

Preface

Overview

"Knowledge and human power are synonymous," once said the great philosopher Francis Bacon. However, based on experience within today's global markets, he would probably say, "The ability to capture, communicate, and leverage knowledge to solve problems is human power." This raises the question of how exactly one can best capture, communicate, and leverage knowledge, especially within the world of systems engineering. Could the answer to this question be the Unified Modeling Language (UML)?

The UML is a modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system-intensive process. It was originally conceived by Rational Software Corporation and the three prominent methodologists in the information systems and technology industry, Grady Booch, James Rumbaugh, and Ivar Jacobson (the Three Amigos). The language has gained significant industry support from various organizations via the UML Partners Consortium and has been submitted to and approved by the Object Management Group (OMG) as a standard.

The UML represents the evolutionary unification of the three most prominent methodologists' experience with other industry engineering best practices. It is a general-purpose, broadly applicable, tool-supported, industry-standardized modeling language. It applies to a multitude of different types of systems, domains, and methods or processes.

UML in a Nutshell is a quick reference to the Unified Modeling Language (UML) version 1.1.

This book is for managers and professionals who seek a broad understanding of what the UML is and is not, and for practitioners (engineers) and professionals who seek a detailed understanding of every part of the UML, including the syntax and semantics of the language and how to apply it to solve problems.

This book consolidates the authoritative UML documentation and specification into a holistic and integrated quick reference. It does not simply describe the subject, but provides the foundation for the subject from a practitioner's perspective. This book is not exclusively for methodologists, managers, analysts, developers, or programmers, but rather is for *all* types of individuals who apply knowledge to solve problems and develop solutions or systems (business, software, hardware, etc.).

This book also goes beyond the standard to show how the UML may be applied in the context of problem solving and the application of knowledge. This book will be most useful to you after you've learned the UML. It is holistic in that it covers every aspect of the UML, integrated in that it shows how all the elements of the UML are cooperatively used, and organized in a quick reference format so that you can consult it. However, this book can also help you learn the UML. It contains a simple yet complete principle-focused and technique-oriented tutorial that introduces the essentials of the UML.

Why This Book?

Working within the business and technology industry in applying object orientation, I was very aware of the issues surrounding nonstandard methods and notation. So when the thoughts of prominent methodologists in the industry started to converge toward a unified standard, I eagerly began applying the results. As I and the business and technology industry began to apply the UML from its earliest version (Unified Method), I saw that the industry was sold on the hype of the UML rather than its real benefits; that is, everyone is trying to sell the UML *standard* without really contemplating what the UML is and how to apply the UML to maximize a return on investment. Most people think it is a good thing to comply with a standard, but very few can truly articulate why!

After utilizing the UML on various projects, I started to realize that the UML is significantly more than a standard or another modeling language. It is a paradigm, philosophy, revolution, and evolution of how we approach problem solving and systems.

Furthermore, most books on the subject focus on the technology that applies (or adheres to) the UML rather than looking at the impact the UML itself has on systems development. These books directly tie the UML to some implementation language (Java, C++, etc.); however, very few, if any, are able to move beyond the notation to the concepts and their true value. This book confronts the problem that users think the UML is just for drawing diagrams, when it is really meant for capturing the knowledge and semantics behind the diagrams.

You will find that the text is pretty compressed in many places. For example, I say that the UML is a language. That's obvious, you may reply. However, don't take for granted what such a statement really means: A language is used for communication. Various practitioners with experience in the object-oriented world classify the UML as just another notation simply because the official UML documentation and most books on the subject emphasize notation rather than the holistic view of the UML. One needs to look beyond the notation and look at what constitutes a notation, why it is used, and how it is best applied.

It is often said that the English language is the world's universal language; now it is virtually certain that the UML will be the information systems and technology world's universal language.

This book, just like every other book ever written, is a snapshot of thoughts in time. If you discover errors in the text, please let me know. The UML will most likely have evolved since the writing of this book; however, this book captures the fundamental concepts that drive this evolution. Therefore, the book should remain valuable to you. Readers who would like to contact me to ask questions or to discuss this book, the UML, object orientation, or other related topics are very welcome to do so at the e-mail address salhir@earthlink.net or to visit my World Wide Web site home page at *http://home.earthlink.net/~salhir*.

Contents

This book consists of three sections or parts and 16 chapters.

Part I, "Introducing the Unified Modeling Language," presents the overall context in which the UML may be applied.

- Chapter 1, *Introduction,* introduces the goals, scope, and formal documentation of the UML, and its history and evolution.
- Chapter 2, *The Big Picture,* introduces the context for understanding and applying the UML. This includes the key concepts underlying the UML. The chapter explores the concepts of problems, solutions, problem solving, life cycles, development cycles and phases, iteration cycles and phases, domains or spaces, systems, architectures, models, architectural views, diagrams, languages, paradigms, artifacts, activities, and heuristics.
- Chapter 3, *Object Orientation,* presents the object-oriented paradigm that forms the basis for the UML. This chapter introduces some of the key concepts and constructs of object orientation. It explores real-world concepts, implementation-world concepts, the function-driven paradigm, the data-driven paradigm, the object-oriented paradigm, and key concepts independent of any implementation language.

Part II, "Using the Unified Modeling Language," presents an integrated view of the UML and how all its constituent parts (diagrams) interact to facilitate its use.

- Chapter 4, *A Unified Modeling Language Tutorial,* presents a tutorial introducing the different diagrams and elements of the UML.
- Chapter 5, *The Unified Modeling Language,* presents an overview of the constituent parts of the UML, how they relate to one another, and how they contribute to a method or process utilizing the UML. This chapter brings together problems, solutions, and problem-solving concepts into a unified view of the UML.

Part III, "The Unified Modeling Language Quick Reference," presents a detailed view of the UML constituents in a quick reference format.

- Chapter 6, *Diagramming and Model Organization,* presents information on diagramming and model organization. This is general information on diagramming and modeling in the UML.

- Chapter 7, *Class and Object Diagrams,* presents information on class and object diagrams. These diagrams are concerned with the static structure of a system, or how it is organized rather than how it behaves.
- Chapter 8, *Use Case Diagrams,* presents information on use case diagrams. These diagrams are concerned with the functions provided by a system to external entities.
- Chapter 9, *Sequence Diagrams,* presents information on sequence diagrams. These diagrams are concerned with the dynamic structure of a system, or how it behaves rather than how it is structured. The chapter focuses on communication between elements within a system.
- Chapter 10, *Collaboration Diagrams,* presents information on collaboration diagrams. Like sequence diagrams, these are concerned with the dynamic structure of a system, or how it behaves rather than how it is structured. The chapter focuses on relationships between communicating elements within a system.
- Chapter 11, *Statechart Diagrams,* presents information on statechart diagrams. These diagrams are concerned with the dynamic structure of a system, or how it behaves in response to external stimuli.
- Chapter 12, *Activity Diagrams,* presents information on activity diagrams. These diagrams are concerned with the dynamic structure of a system, or how it behaves in response to internal processing.
- Chapter 13, *Component Diagrams,* presents information on component diagrams. These diagrams are concerned with the organization of and dependencies among implementation components.
- Chapter 14, *Deployment Diagrams,* presents information on deployment diagrams. These diagrams are concerned with the configuration of processing resource elements and the mapping of implementation components onto them.
- Chapter 15, *Extension Mechanisms,* presents information on extension mechanisms. These mechanisms are used to customize and extend the UML.
- Chapter 16, *The Object Constraint Language,* presents information on the Object Constraint Language. This language is used for expressing constraints and rules on model elements.

The appendix contains references to notable resources on the World Wide Web and various books that may be of interest.

The book makes extensive use of bulleted lists. Although unusual (even unique), the style is suited to the book's purpose and subject in many ways. The information you need to know about each feature falls into several different categories—visual appearance, rule vis-a-vis other features, rules—and is thus easiest to absorb in bite-sized pieces. Furthermore, by comparing the bullets under different paragraphs you can quickly see the ways in which similar features differ.

The UML notation is presented in example-like diagram templates. These diagrams depict all possible parts and options of the notation, and the text indicates which constructs are required or optional as well as any other important information.

Ellipses in the diagram templates indicate that information is repeated. These diagrams are not simply generic templates, since I would have to invent a whole complex language in order to show what's optional and what's required. Furthermore, these diagrams are not strict examples, since there would be too many examples required to depict all possible options. Rather, they are templates exemplifying the concepts.

The book does not contain any source code because focus is given to the modeling language independent of any translation to a specific implementation. Rather than show you how to translate a system to a specific implementation, I show you how to view a system and determine what the right system is to solve a problem.

How to Contact Us

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
1-800-998-9938 (in the U.S. or Canada)
1-707-829-0515 (international/local)
1-707-829-0104 (FAX)

You can also send us messages electronically. To be put on the mailing list or request a catalog, send email to:

info@oreilly.com

To ask technical questions or comment on the book, send email to:

bookquestions@oreilly.com

We have a web site for the book, where we'll list errata and any plans for future editions. You can access this page at:

http://www.oreilly.com/catalog/umlnut/

For more information about this book and others, see the O'Reilly web site:

http://www.oreilly.com

Acknowledgments

There are a number of individuals who made this work possible. It is not only I who deserve credit, but rather those who had to live with me, who demonstrated an abundance of encouragement, patience, understanding, and had to sacrifice a part of themselves to let me accomplish this endeavor.

I thank God for making everything possible, and I thank my teachers for making this real: My father Saad, who taught me the meaning of success by exemplifying it as my mentor and role model; my mother Rabab, who taught me that being

analytical without emotion provides only a limited view of the world and its wonder; my wife Milad, who taught me that reason without passion is insufficient to bring fulfillment; my brother Ghazwan, who taught me to be pragmatic; my brother Phillip, who taught me to be idealistic; and my daughter Nora, who continuously teaches me the truths I failed to learn from my own childhood.

I would also like to thank my mentors, Dr. Carl Victor Page and Dr. George C. Stockman, for teaching me that the theoretical and abstract realm of computer science and the practical and real life of a computer professional are not disjoint, but intersect to establish a foundation for good solutions to complex real-world problems.

I would also like to thank Tim O'Reilly for giving me the opportunity to do this book; my editor, Andy Oram, for his effort and understanding and for showing me the true fabric that makes O'Reilly and Associates and *In a Nutshell* books a success; and all the staff at O'Reilly and Associates for their work in bringing this book to life. I would also like to thank the following reviewers for their feedback: Jeff Estefan, Dr. Brian Henderson-Sellers, and Kevin Jameson.

I will not forget any of you, and I only ask that you please remember me.