

SAMS

UML基础、案例 与应用

[美] Joseph Schmuller 著
李虎 王美英 万里威 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

UML 基础、案例与应用

[美] Joseph Schmuller 著

李 虎 王美英 万里威 译

人民邮电出版社

图书在版编目 (CIP) 数据

UML 基础、案例与应用 / (美) 施穆勒 (Schmuller, J.) 著; 李虎, 王美英, 万里威译.
—北京: 人民邮电出版社, 2002.6

ISBN 7-115-10074-8

I. U... II. ①施... ②李... ③王... ④万... 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 (2002) 第 023702 号

版 权 声 明

Joseph Schmuller : Sams Teach Yourself UML in 24 Hours, Second Edition

Copyright © 2002 by Sams Publishing

Authorized translation from the English language edition published by the Sams Publishing.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

UML 基础、案例与应用

- ◆ 著 [美] Joseph Schmuller
译 李 虎 王美英 万里威
责任编辑 陈冀康
- ◆ 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>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 20
字数: 463 千字 2002 年 6 月第 1 版
印数: 1-5 000 册 2002 年 6 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2001 - 4092 号

ISBN 7-115-10074-8/TP · 2758

定价: 36.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

译者的话

有人认为软件就是机器代码或可执行代码，从某种意义上讲的确如此。但什么是机器代码呢？它是一种描述，一种可以被计算机“阅读”和“理解”的二进制代码序列，这样的二进制代码序列可以由高级编程语言所编制的程序编译而来。

也有人认为软件是用高级程序设计语言（例如 C、C++、Java、SmallTalk）书写的源程序代码。是的，可以这样讲。我们可以继续以这种方式询问什么是广义上的“软件”。那么有关需求、测试、销售、生产、安装和操作又是什么呢？它们是软件吗？事实上，只要用某种方式将它们描述出来，那么所得到的结果就都是“软件”。因此软件开发的过程归根到底是描述的过程。我们经常所说的程序，只是用高级程序设计语言对要解决的问题的一种描述。这种描述实质上是对系统实现的描述，也可以将程序代码说成是用编程语言描述的有关系统实现的模型。那么软件开发过程仅仅是对系统实现的描述吗？需求、设计、测试、安装、部署怎样来描述呢？在面向对象的分析与设计方法中最引人关注的软件制品是模型。模型制品不仅也是软件，而且在现代软件中占有重要的地位。为系统选择和建立模型是当今软件开发中所要做的最重要的工作之一。在开发过程中，每个工作人员都需要有一个独特的系统视角。当设计一个软件时，要描述出所有的典型工作人员（系统分析员、体系结构设计师、程序员、质量保证人员、系统工程师等）和他们的视角。因此，构造软件的过程就是一个构造模型的过程，即采用不同的模型来描述系统所有不同视角的过程。

也许读者会问，既然程序是使用编程语言对系统实现的描述，那么从系统开发过程的其他方面，从不同工作人员的视角出发，有没有描述软件的手段呢？回答是肯定的。描述模型的工具甚至在计算机软件还没有诞生时就有了。体现系统某一方面特征或者从某一视角出发所看到的系统的某一侧面被称为模型。模型是有视角的，从不同的视角可以对同一个系统建立多个模型。就像机械制图中的三视图那样，必须从多个视角出发建立系统的模型才能获得满足我们需要的对系统的描述。建立系统模型的过程被称为建模，描述模型的工具叫做建模语言。在软件出现的早期，编程人员就把他们各自的概念、想法绘制成各种各样的图，或者更进一步用模型来描述。软件开发组织需要一种用模型进行沟通的方法，这种方法不仅用于项目组内部成员之间，还用于与外部的项目相关人员进行沟通。因此必须要有一种建模语言。在 UML 未出现的时候，各开发方法学和软件工程的专家都倡导自己的软件开发方法和相应的建模语言。然而，有这么多种建模语言，究竟采用哪一种好呢？显然，软件产业也需要和其他产业一样，引入国际标准。传统产业在这方面已经做得很好了，而软件产业则刚刚起步。

如果用一句话来概括什么是 UML 的话，那么正如它的名称（统一建模语言，Unified Modelling Language）所描述的那样，它是一种标准的软件建模语言，是一种用于对软件系统的制品（模型）绘制可视化的标准蓝图或者以图表的方式对所加工的产品进行可视化描述的工具。换句话说，它是“软件分析与设计中的标准语言”。

面向对象技术是软件工程领域中的重要技术，它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。尽管 UML 可以应用于任何开发方法中对任何系统建立模型，但它特别适合采用面向对象的思维方式对软件建模。现代软件项目的规模越来越大，越来越复杂，建立简明准确的模型是把握系统的关键。建模语言一直是面向对象技术的研究重点。

幸运的是，美国 Rational 公司的三位著名的科学家 Grady Booch、Jim Rumbaugh 和 Ivar Jacobson 发起提议并广泛征求意见，集众家之长，几经修改，提出了 UML 语言标准草案。它的 1.1 版于 1997 年 11 月 17 日被对象管理组织（简称 OMG）接纳为标准，这标志着 15 年来面向对象技术中关于建模语言的争论告一段落。最近，UML 的最新版本已经被提交给 ISO 国际标准化组织，拟作为系统建模领域的国际正式标准。

需要说明的是，UML 是一种语言，不是一种方法。在原理上，任何软件开发方法都应由建模语言和开发过程两部分构成。其中，建模语言提供这种方法中用户表示分析和设计制品的符号（通常是图形符号）；开发过程则描述分析、设计、实现中所要遵循的步骤和每个步骤所产生的制品。

在世界范围内，至少近十几年内，如同 20 世纪 80 年代的 C 语言和 90 年代的 C++ 语言那样，UML 将长期统领系统建模领域。更为重要的是，据不完全统计，到 1999 年底，全世界已经有 2000 多家公司采纳了 UML 和与 UML 一同推出的统一开发过程（Unified Development Process），且这个数字正以每年 200% 的速度增长。UML 现在已经发展到 1.3 版，不久将发展到 2.0 版。它是软件产业界事实上的工业标准。读者可能已经注意到有许多关于 UML 的书籍上市，因特网上也有很多关于 UML 的讨论。从因特网上也可以搜索到许多著名的国际大公司的一些系统分析和设计的案例，这些案例都是采用 UML 作为标准建模语言描述的。

既然已经有了许多关于 UML 的读物问世，那么本书和它们相比，有哪些特点呢？概括地说，本书有三个最大的特点。首先，以往出版的介绍 UML 的书大都介绍 UML 1.1 版，本书介绍的是最新的 1.3 版。和前面的版本相比，1.3 版中新增加了许多新的内容。其次，本书通俗易懂而又不失严谨准确，不是一上来就介绍 UML 的每种图符和表示法，而是先以现实生活中的事物为例介绍面向对象的基本概念。最后，本书内容丰富全面，不仅介绍了 UML 语言本身，还以完整的开发过程为案例，介绍了如何将 UML 运用到开发过程中。此外，书中还介绍了 UML 在一些热点领域中的应用。总之，通俗易懂、概念准确、理论严谨、内容丰富、案例齐全是本书的特色。

译者在翻译本书时采取了特别慎重和严谨的态度。为了保持这本书中的专业术语的中文译法与最权威的译法一致，译者参考了北京航空航天大学计算机科学与工程系及北京大学计算机科学与技术系组织翻译的一套 UML 教材和指导手册，在此向他们表示感谢。

本书主要部分由李虎翻译，参加翻译工作的还有王美英、万里威等，在此向他们表示感谢。若有不妥之处，敬请读者及有识之士批评指正！

译者
2002 年 3 月

前 言

当我们能够想象出如何运用技术来把事情做得更好时，一个复杂的系统就随之诞生了。开发人员所开发的系统正是要将构想变为现实，因此他们必须要能够充分地理解这种想象力并将其牢记在心中。

一个成功的系统开发项目的成功之处在于它能够在想象者和实现这些想象的系统开发人员之间建立起沟通的桥梁。统一建模语言（Unified Modeling Language, UML）就是一种建立桥梁的工具。它能帮你捕捉住对系统所发挥的想象力，并使你能够用这些想象出来的东西来和项目的风险承担人进行交流。UML 借助于一套符号和图形来帮助我们完成这些工作。每种图形在开发过程中都发挥其各自不同的作用。

不论对第一版还是第二版来说，这本书的目标都是让你通过高效的学习建立起 UML 的牢固基础。在本书的每一章学习内容中都为读者提供一些实例来强化对所学知识的理解，并且在每章后面还留了一些习题让你能够将新知识学以致用。

在写本书的这一版的过程中，我仔细检查了本书的前一版，对其进行了精简，增加和修改了一些必要内容。原书中的一些模型和图在新版中更加详细，并且补充了一些小测验和练习题。同时，根据一些好心读者的指正，我还纠正了书中的一些错误和前后不一致的地方。我将本书分为三部分。第一部分为“基础知识”部分，在这一部分中首先是对 UML 进行了一个综述，然后转向面向对象这个主题，面向对象的概念是建立对象图和类图时要用到的最基本的概念。本部分还讨论了用例（use case）——用于展示从用户的角度所观察到的系统功能的 UML 组件。我还花了额外的时间来讨论和面向对象及用例有关的基本概念，因为在使用 UML 的大部分时间里所要用的东西都建立在这两个基本概念之上。在第一部分剩余的内容中还将介绍其余的 UML 图。

第二部分为“学习案例”。通过一个虚构的学习案例介绍了一种简化的系统开发方法。因此，第二部分说明了如何将 UML 运用到项目开发背景中去。在这部分中你将学习如何运用 UML 的各个组件协同工作来为系统建立模型。

第三部分为“高级应用”部分，我们先是运用 UML 来描述设计模式和嵌入式系统，然后讨论 UML 在其他领域中的应用。

有不少供应商都提供用于创建 UML 图并将这些图组织成为模型的工具软件包。最有名的软件包当数 Rational Rose，另一个是 Select Enterprise。Visual UML 也是一个很有竞争力的产品。此外 GDPro 和 TogetherSoft 公司的建模工具软件也具有一定竞争力。Microsoft 公司已经取得 Rational 公司的许可，使用 Rational 公司的技术开发出了 Visual Modeler 建模工具软件，它是 Rational Rose 的一个子集，被纳入 Visual Studio 集成开发环境中，以供 Visual Studio 的用户使用。就在本书第一版出版之际，MicroSoft 收购了 Visio，在 MicroSoft Visio 高版本的绘图工具中提供了 UML 建模能力。

目 录

第一部分 基础知识

第 1 章 UML 简介	3
1.1 在纷繁复杂中寻求解决问题的办法.....	3
1.2 UML 的诞生.....	4
1.3 UML 的组成.....	5
1.3.1 类图.....	5
1.3.2 对象图.....	6
1.3.3 用例图.....	6
1.3.4 状态图.....	7
1.3.5 顺序图.....	7
1.3.6 活动图.....	9
1.3.7 协作图.....	9
1.3.8 构件图.....	9
1.3.9 部署图.....	10
1.4 其他特征.....	10
1.4.1 包.....	10
1.4.2 注释.....	11
1.4.3 构造型.....	11
1.5 为什么需要这么多种图.....	11
1.6 小结.....	12
1.7 常见问题解答.....	12
1.8 小测验和习题.....	13
1.8.1 小测验.....	13
1.8.2 习题.....	13
第 2 章 理解面向对象	14
2.1 无处不在的对象.....	14
2.2 一些概念.....	16
2.2.1 抽象.....	16
2.2.2 继承.....	16

2.2.3	多态性	17
2.2.4	封装	18
2.2.5	消息传递	19
2.2.6	关联	19
2.2.7	聚集	21
2.3	意义	22
2.4	小结	22
2.5	常见问题解答	23
2.6	小测验和习题	23
第 3 章	运用面向对象思想	24
3.1	类的可视化表示	24
3.2	属性	25
3.3	操作	26
3.4	属性、操作和可视化表达	27
3.5	职责和约束	28
3.6	附加注释	30
3.7	类——应该做什么和如何识别它们	30
3.8	小结	32
3.9	常见问题解答	32
3.10	小测验和习题	33
3.10.1	小测验	33
3.10.2	习题	33
第 4 章	关系	34
4.1	关联	34
4.1.1	关联上的约束	35
4.1.2	关联类	36
4.1.3	链	36
4.2	多重性	37
4.3	限定关联	38
4.4	自身关联	39
4.5	继承和泛化	39
4.5.1	找出继承关系	40
4.5.2	抽象类	41
4.6	依赖	41
4.7	小结	42
4.8	常见问题解答	42
4.9	小测验和习题	42

4.9.1 小测验	43
4.9.2 习题	43
第 5 章 聚集、组成、接口和实现	44
5.1 聚集	44
5.1.1 聚集上的约束	45
5.2 组成	45
5.3 语境	46
5.4 接口和实现	47
5.4.1 可见性	48
5.4.2 作用域	49
5.5 小结	49
5.6 常见问题解答	50
5.7 小测验和习题	50
5.7.1 小测验	50
5.7.2 习题	50
第 6 章 介绍用例	52
6.1 什么是用例	52
6.2 用例的重要性	53
6.3 举例：饮料自动销售机	53
6.3.1 用例“买饮料”	53
6.3.2 其他用例	54
6.4 包含用例	55
6.5 扩展用例	55
6.6 开始用例分析	55
6.7 小结	56
6.8 常见问题解答	56
6.9 小测验和习题	57
6.9.1 小测验	57
6.9.2 习题	57
第 7 章 用例图	58
7.1 用例模型的表示法	58
7.1.1 回顾饮料销售机	59
7.1.2 跟踪场景中的步骤	59
7.2 用例之间的可视化表示	60
7.2.1 包含	60

7.2.2	扩展	61
7.2.3	泛化	61
7.2.4	分组	62
7.3	用例图在开发过程中的作用	62
7.4	运用用例模型：举例	63
7.4.1	理解领域	63
7.4.2	理解用户	63
7.4.3	理解用例	64
7.4.4	进一步深入	64
7.5	“清查存货”	66
7.5.1	结构元素	67
7.5.2	关系	67
7.5.3	分组	67
7.5.4	注释	67
7.5.5	扩展	67
7.5.6	其他	68
7.6	UML “大图”	68
7.7	小结	69
7.8	常见问题解答	69
7.9	小测验和习题	69
7.9.1	小测验	69
7.9.2	习题	70
第 8 章	状态图	71
8.1	什么是状态图	71
8.1.1	符号集	72
8.1.2	在状态图标中增加细节	72
8.1.3	增加转移的细节：事件和动作	73
8.1.4	增加转移的细节：监视条件	74
8.2	子状态	74
8.2.1	顺序子状态	75
8.2.2	并发子状态	75
8.3	历史状态	76
8.4	消息和信号	76
8.5	为什么状态图很重要	77
8.6	UML “大图”	78
8.7	小结	78
8.8	常见问题解答	79
8.9	小测验和习题	79

8.9.1 小测验	79
8.9.2 习题	80
第 9 章 顺序图	81
9.1 什么是顺序图	81
9.1.1 对象	81
9.1.2 消息	81
9.1.3 时间	82
9.2 图形用户界面 (GUI)	82
9.2.1 交互序列	83
9.2.2 顺序图	83
9.2.3 用例	84
9.3 饮料自动销售机: 实例顺序图和一般顺序图	84
9.3.1 实例顺序图	84
9.3.2 一般顺序图	85
9.4 在消息序列中创建对象实例	87
9.5 表示自身调用	89
9.6 UML “大图”	89
9.7 小结	90
9.8 常见问题解答	90
9.9 小测验和习题	90
9.9.1 小测验	90
9.9.2 习题	91
第 10 章 协作图	92
10.1 什么是协作图	92
10.2 图形用户界面 (GUI)	93
10.3 饮料销售机	94
10.4 创建对象	96
10.5 其他概念	96
10.5.1 发送给多对象的消息	97
10.5.2 返回结果	97
10.5.3 主动对象	98
10.5.4 同步	98
10.6 UML 大图	99
10.7 小结	99
10.8 常见问题解答	100
10.9 小测验和习题	100
10.9.1 小测验	100

10.9.2 习题	100
第 11 章 活动图	101
11.1 什么是活动图	101
11.1.1 判定	102
11.1.2 并发路径	102
11.1.3 信号	102
11.2 活动图的应用	103
11.2.1 操作: Fibs	103
11.2.2 过程: 创建一个文档	104
11.3 泳道	106
11.4 混合图	107
11.5 UML 大图	109
11.6 小结	110
11.7 常见问题解答	111
11.8 小测验和习题	111
11.8.1 小测验	111
11.8.2 习题	111
第 12 章 构件图	112
12.1 什么是构件	112
12.2 构件和接口	112
12.2.1 替换和重用	113
12.3 构件的类型	113
12.4 什么是构件图	114
12.4.1 构件的表示法	114
12.4.2 接口的表示法	115
12.5 应用构件图	116
12.5.1 带一个 Java Applet 的 Web 页面	116
12.5.2 带 ActiveX 控件的 Web 页面	117
12.5.3 PowerToys	118
12.6 UML“大图”中的构件图	119
12.7 小结	120
12.8 常见问题解答	120
12.9 小测验和习题	120
12.9.1 小测验	120
12.9.2 习题	120
第 13 章 部署图	122

13.1	什么是部署图	122
13.2	应用部署图	124
13.2.1	家用计算机系统	124
13.2.2	令牌环网	125
13.2.3	ARCnet	126
13.2.4	细缆以太网	126
13.2.5	Metricom 公司的 Ricochet 无线网	127
13.3	UML “大图”中的部署图	128
13.4	小结	129
13.5	常见问题解答	129
13.6	小测验和习题	129
13.6.1	小测验	129
13.6.2	习题	129
第 14 章	理解 UML 语言基础	131
14.1	UML 的组成结构	131
14.2	元模型层	132
14.2.1	基础包	133
14.2.2	行为元素包	134
14.2.3	模型管理包	135
14.3	UML 的扩展	135
14.4	构造型	135
14.4.1	依赖	136
14.4.2	分类	136
14.4.3	类	137
14.4.4	泛化	137
14.4.5	包	137
14.4.6	构件	137
14.4.7	用于其他 UML 元素的构造型	137
14.4.8	图形构造型	138
14.5	约束	138
14.6	标签值	139
14.7	小结	139
14.8	常见问题解答	139
14.9	小测验和习题	140
第 15 章	在开发过程中运用 UML	141
15.1	开发过程方法学：传统的和现代的	141
15.1.1	传统的开发过程方法学	141

15.1.2 新的开发过程方法学	142
15.2 开发过程中必须做什么	142
15.3 GRAPPLE	143
15.4 RAD ³ : GRAPPLE 的结构	144
15.5 需求收集	145
15.5.1 发现业务过程	145
15.5.2 领域分析	145
15.5.3 识别协作系统	146
15.5.4 发现系统需求	146
15.5.5 将结果提交给客户	146
15.6 分析	146
15.6.1 理解系统的用法	147
15.6.2 充实用例	147
15.6.3 细化类图	147
15.6.4 分析对象状态变化	147
15.6.5 定义对象之间的交互	147
15.6.6 分析与协作系统的集成	147
15.7 设计	147
15.7.1 开发和细化对象图	148
15.7.2 开发构件图	148
15.7.3 制定部署计划	148
15.7.4 设计和开发用户界面原型	148
15.7.5 测试设计	148
15.7.6 开始编制文档	148
15.8 开发	148
15.8.1 编制代码	148
15.8.2 测试代码	149
15.8.3 构建用户界面和用户界面到代码的连接及测试	149
15.8.4 完成文档	149
15.9 部署	149
15.9.1 编制备份和恢复计划	149
15.9.2 在硬件上安装最终系统	149
15.9.3 测试安装后的系统	149
15.9.4 庆祝	149
15.10 GRAPPLE 总结	149
15.11 小结	150
15.12 常见问题解答	150
15.13 小测验和习题	151

第二部分 学习案例

第 16 章 学习案例介绍	155
16.1 用 GRAPPLE 开发过程解决问题	155
16.2 发现业务过程	156
16.2.1 招待一位顾客	156
16.2.2 准备饭菜	162
16.2.3 清理餐桌	164
16.3 吸取的经验教训	165
16.4 小结	166
16.5 常见问题解答	166
16.6 小测验和习题	167
16.6.1 小测验	167
16.6.2 习题	167
第 17 章 领域分析	168
17.1 分析业务过程会谈	168
17.2 开发初步类图	169
17.3 对类分组	171
17.4 形成关联	173
17.4.1 Customer 参与的关联	173
17.4.2 Server 参与的关联	175
17.4.3 Chef 参与的关联	177
17.4.4 Busser 参与的关联	178
17.4.5 Manager 参与的关联	178
17.4.6 其他问题	179
17.5 形成聚集和组成	179
17.6 填充类的信息	180
17.6.1 Customer 类	181
17.6.2 Employee 类	181
17.6.3 Check 类	182
17.7 有关模型的一些问题	183
17.7.1 模型词典	183
17.7.2 模型图的组织	183
17.8 吸取的经验教训	183
17.9 小结	184
17.10 常见问题解答	184

17.11 小测验和习题	184
17.11.1 小测验	184
17.11.2 习题	185
第 18 章 收集系统需求	186
18.1 开发系统的映像	186
18.2 收集系统需求	192
18.3 需求联合应用开发会议	193
18.4 结果	195
18.5 下一步该做什么	198
18.6 小结	198
18.7 常见问题解答	198
18.8 小测验和习题	198
18.8.1 小测验	198
18.8.2 习题	199
第 19 章 开发用例	200
19.1 分析和描述用例	200
19.2 用例分析	200
19.3 Server 包	201
19.3.1 用例 “Take an Order”	202
19.3.2 用例 “Transmit the Order to the Kitchen”	202
19.3.3 用例 “Change an Order”	203
19.3.4 用例 “Track Order Status”	203
19.3.5 用例 “Notify Chef about Party Status”	204
19.3.6 用例 “Total Up a Check”	205
19.3.7 用例 “Print a Check”	205
19.3.8 用例 “Summon an Assistant”	206
19.3.9 其余的用例	206
19.4 系统中的构件	207
19.5 小结	207
19.6 常见问题解答	208
19.7 小测验和习题	208
19.7.1 小测验	208
19.7.2 习题	208
第 20 章 交互和状态变化	209
20.1 系统中的工作部件	209

20.1.1	Server 包	209
20.1.2	Chef 包	210
20.1.3	Busser 包	210
20.1.4	Assitant Server 包	210
20.1.5	Assitant Chef 包	210
20.1.6	Bartender Chef 包	210
20.1.7	Coat-Check Clerkf 包	211
20.2	系统中的交互	211
20.2.1	用例 “Take an Order”	211
20.2.2	用例 “Change an Order”	213
20.2.3	用例 “Track Order Status”	215
20.3	结论	215
20.4	小结	216
20.5	常见问题解答	216
20.6	小测验和习题	216
20.6.1	小测验	217
20.6.2	习题	217
第 21 章	设计外观、感觉和部署	218
21.1	GUI 设计的一般原则	218
21.2	用于 GUI 设计的 JAD Session	219
21.3	从用例到用户界面	220
21.4	用于 GUI 设计的 UML 图	222
21.5	描绘出系统的部署	223
21.5.1	网络	224
21.5.2	节点和系统部署图	224
21.6	下一步	225
21.7	听听项目的发起人怎么说	226
21.7.1	扩展销售区的地理范围	226
21.7.2	扩展餐馆的地理范围	227
21.8	小结	228
21.9	常见问题解答	228
21.10	小测验和习题	229
21.10.1	小测验	229
21.10.2	习题	229
第 22 章	理解设计模式	230
22.1	参数化	230
22.2	设计模式	232