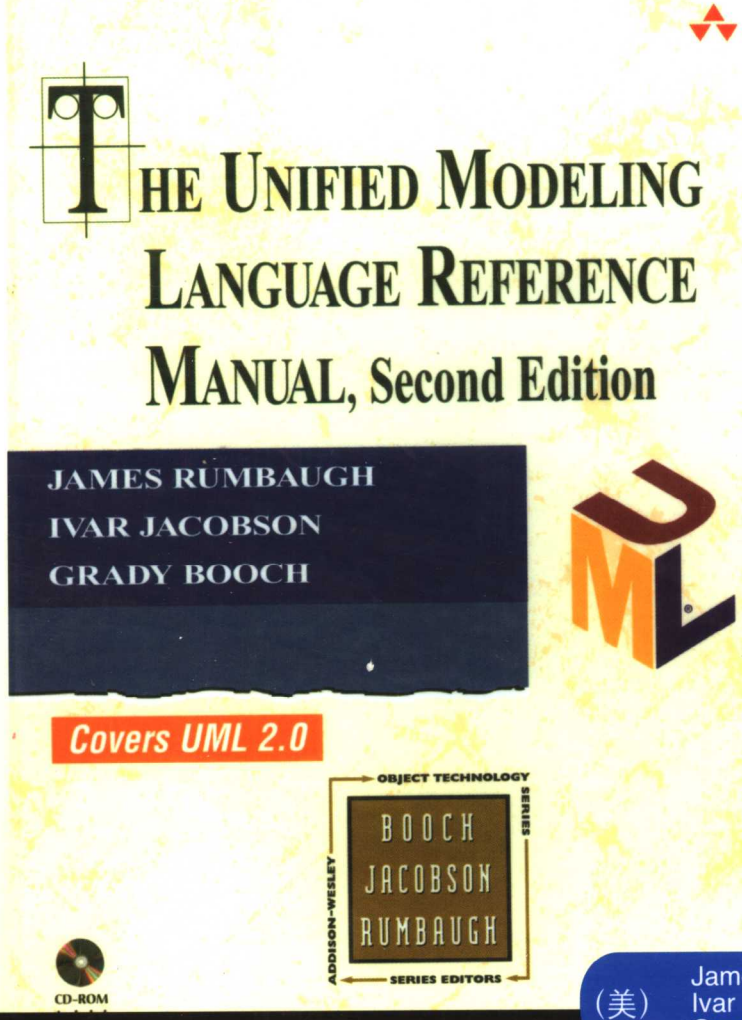
HZ BOOKS

经 典 原 版 书 库

PEARSON Addison Wesley

UML参考手册

（英文版 · 第2版）



附赠 CD-ROM

(美) James Rumbaugh
Ivar Jacobson
Grady Booch 著

机械工业出版社
China Machine Press

经 典 原 版 书 库

UML参考手册

(英文版·第2版)

The Unified Modeling Language Reference Manual

(Second Edition)

(美) James Rumbaugh
Ivar Jacobson
Grady Booch 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2005-4392

图书在版编目（CIP）数据

UML参考手册（英文版·第2版）/（美）朗包（Rumbaugh, J.）等著. -北京：机械工业出版社，2006.1

（经典原版书库）

书名原文：The Unified Modeling Language Reference Manual, Second Edition

ISBN 7-111-17271-X

Ⅰ.U… Ⅱ.朗… Ⅲ.面向对象语言，UML-程序设计-技术手册-英文 Ⅳ.TP312-62

中国版本图书馆CIP数据核字（2005）第100795号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：迟振春

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2006年1月第1版第1次印刷

787mm × 1092mm 1/16 · 46.25印张

印数：0 001-3 000册

定价：75.00元（附光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

Advanced Praise for *The Unified Modeling Language Reference Manual, Second Edition*

"If you are a serious user of UML, there is no other book quite like this one. I have been involved with the UML specification process for some time, but I still found myself learning things while reading through this book—especially on the changes and new capabilities that have come with UML 2.0. The intimate involvement of the author in the creation and continuing evolution of UML, and the encyclopedic scope of his book, make the work a unique contribution to the UML 2.0 literature, as the first edition was for UML 1.0."

—Ed Seidewitz, Chief Architect, InteliData Technologies Corporation

"In addition to the documents of the OMG UML 2.0 Standard, this book is probably the most important source for the Unified Modeling Language. It is a detailed reference, covering the mainstream ideas as well as the delicate niches of the language. The Dictionary of Terms offers precise, comprehensive and, perhaps most important, systematic information on all aspects of the UML 2.0."

—Martin Gogolla, Professor for Computer Science, University of Bremen

"Comprehensive and instructive, written by a person with the insights of not only the technical matters, but also the processes that led to the UML language and its version 2.0. This book should be a companion for every serious UML modeler."

—Øystein Haugen, Ericsson Representative in the OMG UML 2.0 Standardization, Associate Professor, University of Oslo

"This book provides an authoritative and user-oriented account of UML 2.0."

—Dr. Robert France, Department of Computer Science, Colorado State University.

"This is so far the most comprehensive book on UML 2.0. It gives you what the specification does not: real introductions to the various parts of UML, annotated examples, discussions on how to use the new features, and an insight into how and why the new features entered UML 2.0. As one of the persons who was involved in the making of UML 2.0, I can tell that the book is faithful to the specification and to the ideas behind the new features. Read this book instead or as a complement to the specification."

—Birger Møller-Pedersen, Professor, University of Oslo

Preface

Goals

This book is intended to be a complete, useful reference to the Unified Modeling Language (UML) for the developer, architect, project manager, system engineer, programmer, analyst, contracting officer, customer, and anyone else who needs to specify, design, build, or understand complex software systems. It provides a full reference to the concepts and constructs of UML, including their semantics, notation, and purpose. It is organized to be a convenient but thorough reference for the working professional developer. It also attempts to provide additional detail about issues that may not be clear from the standards documents and to provide a rationale for many decisions that went into the UML.

This book is not intended as a guide to the UML standards documents or to the internal structure of the metamodel contained in them. The details of the metamodel are of interest to methodologists and UML tool builders, but most other developers have little need for the arcane details of the Object Management Group (OMG) documents. This book provides all the details of UML that most developers need; in many cases, it makes information explicit that must otherwise be sought between the lines of the original documents. For those who wish to consult the source documents, they are on the OMG web site (www.omg.org).

This book is intended as a reference for those who already have some understanding of object-oriented technology. For beginners, the original books by us and by other authors are listed in the bibliography; although some of the notation has changed, books such as [Rumbaugh-91], [Jacobson-92], [Booch-94], and [Meyer-88] provide an introduction to object-oriented concepts that is still valid and therefore unnecessary to duplicate here. [Blaha-05] updates [Rumbaugh-91] using UML notation. For a tutorial introduction to UML that shows how to model a number of common problems, see *The Unified Modeling Language User Guide* [Booch-99] or *UML Distilled* [Fowler-04].

UML does not require a particular development process. Although UML may be used with a variety of development processes, it was designed to support an iterative, incremental, use-case–driven process with a strong architectural focus—the kind we feel is most suitable for the development of modern, complex systems. To place UML in its context as a tool for software development, this book defines the stages of such a process, but they are not part of the UML standard. *The Unified Software Development Process* [Jacobson-99] describes in detail the kind of process we believe complements the UML and best supports software development.

Second Edition and UML Version

This second edition has been extensively modified from the first edition, which was published in 1999. This edition is based on the OMG "adopted" specification of UML version 2.0, with anticipated changes to the "available" specification being prepared by an OMG Finalization Task Force. Corrections to the book due to changes in the OMG UML specification will be posted on the publisher's web site for this book at www.awprofessional.com/titles/0321245628. The information in the book is accurate as of June 2004.

Original specification documents and up-to-date information about work on UML and related topics can be found on the OMG web site at www.omg.org.

Reference Manual and OMG Specification

UML is a large modeling language with many features. A reference manual that just repeats the original specification documents would not help readers much. As in any dictionary or encyclopedia, we have had to summarize information as clearly as possible while reducing the amount of material included. We have frequently chosen to emphasize common usages by omitting obscure special cases or redundant means of representing some concepts. This does not mean that those capabilities are useless, but most readers should be able to succeed without using them. The *Reference Manual* should not be regarded as the final authority on the UML language, however. As with any standard, the final authority rests with the official specifications, and these should be consulted to resolve disputes.

We have tried to follow these principles:

- Explain the main intent of a concept without getting lost in the details of the metamodel representation.
- Avoid discussion of abstract metaclasses. Modelers must ultimately use concrete metaclasses, which can be described more simply if the internal abstract layers are collapsed.
- Avoid discussion of the packaging of the metamodel. The packages may be important to tool builders, but modelers don't need to know them most of the time. If you need to know, you need to look at the specification in detail anyway.

- Describe concepts from the complete specification. The OMG specification has a number of intermediate layers and compliance points that greatly complicate understanding of UML. We describe UML with all of its features. If your tool does not implement all of the facilities, then some of the features may be unavailable to you, but it doesn't usually hurt to know about them.
- Describe concepts from the viewpoint of their normal usage. Often the OMG specification goes to considerable trouble to express concepts in a general way. This is proper for a specification, but we feel that readers often understand concepts better if they are presented in a specific context and then generalized. If you are worried about the application of a concept in a complex, ambiguous situation and you feel that the *Reference Manual* explanation may be inadequate, check the original specification. Unfortunately, however, even the OMG specification is sometimes ambiguous in complex situations.

Outline of the Book

The UML Reference Manual is organized into four parts: (1) an overview of UML history and of modeling, (2) a survey of UML concepts, (3) an alphabetical dictionary of UML terms and concepts, and (4) brief appendices.

The first part is an overview of UML—its history, purposes, and uses—to help you understand the origin of UML and the need it tries to fill.

The second part is a brief survey of UML concepts so that you can put all the features into perspective. The survey provides a brief overview of the views UML supports and shows how the various constructs work together. This part uses an example that walks through various UML views. It contains one chapter for each kind of UML view. This survey is not intended as a full tutorial or as a comprehensive description of concepts. It serves mainly to summarize and relate the various UML concepts, providing starting points for detailed readings in the dictionary.

The third part contains the reference material organized for easy access to each topic. The bulk of the book is an alphabetical dictionary of all of the concepts and constructs in UML. Each UML term of any importance has its own entry in the dictionary. The dictionary is meant to be complete; therefore, everything in the concept overview in Part 2 is repeated in more detail in the dictionary. The same or similar information has sometimes been repeated in multiple dictionary articles so that the reader can conveniently find it. Some common object-oriented terms that are not official UML concepts are included to provide context in examples and discussions.

Appendices show the UML metamodel and a summary of UML notation. There is a brief bibliography of major object-oriented books, but no attempt has been made to include a comprehensive citation of sources of ideas for UML or other approaches. Many of the books in the bibliography contain excellent lists of references to books and journal articles for those interested in tracking the development of the ideas.

Dictionary Entry Formatting Conventions

The dictionary part of the book is organized as an alphabetical list of entries, each describing one concept in some detail. The articles represent a flat list of UML concepts at various conceptual levels. A high-level concept typically contains a summary of its subordinate concepts, each of which is fully described in a separate article. The articles are highly cross-referenced. The flat dictionary organization permits the description of each concept to be presented at a fairly uniform level of detail, without constant shifts in level for the nested descriptions that would be necessary for a sequential presentation. The hypertext format of the document should also make it convenient for reference. It should not be necessary to use the index much; instead, go directly to the main article in the dictionary for any term of interest and follow cross-references. This format is not necessarily ideal for learning the language; beginners are advised to read the overview description of UML found in Part 2 or to read introductory books on UML, such as the *UML User Guide* [Booch-99].

Dictionary articles have the following divisions, although not all divisions appear in all articles.

Headword and brief definition

The name of the concept appears in boldface, set to the left of the body of the article. A brief definition follows in normal type. This definition is intended to capture the main idea of the concept, but it may simplify the concept for concise presentation. Refer to the main article for precise semantics.

Predefined stereotypes are included as entries. A comment in parentheses following the entry name identifies the modeling element to which they apply.

Semantics

This section contains a detailed description of the meaning of the concept, including constraints on its uses and its execution consequences. Notation is not covered in this section, although examples use the appropriate notation. General semantics are given first. For concepts with subordinate structural properties, a list of the properties follows the general semantics, often under the subheading *Structure*. In most cases, properties appear as a table in alphabetical order by property name, with the description of each property on the right. If a property has an enumerated list of choices, they may be given as an indented sublist. In more complicated cases, a property is given its own article to avoid excessive nesting. When properties require more explanation than permitted by a table, they are described in normal text with run-in headers in boldface italics. In certain cases, the main concept is best described under several logical subdivisions rather than one list. In such cases, additional sections follow or replace the *Structure* subsection. Although several

entry name **Dictionary entry format**

A brief description of the concept in one or two sentences.
See also related concept.

Semantics

A description of the semantics in several paragraphs.

Structure

A list of the subordinate concepts within the main concept.

item	Description of an item. UML metamodel names are usually converted into plain English.
enumerated item	An enumeration with several values. List of values:
value	The meaning of this value of the item.

Another item. More complicated topics are described in separate paragraphs.

Example

An example may be included in semantics, notation, or stand alone.

Notation

Description of the notation, usually including a diagram or syntax.

Presentation options

Describes variant forms of notation, usually optional.

Style guidelines

States recommended practice although not mandatory.

Discussion

The author's opinions or background explanations beyond UML.

History

Changes from UML version 1.x.

stereotype entry (stereotype of Class)

Description of the meaning of the stereotype.

organizational mechanisms have been used, their structure should be obvious to the reader. The names of properties are usually stated in plain language rather than using internal identifiers from the UML metamodel, but the correspondence is meant to be obvious.

Notation

This section contains a detailed description of the notation for the concept. Usually, the notation section has a form that parallels the preceding semantics section, which it references, and it often has the same divisions. The notation section usually includes one or more diagrams to illustrate the concept. The actual notation is printed in black. To help the reader understand the notation, many diagrams contain annotations in blue. Any material in blue is commentary and is not part of the actual notation.

Style guidelines

This optional section describes widespread style conventions. They are not mandatory, but they are followed within the UML specification itself. Recommended presentation guidelines may also be given in a separate section.

Example

This subsection contains examples of notation or illustrations of the use of the concept. Frequently, the examples also treat complicated or potentially confusing situations. If the examples are brief, they may be folded in with another section.

Discussion

This section describes subtle issues, clarifies tricky and frequently confused points, and contains other details that would otherwise digress from the more descriptive semantics section. A minority of articles have a discussion section.

This section also explains certain design decisions that were made in the development of the UML, particularly those that may appear counterintuitive or that have provoked strong controversy. Simple differences in taste are generally not covered.

Sometimes we express an opinion on the value (or lack thereof) of certain concepts. We recognize that others may disagree with these assessments. We have tried to confine opinions to the discussion section.

History

This section describes changes from UML1 to UML2, sometimes including reasons for the changes. Minor changes are not usually listed. Absence of this section does not mean that no changes have occurred.

Syntax Conventions

Syntax expressions. Syntax expressions are given in a modified BNF format in a sans serif font (Myriad). Literal values that appear in the target sentence are printed in black, and the names of syntax variables and special syntax operators are printed in blue.

Text printed in black appears literally in the target string.

Punctuation marks (always printed in black) appear literally in the target string.

Any word printed in blue ink represents a variable that must be replaced by another string or another syntax production in the target string. Words may contain letters and hyphens. If a blue word is italicized or underlined, the actual replacement string must be italicized or underlined.

In code examples, comments are printed in blue to the right of the code text.

Subscripts and L-brackets are used as syntax operators as follows:

$\text{expression}_{\text{opt}}$	The expression is optional.
$\text{expression}_{\text{list,}}$	A comma-separated list of the expression may appear. If there is zero or one repetition, there is no separator. If a different punctuation mark than a comma appears in the subscript, then it is the separator.
$\lfloor = \text{expression} \rfloor_{\text{opt}}$	A pair of right angles ties together two or more terms that are considered a unit for optional or repeated occurrences. In this example, the equal sign and the expression form one unit that may be omitted or included.

Two-level nesting is avoided. Particularly convoluted syntax may be simplified somewhat for presentation, but use of such convoluted syntax is likely to be confusing for humans anyway and should be avoided.

Literal strings. In running text, language keywords, names of model elements, and sample strings from models are shown in a sans serif font (Myriad).

Diagrams. In diagrams, blue text and arrows are annotations, that is, explanations of the diagram notation that do not appear in an actual diagram. Any text and symbols in black are actual diagram notation.

CD

This book is accompanied by a CD containing the full text of the book in Adobe® Reader® (PDF) format. Using Adobe Reader, the viewer can easily search the book for a word or phrase. The CD version also contains a clickable table of contents, index, Adobe Reader bookmarks, and extensive hot links (in red) in the bodies of the articles. Simply click on one of the links to jump to the dictionary article for a word or phrase. We hope that this CD will be a useful reference aid for readers.

Creators of UML

We wish to thank the many collaborators who built the UML specification through years of meetings, heated discussions, writing, and implementation of ideas. The list of contributors has grown significantly since UML1, and the OMG specification no longer lists the major contributors, who number between twenty and fifty depending on the threshold for inclusion, and even more if work influencing UML is included. It no longer appears possible to compile a complete list without overlooking many persons.

Most of all, we want to celebrate the hundreds of persons who contributed to the community of ideas from which UML was drawn—ideas in object-oriented technology, software methodology, programming languages, user interfaces, visual programming, and numerous other areas of computer science. It is impossible to list them all, or indeed to track even the major chains of influence, without a major scholarly effort, and this is an engineering book, not a historical review. Many of them are well known, but many good ideas also came from those who did not have the good fortune to become widely recognized. The bibliography includes a few of the lesser-known books that influenced the authors.

Acknowledgments

We would like to acknowledge the reviewers who made this book possible. For the second edition, they include Conrad Bock, Martin Gogolla, Øystein Haugen, Birger Møller-Pedersen, and Ed Seidewitz. For the first edition, we received feedback from Mike Blaha, Conrad Bock, Perry Cole, Bruce Douglass, Martin Fowler, Eran Gery, Pete McBreen, Gunnar Övergaard, Karin Palmkvist, Guus Ramackers, Tom Schultz, Ed Seidewitz, and Bran Selic.

On a more personal note, I wish to thank Professor Jack Dennis, who inspired my work in modeling and the work of many other students more than thirty years ago. The ideas from his Computations Structures Group at MIT have borne much fruit, and they are not the least of the sources of UML. I must also thank Mary Loomis and Ashwin Shah, with whom I developed the original ideas of OMT, and my former colleagues at GE R&D Center, Mike Blaha, Bill Premerlani, Fred Eddy, and Bill Lorensen, with whom I wrote the OMT book.

Finally, without the patience of my wife, Madeline, and my sons, Nick and Alex, there would have been no UML and no book about it.

James Rumbaugh
Cupertino, California
June 2004

Contents

Preface vii

Part 1: Background

Chapter 1: UML Overview **3**

Brief Summary of UML *3*
UML History *4*
Goals of UML *10*
Complexity of UML *10*
UML Assessment *12*
UML Concept Areas *12*

Chapter 2: The Nature and Purpose of Models **15**

What Is a Model? *15*
What Are Models For? *15*
Levels of Models *17*
What Is in a Model? *19*
What Does a Model Mean? *21*

Part 2: UML Concepts

Chapter 3: UML Walkthrough **25**

UML Views *25*
Static View *28*
Design Views *29*
Use Case View *34*
State Machine View *35*
Activity View *37*
Interaction View *37*
Deployment View *40*
Model Management View *42*
Profiles *43*

Chapter 4: Static View **47**
Overview 47
Classifier 48
Relationships 52
Association 53
Generalization 57
Realization 61
Dependency 62
Constraint 65
Instance 66

Chapter 5: Design View **69**
Overview 69
Structured Classifier 70
Collaboration 71
Patterns 73
Component 73

Chapter 6: Use Case View **77**
Overview 77
Actor 77
Use Case 78

Chapter 7: State Machine View **81**
Overview 81
State Machine 81
Event 82
State 84
Transition 85
Composite State 89

Chapter 8: Activity View **95**
Overview 95
Activity 96
Activities and Other Views 98
Action 98

Chapter 9: Interaction View **101**
Overview 101
Interaction 101
Sequence Diagram 102
Communication Diagram 106