国外优秀信息科学与技术系列教学用书

应用 Use Cases 方法

—— 实践指南

（第二版　影印版）

APPLYING USE CASES

A Practical Guide

(Second Edition)

■ Geri Schneider
Jason P. Winters

PEARSON
Addison
Wesley

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

应用 Use Cases 方法

——实践指南

（第二版　影印版）

APPLYING USE CASES

A Practical Guide

(Second Edition)

Geri Schneider
Jason P. Winters

高 等 教 育 出 版 社

图字：01-2003-1754 号

Applying Use Cases: A Practical Guide, Second Edition
Geri Schneider, Jason P. Winters

图书在版编目(CIP)数据

应用 Use Cases 方法：实践指南 = Applying Use Cases：A Practical Guide：第 2 版 /（美）施奈德（Schneider，G.），（美）温特斯（Winters，J. P.）著．—影印版．—北京：高等教育出版社，2003.8
ISBN 7 - 04 - 013500 - 0

Ⅰ.应...　Ⅱ.①施...②温...　Ⅲ.软件开发—英文
Ⅳ.TP311.52

中国版本图书馆 CIP 数据核字(2003)第 069931 号

出版发行	高等教育出版社	**购书热线**	010 - 64054588
社　　址	北京市西城区德外大街 4 号	**免费咨询**	800 - 810 - 0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010 - 82028899		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京中科印刷有限公司		
开　　本	787 × 1092　1/16	**版　　次**	2003 年 8 月第 1 版
印　　张	16.5	**印　　次**	2003 年 8 月第 1 次印刷
字　　数	320 000	**定　　价**	20.50 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。

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

Foreword

When I first proposed a new set of modeling concepts back in 1967 as the result of my work on large telecommunication switching systems and system design, the idea of use cases as a method of analysis was very sketchy. With the emergence of object-oriented ideas and my subsequent work in applying OO in the 1980s and formalizing the principles underlying Objectory, use case analysis began to take better shape and to play a significant role in the analysis of the problem domain. Today the ideas embodied in use cases have matured, and this technique has become a significant tool that belongs in every analyst's toolkit.

With the incorporation of use cases into the industry standard modeling language, UML, it is time for a new book that illustrates the current notation and semantics of use cases in a practical, easy-to-understand manner. Use case analysis also plays a central role in the new Unified Process for software development. It is, therefore, critical that managers, architects, designers, analysts, domain experts, programmers, and testers understand how to apply use cases.

In *Applying Use Cases,* Geri Schneider and Jason Winters have done an excellent job of introducing this powerful technique and demonstrating its application in real-world settings. Rather than making everything perfect up front, the examples progress in much the same manner you would find in a real project, with early rough models being refined as the team gains understanding of the project. This realism allows the introduction of issues that would arise in actual projects. *Applying Use Cases* is easy to read, but contains a wealth of detail.

This book clearly reflects Geri's experience as a trainer for Rational Software, the time she has spent mentoring and training customers of Wyyzzk Training and Consulting, and the time Jason has spent using the techniques and mentoring engineers at Lucent Technologies. It is an excellent resource for anyone who needs to understand use case analysis, and I recommend it highly.

Ivar Jacobson

Preface to the Second Edition

There have been many changes for us and for the UML since the first edition was released in September 1998. The book has changed to stay current.

The material in the first edition is also in the second edition, but you may find it in a new location. We moved the engineering-oriented material to the end of the book, and the business-oriented material to the beginning. This should make it easier for different audiences to find the material that interests them.

We updated the book to UML 1.3. A lot of the changes are in Chapters 3 and 4 because that is where we described most of the notation. The uses relationship became two relationships in UML 1.3, include and generalization. The extends relationship became extend. In both cases the notation changed as well. The definition of scenarios changed a bit too. What we used to call scenarios are now called paths.

We have added some new material that we found useful and important. Chapter 6 is a new chapter on setting the level of detail in use cases. This includes information on business process-level use cases and maintaining traceability between use cases at different levels of detail. Chapter 7, Documenting Use Cases, includes some ideas on handling login and CRUD (create, read, update, delete) in use cases. Chapter 8, Reviews, has a new section on common mistakes we have seen and how to fix them. We have included more information on sequence diagrams in Chapters 5 and 9.

There have been changes for me and Jason as well. Jason left Octel and is now a staff engineer at Cadence Design Systems. I liked having my own business, but didn't like the bookkeeping, so I took a job running the OO division of Andrews Technology, Inc. We still have Wyyzzk and Jason does some weekend consulting for that business. Things even changed on the publishing side. Addison-Wesley is now part of Pearson Education, and we have a whole new team managing the Object Technology series. They have been wonderful to work with and made the transition as smooth as possible.

One question we get asked a lot is: What do the footprints and people talking icons mean? The footprints mark major steps in the process. The people talking appear next to the storyline.

Thank you for all the e-mail about the book. We don't always get a chance to reply, but we have read all your letters and hope we have answered most of your questions in this second edition.

Many thanks to our distinguished reviewers. They worked as hard as we did to make this book happen:

- Lauren Thayer
- Venkat Narayanan
- Guy Rish
- Kelli A. Houston, Rational Software Corporation
- John Sunda Hsia

Speaking of hard workers, we were most fortunate to be working with Paul Becker, Ross Venables, Tyrrell Albaugh, and her production team at Addison-Wesley. Our most heartfelt thanks for all your support and patience. You guys did all the tough work to make this book a reality.

Finally, a few very special people. Thanks to Lauren Thayer and Kristy Hughes for being my constant friends for almost 15 years. And my cats, Patches and Joker, for keeping me company all those hours on the computer. And as always, thank you to Jason Winters for his love, support, and encouragement. You all are the wind beneath my wings.

Geri Schneider Winters
Santa Clara, California

Preface

You're about to start a new project. Sometimes it seems like colonizing the moon would be easier. But you assemble a stalwart team and prepare to set sail on the good ship *Requirements,* hoping to reach the fabled land of Success. They say there are no failed projects in Success, and the profit margin is so high, the streets are paved with gold.

There are many dangers between here and Success. Many a ship is sunk on the way—some say as many as 80 percent never reach that fabled land. You query those who have tried before. "Use a ship from the OO line," they say. "Booch, OMT, OOSE, UML are all good models to choose from. You'll also need a chart showing risks along the way and an architecture of the major land masses. And finally you'll need to plot a course of use cases to reach your destination."

Use Cases are included in the Unified Modeling Language and are used throughout the Rational Unified Process. They are gaining wide acceptance in many different businesses and industries. Most often, use cases are applied to software projects and enterprise-wide applications.

This book is for anyone interested in applying use cases to project development. While we can't guarantee you will always have successful projects when using use cases, we can give you another way of looking at the projects you are developing and some tools that will make success more likely.

You will get more benefit out of the book if you have some basic knowledge of object-oriented concepts. We will use the Unified Modeling Language for the notation, explaining the notation as we use it. A good book to use for reference on the notation is *UML Distilled* by Fowler. This is an excellent book on the topic and easy to read.

This book is organized using the Rational Unified Process as a framework. Within the phases of the process, we talk about the activities in the phase, focusing on activities based on use cases. We touch lightly on activities that interact with use cases, such as software architecture, project management, and object-oriented analysis and design. These are very important activities, with whole books devoted to each topic. Therefore, in the resource list in Appendix A, you will find our favorite books on these topics.

We have used one example, an order-processing system for a mail order company, throughout the book. This allows us to maintain consistency and build up a reasonably complex example. Parts of the solution are given in the various chapters to illustrate the concepts.

This book is presented as a sequence of steps, though life is never that simple. Each part will contribute to the rest until the system is complete. So if a section says to create an architecture, do what you can at that time, using what you currently know. You will add to it and refine it based on knowledge gained while working through the process.

You don't have to read the whole book before starting with use cases. Chapters 1 through 6 give the basics of working with use cases. We recommend that everyone reads those chapters. Chapter 9 covers architecture and mapping use cases into the architecture. Chapter 7 covers documenting use cases. Chapter 10 covers project planning with use cases, and Chapter 8 covers reviewing the use case documents. Chapter 11 goes into moving from use cases to OOAD.

Ultimately, use cases are about documenting your system. Plan on doing a lot of writing. Appendix A provides a list of books we reference throughout the text, as well as other books we have found useful when developing projects. Appendix B shows the document templates used. These provide an example and a starting point for your own project. Modify them as needed to work with your project.

△ ▽ △ ▽ △

In October of 1995, Rational Software Corporation merged with Objective Systems. Among other things, this merger brought with it Ivar Jacobson and his use cases. In February 1996, I wrote and delivered the first use case course for Rational, which combined use cases with the object-oriented methodologies of Grady Booch and Jim Rumbaugh. Since that time, I have taught and run workshops on use cases with many of Rational's customers, as well as customers of my consulting company, Wyyzzk Training and Consulting. As I have taught them, so they have taught me. This book came out of what I've learned through the workshops.

Contents

Foreword *xi*

Preface to Second Edition *xiii*

Preface *xv*

Chapter 1 1

Getting Started

An Iterative Software Process 2
An Example Project 3
The Project Description 4
Starting Risk Analysis 6
Chapter Review 10

Chapter 2 11

Identifying System Boundaries

Identifying Actors 12
Identifying Use Cases 14
Describing Actors and Use Cases 17
Handling Time 21
Potential Boundary Problems 22
Scoping the Project 23
Chapter Review 24

Chapter 3 27

Documenting Use Cases

The Basic Use Case 27
Pre- and Postconditions *28*
Flow of Events *29*

Guidelines for Correctness and Completeness 31
Presentation Styles 32
Other Requirements 34
Handling Complex Use Cases 34
The Basic Path 35
Alternative Paths 37
Detailing Significant Behavior 40
Documenting Alternatives 42
Scenarios 47
Adding Direction to the Communicates Association 47
Chapter Review 48

Chapter 4 51

Advanced Use Case Documentation Techniques

Include 51
Extend 53
Inheritance 58
Interfaces 59
Chapter Review 65

Chapter 5 67

Diagramming Use Cases

Activity Diagrams 67
Simple Sequence Diagrams 73
Diagramming the User Interface 75
Chapter Review 77

Chapter 6 79

Level of Detail

Determining the Level of Detail 79
Traceability between Use Cases 84
Use Cases for Business Processes 85
Chapter Review 87

Chapter 7 89

Documenting Use Cases

Documentation Templates 89
Other Documents 91
Tool Support for Documents 94
Documenting Login 95
Documenting CRUD 98
Chapter Review 99

Chapter 8 101

Reviews

Review for Completeness 101
Review for Potential Problems 103
Review with End Users 103
Review with Customers 104
Review with Development 104
Reviewers 104
Adding Flexibility to Your System 105
Common Mistakes 107
Work Flow on a Use Case Diagram *107*
Use Cases Too Small *108*
Screens as Use Cases *112*
Using Vague Terms *115*
Business versus Technical Requirements *120*
Chapter Review 122

Chapter 9 123

Dividing Large Systems

Architectural Patterns 123
Three-Tier Architectural Pattern *124*
Pipe and Filter Architectural Pattern *125*
Object-Oriented Architectural Pattern *126*
Order-Processing Architecture Example *126*
Testing the Architecture with Use Cases 129
Sequence Diagrams 133
Defining Interfaces between Subsystems 133
Subordinate Use Cases 136

Creating Subsystem Documentation 140
Subordinate versus Alternative versus Include 141
Chapter Review 142

Chapter 10 143

Use Cases and the Project Plan

Planning the Project 143
Build versus Buy Decisions 149
Prototyping 150
Estimating Work with Use Cases 151
Weighting Actors 151
Weighting Use Cases 152
Weighting Technical Factors 153
Use Case Points 157
Project Estimate 157
Chapter Review 158

Chapter 11 159

Constructing and Delivering a System

Key Abstractions of the Domain 159
Identifying Key Abstractions in Use Cases 160
Diagramming Scenarios with Key Abstractions 161
Diagramming Key Abstractions 163
Use Case versus Subsystem View 164
The Iteration Schedule 166
Delivery and Beyond 167
User Guides and Training 168
Sales Kits and Marketing Literature 168
Use Cases After Delivery 168
Chapter Review 169
Final Wrap-Up 170

Appendix A 171

Resources

Appendix B 175

Documentation Templates

System or Subsystem Documents 175
Use Case Document 176

Appendix C 179

UML Notation

Appendix D 185

Sending Results of the Use Case Estimator

Appendix E 187

Order-Processing System

Order-Processing System 188
Risk Factors 188
System-Level Use Cases 189
Architecture 190

Index 239

Chapter 1

Getting Started

Use cases are used to describe the outwardly visible requirements of a system. They are used in the requirements analysis phase of a project and contribute to test plans and user guides. They are used to create and validate a proposed design and to ensure it meets all requirements. Use cases also are used when creating a project schedule, helping to plan what goes into each release.

This book will give practical guidelines for applying use cases to a project. We will cover a project from its initial inception ("Hey! How about. . . .") to just before we actually start to build a system. We also will look at applying use cases in testing the system code and creating user manuals.

In this book we'll look at use cases from many viewpoints, showing how they contribute to the architecture, scheduling, requirements, testing, and documentation of a project. We'll look at the system from the user's point of view, discuss issues such as boundaries, interfaces, and scoping, and look at how to break a really large system into manageable chunks. We also will look at who would be interested in the documentation you'll be writing and what to look for in a review. We need to consider things such as how to build flexibility into a system, how to make a build-versus-buy decision, and how to turn the documents into an object-oriented design.

This book does not contain in-depth details about software architecture, project planning, testing, process, or methodology. Instead, you will find a listing of books we like on these topics in Resources (Appendix A). There are a number of good books on these topics; the resource list gives you just a starting point.

AN ITERATIVE SOFTWARE PROCESS

Use cases can be used in many processes. Our favorite is a process that is iterative and risk driven. It works well with use cases and object-oriented methodologies. It helps identify and address risks early in the process, leading to more robust and better quality systems. One commonly used iterative and risk driven process is the Rational Unified Process (RUP). We will give a very brief description of RUP here, showing where use cases fit into the process. Subsequent chapters will go into more detail on how use cases are used at each phase.

RUP is divided into four primary phases: inception, elaboration, construction, and transition.

During the inception phase you will determine the scope of the project and create a business case for it. At the end of the inception phase you should be able to answer the question, Does it make good business sense for us to continue with this project?

During the elaboration phase you will do requirements analysis and risk analysis, develop a baseline architecture, and create a plan for the construction phase.

During the construction phase you will progress through a series of iterations. Each iteration will include analysis, design, implementation, and testing.

During the transition phase you will complete the things that make what you developed into a product. These can include beta testing, performance tuning, and creating additional documentation such as training, user guides, and sales kits. You will create a plan for rolling out the product to the user community, whether internal or external.

So where do use cases fit into all this? In the inception phase, high-level use cases are developed to help scope out the project: What should be included in this project, and what belongs to another project? What can you realistically accomplish given your schedule and budget?

In the elaboration phase, you will develop more detailed use cases. These will contribute to the risk analysis and the baseline architecture. The use cases will be used to create the plan for the construction phase.

In the construction phase, you will use use cases as a starting point for design and for developing test plans. More detailed use cases may be developed as part of the analysis of each iteration. Use cases provide some of the requirements that have to be satisfied for each iteration.

In the transition phase, you will use use cases to develop user guides and training.