

# Foreword to the Third Edition

Since ancient times, the most talented architects and the most gifted designers have known the law of parsimony. Whether it is stated as a paradox (“less is more”), or a koan (“Zen mind is beginner’s mind”), its wisdom is timeless: Reduce everything to its essence so that form harmonizes with function. From the pyramids to the Sydney Opera House, from von Neumann architectures to UNIX and Smalltalk, the best architects and designers have strived to follow this universal and eternal principle.

Recognizing the value of shaving with Occam’s Razor, when I architect and read I seek projects and books that adhere to the law of parsimony. Consequently, I applaud the book you are reading now.

You may find my last remark surprising at first. I am frequently associated with the voluminous and dense specifications that define the Unified Modeling Language (UML). These specifications allow tool vendors to implement the UML and methodologists to apply it. For seven years, I have chaired large international standardization teams to specify UML 1.1 and UML 2.0, as well as several minor revisions in between. During this time, the UML has matured in expressiveness and precision, but it has also added gratuitous complexity as a result of the standardization process. Regrettably, standardization processes are better known for design-by-committee compromises than parsimonious elegance.

What can a UML expert familiar with the arcane minutiae of the specification learn from Martin’s distillation of UML 2.0? Quite a bit, as can you. To start with, Martin adroitly reduces a large and complex language into a pragmatic subset that he has proven effective in his practice. He has resisted the easy route of tacking on additional pages to the last edition of his book. As the language has grown, Martin has kept true to his goal of seeking the “fraction of UML that is most useful” and telling you just that. The fraction he refers to is

the mythical 20 percent of UML that helps you do 80 percent of your work. Capturing and taming this elusive beast is no mean accomplishment!

It is even more impressive that Martin achieves this goal while writing in a wonderfully engaging conversational style. By sharing his opinions and anecdotes with us, he makes this book fun to read and reminds us that architecting and designing systems should be both creative and productive. If we pursue the parsimony koan to its full intent, we should find UML modeling projects to be as enjoyable as we found finger-painting and drawing classes in grammar school. UML should be a lightning rod for our creativity as well as a laser for precisely specifying system blueprints so that third parties can bid and build those systems. The latter is the acid test for any bona fide blueprint language.

So, while this may be a small book, it is not a trivial one. You can learn as much from Martin's approach to modeling as you can learn from his explanations of UML 2.0.

I have enjoyed working with Martin to improve the selection and correctness of the UML 2.0 language features explained in this revision. We need to keep in mind that all living languages, both natural and synthetic, must evolve or perish. Martin's choices of new features, along with your preferences and those of other practitioners, are a crucial part of the UML revision process. They keep the language vital and help it evolve via natural selection in the marketplace.

Much challenging work remains before model-driven development becomes mainstream, but I am encouraged by books like this that explain UML modeling basics clearly and apply them pragmatically. I hope you will learn from it as I have and will use your new insights to improve your own software modeling practices.

Cris Kobryn  
Chair, U2 Partners' UML 2.0 Submission Team  
Chief Technologist, Telelogic

# Foreword to the First Edition

When we began to craft the Unified Modeling Language, we hoped that we could produce a standard means of expressing design that would not only reflect the best practices of industry, but would also help demystify the process of software system modeling. We believed that the availability of a standard modeling language would encourage more developers to model their software systems before building them. The rapid and widespread adoption of the UML demonstrates that the benefits of modeling are indeed well known to the developer community.

The creation of the UML was itself an iterative and incremental process very similar to the modeling of a large software system. The end result is a standard built on, and reflective of, the many ideas and contributions made by numerous individuals and companies from the object community. We began the UML effort, but many others helped bring it to a successful conclusion; we are grateful for their contribution.

Creating and agreeing on a standard modeling language is a significant challenge by itself. Educating the development community, and presenting the UML in a manner that is both accessible and in the context of the software development process, is also a significant challenge. In this deceptively short book, updated to reflect the most recent changes to the UML, Martin Fowler has more than met this challenge.

In a clear and friendly style, Martin not only introduces the key aspects of UML, but also clearly demonstrates the role UML plays in the development process. Along the way, we are treated to abundant nuggets of modeling insight and wisdom drawn from Martin's 12-plus years of design and modeling experience.

The result is a book that has introduced many thousands of developers to UML, whetting their appetite to further explore the many benefits of modeling with this now standard modeling language.

We recommend the book to any modeler or developer interested in getting a first look at UML and in gaining a perspective on the key role it plays in the development process.

Grady Booch

Ivar Jacobson

James Rumbaugh