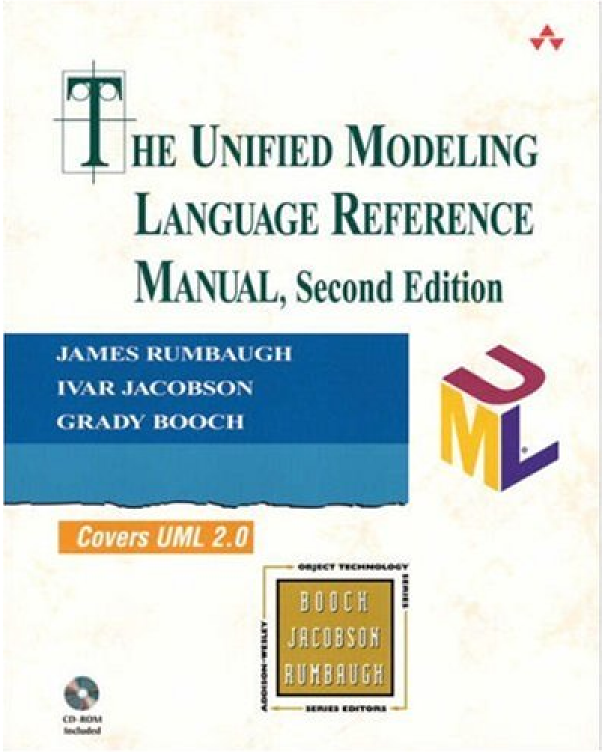
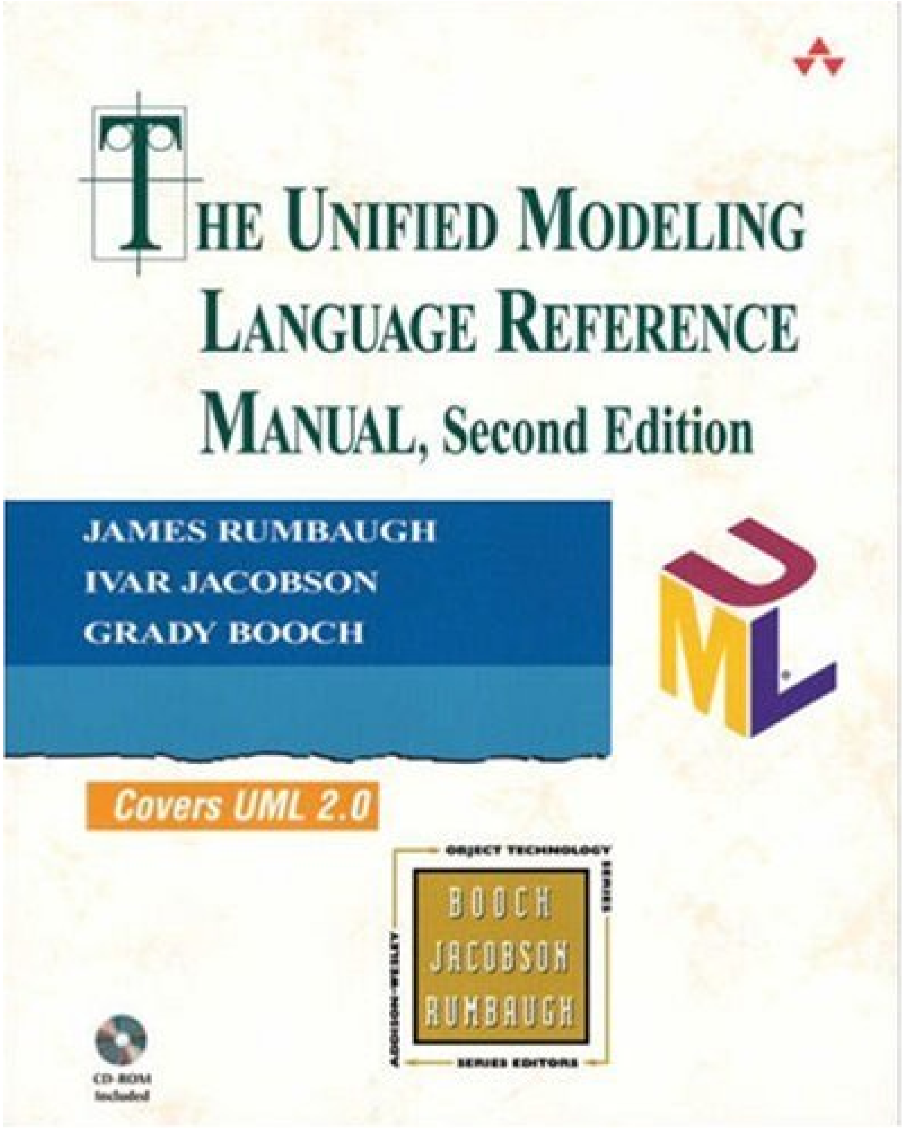
THE UNIFIED MODELING LANGUAGE REFERENCE MANUAL (2ND EDITION) (THE ADDISON-WESLEY OBJECT TECHNOLOGY SERIES) BY JAMES RUMBAUGH, IVAR JACOBS



DOWNLOAD EBOOK : THE UNIFIED MODELING LANGUAGE REFERENCE MANUAL (2ND EDITION) (THE ADDISON-WESLEY OBJECT TECHNOLOGY SERIES) BY JAMES RUMBAUGH, IVAR JACOBS PDF



THE UNIFIED MODELING LANGUAGE REFERENCE MANUAL (2ND EDITION) (THE ADDISON-WESLEY OBJECT TECHNOLOGY SERIES) BY JAMES RUMBAUGH, IVAR JACOBS PDF

Guide The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs will still give you positive value if you do it well. Completing the book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs to read will certainly not come to be the only objective. The objective is by obtaining the favorable worth from guide until the end of the book. This is why; you need to find out even more while reading this The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs This is not just how fast you check out a book and not just has the amount of you completed the books; it has to do with just what you have actually obtained from guides.

Amazon.com Review
Written by the three pioneers behind the Unified Modeling Language (UML) standard, The Unified Modeling Language Reference Manual provides an excellent real-world guide to working with UML. This title provides expert knowledge on all facets of today's UML standard, helping developers who are encountering UML on the job for the first time to be more productive.

The book begins with a history of UML, from structured design methods of the '60s and '70s to the competing object-oriented design standards that were unified in 1997 to create UML. For the novice, the authors illustrate key diagram types such as class, use case, state machine, activity, and implementation. (Of course, learning these basic diagram types is what UML is all about. The authors use an easy-to-understand ticket-booking system for many of their examples.)

After a tour of basic document types, The Unified Modeling Language Reference Manual provides an alphabetical listing of more than 350 UML terms. Entries range from a sentence or two to several pages in length. (Class, operation, and use case are just a few of the important terms that are covered.) Though you will certainly need to be acquainted with software engineering principles, this reference will serve the working software developer well. As the authors note, this isn't UML for Dummies, but neither is it an arcane academic treatise. The authors succeed in delivering a readable reference that will answer any UML question, no matter how common or obscure. --Richard Dragan

From the Inside Flap
Goals

This book is intended to be a complete and 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, syntax, 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 standard 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 do wish to consult the source documents, they are included on the accompanying CD.

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, Booch-94, Jacobson-92, and Meyer-88 provide an introduction to object-oriented concepts that is still valid and therefore unnecessary to duplicate here. 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. Those who already know an object-oriented method, such as OMT, Booch, Objectory, Coad-Yourdon, or Fusion, should be able to read the Reference Manual and use it to understand UML notation and semantics; to learn UML quickly, they may nevertheless find it useful to read the User Guide.

UML does not require a particular development process, and this book does not describe one. Although UML may be used with a variety of development processes, it was designed to support an iterative, incremental, use-casendriven process with a strong architectural focus--the kind we feel is most suitable for the development of modern, complex systems. The Unified Software Development Process Jacobson-99 describes the kind of process we believe complements the UML and best supports software development. Outline of the Book

The UML Reference Manual is organized into three parts: an overview of UML history and of modeling, a survey of UML concepts, and an alphabetical encyclopedia of UML terms and concepts.

The first part is a survey 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 views so that you can put all the concepts into perspective. The survey provides a brief overview of the views UML supports and shows how the various constructs work together. This part begins with an example that walks through various UML views and then 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 and provides starting points for detailed readings in the encyclopedia.

The third part contains the reference material organized for easy access to each topic. The bulk of the book is an alphabetical encyclopedia of all of the concepts and constructs in UML. Each UML term of any importance has its own entry in the encyclopedia. The encyclopedia is meant to be complete; therefore, everything in the concept overview in Part 2 is repeated in more detail in the encyclopedia. The same or similar information has sometimes been included in multiple encyclopedia articles so that the reader can conveniently find it.

The reference part also contains an alphabetic list of UML standard elements. A standard element is a feature predefined using the UML extensibility mechanisms. The standard elements are extensions that are felt to be widely useful.

Appendices show the UML metamodel, a summary of UML notation, and some standard sets of extensions for particular domains. 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. Encyclopedia Article Formatting Conventions

The encyclopedia 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. This flat encyclopedia 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 encyclopedia 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.

Encyclopedic articles have the following divisions, although not all divisions appear in all articles. 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. 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, the properties appear as a table in alphabetical order by property name, with the description of each property on the right. If a property has a brief enumerated list of choices, they may be given as an indented sublist. In more complicated cases, the 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 organizational mechanisms have been used, their structure should be obvious to the reader. 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 ink. To help the reader understand the notation, many diagrams contain annotations in blue ink. Any material in blue is commentary and is not part of the actual notation. 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. 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. Only a fraction of articles have this section. Simple differences in taste are generally not covered. Standard elements

This section lists standard constraints, tags, stereotypes, and other conventions that are predefined for the concept in the article. This section is fairly rare. Syntax Conventions

Syntax expressions. Syntax expressions are given in a modified BNF format in a sans serif font. To avoid confusing literal values and syntax productions, literal values that appear in the target sentence are printed in black ink, and the names of syntax variables and special syntax operators are printed in blue ink.

Text printed in black ink appears in that form in the target string.

Punctuation marks (they are always printed in black) appear 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 ink to the right of the code text. Subscripts and overbars are used as syntax operators (please refer to book for further information).

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

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 ink 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 thumbnails, and extensive hot links in the body of the articles. Simply click on one of the links to jump to the encyclopedia article for the word or phrase.

The CD also contains the full text of the OMG UML specifications, included by the permission of the Object Management Group.

We feel that this CD will be a useful on-line reference to UML for advanced users. For More Information

Additional source files and up-to-date information on further work on UML and related topics can be found on the World Wide Web sites rational and omg. Acknowledgments

We want to thank many people who made the UML possible. First, we must thank Rational Software Corporation, especially Mike Devlin and Paul Levy, who had the vision to bring us together, start the unification work, and stay the course during the four years that were required to bring the work to successful completion. We also thank the Object Management Group for providing the framework that brought together many diverse viewpoints and merged them together into a broad consensus that was much greater than any one contribution.

We particularly want to thank Cris Kobryn, who led the technical team that prepared the UML standard and who managed to achieve a consensus among an extremely strong-willed group of persons (and the three of us were not the least of his problems). His diplomatic skills and technical balance kept the UML effort from foundering amid many differences of opinion. Cris also reviewed the book and provided countless useful suggestions.

We would like to thank Gunnar &#214vergaard for reviewing the book thoroughly, as well as for his perseverance in completing many sections of the UML documents that were not fun to write but were necessary to its formal correctness.

We want to thank Karin Palmkvist for an exceedingly thorough review that uncovered many bugs in technical content, as well as many flaws in grammar, phrasing, and presentation.

We would also like to thank Mike Blaha, Conrad Bock, Perry Cole, Bruce Douglass, Martin Fowler, Eran Gery, Pete McBreen, Guus Ramackers, Tom Schultz, Ed Seidewitz, and Bran Selic for their helpful reviews.

Most of all, we want to thank the scores or even 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 are well known, but many good ideas came from those who did not have the good fortune to be widely recognized.

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 twenty-five 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&#38D 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
November 1998 020130998XP04062001

From the Back Cover

“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.”

—Ed Seidewitz, Chief Architect, IntelliData Technologies Corporation

The latest version of the Unified Modeling Language—UML 2.0—has increased its capabilities as the standard notation for modeling software-intensive systems. Like most standards documents, however, the official UML specification is difficult to read and navigate. In addition, UML 2.0 is far more complex than previous versions, making a thorough reference book more essential than ever.

In this significantly updated and expanded edition of the definitive reference to the standard, James Rumbaugh, Ivar Jacobson, and Grady Booch—the UML’s creators—clearly and completely describe UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Whether you are capturing requirements, developing software architectures, designing implementations, or trying to understand existing systems, this is the book for you.

Highlights include:

- Alphabetical dictionary of articles covering every UML concept
- Integrated summary of UML concepts by diagram type
- Two-color diagrams with extensive annotations in blue
- Thorough coverage of both semantics and notation, separated in each article for easy reference
- Further explanations of concepts whose meaning or purpose is obscure in the original specifications
- Discussion sections offering usage advice and additional insight into tricky concepts
- Notation summary, with references to individual articles
- A hyperlinked version of the book in Adobe Reader format on CD-ROM, an excellent resource for browsing or searching the text for specific information
- An enhanced online index available on the book’s web site allowing readers to quickly and easily search the entire text for specific topics

The result is an indispensable resource for anyone who needs to understand the inner workings of the industry standard modeling language.

THE UNIFIED MODELING LANGUAGE REFERENCE MANUAL (2ND EDITION) (THE ADDISON-WESLEY OBJECT TECHNOLOGY SERIES) BY JAMES RUMBAUGH, IVAR JACOBS PDF

The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs. A job might obligate you to constantly enhance the understanding and also encounter. When you have no adequate time to enhance it straight, you could obtain the encounter as well as expertise from checking out the book. As everybody knows, book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs is popular as the home window to open the globe. It implies that reading publication The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs will certainly provide you a new means to find everything that you require. As the book that we will certainly supply here, The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs

This *The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs* is quite appropriate for you as beginner visitor. The visitors will certainly constantly start their reading routine with the preferred motif. They may rule out the writer and author that develop guide. This is why, this book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs is actually ideal to review. However, the concept that is given in this book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs will reveal you many points. You can begin to like additionally reviewing up until completion of the book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs.

On top of that, we will share you guide The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs in soft file kinds. It will certainly not disrupt you to make heavy of you bag. You need just computer gadget or device. The link that our company offer in this site is available to click and afterwards download this The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs You recognize, having soft file of a book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs to be in your device can make alleviate the viewers. So by doing this, be an excellent visitor currently!

THE UNIFIED MODELING LANGUAGE REFERENCE MANUAL (2ND EDITION) (THE ADDISON-WESLEY OBJECT TECHNOLOGY SERIES) BY JAMES RUMBAUGH, IVAR JACOBS PDF

“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.”

—Ed Seidewitz, Chief Architect, IntelliData Technologies Corporation

The latest version of the Unified Modeling Language—UML 2.0—has increased its capabilities as the standard notation for modeling software-intensive systems. Like most standards documents, however, the official UML specification is difficult to read and navigate. In addition, UML 2.0 is far more complex than previous versions, making a thorough reference book more essential than ever.

In this significantly updated and expanded edition of the definitive reference to the standard, James Rumbaugh, Ivar Jacobson, and Grady Booch—the UML’s creators—clearly and completely describe UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Whether you are capturing requirements, developing software architectures, designing implementations, or trying to understand existing systems, this is the book for you.

Highlights include:

- Alphabetical dictionary of articles covering every UML concept
- Integrated summary of UML concepts by diagram type
- Two-color diagrams with extensive annotations in blue
- Thorough coverage of both semantics and notation, separated in each article for easy reference
- Further explanations of concepts whose meaning or purpose is obscure in the original specifications
- Discussion sections offering usage advice and additional insight into tricky concepts
- Notation summary, with references to individual articles
- A hyperlinked version of the book in Adobe Reader format on CD-ROM, an excellent resource for browsing or searching the text for specific information
- An enhanced online index available on the book’s web site allowing readers to quickly and easily search the entire text for specific topics

The result is an indispensable resource for anyone who needs to understand the inner workings of the industry standard modeling language.

- Sales Rank: #1355053 in Books

- Published on: 2004-07-29
- Original language: English
- Number of items: 1
- Dimensions: 9.54" h x 1.67" w x 7.72" l, 3.10 pounds
- Binding: Hardcover
- 752 pages

Amazon.com Review
Written by the three pioneers behind the Unified Modeling Language (UML) standard, The Unified Modeling Language Reference Manual provides an excellent real-world guide to working with UML. This title provides expert knowledge on all facets of today's UML standard, helping developers who are encountering UML on the job for the first time to be more productive.

The book begins with a history of UML, from structured design methods of the '60s and '70s to the competing object-oriented design standards that were unified in 1997 to create UML. For the novice, the authors illustrate key diagram types such as class, use case, state machine, activity, and implementation. (Of course, learning these basic diagram types is what UML is all about. The authors use an easy-to-understand ticket-booking system for many of their examples.)

After a tour of basic document types, The Unified Modeling Language Reference Manual provides an alphabetical listing of more than 350 UML terms. Entries range from a sentence or two to several pages in length. (Class, operation, and use case are just a few of the important terms that are covered.) Though you will certainly need to be acquainted with software engineering principles, this reference will serve the working software developer well. As the authors note, this isn't UML for Dummies, but neither is it an arcane academic treatise. The authors succeed in delivering a readable reference that will answer any UML question, no matter how common or obscure. --Richard Dragan

From the Inside Flap
Goals

This book is intended to be a complete and 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, syntax, 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 standard 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 do wish to consult the source documents, they are included on the accompanying CD.

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, Booch-94, Jacobson-92, and Meyer-88 provide an introduction to object-oriented concepts that is still valid and therefore unnecessary to duplicate here. 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. Those who already know an object-oriented method, such as OMT, Booch, Objectory, Coad-Yourdon, or Fusion, should be able to read the Reference Manual and use it to understand UML notation and semantics; to learn UML quickly, they may nevertheless find it useful to read the User Guide.

UML does not require a particular development process, and this book does not describe one. Although UML may be used with a variety of development processes, it was designed to support an iterative, incremental, use-casendriven process with a strong architectural focus--the kind we feel is most suitable for the development of modern, complex systems. The Unified Software Development Process Jacobson-99 describes the kind of process we believe complements the UML and best supports software development. Outline of the Book

The UML Reference Manual is organized into three parts: an overview of UML history and of modeling, a survey of UML concepts, and an alphabetical encyclopedia of UML terms and concepts.

The first part is a survey 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 views so that you can put all the concepts into perspective. The survey provides a brief overview of the views UML supports and shows how the various constructs work together. This part begins with an example that walks through various UML views and then 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 and provides starting points for detailed readings in the encyclopedia.

The third part contains the reference material organized for easy access to each topic. The bulk of the book is an alphabetical encyclopedia of all of the concepts and constructs in UML. Each UML term of any importance has its own entry in the encyclopedia. The encyclopedia is meant to be complete; therefore, everything in the concept overview in Part 2 is repeated in more detail in the encyclopedia. The same or similar information has sometimes been included in multiple encyclopedia articles so that the reader can conveniently find it.

The reference part also contains an alphabetic list of UML standard elements. A standard element is a feature predefined using the UML extensibility mechanisms. The standard elements are extensions that are felt to be widely useful.

Appendices show the UML metamodel, a summary of UML notation, and some standard sets of extensions for particular domains. 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. Encyclopedia Article Formatting Conventions

The encyclopedia 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. This flat encyclopedia 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 encyclopedia 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.

Encyclopedic articles have the following divisions, although not all divisions appear in all articles. 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. 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, the properties appear as a table in alphabetical order by property name, with the description of each property on the right. If a property has a brief enumerated list of choices, they may be given as an indented sublist. In more complicated cases, the 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 organizational mechanisms have been used, their structure should be obvious to the reader. 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 ink. To help the reader understand the notation, many diagrams contain annotations in blue ink. Any material in blue is commentary and is not part of the actual notation. 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. 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. Only a fraction of articles have this section. Simple differences in taste are generally not covered. Standard elements

This section lists standard constraints, tags, stereotypes, and other conventions that are predefined for the concept in the article. This section is fairly rare. Syntax Conventions

Syntax expressions. Syntax expressions are given in a modified BNF format in a sans serif font. To avoid confusing literal values and syntax productions, literal values that appear in the target sentence are printed in black ink, and the names of syntax variables and special syntax operators are printed in blue ink.

Text printed in black ink appears in that form in the target string.

Punctuation marks (they are always printed in black) appear 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 ink to the right of the code text. Subscripts and overbars are used as syntax operators (please refer to book for further information).

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

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 ink 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 thumbnails, and extensive hot links in the body of the articles. Simply click on one of the links to jump to the encyclopedia article for the word or phrase.

The CD also contains the full text of the OMG UML specifications, included by the permission of the Object Management Group.

We feel that this CD will be a useful on-line reference to UML for advanced users. For More Information

Additional source files and up-to-date information on further work on UML and related topics can be found on the World Wide Web sites rational and omg. Acknowledgments

We want to thank many people who made the UML possible. First, we must thank Rational Software Corporation, especially Mike Devlin and Paul Levy, who had the vision to bring us together, start the unification work, and stay the course during the four years that were required to bring the work to successful completion. We also thank the Object Management Group for providing the framework that brought together many diverse viewpoints and merged them together into a broad consensus that was much greater than any one contribution.

We particularly want to thank Cris Kobryn, who led the technical team that prepared the UML standard and who managed to achieve a consensus among an extremely strong-willed group of persons (and the three of us were not the least of his problems). His diplomatic skills and technical balance kept the UML effort from foundering amid many differences of opinion. Cris also reviewed the book and provided countless useful suggestions.

We would like to thank Gunnar &#214vergaard for reviewing the book thoroughly, as well as for his perseverance in completing many sections of the UML documents that were not fun to write but were necessary to its formal correctness.

We want to thank Karin Palmkvist for an exceedingly thorough review that uncovered many bugs in technical content, as well as many flaws in grammar, phrasing, and presentation.

We would also like to thank Mike Blaha, Conrad Bock, Perry Cole, Bruce Douglass, Martin Fowler, Eran Gery, Pete McBreen, Guus Ramackers, Tom Schultz, Ed Seidewitz, and Bran Selic for their helpful reviews.

Most of all, we want to thank the scores or even 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 are well known, but many good ideas came from those who did not have the good fortune to be widely recognized.

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 twenty-five 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&#38D 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
November 1998 020130998XP04062001

From the Back Cover

“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.”

—Ed Seidewitz, Chief Architect, IntelliData Technologies Corporation

The latest version of the Unified Modeling Language—UML 2.0—has increased its capabilities as the standard notation for modeling software-intensive systems. Like most standards documents, however, the official UML specification is difficult to read and navigate. In addition, UML 2.0 is far more complex than previous versions, making a thorough reference book more essential than ever.

In this significantly updated and expanded edition of the definitive reference to the standard, James Rumbaugh, Ivar Jacobson, and Grady Booch—the UML’s creators—clearly and completely describe UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Whether you are capturing requirements, developing software architectures, designing implementations, or trying to understand existing systems, this is the book for you.

Highlights include:

- Alphabetical dictionary of articles covering every UML concept
- Integrated summary of UML concepts by diagram type
- Two-color diagrams with extensive annotations in blue
- Thorough coverage of both semantics and notation, separated in each article for easy reference
- Further explanations of concepts whose meaning or purpose is obscure in the original specifications
- Discussion sections offering usage advice and additional insight into tricky concepts
- Notation summary, with references to individual articles
- A hyperlinked version of the book in Adobe Reader format on CD-ROM, an excellent resource for browsing or searching the text for specific information
- An enhanced online index available on the book’s web site allowing readers to quickly and easily search the entire text for specific topics

The result is an indispensable resource for anyone who needs to understand the inner workings of the industry standard modeling language.

Most helpful customer reviews

35 of 35 people found the following review helpful.
Solid reference source
By Martin Fowler
Let's be clear - this book is a reference manual, not a tutorial. Don't use this book to learn what the UML is all about. But when you want to answer a question about how to show something or what something means, then this book is invaluable. It's my first reference choice because, unlike the specification, it is written with explanation in mind. I turn to it more than any other UML book and so far I've found that when this can't answer my question, it's because the UML designers haven't thought about it yet.
So to sum up: if you use the UML seriously, make sure you have a copy handy.

13 of 13 people found the following review helpful.
Power UML developer's companion
By Daniel Moth
This book is by no means an introductory text. It assumes you already know UML. I do not think it would be of any value to managers or students. It is also of little value to developers that are happy downloading the 808 page UML specification and crunching through it. For the power UML engineer that needs to refer to the UML constructs, elements and semantics and discover new ones quickly when designing systems, this book will come in very handy indeed. Unlike other (valuable) UML books, this one will come down from the bookshelf often.

14 of 15 people found the following review helpful.
Accurate, complete (for UML 1.*); not for UML beginners
By Daniel Duffy
This book is one in a series of three by the three amigos. It is certainly the most authoritative and accurate of all three (the other two being very fuzzy in places). The book consists of the following major sections:
I: Background (some history) II: UML concepts (static, use case, statechart and other 'views') III: Reference
This book is pure syntax and can answer most of the questions that you might have about UML syntax. However, this book is not for beginners because it assumes (in my opinion) that you have applied UML to real-life situations. I find the book to be well-written (even if it is fairly dry) and compares favouably with other books in the UML series. There are different ways that you can use this book. First, you can consult it to check of you are using the correct UML syntax in your applications. Second, you can use it to deteremine what you have still to learn in UML (for example, activity diagrams, statecharts). This book should complement the other, more application-specific UML books. For example, it could be seen as a follow-up of Fowler's somewhat outdated UML Primer.
It would have been a good idea if the authors had included a complete test case showing how all the specific 'views' are documented and how they fit together. UML has about 11 different views and which one to use and when will be a major undertaking if you are embarking on a first project.
This book will be outdated as soon as the new UML 2.0 specification is ready. Do the authors have plans for a new version of their book "UML Reference 2.0"?

See all 15 customer reviews...

THE UNIFIED MODELING LANGUAGE REFERENCE MANUAL (2ND EDITION) (THE ADDISON-WESLEY OBJECT TECHNOLOGY SERIES) BY JAMES RUMBAUGH, IVAR JACOBS PDF

Just attach to the web to gain this book **The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs** This is why we imply you to use as well as utilize the industrialized modern technology. Checking out book doesn't mean to bring the printed The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs Established technology has permitted you to read just the soft documents of the book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs It is same. You may not need to go and get traditionally in looking the book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs You may not have enough time to invest, may you? This is why we give you the most effective means to obtain the book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs currently!

Amazon.com Review
Written by the three pioneers behind the Unified Modeling Language (UML) standard, The Unified Modeling Language Reference Manual provides an excellent real-world guide to working with UML. This title provides expert knowledge on all facets of today's UML standard, helping developers who are encountering UML on the job for the first time to be more productive.

The book begins with a history of UML, from structured design methods of the '60s and '70s to the competing object-oriented design standards that were unified in 1997 to create UML. For the novice, the authors illustrate key diagram types such as class, use case, state machine, activity, and implementation. (Of course, learning these basic diagram types is what UML is all about. The authors use an easy-to-understand ticket-booking system for many of their examples.)

After a tour of basic document types, The Unified Modeling Language Reference Manual provides an alphabetical listing of more than 350 UML terms. Entries range from a sentence or two to several pages in length. (Class, operation, and use case are just a few of the important terms that are covered.) Though you will certainly need to be acquainted with software engineering principles, this reference will serve the working software developer well. As the authors note, this isn't UML for Dummies, but neither is it an arcane academic treatise. The authors succeed in delivering a readable reference that will answer any UML question, no matter how common or obscure. --Richard Dragan

From the Inside Flap
Goals

This book is intended to be a complete and 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, syntax, 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 standard 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 do wish to consult the source documents, they are included on the accompanying CD.

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, Booch-94, Jacobson-92, and Meyer-88 provide an introduction to object-oriented concepts that is still valid and therefore unnecessary to duplicate here. 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. Those who already know an object-oriented method, such as OMT, Booch, Objectory, Coad-Yourdon, or Fusion, should be able to read the Reference Manual and use it to understand UML notation and semantics; to learn UML quickly, they may nevertheless find it useful to read the User Guide.

UML does not require a particular development process, and this book does not describe one. Although UML may be used with a variety of development processes, it was designed to support an iterative, incremental, use-casendriven process with a strong architectural focus--the kind we feel is most suitable for the development of modern, complex systems. The Unified Software Development Process Jacobson-99 describes the kind of process we believe complements the UML and best supports software development. Outline of the Book

The UML Reference Manual is organized into three parts: an overview of UML history and of modeling, a survey of UML concepts, and an alphabetical encyclopedia of UML terms and concepts.

The first part is a survey 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 views so that you can put all the concepts into perspective. The survey provides a brief overview of the views UML supports and shows how the various constructs work together. This part begins with an example that walks through various UML views and then 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 and provides starting points for detailed readings in the encyclopedia.

The third part contains the reference material organized for easy access to each topic. The bulk of the book is an alphabetical encyclopedia of all of the concepts and constructs in UML. Each UML term of any importance has its own entry in the encyclopedia. The encyclopedia is meant to be complete; therefore, everything in the concept overview in Part 2 is repeated in more detail in the encyclopedia. The same or similar information has sometimes been included in multiple encyclopedia articles so that the reader can conveniently find it.

The reference part also contains an alphabetic list of UML standard elements. A standard element is a feature predefined using the UML extensibility mechanisms. The standard elements are extensions that are felt to be widely useful.

Appendices show the UML metamodel, a summary of UML notation, and some standard sets of extensions for particular domains. 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. Encyclopedia Article Formatting Conventions

The encyclopedia 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. This flat encyclopedia 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 encyclopedia 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.

Encyclopedic articles have the following divisions, although not all divisions appear in all articles. 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. 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, the properties appear as a table in alphabetical order by property name, with the description of each property on the right. If a property has a brief enumerated list of choices, they may be given as an indented sublist. In more complicated cases, the 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 organizational mechanisms have been used, their structure should be obvious to the reader. 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 ink. To help the reader understand the notation, many diagrams contain annotations in blue ink. Any material in blue is commentary and is not part of the actual notation. 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. 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. Only a fraction of articles have this section. Simple differences in taste are generally not covered. Standard elements

This section lists standard constraints, tags, stereotypes, and other conventions that are predefined for the concept in the article. This section is fairly rare. Syntax Conventions

Syntax expressions. Syntax expressions are given in a modified BNF format in a sans serif font. To avoid confusing literal values and syntax productions, literal values that appear in the target sentence are printed in black ink, and the names of syntax variables and special syntax operators are printed in blue ink.

Text printed in black ink appears in that form in the target string.

Punctuation marks (they are always printed in black) appear 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 ink to the right of the code text. Subscripts and overbars are used as syntax operators (please refer to book for further information).

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

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 ink 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 thumbnails, and extensive hot links in the body of the articles. Simply click on one of the links to jump to the encyclopedia article for the word or phrase.

The CD also contains the full text of the OMG UML specifications, included by the permission of the Object Management Group.

We feel that this CD will be a useful on-line reference to UML for advanced users. For More Information

Additional source files and up-to-date information on further work on UML and related topics can be found on the World Wide Web sites rational and omg. Acknowledgments

We want to thank many people who made the UML possible. First, we must thank Rational Software Corporation, especially Mike Devlin and Paul Levy, who had the vision to bring us together, start the unification work, and stay the course during the four years that were required to bring the work to successful completion. We also thank the Object Management Group for providing the framework that brought together many diverse viewpoints and merged them together into a broad consensus that was much greater than any one contribution.

We particularly want to thank Cris Kobryn, who led the technical team that prepared the UML standard and who managed to achieve a consensus among an extremely strong-willed group of persons (and the three of us were not the least of his problems). His diplomatic skills and technical balance kept the UML effort from foundering amid many differences of opinion. Cris also reviewed the book and provided countless useful suggestions.

We would like to thank Gunnar &#214vergaard for reviewing the book thoroughly, as well as for his perseverance in completing many sections of the UML documents that were not fun to write but were necessary to its formal correctness.

We want to thank Karin Palmkvist for an exceedingly thorough review that uncovered many bugs in technical content, as well as many flaws in grammar, phrasing, and presentation.

We would also like to thank Mike Blaha, Conrad Bock, Perry Cole, Bruce Douglass, Martin Fowler, Eran Gery, Pete McBreen, Guus Ramackers, Tom Schultz, Ed Seidewitz, and Bran Selic for their helpful reviews.

Most of all, we want to thank the scores or even 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 are well known, but many good ideas came from those who did not have the good fortune to be widely recognized.

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 twenty-five 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&#38D 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
November 1998 020130998XP04062001

From the Back Cover

“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.”

—Ed Seidewitz, Chief Architect, IntelliData Technologies Corporation

The latest version of the Unified Modeling Language—UML 2.0—has increased its capabilities as the standard notation for modeling software-intensive systems. Like most standards documents, however, the official UML specification is difficult to read and navigate. In addition, UML 2.0 is far more complex than previous versions, making a thorough reference book more essential than ever.

In this significantly updated and expanded edition of the definitive reference to the standard, James Rumbaugh, Ivar Jacobson, and Grady Booch—the UML’s creators—clearly and completely describe UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Whether you are capturing requirements, developing software architectures, designing implementations, or trying to understand existing systems, this is the book for you.

Highlights include:

- Alphabetical dictionary of articles covering every UML concept
- Integrated summary of UML concepts by diagram type
- Two-color diagrams with extensive annotations in blue
- Thorough coverage of both semantics and notation, separated in each article for easy reference
- Further explanations of concepts whose meaning or purpose is obscure in the original specifications
- Discussion sections offering usage advice and additional insight into tricky concepts
- Notation summary, with references to individual articles
- A hyperlinked version of the book in Adobe Reader format on CD-ROM, an excellent resource for browsing or searching the text for specific information
- An enhanced online index available on the book’s web site allowing readers to quickly and easily search the entire text for specific topics

The result is an indispensable resource for anyone who needs to understand the inner workings of the industry standard modeling language.

Guide The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs will still give you positive value if you do it well. Completing the book The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs to read will certainly not come to be the only objective. The objective is by obtaining the favorable worth from guide until the end of the book. This is why; you need to find out even more while reading this The Unified Modeling Language Reference Manual (2nd Edition) (The Addison-Wesley Object Technology Series) By James Rumbaugh, Ivar Jacobs This is not just how fast you check out a book and not just has the amount of you completed the books; it has to do with just what you have actually obtained from guides.