对可选 受限关联	288	17.4.2 建立状态图	325
16.3.3 可选对强制或者可选 受限关联	288	17.4.3 物理模型	326
16.3.4 多对可选的受限关联	289	17.5 新手训练营	327
16.3.5 多对受限关联	290	第 18 章 Web 应用程序设计	329
16.4 UML 关系的实现	290	18.1 Web 应用程序的结构	330
16.4.1 泛化关系的实现	290	18.1.1 瘦客户端模式	330
16.4.2 聚合与组合关系的 实现	291	18.1.2 胖客户端模式	331
16.5 特殊类的实现	291	18.1.3 Web 传输模式	332
16.5.1 接口	292	18.2 Web 应用系统的 UML 建模 方法	332
16.5.2 枚举	292	18.2.1 Web 页面建模	332
16.5.3 包	293	18.2.2 表单建模	333
16.5.4 模板	293	18.2.3 组件建模	333
16.6 建模实例：创建图书管理系统 静态结构模型	294	18.2.4 框架建模	334
16.6.1 定义系统中的 对象和类	294	18.3 UML 在学生成绩管理系统建模中的 运用	334

18.3.1	系统需求分析	334	19.2.1	MP3 播放器的工作原理	345
18.3.2	系统设计	335	19.2.2	外部事件	346
18.4	系统详细设计和部署	339	19.2.3	识别用例	347
18.4.1	系统详细设计	340	19.2.4	使用顺序图描述用例	348
18.4.2	系统部署	341	19.3	系统的静态模型	349
第 19 章	嵌入式系统设计	342	19.3.1	识别系统中的对象或类	349
19.1	嵌入式系统概述	343	19.3.2	绘制类图	349
19.1.1	嵌入式系统的技术特点	343	19.4	系统的动态模型	352
19.1.2	嵌入式系统的开发技术	344	19.4.1	状态机图	352
19.2	嵌入式系统的需求分析	345	19.4.2	协作图	353
			19.5	体系结构	354

准备篇

第 1 章

UML 概述



统一建模语言 (Unified Modeling Language, UML) 是一种支持模型化和软件系统开发的图形化语言, 为软件开发的所有阶段提供模型化和可视化支持, 包括由需求分析到规格, 到构造和配置。面向对象的分析与设计方法的发展在 20 世纪 80 年代末至 20 世纪 90 年代中出现了一个高潮, UML 便是这个高潮的产物, 它能让系统构造者用标准的、易于理解的方式建立起能够表达出他们想象力的系统蓝图, 并且提供了便于不同的人之间有效共享和交流设计结果的机制。

UML 是面向对象软件的标准化建模语言。UML 由于不仅具有简单、统一的特点，而且还能表达软件设计中的动态和静态信息，因此已成为现在诸多领域内建模的首选标准。

1.1.1 UML 的发展历程

UML 起源于多种面向对象建模方法，而面向对象建模语言最早出现于 20 世纪 70 年代中期，到 80 年代末发展极为迅速。据统计，从 1989 年至 1994 年，面向对象建模语言的数量从不到 10 种增加到 50 多种。在众多的建模语言中，各类语言的创造者极力推崇自己的语言，并不断地发展完善它。但由于各种建模语言固有的差异和优缺点，使得使用者很难根据应用的特点选择合适的建模语言。

UML 是 Grady Booch(Booch)、James Rumbaugh (OMT)和 Ivar Jacobson(OOSE)智慧的结晶。其中，OMT 擅长分析，Booch 擅长设计，而 OOSE 擅长业务建模。James Rumbaugh 于 1994 年离开 GE 公司加入 Booch 所在的 Rational 公司，他们一起研究一种统一的方法。

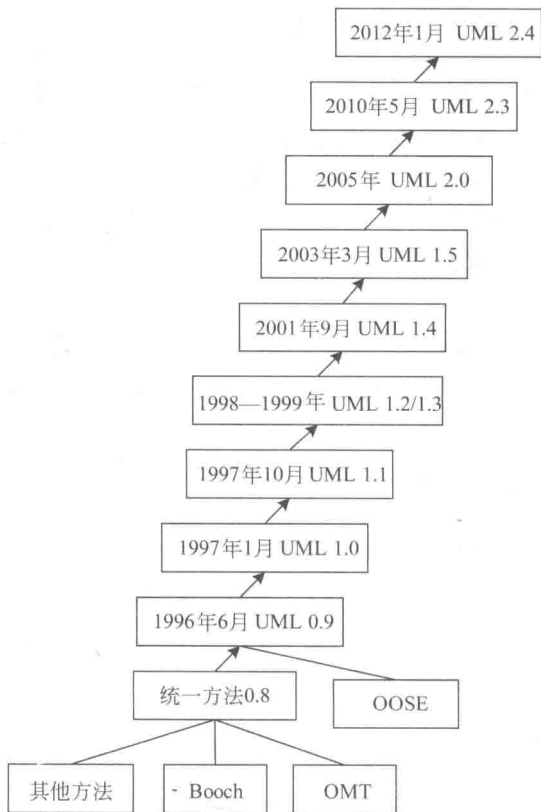
1995 年完成“统一方法 (Unified Method)” 0.8 版。之后 Ivar Jacobson 加入，吸取了他的用例 (Use Case) 思想，于 1996 年完成“统一建模语言” 0.9 版。

1997 年 1 月，UML 版本 1.0 被提交给 OMG (对象管理组织)，作为软件建模语言标准化的候选。随后一些重要的软件开发者和系统集成商成为“UML 伙伴 (UML Partners)”，其中有 Microsoft、IBM 和 HP。经过应用并吸收了开发和其他诸多意见后，于 1997 年 9 月再次提交给 OMG，11 月 7 日正式被 OMG 采纳作为业界标准。

2001 年，UML 1.4 版本被核准推出。2005 年，UML 2.0 标准版发布。UML 2.0 建立在 UML 1.x 基础之上，大多数的 UML 1.x 模型在 UML 2.0 中都可使用，但 UML 2.0 在结构建模方面有了一系列

重大的改进，包括结构类、精确的接口和端口、拓展性、交互片断和操作符，以及基于时间建模能力的增强。

UML 版本变更得比较慢，主要因为建模语言的抽象级别高，所以相对而言，实现语言（如 C#、Java 等）版本变更更加频繁。2010 年 5 月发布了 UML 2.3。2012 年 1 月，UML 2.4 的所有技术环节已经完成，目前只需等待进入 OMG 的投票流程，然后将发布为最新的 UML 规约。同时，UML 也被 ISO 吸纳为标准 ISO/IEC 19501 和 ISO/IEC 19505。UML 的发展历程如下图所示。



1.1.2 UML 统一的作用

UML 的中文含义为统一建模语言。“统一”在 UML 中具有特殊的作用和含义，主要体现在如下

6 个方面。

1. 方法和表示法方面

在以往出现的方法和表示法方面, UML 合并了许多面向对象方法中被普遍接受的概念, 对每种概念, UML 都给出了清晰的定义、表示法和有关术语。使用 UML 可以对已有的各种方法建立的模型进行描述, 并比原来的方法描述得更好。

2. 软件周期方面

在软件开发生命期方面, UML 对开发的要求具有无缝性。开发过程中的不同阶段可以采用相同的一整套概念和表示法, 在同一个模型中, 它们可以混合使用, 而不必转换概念和表示法。这种无缝性对迭代的增量式软件开发至关重要。

3. 应用领域方面

在应用领域方面, UML 适用于各种领域的建模, 包括大型的、复杂的、实时的、分布的、集中式数据或计算的、嵌入式的系统等。

4. 编程语言和开发平台方面

在实现的编程语言和开发平台方面, UML 可应用于运行各种不同的编程实现语言和开发平台的系统。

5. 开发过程方面

在开发过程方面, UML 是一种建模语言, 不是对开发过程的细节进行描述的工具。就像通用程序设计语言可以进行许多风格的程序设计一样。

6. 内部概念方面

在内部概念方面, 在构建 UML 元模型的过程中, 应特别注意揭示和表达各种概念之间的内在联系。试图用多种适用于已知和未知情况的办法把握建模中的概念, 这个过程会增强对概念及其适用性的理解。这不是统一各种标准的初衷, 但却是统一各种标准最重要的结果之一。

1.1.3 UML 体系结构

UML 从 4 个抽象层次上对建模语言的概念、模型元素和结构等进行了全面的定义, 并规定了相应的表示方法和图形符号, 它们分别如下。

- 元元模型层 (Metameta Model) 位于结构的最上层, 组成 UML 的最基本元素“事

物 (Thing)”, 代表要定义的所有事物。

- 元模型层 (Meta Model) 组成 UML 的基本元素, 包括面向对象和面向组件的概念。这一层的每个概念都是元元模型层中“事物”的实例。
- 模型层 (Model) 组成 UML 的模型, 这一层中的概念都是元模型层中概念的实例化。该层的模型通常叫作类模型 (Class Model) 或类型模型 (Type Model)。
- 用户模型层 (User Model) 该层的每个实例都是模型层和元模型层概念的实例。该层中的模型通常叫作对象模型 (Object Model) 或实例模型 (Instance Model)。

上述 4 层体系结构定义了 UML 的所有内容。

具体来说, UML 的核心由视图 (Views)、图 (Diagrams)、模型元素和通用机制组成。

- 视图 视图是表达系统某一个方面特征的 UML 建模元素的子集, 它并不是具体的图, 而是由一个或多个图组成对系统某个角度的抽象。建造完整整个系统时, 通过定义多个反映系统不同方面的视图, 才能做出完整、精确的描述。
- 图 图由各种图片组成, 用于描述一个视图内容。图并不仅仅是一幅图片, 而是在某一个抽象层上对建模系统的抽象表示。UML 中共定义了 9 种基本图, 结合这些图可以描述系统所有的视图。
- 模型元素 UML 中的模型元素包括事物和事物之间的联系。事物描述了面向对象概念, 如类、对象、消息和关系等。事物之间的联系能够把事物联系在一起, 组成有意义的结构模型。常见的联系包括关联关系、依赖关系、泛化关系、实现关系和聚合关系等。
- 通用机制 通用机制用于为模型元素提供额外信息, 如注释、模型元素的语义等, 同时它还提供扩展机制, 允许用户对 UML 进行扩展, 以便适应特殊的方法、组织或用户。

1.1.4 UML 建模流程

了解了 UML 的体系结构之后, 还需要了解一下 UML 建模的流程, 为使用 UML 建模奠定基础。

进行面向对象软件开发建模时, 需要按 5 个步骤进行, 每步都需要与 UML 进行紧密结合, 这 5 步分别是: 需求分析、分析、设计、构造和测试。

1. 需求分析

UML 的用例图可以表示用户的需求。通过用例建模, 可以对外部的角色以及它们所需要的系统功能建模。角色和用例是用它们之间的关系通信建模的。每个用例指定了用户的需求: 用户要求系统做什么。

2. 分析

分析阶段主要考虑所要解决的问题, 可以用 UML 的逻辑视图和动态视图来描述。在该阶段只为问题域类建模, 不定义软件系统解决方案的细节, 如用户接口的类、数据库等。

3. 设计

在设计阶段, 把分析阶段的成果扩展成技术解

决方案。加入新的类来提供技术基础结构、用户接口、数据库等。设计阶段结果是构造阶段的详细规格说明。

4. 构造

在该阶段中, 把设计阶段的类转移成某种面向对象程序设计语言的代码。在对 UML 表示的分析和设计模型进行转换时, 最好不要直接把模型转换成代码。因为在早期阶段, 模型是理解系统并对系统进行结构化的手段。

5. 测试

系统测试通常分为单元测试、集成测试、系统测试和接受测试几个不同的级别。单元测试是对一个类或一组类进行测试, 通常由程序员进行; 集成测试通常测试集成组件和类, 看它们之间是否能恰当地协作; 系统测试验证系统是否具有用户所要求的所有功能; 接受测试验证系统是否满足所有需求, 通常由用户完成。不同的测试小组可以使用不同的 UML 图作为工作基础: 单元测试使用类图和类的规格说明; 典型的集成测试使用组件图和协作图; 系统测试则使用用例图来确定系统行为是否符合图中的定义。

1.2 UML 的组成

至此, 我们已经对 UML 的发展过程有了一定了解, 并且认识了 UML 体系结构中每层的作用。除了上述了解的 UML 基本概述外, 还需要了解一下 UML 的组成。

UML 的组成包括事物、关系和图。其中, 事物是 UML 中的重要组成部分, 关系具有联系元素的作用, 而图则是很多有相互关系的事物的组。

1.2.1 事物

UML 中包括构件事物、行为事物、分组事物和注释事物。

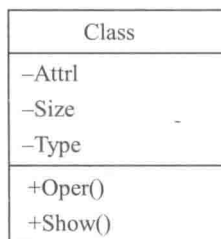
1. 构件事物

构件事物是 UML 模型的静态部分、描述概念或物理元素, 主要包括类、接口、协作、用例、组

件、节点和活动类。

□ 类

类是对具有相同属性、方法、关系和语义的一组对象的抽象。一个类可以实现一个或多个接口。UML 中类的符号如下图所示。



□ 接口

接口是为类或组件提供特定服务的一组操作