

基于面向对象的 软件分析与设计实验教程

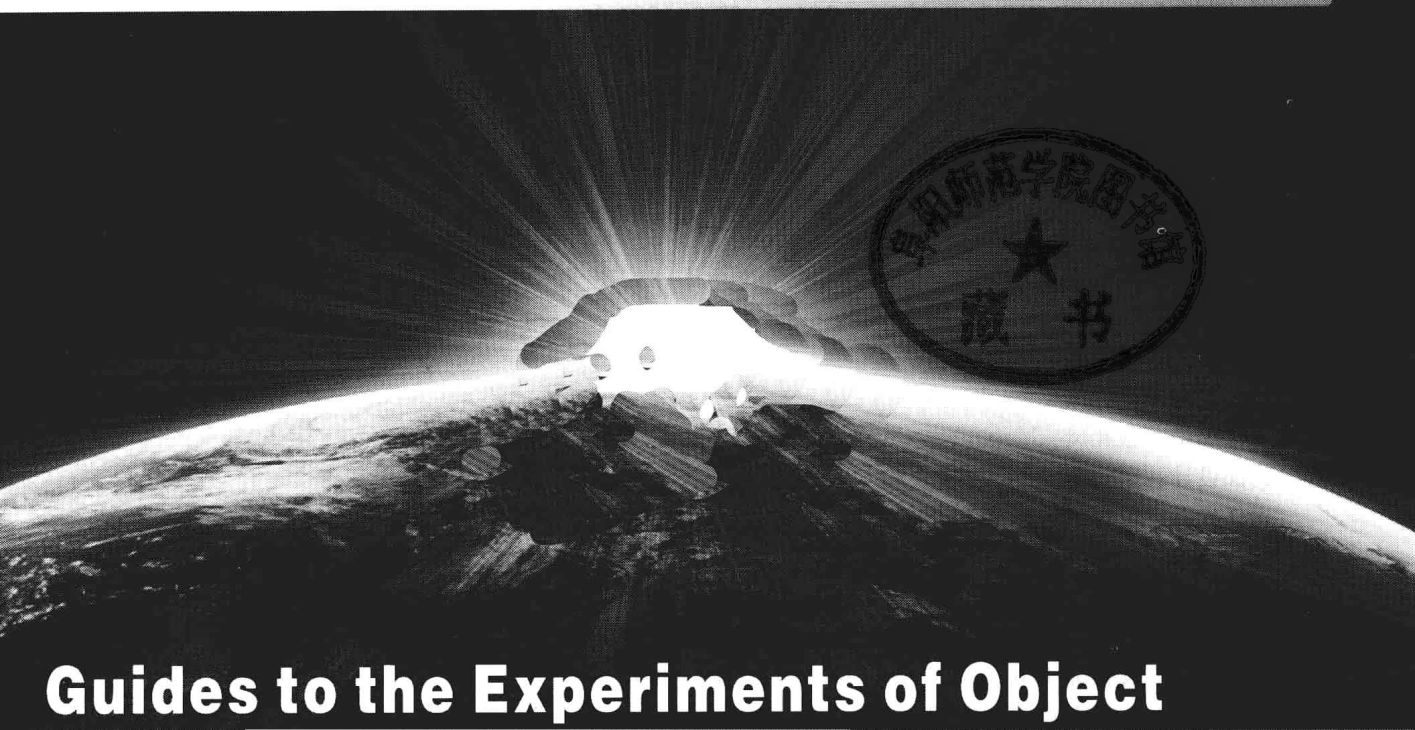
衣 杨 肖志娇 顾春琴 编著

Guides to the Experiments of Object

中山大学出版社

基于面向对象的 软件分析与设计实验教程

衣 杨 肖志娇 顾春琴 编著



Guides to the Experiments of Object

Analysis & Design

中山大学出版社

·广州·

版权所有 翻印必究

图书在版编目 (CIP) 数据

基于面向对象的软件分析与设计实验教程/衣杨, 肖志娇, 顾春琴编著. —广州: 中山大学出版社, 2011. 3

ISBN 978 - 7 - 306 - 03809 - 8

I. 基… II. ①衣… ②肖… ③顾… III. 面向对象语言—软件开发—教材
IV. TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 241386 号

出 版 人: 祁 军

策划编辑: 张礼凤

责任编辑: 张礼凤

封面设计: 贾 萌

责任校对: 曾育林

责任技编: 何雅涛

出版发行: 中山大学出版社

电 话: 编辑部 020 - 84111996, 84113349, 84111997, 84110779

发行部 020 - 84111998, 84111981, 84111160

地 址: 广州市新港西路 135 号

邮 编: 510275 传 真: 020 - 84036565

网 址: <http://www.zsup.com.cn> E-mail: zdcbs@mail.sysu.edu.cn

印 刷 者: 广州中大印刷有限公司

规 格: 787mm × 960mm 1/16 19.5 印张 475 千字

版次印次: 2011 年 3 月第 1 版 2011 年 3 月第 1 次印刷

印 数: 1 ~ 3000 册

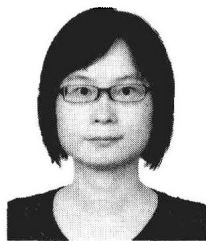
定 价: 39.80 元

如发现本书因印装质量影响阅读, 请与出版社发行部联系调换

作者简介



衣杨 博士，现为中山大学信息科学与技术学院计算机科学系副教授，硕士生导师。1989年7月毕业于复旦大学电子工程系，获得理学学士学位；分别于1996年3月和2002年3月获得东北大学系统工程专业硕士学位和博士学位。2002年3月至2004年4月，任中山大学计算机科学系讲师；2004年5月至今，任中山大学计算机科学系副教授。2005年10月至2006年11月，在美国EWU计算机科学系做访问学者。担任2006—2010年教育部高等学校计算机科学与技术专业教学指导分委员会专家组成员、中国计算机学会高级会员。发表学术论文60多篇，其中SCI收录论文10篇；撰写学术专著2部。曾获得广州市人民政府颁发的2005年“广州市科技进步奖”二等奖（排名第四），广东省教育厅颁发的2009年广东省教学成果一等奖（排名第三），申请并获得一项实用新型专利，拥有一项软件著作权，2008年获得教育部IBM大学合作项目优秀教师奖。承担本科生软件工程、系统分析与设计以及研究生数据建模与优化、最优化理论与计算方法、系统可视化分析与设计等课程的教学。



肖志娇 湖南人，1980年生，博士，讲师，中国计算机学会会员。2007年获中山大学计算机软件与理论博士学位。2007年7月至今在深圳大学计算机与软件学院任教。2009年1月，到英国进行短期的学习和访问。主要从事工作流、Petri网、形式化建模与智能优化等方面的研究。近几年来在国内外有影响力的学术刊物和国际会议上发表论文10多篇，其中以第一作者被SCI、EI、ISTP收录论文多篇。



顾春琴 江苏南通人，获中山大学硕士、博士学位。现任仲恺农业工程学院计算机科学系讲师，主要从事计算机协同技术、智能计算等方面的研究。主讲数据结构、软件工程、面向对象程序设计等专业课程。在国内外各类期刊发表科研论文20多篇，其中被SCI、EI等收录10多篇。

前 言

《基于 UML 可视化设计实验教程》自出版以来（2006 年初版、2007 年修订），得到了各高校教师以及广大学生的好评和支持，他们对本书提出了宝贵的修改意见与建议，在此，笔者对多年来关心、支持并对本教材提出修改意见和建议的师生表示衷心感谢。

本书是在《基于 UML 可视化设计实验教程》的基础上修订而成。此次修订，增加了对“软件分析与设计一般性过程”的分析、归纳，并提出了“基于面向对象的软件分析与设计文档规约”，充实了若干难易程度不同、应用领域不同的新颖案例，而且本教材的分析与设计文档更加严谨和规范，从而使本教材的层次更加清晰、通俗易懂，既便于教师教学又便于学生学习。

本教材分为两个部分。第一部分的核心任务是加强学生对面向对象方法学以及软件分析与设计过程方法学的理解，使学生对相关的基础理论、基本知识、主流技术方法有一个系统、完整、深刻、具体的认识。在第二部分，根据当前软件系统应用情况，选取了电子商务、网上书店、小型人力资源计划系统（Enterprise Resource Planning, ERP）、小型信息管理系统（Management Information Systems, MIS）、在线考试系统、考勤系统、个人理财系统等 11 个功能比较完善的案例，案例应用领域背景丰富，具有相对多个层次的技术复杂度，可适应不同层次的教学需求。

为了方便师生使用本教材，笔者精心准备了与本教材中案例匹配的辅助材料，主要是案例的实现代码，提供给中山大学出版社，有需要的读者可自行索取。

鉴于本人水平有限，加之软件工程、系统分析与设计的理论、技术和方法发展日新月异，书中难免存在错误和疏漏之处，敬请读者指正。笔者电子邮件地址：issyy@mail.sysu.edu.cn。

衣 杨
2010 年 10 月

目 录

上编 “基于面向对象的软件分析与设计可视化建模” 规约

第 1 章 基于面向对象的软件分析与设计一般性步骤	(1)
1.1 基于面向对象的软件分析与设计步骤	(1)
1.1.1 问题陈述 (Problem Statement)	(1)
1.1.2 需求分析 (Requirement Analysis)	(1)
1.1.3 架构分析 (Architectural Analysis)	(1)
1.1.4 用例分析 (UseCase Analysis)	(2)
1.1.5 设计元素的设计 (Design Element)	(2)
1.1.6 系统运行时刻架构设计 (Run Time Architectural Design)	(2)
1.1.7 分布式系统的设计 (Describe Distribution)	(2)
1.1.8 用例设计 (UseCase Design)	(2)
1.1.9 类的设计 (Class Design)	(2)
1.1.10 数据库的设计 (Data Base Design)	(3)
1.2 基于面向对象软件分析与设计的若干规则	(3)
1.3 基于面向对象软件分析与设计规约若干关注点	(4)
第 2 章 基于面向对象的软件分析与设计文档规约	(5)
第 3 章 术语	(7)

下编 典型案例

案例 1 “网上书店系统” (WebLab) 分析与设计	(9)
1.1 WebLab 问题陈述	(10)
1.1.1 WebLab 系统涉众	(10)
1.1.2 WebLab 功能概述	(10)
1.2 WebLab 需求分析	(10)
1.2.1 WebLab 用例析取	(10)
1.2.2 WebLab 用例规约	(11)
1.2.3 WebLab 补充规约	(19)

1.2.4	WebLab 术语	(20)
1.3	WebLab 架构分析与设计	(20)
1.3.1	WebLab 架构描述	(20)
1.3.2	WebLab 架构设计	(20)
1.3.3	WebLab 关键抽象	(21)
1.4	WebLab 用例分析	(21)
1.4.1	WebLab 分析类的析取	(21)
1.4.2	WebLab 分析类机制	(22)
1.5	WebLab 子系统设计及子系统中类的设计	(22)
1.5.1	WebLab 用户管理子系统	(23)
1.5.2	WebLab 图书管理子系统	(24)
1.5.3	WebLab 订单管理子系统	(24)
1.5.4	WebLab 购物篮管理子系统	(25)
1.5.5	WebLab 分析类到设计元素的映射	(26)
1.6	WebLab 实现技术	(26)
案例 2	“进口设备跟踪管理系统” (EI _{mp} tMIS) 分析与设计	(27)
2.1	EI _{mp} tMIS 问题陈述	(28)
2.1.1	EI _{mp} tMIS 系统涉众	(28)
2.1.2	EI _{mp} tMIS 系统功能	(28)
2.1.3	系统基本流程	(28)
2.2	EI _{mp} tMIS 需求分析	(31)
2.2.1	EI _{mp} tMIS 用例析取	(31)
2.2.2	EI _{mp} tMIS 用例规约	(31)
2.2.3	EI _{mp} tMIS 补充规约	(41)
2.2.4	EI _{mp} tMIS 术语	(41)
2.3	EI _{mp} t 架构分析	(42)
2.3.1	EI _{mp} tMIS 架构描述	(42)
2.3.2	EI _{mp} tMIS 架构设计	(43)
2.3.3	EI _{mp} tMIS 关键抽象	(44)
2.4	EI _{mp} tMIS 用例分析	(44)
2.4.1	EI _{mp} tMIS 系统实体类 (数据类) 的设计	(44)
2.4.2	EI _{mp} tMIS 用例类的设计	(45)
2.4.3	EI _{mp} tMIS 分析类机制	(49)
2.5	EI _{mp} tMIS 运行时架构设计	(49)
2.6	EI _{mp} tMIS 数据设计	(50)
2.6.1	EI _{mp} tMIS 数据库的设计	(50)
2.6.2	EI _{mp} tMIS 数据字典	(50)

2.6.3 ElmptMIS 数据表描述	(51)
2.7 ElmptMIS 实现技术	(57)
附录	(57)
案例3 “MircoTalk 微博网站” (mTalk) 分析与设计	(68)
3.1 mTalk 问题陈述	(69)
3.1.1 微博发展的背景	(69)
3.1.2 mTalk 网站拟实现的功能	(69)
3.1.3 mTalk 系统涉众	(70)
3.1.4 mTalk 功能概述	(70)
3.2 mTalk 需求分析	(70)
3.2.1 mTalk 软件概要结构	(70)
3.2.2 mTalk 用例析取	(71)
3.2.3 mTalk 用例规约	(73)
3.2.4 mTalk 补充规约	(80)
3.3 mTalk 总体设计	(82)
3.3.1 mTalk 功能总体设计	(82)
3.3.2 mTalk 界面设计	(83)
3.4 mTalk 系统设计	(84)
3.4.1 mTalk 命名规则	(84)
3.4.2 mTalk 技术准备	(85)
3.4.3 mTalk 数据库设计	(85)
3.4.4 mTalkVOPC (参与类) 的析取	(86)
3.5 类的设计	(88)
3.5.1 mTalk 消息发布模块类的设计	(88)
3.5.2 mTalk 用户登录模块类的设计	(89)
3.5.3 mTalk 用户管理模块类的设计	(91)
3.5.4 mTalk 类的设计	(93)
3.6 mTalk 系统部署	(94)
3.6.1 mTalk 运行环境	(94)
3.6.2 mTalk 系统安装说明	(94)
3.6.3 mTalk 支持软件配置	(108)
3.6.4 mTalk 发布网站	(114)
3.6.5 mTalk 网站使用说明	(115)
案例4 “网吧行业在线培训系统” (Cyber-online Training) 分析与设计	(116)
4.1 Cyber-online Training 问题陈述	(117)
4.1.1 Cyber-online Training 待解决问题	(117)

4.1.2	Cyber-online Training 系统涉众	(117)
4.1.3	Cyber-online Training 功能概述	(118)
4.2	Cyber-online Training 需求分析	(118)
4.2.1	Cyber-online Training 用例析取	(118)
4.2.2	Cyber-online Training 用例规约	(121)
4.2.3	Cyber-online Training 术语	(130)
4.3	Cyber-online Training 系统设计	(131)
4.3.1	Cyber-online Training 系统架构设计	(131)
4.3.2	Cyber-online Training 系统分析类的设计	(132)
4.3.3	Cyber-online Training 系统类的操作及关联设计	(133)
4.4	Cyber-online Training 系统部署	(150)
案例 5	“考勤管理系统” (RollBookMIS) 分析与设计	(152)
5.1	RollBookMIS 问题陈述	(153)
5.1.1	RollBookMIS 系统涉众	(153)
5.1.2	RollBookMIS 功能概述	(153)
5.2	RollBookMIS 需求分析	(153)
5.2.1	RollBookMIS 用例析取	(153)
5.2.2	RollBookMIS 用例规约	(154)
5.2.3	RollBookMIS 补充规约	(160)
5.2.4	RollBookMIS 术语	(160)
5.3	RollBookMIS 架构分析与设计	(161)
5.4	RollBookMIS 分析类的设计	(162)
5.4.1	RollBookMIS 关键抽象	(162)
5.4.2	RollBookMIS 实体类的设计	(162)
5.4.3	RollBookMIS 用例分析	(163)
5.5	RollBookMIS 分析类机制的设计	(167)
5.6	RollBookMIS 类的设计	(167)
5.7	RollBookMIS 子系统的划分与设计	(167)
5.7.1	RollBookMIS 子系统的划分	(167)
5.7.2	RollBookMIS 子系统的设计	(169)
5.8	RollBookMIS 分析类到设计元素的映射	(170)
案例 6	“航空订票系统” (aTickets) 分析与设计	(171)
6.1	aTickets 问题陈述	(172)
6.1.1	aTickets 系统涉众	(172)
6.1.2	aTickets 功能概述	(172)
6.2	aTickets 需求分析	(172)

6.2.1	aTickets 用例析取	(172)
6.2.2	aTickets 用例规约	(173)
6.2.3	aTickets 补充规约	(181)
6.2.4	aTickets 术语	(182)
6.3	aTickets 架构分析与设计	(182)
6.3.1	aTickets 架构描述	(182)
6.3.2	aTickets 架构设计	(183)
6.3.3	aTickets 关键抽象	(183)
6.4	aTickets 系统分析类的设计	(185)
6.4.1	分析类及其功能	(185)
6.4.2	预订机票用例分析	(186)
6.5	aTickets 系统类的设计	(187)
案例 7	“ERP 模拟游戏系统” (ERPGame) 分析与设计	(189)
7.1	ERPGame 问题陈述	(190)
7.1.1	ERPGame 背景	(190)
7.1.2	ERPGame 设计目标	(190)
7.1.3	ERPGame 系统涉众	(191)
7.1.4	ERPGame 功能概述	(191)
7.2	ERPGame 需求分析	(194)
7.2.1	ERPGame 用例析取	(194)
7.2.2	ERPGame 用例规约	(195)
7.2.3	ERPGame 补充规约	(208)
7.2.4	ERPGame 术语	(209)
7.3	ERPGame 架构分析与设计	(210)
7.4	ERPGame 关键抽象	(211)
7.5	ERPGame 分析类的设计	(211)
7.6	ERPGame 类的设计	(213)
附录	ERP 模拟游戏简介	(215)
案例 8	“在线考试系统” (Online Testing) 分析与设计	(216)
8.1	Online Testing 问题陈述	(217)
8.1.1	Online Testing 背景	(217)
8.1.2	Online Testing 系统涉众	(217)
8.1.3	Online Testing 功能概述	(217)
8.2	Online Testing 需求分析	(217)
8.2.1	Online Testing 用例析取	(217)
8.2.2	Online Testing 用例规约	(218)

8.2.3	Online Testing 补充规约	(227)
8.2.4	Online Testing 术语	(228)
8.3	Online Testing 系统架构设计	(228)
8.3.1	Online Testing 架构分析与设计	(228)
8.3.2	Online Testing 关键抽象	(230)
8.4	Online Testing 用例分析	(230)
8.4.1	Online Testing 定义分析类	(230)
8.4.2	Online Testing 分析机制的设计	(232)
8.4.3	Online Testing 系统类的设计	(232)
8.5	Online Testing 子系统设计	(233)
8.5.1	Online Testing 子系统的划分	(233)
8.5.2	Online Testing 子系统设计	(233)
案例9	“校园跳蚤市场系统”(School FLMKT) 分析与设计	(235)
9.1	School FLMKT 问题陈述	(236)
9.1.1	School FLMKT 待解决问题	(236)
9.1.2	School FLMKT 系统涉众	(236)
9.1.3	School FLMKT 功能概述	(236)
9.2	School FLMKT 需求分析	(237)
9.2.1	School FLMKT 用例析取	(237)
9.2.2	School FLMKT 用例规约	(238)
9.2.3	School FLMKT 补充规约	(241)
9.2.4	School FLMKT 术语	(241)
9.3	School FLMKT 架构设计	(242)
9.3.1	School FLMKT 系统框架	(242)
9.3.2	School FLMKT 系统关键抽象	(242)
9.4	School FLMKT 用例分析	(243)
9.4.1	School FLMKT 系统分析类的设计	(243)
9.4.2	School FLMKT 分析类及其功能	(243)
9.4.3	School FLMKT 分析类的分析机制	(245)
9.5	School FLMKT 用例实现	(246)
9.5.1	Buyer Activities 用例的类的设计	(246)
9.5.2	Salemen Management 用例的类的设计	(246)
9.5.3	Admin Management 用例的类的设计	(247)
9.6	School FLMKT 子系统分析与设计	(248)
9.6.1	School FLMKT 子系统的划分	(249)
9.6.2	School FLMKT 子系统的设计	(249)
9.7	School FLMKT 数据库的设计	(250)

案例 10 “快餐店管理系统” (FstFdMIS) 分析与设计	(252)
10.1 FstFdMIS 问题陈述	(253)
10.1.1 FstFdMIS 背景	(253)
10.1.2 FstFdMIS 系统涉众	(253)
10.1.3 FstFdMIS 功能概述	(253)
10.2 FstFdMIS 需求分析	(253)
10.2.1 FstFdMIS 用例析取	(253)
10.2.2 FstFdMIS 用例规约	(254)
10.2.3 FstFdMIS 补充规约	(266)
10.2.4 FstFdMIS 术语	(266)
10.3 FstFdMIS 架构设计	(266)
10.3.1 FstFdMIS 系统架构	(266)
10.3.2 FstFdMIS 系统关键抽象	(267)
10.4 FstFdMIS 用例分析	(268)
10.4.1 FstFdMIS 分析类及其功能	(268)
10.4.2 FstFdMIS 分析类机制的设计	(270)
10.4.3 FstFdMIS 系统类的设计	(270)
10.4.4 FstFdMIS 分析类到设计元素的映射	(272)
10.5 FstFdMIS 子系统及其接口设计	(272)
10.5.1 FstFdMIS 子系统设计	(272)
10.5.2 FstFdMIS 定义设计元素及其包	(274)
案例 11 “个人事务管理系统” (MyKit) 分析与设计	(275)
11.1 MyKit 问题陈述	(276)
11.1.1 MyKit 背景	(276)
11.1.2 MyKit 目标	(276)
11.1.3 MyKit 系统涉众	(276)
11.1.4 MyKit 系统功能概述	(276)
11.2 MyKit 需求分析	(277)
11.2.1 MyKit 用例析取	(277)
11.2.2 MyKit 用例规约	(278)
11.2.3 MyKit 补充规约说明	(281)
11.2.4 MyKit 补充规约	(281)
11.2.5 MyKit 术语	(282)
11.3 MyKit 架构分析与设计	(282)
11.3.1 MyKit 关键抽象	(282)
11.3.2 MyKit 架构描述	(283)

11.3.3	MyKit 架构设计	(283)
11.4	MyKit 用例分析	(284)
11.4.1	MyKit 的用例顺序图分析	(284)
11.4.2	MyKit 的用例类图分析	(284)
11.4.3	MyKit 分析类机制	(286)
11.5	MyKit 子系统设计	(287)
11.5.1	MyKit 的子系统及接口设计	(287)
11.5.2	MyKit 的子系统主要流程	(287)
11.5.3	MyKit 的子系统内部设计	(288)
11.6	MyKit 运行时架构设计	(291)
11.6.1	MyKit 运行时进程分析	(291)
11.6.2	MyKit 日记管理模块运行时架构设计	(292)
11.6.3	MyKit 日程管理模块运行时架构设计	(293)
11.6.4	MyKit 通讯录管理模块运行时架构设计	(293)
11.7	MyKit 数据库设计	(294)
11.7.1	MyKit ER 图的设计	(294)
11.7.2	MyKit 数据字典	(295)
11.7.3	数据表结构	(295)

上编 “基于面向对象的软件分析与设计可视化建模” 规约

软件系统的分析与设计过程比较复杂，是项目成功至关重要的一个环节。其过程是一个迭代、逐步细化、完善的过程。系统分析与设计的全过程，通常至少经过3次迭代。

本编结合作者的学习和工作积累，综合了诸如 IBM、Microsoft 等国际著名 IT 企业的软件分析与设计规约，归纳了“基于面向对象的软件分析与设计可视化建模”的一般性步骤，设计了系统分析与设计文档的规约。

第1章 基于面向对象的软件分析与设计一般性步骤

1.1 基于面向对象的软件分析与设计步骤

1.1.1 问题陈述 (Problem Statement)

应输出的制品包括：用文字结合图表的形式，从各个项目关系人，尤其是用户（或者使用者）的角度，描述系统要做什么。

1.1.2 需求分析 (Requirement Analysis)

应给出的制品包括：①用例图 (UseCase Diagram)、②活动图 (Activity Diagram)、③用例规约 (UseCase Specification)、④补充规约 (Supplementary Specification) 和⑤术语表 (Glossary) 等。

1.1.3 架构分析 (Architectural Analysis)

应给出的制品包括：①架构文档 (Architectural Document)，用包或类图的形式表述架构 (Architectural Diagram)；②软件系统的设计模式 (Design Pattern)；③系统关键抽象图 (Key Abstraction) (系统实体类图)；④用例实现 (UseCase Realization)，用顺序图或协作图描述用例实现的过程 (Sequence Diagram、Collaboration Diagram)。

1.1.4 用例分析 (UseCase Analysis)

应给出的制品包括：①用例的交互图 (UseCase Interaction Diagram)。②用例中的事件流图 (UseCase flows of events)。③设计系统的分析类，用 VOPC 图描述系统的分析类 (VOPC, 参与类类图, View of Participating Class)。在 VOPC 图中，应写出类的构造型 (Stereotypes, 分为边界类、控制类和实体类)、类的操作 (Responsibilities)、类的属性 (Attributes) 以及类之间的关联 (Relationships)。④对分析类的机制进行设计。

1.1.5 设计元素的设计 (Design Element)

应给出的软件制品包括：①设计类 (Design Classes) 图；②子系统 (Subsystems) 和接口 (Their Interfaces) 图，应描述子系统和接口之间的关联 (Their Relationships with other Design Elements)；③至少一个映射表，将分析类映射到设计元素，推进系统设计的过程。

1.1.6 系统运行时刻架构设计 (Run Time Architectural Design)

应给出的软件制品为：一个或多个类图或进程 (线程) 图 (Class Diagram)，描述并发进程或线程，将存在并发、冲突、互斥、死锁等潜在可能性的进程或子系统，用类图进行描述。在类图或进程图中，应明确参与类 (子系统、进程、线程)、类的属性和操作，以及类 (子系统、进程、线程) 之间的关联。

1.1.7 分布式系统的设计 (Describe Distribution)

应给出的软件制品为：一个或多个部署图 (Deployment Diagram)，描述节点 (Nodes)、连接 (Connections) 以及在节点上部署的软件、硬件、协议等制品。

1.1.8 用例设计 (UseCase Design)

应给出的软件制品包括：一个或多个用例及系统的 VOPC 类图。在该类图中，不但应明确类的操作、属性、管理，而且，应完全实现用例规约中的功能需求和补充规约中的非功能需求，达到客户要求的软件质量。

子系统设计 (Subsystem Design) 应给出的软件制品包括：①顺序图，描述子系统及接口实现功能的过程；②设计元素图 (类图)，包括设计类、子系统、子系统接口及其之间的关联；③设计元素图 (类图)，描述本系统对外界其他系统的依赖 (关联) 关系。

1.1.9 类的设计 (Class Design)

应给出的软件制品为：最终的 VOPC 图，完成类的所有参数，其中包括类、类的属性和操作、类之间的关系 [泛化、依赖、聚合、关联 (Generalization, Dependency, Aggregation, Association)]、类的策略模式等。

1.1.10 数据库的设计 (Data Base Design)

应给出的软件制品为：基于编程环境和平台，设计系统的数据库表或数据对象。

1.2 基于面向对象软件分析与设计的若干规则

本书提炼并设计了基于面向对象的软件分析与设计应遵循的若干规则。这些规则对于输出严谨、完善、科学、正确的系统分析与设计文档至关重要。

规则1：软件分析设计过程是一个迭代过程。图1-1描述了软件分析与设计的迭代过程。

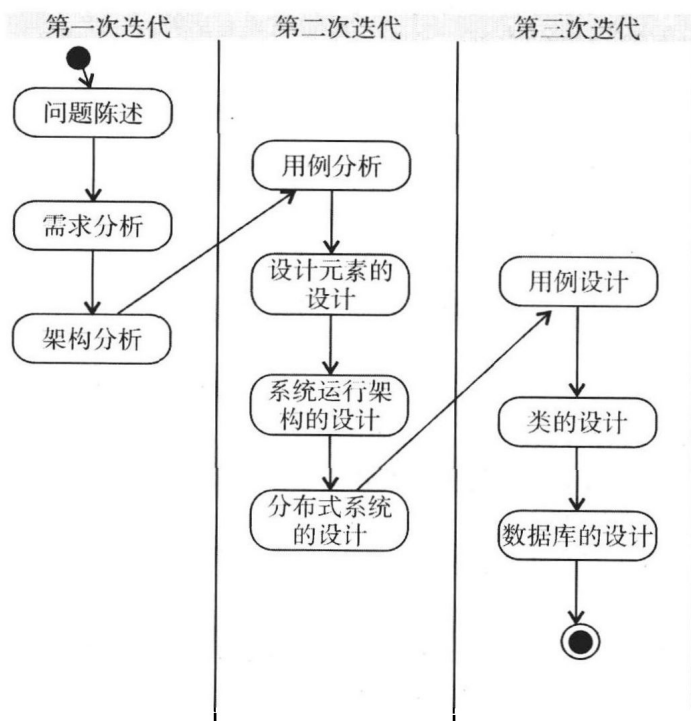


图1-1 基于面向对象软件分析与设计可视化建模迭代过程

规则2：软件分析与设计，采用“敏捷开发+RUP”的理念，在具体应用中，可以对某些步骤进行适当删减。例如，一些简单的小系统，上述的一些设计步骤可以合并，或省略。

规则3：大量软件工程实践表明，在需求文档中，至少要完成的制品包括：问题陈述、用例析取、用例规约、补充规约、术语表。上述需求建模制品总和是构成一个完善、可靠、有效的需求文档的必要内容。

规则4：软件工程实践表明，在设计文档中，必须包含的制品包括：系统架构文档、系统架构设计图、系统关键抽象（实体类）类图、系统分析类的设计、类的机制分析与设计、子系统的划分与设计、接口设计、设计类的设计等。

规则 5：系统运行时刻架构设计、系统部署设计可以丰富并充实系统的设计方案，使设计方案更加完善和严谨。

1.3 基于面向对象软件分析与设计规约若干关注点

(1) 在“架构分析”中，应关注：

- 确认所有的与系统有关的领域知识、关键概念都被找出来了
- 确认类的名称是否能反映类的角色
- 确认各个抽象水平的架构图是否协调一致

(2) 在“设计元素的设计”中，应关注：

- 确定子系统的定义是符合逻辑的
- 确认子系统的接口已经定义
- 确定分析类映射到设计元素的合理性
- 确认划分子系统或包是符合逻辑的
- 思考在一个子系统或包中，是否有类或几个类可以与其他分离，形成独立的部分

(3) 在“系统运行时架构的设计”中，应关注：

- 确认描述的存在并发的进程或线程的构造型是正确的
- 如果描述了一个并发进程，确定是否需要用聚合（组合）描述该进程与其线程

的关系

- 确认所有并发、冲突、互斥、死锁的进程、线程都已经被定义了

(4) 在“分布式系统的设计”中，应关注：

- 确认已经对所有的节点及其关联都建模了
- 确认所有进程和线程已经被部署到节点了
- 确认分布式设计方案的合理性

(5) 在“用例设计”中，应关注：

- 确认用例中所有的基本事件流和备选事件流都已经被处理了
- 确认用例需实现的功能都被分配到类
- 确认类的操作是合理的
- 确认已经对接口做了正确的建模

(6) 在“类的设计”中，应关注：

- 类的名称能反映类的实质且容易被理解
- 确认类的属性都是单一的逻辑概念
- 确认类的属性是可理解的，且正确代表了其需存储的信息
- 确认类的状态是合理的，且其名称是可理解的
- 确定类之间的关系是正确的
- 确定类的多策略模式是正确的
- 确定类的泛化（继承）确实捕获了若干类的共性
- 确定（类、包）继承关系中确实存在明显的层次结构