

国外优秀信息科学与技术系列教学用书

面向对象 分析与设计导论

——使用 UML 和统一过程

(影印版)

AN INTRODUCTION TO OBJECT-ORIENTED ANALYSIS
AND DESIGN WITH UML AND THE UNIFIED PROCESS

■ Stephen R. Schach

 Education



高等教育出版社
Higher Education Press

国外优秀信息科学与技术系列教学用书

面向对象分析与设计导论

——使用 UML 和统一过程

(影印版)

AN INTRODUCTION TO OBJECT ORIENTED
ANALYSIS AND DESIGN WITH UML AND
THE UNIFIED PROCESS

江苏工业学院图书馆
藏书章

Stephen R. Schach



高等教育出版社
HIGHER EDUCATION PRESS

图字: 01-2004-6713 号

An Introduction to Object-Oriented Analysis and Design with UML and the Unified Process

Stephen R. Schach

原版 ISBN:0-07-282646-0

Copyright © 2004 by the McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Authorized English language reprint edition jointly published by McGraw-Hill Education (Asia) Co. and Higher Education Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SARs and Taiwan areas. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书英文影印版由高等教育出版社和美国麦格劳-希尔教育出版(亚洲)公司合作出版。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售。未经许可之出口,视为违反著作权法,将受法律之制裁。

未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

面向对象分析与设计导论:使用UML和统一过程 = An Introduction to Object-Oriented Analysis and Design with Uml and the Unified Process / (美)沙赫查 (Schach, S. R.). —影印版. —北京:高等教育出版社, 2005.1

ISBN 7-04-016338-1

I.面... II.沙... III.面向对象语言-程序设计
—教材—英文 IV.TP312

中国版本图书馆CIP数据核字(2005)第000060号

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 100011
总 机 010-58581000

经 销 新华书店北京发行所
印 刷 北京外文印刷厂

开 本 787×1092 1/16
印 张 25.75
字 数 560 000

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

版 次 2005年1月第1版
印 次 2005年1月第1次印刷
定 价 39.00元(含光盘)

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: 16338-00

出版说明

20 世纪末,以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用,带动了世界范围信息产业的蓬勃发展,为许多国家带来了丰厚的回报。

进入 21 世纪,尤其随着我国加入 WTO,信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展,但与发达国家相比,甚至与印度、爱尔兰等国家相比,还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力,最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材,在有条件的学校推动开展英语授课或双语教学,是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此,教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求,一是要高水平,二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下,经过比较短的时间,第一批由教育部高等教育司推荐的 20 多种引进教材已经陆续出版。这套教材出版后受到了广泛的好评,其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品,代表了目前世界信息科学技术教育的一流水平,而且价格也是最优惠的,与国内同类自编教材相当。这套教材基本覆盖了计算机科学与技术专业的课程体系,体现了权威性、系统性、先进性和经济性等特点。

目前,教育部正在全国 35 所高校推动示范性软件学院的建设,这也是加快培养信息科学技术人才的重要举措之一。为配合软件学院的教学工作,结合各软件学院的教学计划和课程设置,高等教育出版社近期聘请有关专家和软件学院的教师遴选推荐了一批相应的原版教学用书,正陆续组织出版,以方便各软件学院开展双语教学。

我们希望这些教学用书的引进出版,对于提高我国高等学校信息科学技术的教学水平,缩小与国际先进水平的差距,加快培养一大批具有国际竞争力的高质量信息技术人才,起到积极的推动作用。同时我们也欢迎广大教师和专家们对我们的教材引进工作提出宝贵的意见和建议。联系方式: hep.cs@263.net。

高等教育出版社

二〇〇二年九月

序 言

本书是面向对象系统分析和设计的第一门课程的教材。书中内容遵照 IS'2002¹(课程 IS'01.7, 分析和逻辑设计)的教学指导方针;特别地,学生不必具有任何编程知识。本书使用统一建模语言(UML)为信息系统建模,使用统一过程作为方法学。

本书的主要目标是,确保在课程结束时每一位学生都能够进行面向对象分析和设计。这一目标是通过下列三种不同的方式达到的:

- **教学方法:** 详细解释面向对象分析与设计的步骤。在介绍了每一个步骤之后,通过在两个连续的案例研究的第一个案例研究中应用这一步骤,来说明该步骤。然后,当解释并举例说明了每一步骤后,就将所有步骤应用到第二个案例研究中。
- **内容组织:** 通过使第一部分尽量简短,本书尽可能早地开始介绍面向对象分析和设计(本书第二部分)的内容。这样,学生就可以把半学年或三个月的小学期中的大部分时间用于掌握面向对象分析和设计的内容。
- **实践:** 精通面向对象分析和设计的惟一途径就是反复地实践它。相应地,我们在每一章的末尾都给出了不同类型的习题。首先,是针对面向对象分析和设计的练习;其次,是基于两个连续案例研究的习题,这两个案例研究均以最详尽的形式给出。第三,是一个基于团队的学期项目,学生在该项目中要从头开始进行系统分析和设计。教师可以从习题中选择一道或多道题作为作业布置给学生。

在 20 世纪 90 年代,活跃在面向对象技术领域中的三位著名人物是 Grady Booch(他的方法称为“Booch 方法”)、Ivar Jacobson(“Objectory”)和 Jim Rumbaugh(“OMT”)。Booch、Jacobson 和 Rumbaugh 随后在 Rational 公司进行了强强联合,为信息技术做出了两个里程碑式的贡献:统一建模语言(UML)和统一过程。

在 1997 年推出后的几个月里,UML 1.0 几乎风靡全球。在今天,编写一本尝试用任何其他建模语言来表示信息系统的教材都是不可想象的。UML 现在是对象管理组(OMG)的一项产品,OMG 是由全球领先的软件技术公司组成的一个联盟。本书使用 UML 1.4,这是编写此书时的最新版本。

Booch、Jacobson 和 Rumbaugh 随后开发了一种方法,它整合(“统一”)了他们各自(但相似)的三种方法。这种统一的方法最初称为“统一软件开发过程(USDP)”或“Rational 统一过程(RUP)”。目前,它也称为“统一过程(United Process)”。就像 UML 一样,它没有替代的选择——对于 21 世纪介绍信息系统的面向对象分析和设计的教材而言,目前最佳的方法是统一过程。三位作者现在已不再为他们原来的三种方法提供支持。

本书分为三个部分。如前所述,第一部分尽可能简短,它包含足以让学生理解利用

¹ IS'2002 即 Information System 2002,由 ACM、AIS 和 ATIP 联合制定。原文此处为 IS'2001,疑有误。——编者注

统一过程和 UML 进行面向对象分析和设计所需要的最少的信息。第二部分介绍如何把 UML 作为建模语言,使用统一过程来进行面向对象分析和设计。第三部分讨论在系统分析和设计课程中应当讲述的 9 大主题,如项目管理、计划和团队组织等。也就是说,第二部分讲授必要的专门技能,而第三部分则强调每一名优秀的系统分析师必须具备的管理和交际技能。此外,第三部分还讲述了更深入的专门技能,如测试和用户界面设计。本书余下的内容重点介绍 21 世纪信息系统的必要知识:可移植性、重用、互操作性、万维网以及过程改进,包括 CMM。最后,如果教师希望介绍更多的技术概念,还可以介绍模块化、多态性和动态绑定等,最后一章包含关于这些主题的介绍性内容。

如前所述,遵照 IS'2002 的教学指导方针,学习本书不需要有编程经验。但是,如果学生具有编程经验,则有三个额外的习题可以作为作业。更确切地讲,学期项目中包含大量的每位学生都可以完成的任务,如确定需求和进行面向对象分析。但是,学期项目还包含三个可选的任务,包括一个完整的实现,如果学生具有必要的编程经验,教师就可以选择将这些任务作为作业布置给学生。

习题

在每章结尾都提供了一些复习题和习题。可以从本教材相关的章节中找到这些复习题的答案。习题具有不同的类型。如前所述,既有针对面向对象分析和设计的练习,又有基于两个连续案例研究的习题。这些习题包含试验性的问题、数值问题以及用于测试学生对课堂上讲授内容掌握程度的问题。最后,在大多数章结尾都提供有学期项目的部分任务。

教师手册中具有所有习题的答案,包括学期项目中所有任务的解决方案,甚至包括学期项目的 C++ 和 Java 实现。教师可以与当地的 McGraw-Hill 代表处联系,索取教师手册。

补充材料

在本书的配套 Web 站点(www.mhhe.com/schach)上可以找到本教材完整的讲义,它是以 PowerPoint 幻灯片形式提供的。此外,可以为采用本书的教师还提供一张教师光盘,其中包含有 PowerPoint 讲义以及下列额外的资源:

- 教师手册包含使用本书的建议、教学建议以及各章末所有习题的解答。
- 一个包含判断题、多项选择题和完形填空题的测试题库;一个计算机化的测试题库,通过 Brownstone Diploma 软件可在局域网上进行测试管理。这些测试题也可以打印成标准试卷,或发布到 Web 站点上供学生访问。

采用该书作教材的教师可向 McGraw-Hill 公司北京代表处联系索取教学课件资源,传真:(010)6263 8354,电子邮件:webmaster@mcgraw-hill.com.cn。

Preface

This book is a textbook for the first course in object-oriented systems analysis and design. The material conforms to the guidelines of IS'2001 (Course IS'01.7, Analysis and Logical Design); in particular, the student does not need to have any programming knowledge. The book uses the Unified Modeling Language (UML) to model information systems and the Unified Process as the methodology.

The primary objective of this book is to ensure that, by the end of the course, every student is able to perform object-oriented analysis and design. This objective is achieved in three different ways:

- **Pedagogics:** The steps of object-oriented analysis and design are carefully explained. After each step has been presented, it is illustrated by applying it to the first of the two running case studies. Then, when each step has been explained and illustrated separately, all the steps are applied to the second running case study.
- **Organization:** The material on object-oriented analysis and design (Part 2 of the book) is presented as early as possible by keeping Part 1 of the book as short as possible. The result is that the student has most of the semester or quarter to master the material on object-oriented analysis and design.
- **Practice:** The only way to become proficient in object-oriented analysis and design is by doing it repeatedly. Accordingly, there are different types of problems at the end of each chapter. First, there are exercises in object-oriented analysis and design. Second, there are problems based on the two running case studies, both of which are presented in the fullest detail. Third, there is a team-based term project in which the students perform systems analysis and design from scratch. The instructor has the option of assigning one or more problems of any type.

During the 1990s, there were three major figures in object-oriented technology, namely, Grady Booch (his approach is referred to as “Booch’s method”), Ivar Jacobson (“Objectory”), and Jim Rumbaugh (“OMT”). Booch, Jacobson, and Rumbaugh then joined forces at Rational, Inc., and made two landmark contributions to information technology: the Unified Modeling Language (UML) and the Unified Process.

Within a few months of its introduction in 1997, UML version 1.0 literally swept the world. It is inconceivable that a textbook written today would attempt to represent an information system using any other modeling language. UML is now a product of the Object Management Group (OMG), a consortium of the leading software technology firms worldwide. This book uses version 1.4 of UML, the latest version at the time of writing.

Next, Booch, Jacobson, and Rumbaugh developed a methodology that integrated (“unified”) their three separate (but similar) methodologies. This unified methodology was first known as the Unified Software Development Process (USDP) or the Rational Unified Process (RUP). Currently it is also called the Unified Process. Just as with UML, there is now no alternative—currently the best methodology for a twenty-first-century textbook on object-oriented analysis and design of information systems is the Unified Process. The other three methodologies are now no longer supported by their respective authors.

The book is divided into three parts. Part 1, as previously stated, has been kept as short as possible. It contains the minimum information needed to enable students to understand object-oriented analysis and design with the Unified Process and UML. Part 2 is on how to perform object-oriented analysis and design using the Unified Process with UML as the

modeling language. Part 3 presents nine major topics that should be taught in a course on systems analysis and design, such as project management, planning, and team organization. That is, Part 2 teaches the necessary technical skills, whereas Part 3 stresses the managerial and interpersonal skills that every good systems analyst must have. In addition, further technical skills are taught in Part 3, such as testing and user-interface design. As with the rest of this book, there is an emphasis on the knowledge needed for information systems for the twenty-first century, including portability, reuse, interoperability, the World Wide Web, and process improvement, including CMM. Finally, if an instructor wishes to cover more technical concepts such as modularity and polymorphism and dynamic binding, the last chapter contains introductory-level material on these topics.

As previously mentioned, following the IS'2001 guidelines, programming is not a prerequisite for this book. However, if the class has programming experience, there are three additional problems that can be assigned. More specifically, the term project consists of a number of components that every student can complete, such as determining the requirements and performing the object-oriented analysis. However, the term project also includes three optional components, including a complete implementation, that the instructor may choose to assign if the class has the necessary programming experience.

PROBLEM SETS

There are review questions and problems at the end of each chapter. The review questions can all be answered from the relevant section of the textbook. The problems are of different types. As previously mentioned, there are exercises in object-oriented analysis and design, as well as problems based on the two running case studies. There are also essay-type problems, numerical problems, and problems that test how thoroughly the students understand what was taught in class. Finally, at the end of most chapters there are components of the term project.

The Instructor's Manual has solutions to every problem, including all the components of the term project. There are even implementations of the term project in C++ and Java. The instructor should contact his or her McGraw-Hill representative to obtain the Instructor's Manual.

SUPPLEMENTARY MATERIAL

A complete set of lecture notes in the form of PowerPoint slides are available at the website for this book, www.mhhe.com/schach. In addition, an Instructor CD-Rom is available to adopters. The Instructor CD-Rom contains the PowerPoint lecture notes, together with the following additional resources:

- An Instructor's Manual containing suggestions for using the book, teaching suggestions, and answers to all the end of chapter material.

- A testbank containing true-false, multiple choice, and fill-in-the-blank questions, as well as a computerized testbank with Brownstone Diploma software offering fully networkable LAN test administration. Tests also can be printed for standard paper delivery or posted to a website for student access.

Acknowledgments

I should like to thank the reviewers who have commented on the various earlier drafts of this book. They are

Thad Crews
Western Kentucky University
Kevin P. Duffy
The University of Texas at Arlington
Jeff Gray
University of Alabama at Birmingham
Kris Howell
University of Southern Colorado
Jon Jasperson
University of Oklahoma
Geoffrey Kennedy
University of Sydney
Synovie Merchant
California State University—Sacramento
Megan Murray
Kennesaw State University
Jeffrey Parsons
Memorial University of Newfoundland
David L. Paul
University of Denver
Elizabeth Perry
Binghamton University
Carl Scott
University of Houston
J. Michael Tarn
Western Michigan University
Amir Tomer
The Technion
Steve Walczak
Colorado University at Denver

Each of these reviewers, without exception, provided numerous helpful suggestions that have improved this book, and I am grateful to all of them. Nevertheless, I wish to single out Dr. Kris Howell, whose comments, suggestions, and criticisms were of immeasurable assistance to me.

A special thank you is extended to the following focus group participants:

Jeffrey Parsons
Memorial University of Newfoundland
Parag C. Pendharkar
Penn State University
Carl Scott
University of Houston

Ron Thompson
Wake Forest University
 Steve Walczak
Colorado University at Denver

I am indebted to three individuals who also have made significant contributions to many of my earlier books. Dr. Jeff Gray was previously a co-author of four Instructor's Manuals. For this book he has served as a reviewer, and he also was responsible for Appendices B through E; that is, he performed the design and implementation of both the case studies. My son David and my daughter Lauren are two of the co-authors of the Instructor's Manual. Lauren also contributed to the PowerPoint lecture notes. I thank Jeff, David, and Lauren for their usual meticulous work.

Kris Irwin took over Jeff's previous role—she provided a complete solution to the term project, including implementing it in both Java and C++. It was a pleasure working with an information technology professional of Kris's caliber. I also thank Kris for her many suggestions that led to improvements in the term project.

The copyeditor was Betsy Blumenthal. I thank her for her superlative work and her many helpful suggestions.

I turn now to my publisher, McGraw-Hill. The aim of object-oriented analysis is to determine *what* the target information system should do, and the aim of object-oriented design is to determine *how* the information system should do it. From the start of this project, publisher George Werthman and senior editor Rick Williamson made it clear *what* book they wanted me to write, but they left *how* to write it entirely up to me. It has been an eye-opening education to work with George and Rick. It also has been tremendous fun.

I should like to single out project manager Natalie Ruffatto. Natalie was a tower of strength throughout the production process. I deeply appreciate all that she has done.

I am grateful to my development editor, Kelly Delso, for her help from start to finish. I should also like to thank other McGraw-Hill professionals: Paul Ducham, senior sponsoring editor; Greta Kleinert, marketing manager; Greg Bates, media producer; Debra R. Sylvester, production supervisor; Adam Rooke, designer; and Rose Range, supplement coordinator.

I am also grateful to Jade Moran, project manager, and the compositors at Interactive Composition Corporation for their outstanding work. They were always willing to go the extra mile—and more.

I would like to thank the numerous instructors from all over the world who have sent me e-mail about my earlier books. I have used some of their comments, criticisms, and suggestions in this book. I look forward to receiving instructors' feedback on this book, too. My e-mail address is srs@vuse.vanderbilt.edu.

I also want to thank Kai Chen, Fang Fang, David Gannaway, Yun Huang, Kris Irwin (again), Michael Kearney, Johann Klemmack, Tao Lu, Yolanda McMillian, Sarah Miracle, Melissa Nestor, and Glenn Stein, the students in a graduate course on the Unified Process that I recently taught at Vanderbilt University. Their comments and critical analyses have helped to shape my views on the Unified Process, and thereby enriched this book.

I am grateful for the questions and suggestions about earlier books that students have e-mailed to me, and I look forward to receiving student feedback on this book.

I should like to thank Howard Aksen. In 1988 he persuaded me to write my first book, *Software Engineering*. Without Howard's urging, I would never have had 15 years of sheer enjoyment writing textbooks.

Finally, as always, I would like to thank my family for their unfailing support. Once again, writing this book has been a family activity. For the 11th time, it is my privilege to dedicate this book to my wife, Sharon, and my children, David and Lauren, with love.

Stephen R. Schach

Brief Contents

Preface v

Acknowledgments vii

PART ONE

Introduction to UML and the Unified Process 1

- 1 Introduction to Information Systems 3
- 2 How Information Systems Are Developed 17
- 3 The Object-Oriented Paradigm, UML, and the Unified Process 35

PART TWO

UML and the Unified Process 51

- 4 The Requirements Workflow I 53
The Requirements Workflow II 75
- 6 The Object-Oriented Analysis Workflow I 99
- 7 The Object-Oriented Analysis Workflow II 129
- 8 The Object-Oriented Design Workflow 161
- 9 The Workflows and Phases of the Unified Process 177
- 10 More on UML 189

PART THREE

Major Topics in Systems Analysis and Design 205

- 11 CASE 207
- 12 Teams 231

13 Testing 243

14 Management Issues 259

15 Planning and Estimating 287

16 Maintenance 311

17 User-Interface Design 325

18 Introduction to Web-Based Information Systems 337

19 Introduction to Database Management Systems 347

20 Technical Topics 357

APPENDIXES

- A Term Project: Chocoholics Anonymous** 367
- B Object-Oriented Design: Osbert Oglesby Case Study** 371
- C Object-Oriented Design: MSG Foundation Case Study** 379

INDEX 385

Contents

Preface v

Acknowledgments vii

PART ONE

INTRODUCTION TO UML AND THE UNIFIED PROCESS 1

Chapter 1

Introduction to Information Systems 3

1.1 Categories of Information Systems 5

1.2 Traditional Information System
Development 6

1.2.1 *The Requirements Phase* 6

1.2.2 *The Analysis Phase* 6

1.2.3 *The Design Phase* 7

1.2.4 *The Implementation Phase* 7

1.2.5 *The Maintenance Phase* 7

1.2.6 *Retirement* 8

1.3 Why There Is No Planning Phase 8

1.4 Why There Is No Testing Phase 8

1.5 Why There Is No Documentation Phase 9

1.6 Systems Analysis and Design 10

1.7 Maintenance 10

1.8 Information Technology Professionals 12

Key Terms 15

Review Questions for Chapter 1 15

Problems 16

References 16

Chapter 2

How Information Systems Are Developed 17

2.1 Information System Development
in Theory 17

2.2 Winburg Mini Case Study 18

2.3 Lessons of the Winburg Mini Case Study 21

2.4 Teal Tractors Mini Case Study 22

2.5 Iteration and Incrementation 23

2.6 Iteration: The Newton-Raphson Algorithm 27

2.7 The Winburg Mini Case Study Revisited 28

2.8 Other Aspects of Iteration
and Incrementation 29

2.9 Managing Iteration and Incrementation 31

2.10 Maintenance Revisited 32

Key Terms 32

Review Questions for Chapter 2 33

Problems 33

References 33

Chapter 3

The Object-Oriented Paradigm, UML, and the Unified Process 35

3.1 The Traditional Paradigm versus the
Object-Oriented Paradigm 36

3.2 Objects and Classes 38

3.3 Inheritance 40

3.4 Generalization, Aggregation,
and Association 42

3.5 Examples of UML Class Modeling 44

3.6 Information Hiding 46

3.7 The Unified Process 47

3.8 Iteration and Incrementation within the
Unified Process 48

Key Terms 49

Review Questions for Chapter 3 50

Problems 50

References 50

PART TWO

UML AND THE UNIFIED PROCESS 51

Chapter 4

The Requirements Workflow I 53

4.1 Determining What the Client Needs 53

4.2 Overview of the Requirements Workflow 54

4.3 Understanding the Domain 55

4.4 Initial Understanding of the Domain:
Osbert Oglesby Case Study 56

4.5 Business Model 57

4.5.1 *Interviewing* 57

4.5.2 *Other Techniques* 58

4.5.3 *Use Cases* 59

4.6 Initial Business Model: Osbert Oglesby
Case Study 60

4.7 Initial Requirements 63

4.8 Initial Requirements: Osbert Oglesby
Case Study 63

- 4.9** Continuing the Requirements Workflow: Osbert Oglesby Case Study 65
- 4.10** It Ain't Over Till It's Over 70
 - Key Terms 72
 - Case Study Key Terms 72
 - Review Questions for Chapter 4 72
 - Problems 73
 - Reference 73

Chapter 5

The Requirements Workflow II 75

- 5.1** MSG Foundation Case Study 75
- 5.2** Initial Understanding of the Domain: MSG Foundation Case Study 76
- 5.3** Initial Business Model: MSG Foundation Case Study 77
- 5.4** Initial Requirements: MSG Foundation Case Study 80
- 5.5** Continuing the Requirements Workflow: MSG Foundation Case Study 81
- 5.6** Revising the Requirements: MSG Foundation Case Study 83
- 5.7** Refining the Revised Requirements: MSG Foundation Case Study 92
 - Key Term 97
 - Case Study Key Terms 97
 - Review Questions for Chapter 5 97
 - Problems 97
 - Systems Analysis and Design Projects* 97
 - Term Project* 98

Chapter 6

The Object-Oriented Analysis Workflow I 99

- 6.1** The Analysis Workflow 99
- 6.2** Extracting Entity Classes 101
- 6.3** Initial Functional Model: Osbert Oglesby Case Study 103
- 6.4** Initial Class Diagram: Osbert Oglesby Case Study 104
- 6.5** Initial Dynamic Model: Osbert Oglesby Case Study 109
- 6.6** Extracting Boundary Classes 111
- 6.7** Extracting Control Classes 112
- 6.8** Refining the Use Cases 112
- 6.9** Use-Case Realization 113
 - 6.9.1 *Buy a Masterpiece Use Case* 115
 - 6.9.2 *Buy a Masterwork Use Case* 119

- 6.9.3 *Buy Other Painting Use Case* 122
- 6.9.4 *The Remaining Five Use Cases* 123

- 6.10** Incrementing the Class Diagram 124
- 6.11** Where's the Specification Document? 125
 - Key Terms 127
 - Review Questions for Chapter 6 127
 - Problems 127
 - Reference 128

Chapter 7

The Object-Oriented Analysis Workflow II 129

- 7.1** Extracting Entity Classes: MSG Foundation Case Study 129
- 7.2** Initial Functional Model: MSG Foundation Case Study 130
- 7.3** Initial Class Diagram: MSG Foundation Case Study 132
- 7.4** Back to the Requirements Workflow 133
- 7.5** Initial Dynamic Model: MSG Foundation Case Study 134
- 7.6** Revising the Entity Classes: MSG Foundation Case Study 136
- 7.7** Extracting Boundary Classes: MSG Foundation Case Study 137
- 7.8** Extracting Control Classes: MSG Foundation Case Study 138
- 7.9** Use-Case Realization: MSG Foundation Case Study 138
 - 7.9.1 *Estimate Funds Available for Week Use Case* 138
 - 7.9.2 *Manage an Asset Use Case* 144
 - 7.9.3 *Update Estimated Annual Operating Expenses Use Case* 146
 - 7.9.4 *Produce a Report Use Case* 147
- 7.10** Incrementing the Class Diagram 151
- 7.11** More on Actors 152
- 7.12** More on Use Cases 155
- 7.13** Risk 156
 - 7.13.1 *Rapid Prototyping* 156
 - 7.13.2 *Scenarios and the Client's Needs* 158
- Key Terms 159
- Review Questions for Chapter 7 159
- Problems 159
- Systems Analysis and Design Projects* 160
- Term Project* 160
- Reference 160

Chapter 8

The Object-Oriented Design Workflow 161

- 8.1** The Design Workflow 161
- 8.2** Traditional versus Object-Oriented Design 163
- 8.3** Formats of the Attributes 164
 - 8.3.1 *Formats of the Attributes of the Osbert Oglesby Case Study* 164
 - 8.3.2 *Formats of the Attributes of the MSG Foundation Case Study* 165
- 8.4** Allocation of Operations to Classes 167
 - 8.4.1 *Responsibility-Driven Design* 167
 - 8.4.2 *Inheritance* 167
- 8.5** Allocation of Operations: Osbert Oglesby Case Study 169
 - 8.5.1 *Responsibility-Driven Design* 169
 - 8.5.2 *Inheritance* 169
- 8.6** CRC Cards 172
 - Key Terms 174
 - Review Questions for Chapter 8 174
 - Problems 174
 - Systems Analysis and Design Projects* 175
 - Term Project* 175
 - References 175

Chapter 9

The Workflows and Phases of the Unified Process 177

- 9.1** The Workflows of the Unified Process 177
 - 9.1.1 *The Requirements Workflow* 178
 - 9.1.2 *The Analysis Workflow* 178
 - 9.1.3 *The Design Workflow* 179
 - 9.1.4 *The Implementation Workflow* 179
 - 9.1.5 *The Test Workflow* 179
- 9.2** The Phases of the Unified Process 180
 - 9.2.1 *The Inception Phase* 180
 - 9.2.2 *The Elaboration Phase* 183
 - 9.2.3 *The Construction Phase* 183
 - 9.2.4 *The Transition Phase* 184
- 9.3** Why a Two-Dimensional Model? 185
 - Key Terms 188
 - Review Questions for Chapter 9 188
 - Problems 188
 - Term Project* 188
 - References 188

Chapter 10

More on UML 189

- 10.1** UML Is *Not* a Methodology 189
- 10.2** Class Diagrams 190
 - 10.2.1 *Aggregation* 191
 - 10.2.2 *Multiplicity* 192
 - 10.2.3 *Composition* 193
 - 10.2.4 *Generalization* 194
 - 10.2.5 *Association* 194
- 10.3** Notes 195
- 10.4** Use-Case Diagrams 195
- 10.5** Stereotypes 195
- 10.6** Interaction Diagrams 196
- 10.7** Statecharts 197
- 10.8** Activity Diagrams 200
- 10.9** Package Diagrams 202
- 10.10** Component Diagrams 202
- 10.11** Deployment Diagrams 202
- 10.12** Review of UML Diagrams 202
- 10.13** UML and Iteration 203
 - Key Terms 203
 - Review Questions for Chapter 10 204
 - Problems 204
 - Reference 204

PART THREE

MAJOR TOPICS IN SYSTEMS ANALYSIS AND DESIGN 205

Chapter 11

CASE 207

- 11.1** Taxonomy of CASE 208
- 11.2** The Scope of CASE 210
- 11.3** Versions 211
 - 11.3.1 *Revisions* 211
 - 11.3.2 *Variations* 212
 - 11.3.3 *Version-Control Tools* 213
- 11.4** Configuration Control 215
- 11.5** Build Tools 216
- 11.6** CASE Environments 217
- 11.7** Environments for Information Systems 218
- 11.8** Potential Problems with Environments 219
- 11.9** Productivity Gains with CASE Technology 220
- 11.10** CASE and Aesthetics 220

Key Terms	229
Review Questions for Chapter 11	229
Problems	229
<i>Term Project</i>	229
References	229

Chapter 12

Teams 231

12.1	Team Organization	231
12.2	Traditional Chief Programmer Teams	233
12.3	Modern Hierarchical Teams	234
12.4	Other Ways of Organizing Teams	236
12.4.1	<i>Synchronize-and-Stabilize Teams</i>	236
12.4.2	<i>Extreme Programming Teams</i>	237
	Key Terms	240
	Review Questions for Chapter 12	240
	Problems	240
	<i>Term Project</i>	241
	References	241

Chapter 13

Testing 243

13.1	Introduction to Testing	243
13.2	Quality Issues	245
13.2.1	<i>Quality Assurance</i>	245
13.2.2	<i>Quality Assurance Terminology</i>	246
13.2.3	<i>Managerial Independence</i>	246
13.3	Nonexecution-Based Testing	248
13.3.1	<i>Principles of Inspections</i>	248
13.3.2	<i>How Inspections Are Performed</i>	249
13.4	Execution-Based Testing	251
13.4.1	<i>The Relevance of Execution-Based Testing</i>	251
13.4.2	<i>Principles of Execution-Based Testing</i>	251
13.5	The Two Basic Types of Test Cases	252
13.6	What Execution-Based Testing Should Test	252
13.6.1	<i>Utility</i>	252
13.6.2	<i>Reliability</i>	252
13.6.3	<i>Robustness</i>	253
13.6.4	<i>Performance</i>	253
13.6.5	<i>Correctness</i>	253
13.7	Who Should Perform Execution-Based Testing?	254
13.8	When Testing Stops	255
	Key Terms	256
	Review Questions for Chapter 13	256

Problems	256
<i>Term Project</i>	257
References	257

Chapter 14

Management Issues 259

14.1	Cost-Benefit Analysis	259
14.2	Risk Analysis	260
14.3	Improving the Process	261
14.3.1	<i>Capability Maturity Models</i>	262
14.3.2	<i>Other Process Improvement Initiatives</i>	264
14.3.3	<i>Costs and Benefits of Process Improvement</i>	265
14.3.4	<i>CMM and CASE</i>	266
14.4	Metrics	266
14.5	CPM/PERT	267
14.6	Choice of Programming Language	268
14.7	Reuse	272
14.7.1	<i>Reuse Concepts</i>	272
14.7.2	<i>Impediments to Reuse</i>	274
14.8	Reuse Case Studies	275
14.8.1	<i>Raytheon Missile Systems Division</i>	275
14.8.2	<i>Toshiba Software Factory</i>	276
14.8.3	<i>NASA Software</i>	277
14.8.4	<i>GTE Data Services</i>	277
14.8.5	<i>Hewlett-Packard</i>	278
14.8.6	<i>European Space Agency</i>	278
14.9	Portability	279
14.9.1	<i>Hardware Incompatibilities</i>	279
14.9.2	<i>Operating System Incompatibilities</i>	280
14.9.3	<i>Compiler Incompatibilities</i>	280
14.10	Why Portability?	281
	Key Terms	282
	Review Questions for Chapter 14	283
	Problems	283
	<i>Term Project</i>	284
	References	284

Chapter 15

Planning and Estimating 287

15.1	Planning and the Information System Life Cycle	287
15.2	Estimating Duration and Cost	289
15.2.1	<i>Metrics for the Size of an Information System</i>	290

- 15.2.2 *Techniques of Cost Estimation* 292
- 15.2.3 *COCOMO* 294
- 15.2.4 *COCOMO II* 295
- 15.2.5 *Tracking Duration and Cost Estimates* 295
- 15.3** Components of a Project Management Plan 296
- 15.4** Project Management Plan Framework 297
- 15.5** IEEE Project Management Plan Framework 299
- 15.6** Project Management Plan: Osbert Oglesby Case Study 301
- 15.7** Planning of Testing 305
- 15.8** Training Requirements 306
- 15.9** Documentation Standards 306
- 15.10** CASE Tools for Planning and Estimating 307
- 15.11** Testing the Project Management Plan 307
 - Key Terms 308
 - Review Questions for Chapter 15 308
 - Problems 308
 - Term Project* 309
 - References 309

Chapter 16

Maintenance 311

- 16.1** Why Maintenance Is Necessary 311
- 16.2** Development and Maintenance 312
- 16.3** What Is Required of Maintainers? 313
- 16.4** Temperate Fruit Committee Mini Case Study 315
- 16.5** The Management of Maintenance 316
 - 16.5.1 *Fault Reports* 316
 - 16.5.2 *Authorizing Changes to the Information System* 317
 - 16.5.3 *Ensuring Maintainability* 318
 - 16.5.4 *The Problem of Repeated Maintenance* 319
- 16.6** Maintenance and the Object-Oriented Paradigm 319
- 16.7** Maintenance Skills versus Development Skills 320
- 16.8** Reverse Engineering 321
- 16.9** Testing during Maintenance 321
- 16.10** CASE Tools for Maintenance 322
 - Key Terms 323
 - Review Questions for Chapter 16 323
 - Problems 323
 - References 323

Chapter 17

User-Interface Design 325

- 17.1** Input and Output Devices 325
- 17.2** Consistency 327
- 17.3** Correcting Mistakes 327
- 17.4** Metaphors 328
- 17.5** Layout and Aesthetics 330
- 17.6** Dynamic Interfaces 331
- 17.7** The Unified Process and the User Interface 332
 - 17.7.1 *Requirements Workflow* 332
 - 17.7.2 *Prototyping the User Interface* 332
 - 17.7.3 *User-Interface Design* 333
 - 17.7.4 *Iteration* 334
- Key Terms 334
- Review Questions for Chapter 17 334
- Problems 334
- Term Project* 335
- References 335

Chapter 18

Introduction to Web-Based Information Systems 337

- 18.1** Prelude to the Web 337
- 18.2** Security Issues 342
- 18.3** Analysis and Design Implications of Networks 343
- 18.4** Web-Based Information Systems 345
 - Key Terms 345
 - Review Questions for Chapter 18 345
 - Problems 345
 - Term Project* 346

Chapter 19

Introduction to Database Management Systems 347

- 19.1** Files and Their Problems 348
- 19.2** Tables 350
- 19.3** Traditional Database Systems 352
- 19.4** Object-Oriented Database Systems 354
- 19.5** Database Design and the Unified Process 355
- 19.6** The Future of Object-Oriented Database Management Systems 355
 - Key Terms 355
 - Review Questions for Chapter 19 355
 - Problems 356
 - Term Project* 356

Chapter 20

Technical Topics 357

- 20.1** Source Code and Compiled Code 357
- 20.2** Modularity 358
- 20.3** Polymorphism and Dynamic Binding 361
- 20.4** Example of Polymorphism and
Dynamic Binding 363
- 20.5** Maintenance Implications of Polymorphism
and Dynamic Binding 365
 - Key Terms 365
 - Review Questions for Chapter 20 365
 - Problems 366
 - Reference 366

Appendix A

Term Project: Chocoholics Anonymous 367

Appendix B

Object-Oriented Design: Osbert Oglesby Case Study 371

Appendix C

Object-Oriented Design: MSG Foundation Case Study 379

Index 385