

高职高专计算机实用规划教材

—— 案例驱动与项目实践



本书特色：

- 所有概念、技术、建模方法都通过实例来演示
- 以网上书店为例，介绍领域建模、用例建模和动态建模
- 在建模过程中充分体现建模技术和建模方法的具体运用

UML

王爱国 陈辉林 编 著

基础与建模实践



NLIC2970801727

■ 案例驱动

■ 项目实践



提供代码、电子教案下载
<http://www.tup.com.cn>

清华大学出版社

高职高专计算机实用规划教材——案例驱动与项目实践

UML 基础与建模实践

王爱国 陈辉林 编 著

清华大学出版社
北 京

内 容 简 介

本书结合大量软件分析与设计实例,重点介绍了UML语言体系结构、UML元素语义、语法,并通过实际的软件开发实例对UML建模方法进行了详细的讲解。全书文字表述通俗易懂,技术讲解深入浅出。

全书分为3篇,共15章。第1篇(第1章~第12章)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13章)UML建模实践,以网上书店为例,重点介绍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。第3篇(第14章和第15章)UML高级技术,内容包括:Rose的双向工程和统一软件过程(RUP)。通过对本书的学习,读者可以系统地掌握UML语言的阅读方法和建模方法。

本书重点突出了UML语言的表示方法和建模方法。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基础与建模实践/王爱国,陈辉林编著. —北京:清华大学出版社,2012

(高职高专计算机实用规划教材——案例驱动与项目实践)

ISBN 978-7-302-28529-8

I ①U… II. ①王… ②陈… III ①面向对象语言,UML—程序设计—高等职业教育—教材
IV ①TP312

中国版本图书馆CIP数据核字(2012)第066242号

责任编辑:汤涌涛

封面设计:杨玉兰

责任校对:王 晖

责任印制:王静怡

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

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:北京世知印务有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:15.5

字 数:372千字

版 次:2012年5月第1版

印 次:2012年5月第1次印刷

印 数:1~4000

定 价:29.00元

产品编号:044409-01

前 言

“UML 基础与建模实践”是计算机专业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，但是几乎没有一本书在同一个软件分析与设计实例中全面、具体、正确地涵盖三种重要的建模过程：领域建模、用例建模、动态建模，并且，在建模过程中，技术应用和建模方法是脱节的。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 语言来实现面向对象的分析和设计。

学生在建模实践中出现的问题主要集中在以下三点：第一，没有真正理解 UML 表示法，不知道如何使用它们。第二，对统一开发过程(RUP)不理解，不知道在什么情况下采用哪种模型来构造系统。第三，不知道如何运用建模技术和建模方法。

本书以网上书店系统为例，全面介绍领域建模、用例建模和动态建模，并且在建模过程中充分体现建模技术和建模方法的具体运用。本教材为大、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。

本书特色

本书以网上书店系统为例，为读者提供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，在建模过程中提供了启发性技术和建模方法。本书体系结构安排合理，知识组织风格统一，表达通俗易懂，讲解深入浅出，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 语言的讲解中，充分体现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，使读者易于理解。

(3) 知识表达方面，采用框架到细节方法。即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，这样就将复杂的建模技术、建模方法简单化，抽象问题具体化。

(4) 提供完整的建模实例。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。

读者对象

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 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，既适合 UML 语言初学者阅读，也适合系统分析师、设计者和系统测试者阅读；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。

作者情况

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。

本书第 1 章由陈辉林编写，第 2 章～第 15 章由王爱国编写，全书由王爱国统稿。

联系方式

本书中的实例虽然经过了多次测试，但难免会存在疏漏和错误，恳请读者批评指正。如有建议或在学习中遇到疑难问题，欢迎大家发电子邮件与本人(11069881@qq.com)联系。

编 者



目 录

第 1 篇 UML 语言基础

第 1 章 UML 概述3	2.6.2 修饰.....27
1.1 什么是 UML.....3	2.6.3 通用划分.....27
1.1.1 UML 简史.....3	2.6.4 扩展机制.....27
1.1.2 UML 定义.....4	2.7 系统建模与视图.....28
1.1.3 UML 的特点.....4	2.8 小结.....29
1.2 什么是模型.....5	2.9 习题.....29
1.2.1 模型的用途.....7	第 3 章 类图30
1.2.2 建模目标.....8	3.1 什么是类图.....30
1.2.3 建模原则.....8	3.2 类的表示.....31
1.2.4 用 UML 建模的好处.....8	3.3 类图中的元素.....33
1.2.5 模型种类.....9	3.4 类图中的关系.....37
1.3 UML 工具.....9	3.4.1 依赖关系.....37
1.3.1 UML 工具介绍.....10	3.4.2 泛化关系.....38
1.3.2 如何选择 UML 工具.....10	3.4.3 实现关系.....39
1.4 UML 语言应用.....11	3.4.4 关联关系.....40
1.5 小结.....12	3.4.5 关联的属性.....42
1.6 习题.....12	3.5 理解类图.....44
第 2 章 UML 语言体系结构13	3.6 小结.....45
2.1 UML 语言组成.....13	3.7 习题.....45
2.2 元素符号.....14	第 4 章 对象图46
2.2.1 结构符号.....14	4.1 什么是对象.....46
2.2.2 行为符号.....18	4.2 什么是对象图.....46
2.2.3 分组符号.....18	4.3 对象的表示.....47
2.2.4 注释符号.....19	4.4 对象图中的关系.....48
2.3 关系符号.....19	4.5 类图与对象图.....49
2.4 UML 图.....22	4.6 小结.....50
2.4.1 UML 图分类.....22	4.7 习题.....50
2.4.2 视图.....25	第 5 章 包图51
2.5 规则.....25	5.1 什么是包.....51
2.6 公共机制.....26	
2.6.1 规格描述.....26	



5.2 什么是包图.....	52	6.9 用例建模要点.....	80
5.3 包的表示.....	53	6.10 小结.....	81
5.3.1 包命名.....	53	6.11 习题.....	81
5.3.2 包中的元素.....	54	第7章 交互图.....	82
5.3.3 用构造型表示包.....	55	7.1 顺序图.....	82
5.4 包图中的关系.....	56	7.1.1 什么是顺序图.....	82
5.4.1 依赖关系.....	56	7.1.2 顺序图的表示.....	83
5.4.2 泛化关系.....	57	7.1.3 表示分支、并发和循环.....	86
5.5 包的传递性.....	58	7.1.4 顺序图应用.....	93
5.6 创建包图的方法.....	58	7.2 协作图.....	95
5.6.1 标识候选包.....	59	7.2.1 协作图的表示.....	95
5.6.2 调整候选包.....	59	7.2.2 表示循环和分支.....	97
5.6.3 消除包的循环依赖.....	60	7.2.3 顺序图与协作图的关系.....	98
5.7 包图应用.....	60	7.3 定时图.....	98
5.7.1 对成组元素建模.....	61	7.4 小结.....	101
5.7.2 对体系结构建模.....	63	7.5 习题.....	101
5.8 小结.....	63	第8章 活动图.....	102
5.9 习题.....	64	8.1 什么是活动图.....	102
第6章 用例图.....	65	8.2 活动图的表示.....	102
6.1 什么是用例图.....	65	8.3 活动图应用.....	104
6.2 参与者与用例.....	66	8.3.1 简单活动图.....	105
6.2.1 参与者的表示.....	66	8.3.2 展示泳道的活动图.....	106
6.2.2 用例的表示.....	67	8.3.3 展示对象流的活动图.....	107
6.3 参与者之间的关系.....	68	8.3.4 展示参数的活动图.....	108
6.3.1 识别参与者.....	68	8.3.5 展示别针的活动图.....	108
6.3.2 参与者之间的泛化关系.....	69	8.3.6 展示中断的活动图.....	109
6.4 用例之间的关系.....	69	8.3.7 展示异常的活动图.....	110
6.4.1 包含关系.....	69	8.3.8 展示扩展区的活动图.....	111
6.4.2 扩展关系.....	71	8.3.9 展示信号的活动图.....	112
6.4.3 泛化关系.....	72	8.3.10 展示嵌套活动图.....	113
6.5 参与者与用例之间的关系.....	72	8.4 构建活动图.....	114
6.6 组织用例.....	73	8.4.1 对工作流程建模.....	114
6.7 用例规格描述.....	74	8.4.2 对操作流程建模.....	115
6.7.1 事件流.....	75	8.5 小结.....	115
6.7.2 用例模板.....	75	8.6 习题.....	116
6.7.3 用例优先级.....	77	第9章 交互概况图.....	117
6.7.4 用例粒度.....	77	9.1 什么是交互概况图.....	117
6.8 用例描述实例.....	78		

9.2 交互概况图应用	118	11.2.2 构件图的组成元素	141
9.3 小结	120	11.3 构件的表示	141
9.4 习题	120	11.3.1 没有标识接口的构件	141
第 10 章 状态机图	121	11.3.2 标识接口的构件	141
10.1 什么是状态机图	121	11.4 构件间的关系	142
10.2 状态图的表示	122	11.5 构件图分类	143
10.2.1 状态的表示	122	11.5.1 简单构件图	143
10.2.2 迁移的表示	123	11.5.2 嵌套的构件图	144
10.2.3 分支的表示	127	11.6 制品	144
10.3 迁移的分类	128	11.6.1 制品的表示	145
10.3.1 外部迁移	128	11.6.2 制品的标准元素	145
10.3.2 内部迁移	128	11.6.3 制品的种类	145
10.3.3 自动迁移	129	11.6.4 制品与类的区别	146
10.3.4 复合迁移	129	11.7 建模应用	146
10.4 状态的分类	130	11.7.1 对可执行程序建模	146
10.4.1 简单状态	130	11.7.2 对源代码建模	147
10.4.2 复合状态	130	11.8 小结	148
10.4.3 历史状态	132	11.9 习题	148
10.4.4 子状态机间的异步通信	133	第 12 章 部署图	149
10.4.5 建立状态机图的步骤	134	12.1 什么是部署图	149
10.5 状态机图应用	134	12.1.1 部署图的作用	149
10.6 小结	135	12.1.2 部署图的组成元素	150
10.7 习题	135	12.2 部署图的表示	150
第 11 章 构件图	136	12.2.1 节点	150
11.1 什么是接口和构件	136	12.2.2 连接	151
11.1.1 接口	136	12.3 部署图应用	152
11.1.2 端口	137	12.3.1 设计阶段	152
11.1.3 构件	138	12.3.2 实现阶段	153
11.1.4 构件类型	139	12.4 小结	153
11.2 什么是构件图	140	12.5 习题	153
11.2.1 构件图的作用	140		

第 2 篇 UML 建模实践

第 13 章 网上书店	157	13.1.2 领域建模过程	157
13.1 领域建模与分析过程	157	13.2 用例建模与分析过程	164
13.1.1 领域建模方法	157	13.2.1 用例建模方法	165





13.2.2 用例建模过程.....	165	13.3.2 动态建模过程.....	186
13.3 动态建模与分析过程.....	186	13.4 小结.....	194
13.3.1 动态建模方法.....	186	13.5 习题.....	194

第3篇 UML 高级技术

第14章 Rose 的双向工程.....	197	15.2.3 RUP 中的迭代模型.....	211
14.1 双向工程简介.....	197	15.3 RUP 中的核心 workflow.....	212
14.2 正向工程.....	197	15.3.1 需求 workflow.....	212
14.3 逆向工程.....	200	15.3.2 分析 workflow.....	216
14.4 实例应用.....	201	15.3.3 设计 workflow.....	218
14.5 小结.....	205	15.3.4 实现 workflow.....	221
14.6 习题.....	205	15.3.5 测试 workflow.....	225
第15章 统一软件过程(RUP).....	206	15.4 RUP 裁剪.....	229
15.1 当前流行的软件过程.....	206	15.5 RUP 的十大要素.....	229
15.2 统一软件过程概述.....	207	15.6 小结.....	232
15.2.1 RUP 的核心 workflow.....	208	15.7 习题.....	232
15.2.2 RUP 的4个阶段.....	210	附录A UML 图总结.....	233
		参考文献.....	240

第 1 篇

UML 语言基础

本篇包括 1~12 章。

第 1 章介绍 UML 的发展历程、UML 语言的定义、UML 特点、UML 工具、UML 应用领域和 UML 模型种类。

第 2 章介绍 UML 语言体系结构，即 UML 语言的组成元素、语法规则、图形种类、公共机制和 UML 视图。

第 3~12 章介绍 UML 语言的 13 种图的定义、图形符号表示方法、建模方法。

学习本篇后，读者能理解和掌握 UML 语言的基本元素、关系元素、13 种图元素及 UML 视图的语义和语法，并可以在软件开发过程中使用 UML 模型元素对软件进行建模。

第 1 章 UML 概述

UML(Unified Modeling Language, 统一建模语言)是一种通用的设计语言, 通过这种语言可以对任何具有静态结构和动态行为的系统进行建模。

1.1 什么是 UML

UML 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, 我们常采用该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, 然而从 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。在众多的建模语言中, 语言的创造者努力推崇自己的产品, 并在实践中不断完善。但是, OO 方法(Object-Oriented Method, 面向对象的方法)的用户并不了解不同建模语言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, 于是爆发了一场“方法大战”。20 世纪 90 年代, 一批新软件开发方法出现了, 其中最引人注目的是 Booch 1993、OMT-2 和 OOSE 等。

Grady 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, 他将之前面向 Ada 的工作扩展到面向整个对象设计领域。Booch 1993 较适用于系统的设计和构造。

James Rumbaugh 等人提出了面向对象的建模技术(OMT, 一种软件开发方法), 该方法采用了面向对象的概念, 并引入各种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。

Ivan Jacobson 于 1994 年提出了 OOSE 方法, 该方法最大的特点是面向用例(Use-Case), 并在用例的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。OOSE 较适用于商业工程和需求分析。

此外, 还有 Coad/Yourdon 方法, 即著名的 OOA/OOD, 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,极大地妨碍了用户之间的交流。因此,在客观上极有必要组织联合设计小组,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,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James 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作为其商业策略已日趋明显。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、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)及UML 1.1(1997年11月)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,而且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,但对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。

作为一种建模语言,UML是一种标准的图形符号,它的定义包括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语言的主要特点可以归结为以下4点。

- UML统一了Booch、OMT和OOSE等方法中的基本概念。
- 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。它的作用域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。

总之,UML 是一种先进、实用的标准建模语言,但其中某些概念尚待实践来验证,也就是说,UML 必然存在一个进化过程。

1.2 什么是模型

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。在日常生活中常见的模型有很多,它们都是对事物的简化表示,下面是生活中经常遇到的 4 种模型。

(1) 交通模型:道路交通图、道路交通标志等,图 1-1 所示的就是广州地铁的模型。

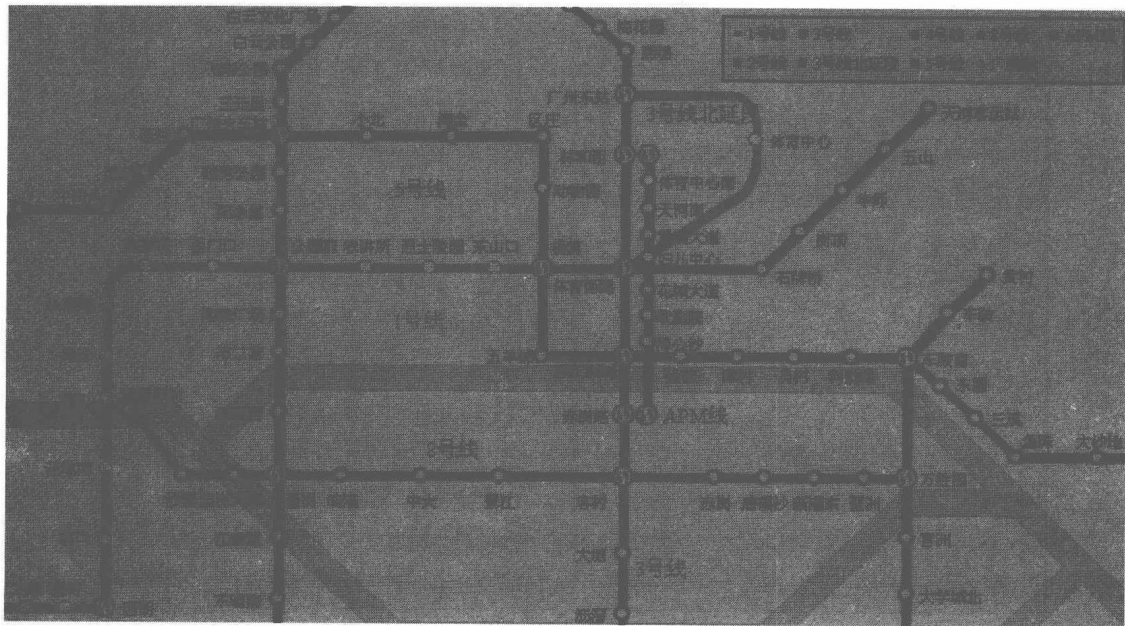


图 1-1 广州地铁模型

(2) 建筑模型:建筑物原型、沙盘等用来描述建筑物的图形,如图 1-2 所示的模型就是描述某集团公司的建筑物模型。



图 1-2 建筑物模型

(3) 设计模型：用来描述管线图、电路板设计图等的图形。图 1-3 所示的就是描述某个局部电路的设计模型。

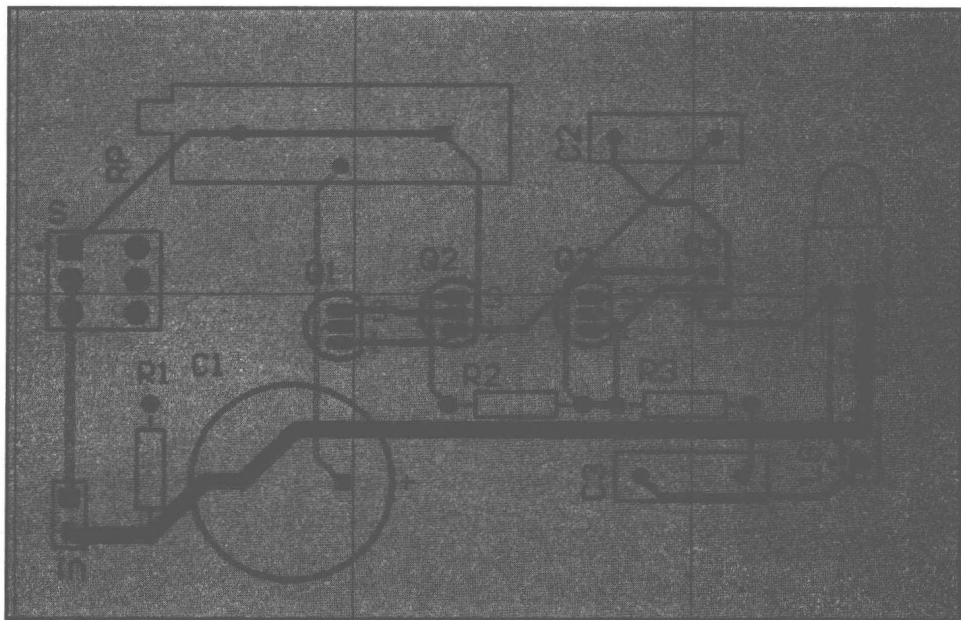


图 1-3 电路设计模型

(4) 数据分析模型：如图 1-4 所示就是描述某公司 4 种产品年销售所占份额的数据分析模型。

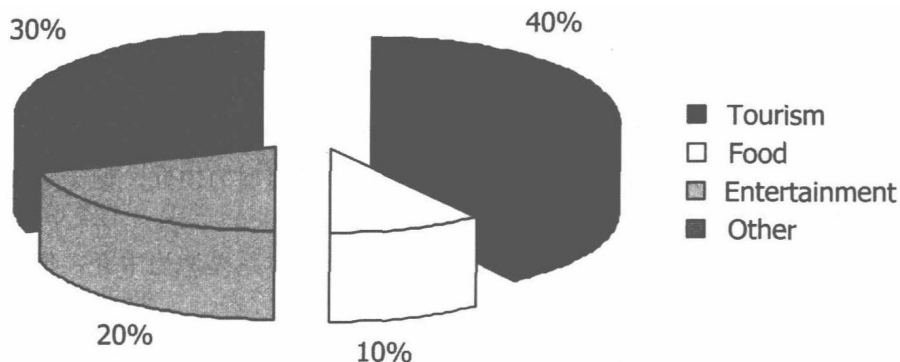


图 1-4 数据分析模型

上述 4 种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内容、业务规则、业务流程和 workflows 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,即解域)。

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,便于设计师修改和构造软件系统。

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。对业务进行建模,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地实现,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，便于领域专家、用户和需求分析师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，便于设计师对软件系统的宏观理解。
- (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的详细理解和沟通。
- (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。便于分析师和设计师沟通、理解。

1.2.3 建模原则

我们知道，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，设计师能方便地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 用 UML 建模的好处

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。使用 UML 建模的原因主要有以下 3 个。

1. UML 是软件开发者的通用语言

UML 为业务分析员、架构设计师、代码实现者、测试者、数据库设计师和许许多多参与软件设计和开发的人员提供了一种公共“语言”，通过 UML 语言，他们能够理解业务、需求、软件结构和组成。