

高等学校软件工程系列教材

UML软件建模教程

卫红春 主编



高等教育出版社
HIGHER EDUCATION PRESS

高等学校软件工程系列教材

UML 软件建模教程

UML Ruanjian Jianmo Jiaocheng

卫红春 主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是一本基于 UML 的软件建模技术教程。全书共分为三篇。第一篇软件建模技术概论(第 1 章),介绍软件建模技术的要素、软件模型的概念和内容以及软件建模技术的形成与发展。第二篇软件建模语言 UML(第 2~10 章),介绍 UML 的相关知识。第三篇建立软件模型(第 11~14 章),结合一个工程案例介绍软件的业务建模、需求建模、分析建模和设计建模。

作者一直从事软件建模技术课程的教学和工程研发工作,从本世纪初就开始跟踪 UML 的发展,并积累了一些基于 UML 的软件建模工程和教学经验,本书即作者十多年工作的总结,融入了作者对软件建模技术的独到见解。

本书可作为软件工程、计算机科学与技术及相关专业的本科生教材和研究生的参考教材,也可作为软件开发人员学习软件建模技术和从事工程开发的参考用书。

图书在版编目(CIP)数据

UML 软件建模教程 / 卫红春主编. -- 北京: 高等教育出版社, 2012.8

ISBN 978-7-04-035914-5

I. ①U… II. ①卫… 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(2012)第 181381 号

策划编辑 倪文慧
插图绘制 尹莉

责任编辑 倪文慧
责任校对 杨雪莲

封面设计 于文燕
责任印制 尤静

版式设计 杜微言

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 化学工业出版社印刷厂
开 本 850 mm × 1168 mm, 1/16
印 张 22 25
字 数 500 千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landracom.com>
<http://www.landracom.com>
版 次 2012 年 8 月第 1 版
印 次 2012 年 8 月第 1 次印刷
定 价 35.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物料号 35914-00

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010) 58581897 58582371 58581879

反盗版举报传真 (010) 82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

前 言

20 世纪 90 年代中期,软件建模开始受到软件工程领域的重视,统一建模语言 UML 的出现,推进了软件建模技术在软件工程领域的普及。21 世纪初,国际对象管理组织(OMG)提出了模型驱动架构(MDA),强调软件模型在软件开发过程中的中心作用,由软件模型来驱动整个软件开发工作。2011 年,我国把软件工程提升为与计算机科学与技术并列的一级学科,软件建模与分析也被列入软件工程教育知识体系。从工程应用看,进入 21 世纪之后,国内外各软件公司均把软件建模作为软件开发的重要工作,软件模型在规范软件开发工作、方便开发团队人员交流、提高软件开发效率、便于软件维护等方面发挥着越来越重要的作用。

随着软件模型在软件工程中作用的凸显,目前,在我国高校软件工程、计算机科学与技术等专业的教育中,加强软件建模技术基础和建模能力培养已成为一项重要工作,各高校相继开设了以 UML 为基础的软件建模技术课程,大部分高校的软件工程专业已将该课程作为专业核心课。为了满足软件建模技术课程的教学需要,作者编写了本教材。教材编写的基本思路主要包括以下几方面:

(1) 系统化软件建模技术课程知识领域。软件建模技术作为软件工程学科的重要内容,自身应具有完整和合理的知识结构。软件建模技术包括软件建模方法、软件建模过程、软件建模语言和软件建模工具等基本内容,并由这 4 部分内容形成了相对完整的软件建模技术知识体系,本书将这 4 方面的知识内容贯穿到教材的全过程中。

(2) 理论与实践并重。软件建模技术需要基础理论支撑,但它更是一种工程技能。本书在阐述软件建模理论的基础上,重视软件建模技术实践能力的教学和培养。前 10 章侧重于软件建模方法和软件建模语言的理论知识教学,但在介绍理论知识的过程中并没有忽视建模技能的培养。后 4 章重点讲述在软件开发的阶段建立软件模型的技能和方法,其中包含作者多年软件建模工程实践的体会,也是本书的特色所在。

(3) 重视案例教学。好的教学案例对学生理解课程内容有事半功倍的作用。本书选用作者指导学生开发的“e 速快餐系统”作为一个完整的教学案例贯穿教材主要内容,相信该案例对学生理解教学内容,掌握软件建模技术将会起到十分有益的作用。

(4) 反映软件建模技术的最新发展。软件建模技术以及 UML 一直在发展之中,本书采用 UML 2.0 作为教学内容,反映了软件建模的新发展。同时第一章介绍了在软件建模领域出现的模型驱动架构(MDA)、面向领域建模以及基于元模型的软件建模等最新技术动态,让学生对软件建模技术领域的发展动态有概要了解。

本书包括 3 篇。

第一篇软件建模技术概论(第 1 章),介绍软件建模技术领域的概念和基本内容,并概要

II 前言

介绍了软件建模技术所包括的软件建模方法、软件建模过程、软件建模语言和软件建模工具等 4 方面的基本内容。

第二篇软件建模语言 UML(第 2~10 章),完整地介绍了 UML 2.0 版本的全部内容,是学习软件建模的基础。

第三篇建立软件模型(11~14 章),详细介绍了软件的业务建模、需求建模、分析建模和设计建模的基本方法,并将“e 速快餐系统”案例穿插于其中,让学生通过案例来学习如何进行建模实践。

本书可作为软件工程专业、计算机科学与技术专业以及相关专业的本科生教材,也可作为软件开发人员学习软件建模技术、从事软件开发的参考书。使用本书进行课程教学建议学时为 46~54 学时,其中课堂教学 28~30 学时,建模工具和 UML 建模实践 12~14 学时,建模综合实验 6~10 学时。为便于教学,随书提供配套的 PPT 课件和用 Enterprise Architect 8.0 建立的“e 速快餐系统”软件模型。

本书由卫红春主编并编写了除第 7 章之外的全部内容,王学龙参与编写了第 7 章,原博、董鹏和聂勃三位同学承担了 e 速快餐系统案例的建模工作,原博同学为模型后期的完善和优化做了大量工作,并参与编写了第 14 章的部分案例。杨怀洲、李宥谋两位老师认真阅读了书稿,并提出了许多建设性意见,程国建老师对教材结构也提出了自己的意见。马丁同学在紧张地考博期间还认真阅读了全稿内容,修改了其中的一些错误。在此,对以上老师和同学表示衷心的感谢。

北京工业大学于学军副教授认真审阅了书稿并提出了宝贵的意见,高等教育出版社的有关人员对书稿做了大量的编辑工作,在此向她们一并致以诚挚的谢意。

由于作者水平有限,书中缺点和欠妥之处在所难免,恳请指正。作者的电子邮箱是:hchwei@xsyu.edu.cn。

卫红春

2012 年 6 月于西安

目 录

第一篇 软件建模技术概论

第 1 章 概论	1	1.3.1 软件建模技术的形成与发展	7
1.1 模型	1	1.3.2 软件建模技术的内容	10
1.2 软件模型	3	1.4 软件建模方法	11
1.2.1 软件模型的概念和特征	3	1.5 软件建模过程	14
1.2.2 软件模型在软件开发中的作用	4	1.6 软件建模语言	18
1.2.3 软件模型的内容	5	1.7 软件建模工具	18
1.3 软件建模技术	7	本章小结	19
		习题 1	20

第二篇 软件建模语言 UML

第 2 章 UML 概要介绍	23	3.2 参与者	44
2.1 概述	23	3.3 用例	45
2.2 UML 的构成	25	3.4 用例之间的关系	47
2.3 UML 基础	26	3.5 用例叙述	49
2.3.1 元模型	26	3.6 用例图的作用	51
2.3.2 UML 的核心概念	27	3.7 用例图实例	52
2.4 基本元素	31	3.7.1 图书馆图书借阅管理	52
2.5 图	34	3.7.2 旅游宾馆客房预订管理	54
2.5.1 静态结构图	34	本章小结	56
2.5.2 动态行为图	35	习题 3	57
2.6 语义规则	38	第 4 章 类图与对象图	60
2.7 公共机制	39	4.1 对象与实例	60
本章小结	41	4.2 类元与类	61
习题 2	41	4.3 接口	65
第 3 章 用例图	43	4.4 关联关系	68
3.1 概述	43	4.5 组成关系	73
		4.6 泛化关系	74

4.7 依赖关系	79	第 7 章 状态机图	127
4.8 类图	84	7.1 几个基本概念	127
4.8.1 类图的作用	84	7.2 状态	128
4.8.2 类图实例	85	7.3 状态转换	129
4.9 对象图	90	7.4 事件	129
本章小结	91	7.5 复合状态	131
习题 4	91	7.6 并发状态	133
第 5 章 包图与复合结构图	94	7.7 控制结点	134
5.1 包图	94	7.8 子机状态	136
5.1.1 概述	94	7.9 状态机	137
5.1.2 包的关系	96	7.10 状态机图实例	141
5.1.3 包的设计原则	99	本章小结	142
5.1.4 包图的作用	100	习题 7	143
5.2 复合结构图	101	第 8 章 交互图	145
5.3 协作	104	8.1 交互	145
本章小结	105	8.2 顺序图	148
习题 5	105	8.3 通信图	162
第 6 章 活动图	107	8.4 交互概览图	163
6.1 概述	107	8.5 时序图	170
6.2 活动	108	本章小结	174
6.3 活动结点	109	习题 8	174
6.3.1 动作结点	109	第 9 章 构件图	177
6.3.2 控制结点	110	9.1 构件	177
6.3.3 对象结点	113	9.2 构件的接口与端口	178
6.4 活动边	116	9.3 构件的结构与视图	179
6.5 活动分区	118	9.4 构件的关系	180
6.6 可中断活动区间和异常	118	9.5 构件连接器	182
6.7 活动图的作用	120	9.6 构件图	183
6.8 活动图实例	122	本章小结	183
本章小结	124	习题 9	184
习题 6	125		

第 10 章 部署图	185	10.3 结点	188
10.1 概述	185	10.4 部署	190
10.2 制品	186	本章小结	193
		习题 10	194
第三篇 建立软件模型			
第 11 章 业务建模	195	13.1 概述	251
11.1 概述	195	13.2 事务模式	252
11.2 业务愿景建模	197	13.2.1 事务模式的四要素	253
11.3 涉众建模	199	13.2.2 事务之间的关系	256
11.4 业务过程建模	201	13.3 PIM 类模型	259
11.4.1 业务过程模型	201	13.4 用例分析建模	264
11.4.2 业务过程建模	204	13.5 “e 速快餐系统”分析模型	268
11.5 业务对象建模	207	本章小结	283
11.6 业务规则建模	208	习题 13	284
11.7 “e 速快餐系统”业务模型	210	第 14 章 设计建模	285
本章小结	218	14.1 概述	285
习题 11	218	14.2 软件架构模型	286
第 12 章 需求建模	219	14.2.1 分层软件架构	286
12.1 概述	219	14.2.2 软件逻辑架构	288
12.2 用例建模	221	14.3 类与接口设计模型	289
12.3 用例建模的几个问题	233	14.4 用例设计模型	297
12.4 非功能需求建模	240	14.5 数据库设计模型	299
12.5 “e 速快餐系统”需求模型	241	14.6 界面模型	302
本章小结	250	14.7 构件模型	306
习题 12	250	14.8 部署模型	310
第 13 章 分析建模	251	14.9 “e 速快餐系统”设计模型	312
		本章小结	334
		习题 14	335
附录 英汉名词对照	336		
参考文献	346		

第一篇 软件建模技术概论

第1章 概 论

本章导读

本章从模型入手,引入软件模型的概念,并阐述了软件建模的必要性和软件模型的内容,然后重点介绍软件建模方法、软件建模过程、软件建模语言和软件建模工具等软件建模技术的四大要素。

主要知识点

- 模型
- 软件建模要素
- 软件建模过程
- 软件建模工具
- 软件模型
- 软件建模方法
- 软件建模语言

1.1 模 型

模型(model)是对现实系统本质特征的一种抽象、简化和直观的描述。现实系统的复杂性和内隐性,使得人们难于直接认识和把握,为了使人们能够直观和明了地认识和把握现实系统,就需要借助于模型。例如,地球仪是地形地貌的模型,人们通过地球仪这种模型可以非常直观地了解一个地区的地形和地貌。有些现实系统内部结构非常复杂,在生产和加工时,就需要通过模型来展现这些系统的结构和内部构成。例如,在生产汽车过程中,需要预先设计出反映汽车构造的设计图纸,然后生产车间再根据设计图纸组织生产,设计图纸就是所要生产的汽车的模型。

模型不必囊括现实系统的全部特征,但它要反映现实系统的本质特征。模型一定比现实系统简单,如果比现实系统还复杂,就失去了模型的意义。所谓好模型就是既能够反映现实系统的本质特征,又尽量地简化,且能以直观形式表现的模型。模型具有不同的抽象度,模型的抽象程度越高,其距现实系统的距离就越远,建模时所需考虑的因素也就越少。

1. 模型的特征

(1) 反映性

一般而言现实系统是复杂的、内隐的,人们无法采用常规方式和途径来全面、深入地认识现实系统,这就需要借助于模型来直观、全面地理解和把握现实系统。例如,利用一张地图就能够很清楚地让人们从整体上了解这一段区域的地形和地貌情况。

(2) 直观性

只有直观的形式才便于人们理解现实系统,模型会以直观的形式把现实系统的结构和内在要素的联系展现出来。

(3) 简化性

模型在反映现实系统时作了大量简化,所以模型不会比现实系统复杂,如果比现实系统还复杂,就失去了模型的意义。

(4) 抽象性

模型总要忽略现实系统中的一些次要细节,抓住现实系统的本质特征进行重点描述,如数学模型、逻辑模型都是以抽象的形式来描述现实系统。

2. 建模过程

按照确定的方法和规则建立模型的工作被称为建模(modeling)。模型对现实系统的反映不是简单地复现和照搬,而是对现实系统的抽象、概括和提炼。所以需要认真分析现实系统,并对现实系统进行抽象和概括,然后以科学和直观的形式创建模型。建模的过程是一个反复和逐步求精的过程。图 1.1 说明了建模的过程。

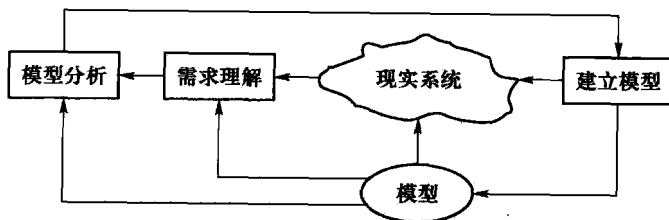


图 1.1 建模过程

建立模型之初,首先要深入理解建模的需求,理解所要建立模型的现实系统,并清楚要建立一个什么样的模型,接下来需要对建模工作进行深入分析,并在分析的基础上建立起所需要的模型。建模的过程是一个逐步深入、逐步求精的过程,对建立的模型需要再次进行分析修改,通过多次反复,最后建立起能够满足实际需要的模型来。

3. 模型分类

从不同的角度考虑,可将模型划分为多种类型。

从抽象程度上可以把模型分为概念模型、逻辑模型和物理模型。概念模型(concept model)是人们根据所要表达的目标和人们已有的知识经验构造出来的一种系统雏形。它是对所描述的

现实系统的主要特征的一种概括性描述。概念模型是最抽象的模型,一般只能表述现实系统的主要特征,是对现实系统的一种概要式的反映。逻辑模型(logical model)是在概念模型的基础上,从原理上证明是合理可行的系统,它考虑了系统的目标合理性、结构合理性、功能合理性和实现合理性。物理模型(physical model)是在逻辑模型的基础上充分考虑环境,并对细节都做了精心设计后所实现的实在模型。

从形式上可以把模型分为实体模型、结构模型、仿真模型和数学模型几种类型。实体模型(entity model)是现实系统的实物再现,其抽象程度最低。例如用来反映城市发展规划的沙盘,美术学院学生作画用的雕塑都属于实体模型。结构模型(structure model)用来反映系统要素之间的空间或逻辑关系,软件结构图、建筑结构图等都属于结构模型。仿真模型(simulation model)是利用计算机程序的图形图像显示或虚拟现实技术在计算机上模仿现实系统的外形、结构和内在特征的一类模型。数学模型(mathematics model)是用数字和符号来描述现实系统的各种要素及其数量关系的一种模型,它通常表现为定律、定理、公式、算法等。

1.2 软件模型

软件属于智能型系统,在软件中蕴藏着大量的信息、知识、方法和技术。软件无论是在开发过程中,还是在开发成功之后,都不具备其他物体的有形性。软件的无形性和包藏性,给软件开发带来了极大的困难,使得在整个软件开发过程中,人们对它难以把握和描述。为了工程化、有效地开发软件,人们除了寻求有效的开发方法,严密地组织开发过程之外,还需要在开发的各个阶段以某种有效的形式把软件描述和表现出来,这样开发人员才能够有针对性地进行交流和讨论。因此,软件开发必须建立模型。

1.2.1 软件模型的概念和特征

软件模型(software model)是指通过软件建模语言对软件的功能和性能等外特性,软件的要素和结构,以及软件的动态行为特性所给出的抽象和规范描述。按照确定的规则和方法建立软件模型的工作被称为软件建模(software modeling)。

较之于其他模型,软件模型具有以下特征。

1. 建模对象特殊

软件的建模对象是软件。软件是一种特殊的事物,它以一种可程式化和可动态复现的方式来反映人头脑中的可计算性意识,并通过这些意识的动态性外化,达到对客观以及人的主观意识的反映。软件与一般事物比较有其显著的特性。第一,软件是对意识的反映,而意识又是对客观的反映,因此软件承担着反映意识进而反映客观的任务,并且软件对客观的反映并不简单是对客观的模拟和复现,而是通过虚拟的形式,对客观一种创造性的反映。第二,软件没有一般客观事物的形态外显性,软件是无形的,它被承载在介质之中,这样使得软件生产过程难以把握。第三,软件的规模多样,大规模的软件需要工程团队经历很长时间方能研制成功。

2. 复杂性

因为软件是人的意识对客观存在的反映,因此软件具有复杂性。软件的复杂性决定了软件模型的复杂性。软件模型的复杂性体现在下面几个方面:第一,软件模型的反映对象难以把握。这是由于软件没有确定形态而决定的,软件自身没有形态,因此采用怎么样一种直观、简明和形象的方式来对其建模,就是一个很复杂的问题,可以说这个问题其实到现在仍然没有得到很好的解决。第二,软件要描述人类智能所能涉及的所有领域,在这其中要找出来一种能够描述所有领域问题的软件模型难度是很大的。

3. 多样性

软件模型并不是一种形式,而是多种形式。从问题本身看,软件要把所要表述的问题表达清楚,需要通过不同侧面来描述问题。例如,反映软件功能的被称为软件的功能模型,反映软件结构的被称为软件的结构模型,反映软件内部实体要素及其关系的被称为软件的对象模型,反映软件实体状态的被称为软件的状态模型等。软件的开发有多项工作,有业务分析、需求分析、系统分析、系统设计、实现和测试等,对应这些工作也应该建立对应的业务模型、需求模型、分析模型、设计模型和测试模型等。

1.2.2 软件模型在软件开发中的作用

软件模型在软件开发中的作用表现在以下几个方面。

1. 人员交流的媒介

在软件开发过程中会有大量的人员,包括用户、管理人员、项目主管、系统分析员、架构师、程序员、测试人员、网络和数据库专家、技术顾问等,软件模型为参与软件开发、维护、应用和管理人员的交流提供了一个统一的交流媒介。通过这个媒介,相关人员可以彼此沟通、理解需求、澄清问题。

2. 软件的中间形态

软件在开发过程中要经历分析、设计、实现和测试等阶段,从软件策划开始,到得到最终软件产品期间,软件模型将作为软件的中间形态来反映软件的生产过程。软件模型也是规范化的软件文档的主要内容。

作为软件的中间形态,软件模型要完整地反映软件所要服务的业务领域的领域知识、业务流程、业务实体等业务领域的各种知识和信息;软件模型要反映用户对所开发软件的目标、功能、性能等需求;软件模型要反映设计人员对软件的设计思想、技术方案和设计意图;软件模型还要反映软件实现的网络、硬件和支撑软件等实现环境。

3. 全面把握复杂系统、优化方案和保障软件质量

大部分软件属于大型复杂系统,人们难于理解和把握,分层的软件模型和软件模型所展现的不同视图可以帮助人们全面把握复杂系统;通过研究一个大型软件系统的模型可以提出多个设计方案进行比较分析,并给出优化的设计方案;软件模型也是保障软件质量的有效措施,通过对模型的正确性和完备性的审查,可以预先发现软件中潜在的问题,以保障软件质量。

4. 软件升级和维护的依据

软件在其生存周期中,将根据需求的变化不断地进行维护和升级,软件模型是软件维护和升级的重要依据。

1.2.3 软件模型的内容

软件的特点决定了软件的构成要素具有多样性,以及软件开发工作的时段性和复杂性。软件模型不是单一模型,而应该是由多个子模型构成的一体化模型。一个完整的软件模型应该能够反映软件的功能、对象、数据、过程、状态、交互、结构和界面等侧面,并包括业务模型、需求模型、分析模型、设计模型和测试模型等。

1. 表现不同侧面的软件模型

一个完整的软件模型应该展现软件的不同侧面,包含展现软件每一个侧面的模型。

(1) 功能模型

软件功能是软件所展现的各种效用,表现为软件为用户提供的各种服务。规范、清晰地描述软件的功能是软件模型必须具备的基本能力。功能模型是软件模型的基本内容。

(2) 非功能模型

软件除了功能之外,还包括效率、可靠性、安全性、可用性、适应性等约束系统、支持功能实现的非功能指标,软件模型除了能够表现软件功能之外,也需要表现软件的非功能指标。

(3) 对象模型

软件需要反映应用需求中大量的客观实体,并把客观实体作为对象在软件空间中表现出来。在面向对象方法中,对象是软件的基本要素,因此,对象模型是软件模型的核心内容。

(4) 数据模型

数据是对客观事物的符号表示,数据模型是传统软件模型的重要内容。在面向对象方法中,对象涵盖了数据以及对数据的加工处理,因此数据的概念相对被弱化了,可以用对象建模来部分代替数据建模。但数据建模仍不能被完全取代,其原因是,虽然在对象级上可以用对象来代替传统意义上的数据,但在对象内部的属性和操作中,仍不能取消常量、变量等基本数据;另外,目前对象数据库技术还不成熟,在大部分软件中仍然采用关系数据库,关系数据库存在概念模型、逻辑模型和物理模型等数据模型,因此数据建模还不能被对象建模完全取代。

(5) 过程模型

软件参与业务活动的业务流程、软件程序的执行流程、软件程序中的算法流程等均表现为活动过程,要表现这些动态活动过程,就需要进行过程建模。

(6) 状态模型

事物在其生存周期中总会处在不同的状态之中,并因为某种激发条件而引起状态的变换。要在软件空间中完整地表现业务需求,就应该能够反映客观事物所处的状态以及状态的变换。

(7) 交互模型

软件有其确定的处理逻辑。一个软件功能总是由若干个相关的对象通过动态操作交互来完

成。交互建模用于描述软件各功能的交互处理逻辑,是软件建模的重要内容。

(8) 架构模型

软件属复杂系统,需要对软件架构进行建模。近年来,软件界已经开始重视软件架构的研究,但人们对软件架构的认识还未取得共识,软件架构目前还没有取得一致性的研究成果。但是,软件架构建模的作用是毋庸置疑的。

(9) 界面模型

软件通过界面向用户展现其功能和特性,用户也是通过界面来感知和使用软件的。界面设计是软件开发的一项重要工作,需要对软件界面进行建模。

2. 表现不同开发工作的软件模型

软件开发涉及多项开发工作,包括业务分析、需求分析、系统分析、系统设计、软件实现和测试等。软件的每一项开发工作都需要建立对应的软件模型。因此,从软件开发角度看,应该建立与这些工作对应的如下软件模型。

(1) 业务模型

业务模型(business model)也被称为业务领域模型,是描述软件服务的业务领域的业务背景、业务资源、业务过程和业务规则等的一种抽象模型,业务模型是对业务的抽象和简化描述。因为软件可以服务所有业务领域,各业务领域的业务知识和业务活动差异很大,因此要寻求一种统一的建模规则和建模语言十分困难,目前虽然出现了UML业务建模、扩展的UML业务建模、BPMN(Business Process Modeling Notation)业务建模等符号规则,但似乎还没有达到完全实用的程度。

(2) 需求模型

需求模型(requirement model)是通过建模语言描述软件需求的模型。在需求模型中,需要对需求结构、软件功能、软件性能等内容进行建模。

(3) 分析模型

分析模型(analysis model)是在系统分析工作过程中建立的软件模型,分析模型包括PIM类模型和用例分析模型等内容。业务模型和需求模型是建立分析模型的基础,分析模型又是建立软件设计模型的依据。

(4) 设计模型

设计模型(design model)是在设计阶段建立的软件模型,系统设计的每一方面的工作都应该建立对应模型,设计模型包括软件架构模型、类及接口设计模型、用例设计模型、数据库设计模型、界面设计模型、构件模型和部署模型等。

(5) 测试模型

测试模型(test model)描述软件的规范化测试设计,因为测试过程独立于软件开发过程,测试模型主要体现在测试策划和测试设计等方面,并把测试设计应用于测试过程。本书将不介绍测试建模方法。

1.3 软件建模技术

1.3.1 软件建模技术的形成与发展

1. 20世纪90年代中期之前的软件建模技术

要有效地开发软件,就需要对软件进行建模,因此,在计算机诞生之后,软件建模就成为软件开发工作面临的实际问题。但是在20世纪50年代末之前,并没有提出软件概念,软件的开发方法以及软件的工程化也没有受到应有的重视。该时期出现的程序流程图实际上就属于软件模型的内容。

20世纪60年代到70年代,因软件危机的出现而引起了人们对软件工程的重视。在这个时期,虽然提出了软件工程的概念,但人们对软件工程学科内涵的理解还处于初期阶段。软件工程学科包含哪些基本内容,软件开发应该重视哪些方面的问题,在计算机界并没有形成共识。当时,人们把软件开发成败的原因更多地归于软件开发方法。因此,以软件开发方法为核心带动软件工程学科发展成为这个时期软件工程学科的基本特征。该时期出现了很多具有代表性并对以后软件工程产生了重大影响的软件开发方法。

软件模型与软件开发方法的关系就像音乐中的乐谱与作曲方法一样,二者密不可分。软件开发方法必然涉及软件开发的过程和步骤,在各个开发阶段对软件的中间形态如何进行描述,这些问题既属于软件开发方法的问题,又属于软件建模的问题。在这个时期出现的结构化方法、JSD方法(Jackson System Development method, Jackson系统开发方法)等软件开发方法中,已经大量采用符号、图形、规则等描述规范和方法,这些就是软件建模的内容。如在结构化方法中,用来描述软件需求的数据流图、描述软件结构的模块图、描述软件模块关系的IPO(Input Process Output, 输入处理输出)图、描述软件流程的程序框图等都应该属于软件建模的内容。该时期因软件开发方法的出现和广泛应用,虽然已经出现了大量软件建模的符号、规则和方法,但学术界还仅认为这是软件开发方法的内容,并没有从软件建模的角度对其进行认识。因此,软件建模作为软件工程的一个重要的学科内容,在这个时期并没有受到应有的重视。

20世纪80年代到90年代初是软件工程学科发展的关键时期。软件开发相关技术得到迅速发展,软件应用向深度和广度发展,软件的规模和范围不断增大,并由独立系统向开放、互联、集成的一体化系统发展;软件产业化速度加快,软件体系结构开始受到重视,软件开发环境逐步成熟。但这个时期,学术界仍然把软件建模混淆在软件开发方法之中,没有对软件建模的语言、技术和方法给予应有的重视。

2. 20世纪90年代中期之后的软件建模技术

(1) UML对软件建模技术的推进

20世纪90年代中期之后,软件建模开始受到重视,软件建模技术也从软件开发方法中被分离出来,作为软件工程学科的重要内容。1998年出版的《计算机科学技术百科全书》中,对软件

工程的研究内容是这样定义的：“软件工程的研究内容主要包括：软件工作模型，软件开发方法，软件过程，软件工具，软件开发环境，计算机辅助软件工程以及软件经济学等。”

在 UML 提出之前，软件开发方法和软件建模语言在全世界没有得到统一，到 1994 年属于面向对象的软件开发方法就达 50 余种。在当时众多的软件开发方法中，比较引人注目的是 Booch 1993、OMT-2 和 OOSE 等。Booch 1993 是由 G.Booch 提出的。G.Booch 是面向对象方法最早的提出人之一，由他提出了面向对象软件工程的观念。1991 年，他将以前面向 Ada 的工作扩展到整个面向对象设计领域，Booch 1993 方法的特点是适合于系统的设计和构造。J.Rumbaugh 等人提出了面向对象建模技术 (OMT)，OMT 方法是 20 世纪 90 年代初在国际具有广泛影响的面向对象方法，并引起了国内软件工程领域的高度重视。这种方法把对象模型、功能模型、动态模型、数据模型贯穿到软件开发的全过程，软件开发人员不必在不同开发阶段进行概念和符号的转换。I.Jacobson 于 1994 年提出了面向对象的软件工程 (OOSE) 方法，该方法的最大的特点是采用用例 (use case) 驱动的开发理念。并在用例的描述中引入了外部角色的概念。用例概念的引入为需求的精确描述找到一种简单易行的方法，并由用例贯穿于整个开发过程。包括对系统的测试和验证。OOSE 比较适合支持商业工程和需求分析。此外，还有 Coad/Yourdon 方法，即著名的面向对象分析与设计 (OOA/OOD)。该方法简单、易学，适合于初学者使用，但由于该方法在处理能力方面的局限，没有得到工程领域的广泛使用。

软件开发方法和建模语言的不统一给软件开发、管理和维护带来了很多问题，采用面向对象方法的开发人员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”。这种现状严重制约了软件规模化开发和软件工程学科的发展。在当时的情况下，迫切需要联合多方面的力量，在众多方法基础上，取精去粗、整合优化，提出统一的软件开发方法。1994 年 10 月，G.Booch 和 J.Rumbaugh 两人加入 Rational 公司，开始致力于这项浩大的工程。他们首先将 Booch93 和 OMT-2 统一起来，并于 1995 年 10 月发布了第一个称之为统一方法 Unified Method 0.8 的公开版本。

Unified Method 0.8 是一个不成熟的软件开发方法，该方法也没有引起软件界的广泛关注。在当时的条件下，提出一种能够被国际广泛接受的统一的软件开发方法的条件并不成熟，因此，研究人员把研究重点转变到提出一种独立于软件开发方法和软件开发过程的统一的软件建模语言上。1995 年秋，I.Jacobson 也加入 Rational 公司参与这一工作。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 UML 0.9 和 UML 0.91 两个版本，并正式命名为 UML (Unified Modeling Language)。两个版本发布之后，得到了来自社会的积极反应，并成立了 UML 成员协会以加强和促进 UML 的制定工作。当时的成员有 DEC、HP、I-Logix、Itellicorp、IBM、ICON Computing、MCI Systemhouse、Microsoft、Oracle、Rational Software、TI 以及 Unisys 等。这一机构对 UML 1.0 和 UML 1.1 的制定和发布起了重要的促进作用。到 1996 年 10 月，UML 已经获得了工业界、科技界和应用界的广泛支持，相继有 700 多个公司表示采用 UML 作为建模语言。1997 年 11 月，对象管理组织 (OMG) 采纳 UML 1.1 作为基于面向对象技术的标准建模语言。