

UML与面向对象设计影印丛书

UML与J2EE企业 应用程序开发

DEVELOPING ENTERPRISE
JAVA APPLICATIONS
WITH J2EE™ AND UML

(美) KHAWAR ZAMAN AHMED
CARY E. UMRYSH

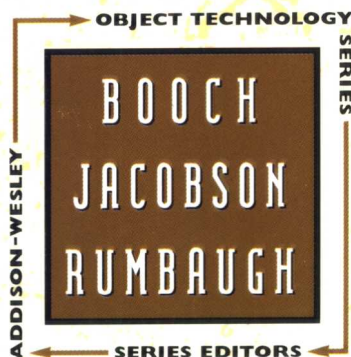
编著

Foreword by Grady Booch



科学出版社

www.sciencep.com



UML 与面向对象设计影印丛书

UML 与 J2EE 企业应用程序开发

Developing Enterprise Java Applications with J2EE and UML

(美) Khawar Zaman Ahmed 编著
Cary E. Umrysh



科学出版社

北 京

图字：01-2003-6987 号

内 容 简 介

本书通过一个完整的实例，系统介绍了用 J2EE 开发企业级软件时，将 UML 建模技术应用到软件开发过程各个阶段的方法。本书首先介绍了 J2EE 的基本概念和主要技术，以及 UML 中的各种设计视图和基本原理，在此基础上，深入讲解了如何使用 UML 进行分析和设计，以及如何使用 UML 为 J2EE 主要技术建模。

本书不仅适合于初学 UML 和 J2EE 的读者，而且对于 J2EE 程序开发人员和软件工程项目管理人员也有很大的参考价值。

English reprint copyright©2003 by Science Press and Pearson Education Asia Limited.

Original English language title: Developing Enterprise Java Applications with J2EE™ and UML, 1st Edition
by Khawar Zaman Ahmed and Cary E. Umrysh, Copyright©2002

ISBN 0-201-73829-5

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley Publishing Company, Inc.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macao SAR).

仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签。无标签者不得销售。

图书在版编目 (CIP) 数据

UML 与 J2EE 企业应用程序开发=Developing Enterprise Java Applications with J2EE and UML/ (美) Khawar Zaman Ahmed 等编著. —影印本. —北京: 科学出版社, 2004

(UML 与面向对象设计影印丛书)

ISBN 7-03-012466-9

I. U... II. Z... III. ①面向对象语言, UML—程序设计—英文②Java 语言—程序设计—英文
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 099951 号

策划编辑: 李佩乾/责任编辑: 舒 立
责任印制: 吕春珉/封面制作: 东方人华平面设计室

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

2004 年 1 月 第 一 版 开本: 787×1092 1/16

2004 年 1 月第一次印刷 印张: 22 1/4

印数: 1—3 000 字数: 422 000

定价: 39.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

影印前言

随着计算机硬件性能的迅速提高和价格的持续下降,其应用范围也在不断扩大。交给计算机解决的问题也越来越难,越来越复杂。这就使得计算机软件变得越来越复杂和庞大。20 世纪 60 年代的软件危机使人们清醒地认识到按照工程化的方法组织软件开发的必要性。于是软件开发方法从 60 年代毫无工程性可言的手工作坊式开发,过渡到 70 年代结构化的分析设计方法、80 年代初的实体关系开发方法,直到面向对象的开发方法。

面向对象的软件开发方法是在结构化开发范型和实体关系开发范型的基础上发展而来的,它运用分类、封装、继承、消息等人类自然的思维机制,允许软件开发者处理更为复杂的问题域和其支持技术,在很大程度上缓解了软件危机。面向对象技术发端于程序设计语言,以后又向软件开发的早期阶段延伸,形成了面向对象的分析和设计。

20 世纪 80 年代末 90 年代初,先后出现了几十种面向对象的分析设计方法。其中,Booch, Coad/Yourdon、OMT 和 Jacobson 等方法得到了面向对象软件开发界的广泛认可。各种方法对许多面向对象的概念的理解不尽相同,即便概念相同,各自技术上的表示法也不同。通过 90 年代不同方法流派之间的争论,人们逐渐认识到不同的方法既有其容易解决的问题,又有其不容易解决的问题,彼此之间需要进行融合和借鉴;并且各种方法的表示也有很大的差异,不利于进一步的交流与协作。在这种情况下,统一建模语言(UML)于 90 年代中期应运而生。

UML 的产生离不开三位面向对象的方法论专家 G. Booch、J. Rumbaugh 和 I. Jacobson 的通力合作。他们从多种方法中吸收了大量有用的建模概念,使 UML 的概念和表示法在规模上超过了以往任何一种方法,并且提供了允许用户对语言做进一步扩展的机制。UML 使不同厂商开发的系统模型能够基于共同的概念,使用相同的表示法,呈现彼此一致的模型风格。1997 年 11 月 UML 被 OMG 组织正式采纳为标准的建模语言,并在随后的几年中迅速地发展为事实上的建模语言国际标准。

UML 在语法和语义的定义方面也做了大量的工作。以往各种关于面向对象方法的著作通常是以比较简单的方式定义其建模概念,而以主要篇幅给出过程指导,论述如何运用这些概念来进行开发。UML 则以一种建模语言的姿态出现,使用语言学中的一些技术来定义。尽管真正从语言学的角度看它还有许多缺陷,但它在这方面所做的努力却是以往的各种建模方法无法比拟的。

从 UML 的早期版本开始,便受到了计算机产业界的重视,OMG 的采纳和大公司的支持把它推上了实际上的工业标准的地位,使它拥有越来越多的用户。它被广泛地用

于应用领域和多种类型的系统建模,如管理信息系统、通信与控制系统、嵌入式实时系统、分布式系统、系统软件等。近几年还被运用于软件再工程、质量管理、过程管理、配置管理等方面。而且它的应用不仅仅限于计算机软件,还可用于非软件系统,例如硬件设计、业务处理流程、企业或事业单位的结构与行为建模,等等。

在 UML 陆续发布的几个版本中,逐步修正了前一个版本中的缺陷和错误。即将发布的 UML2.0 版本将是对 UML 的又一次重大的改进。将来的 UML 将向着语言家族化、可执行化、精确化等理念迈进,为软件产业的工程化提供更有力的支撑。

本丛书收录了与面向对象技术和 UML 有关的十几本书,反映了面向对象技术最新的发展趋势以及 UML 的新的研究动态。其中涉及对面向对象建模理论研究与实践的有这样几本书:《面向对象系统架构及设计》主要讨论了面向对象的基本概念、静态设计、永久对象、动态设计、设计模式以及体系结构等近几年来面向对象技术领域中的新的理论知识与方法;《用 UML 进行用况对象建模》主要介绍了面向对象的需求阶段、分析阶段、设计阶段中用况模型的建立方法与技术;《高级用况建模》介绍了在建立用况模型中需要注意的高级的问题与技术;《UML 面向对象设计基础》则侧重于经典的面向对象理论知识的阐述;《UML 参考手册》列出了 UML 的所有术语和标准元素,从语义、表示法和用途等方面详尽地介绍了 UML 的构成和概念。

涉及 UML 在特定领域的运用的有这样几本:《UML 实时系统开发》讨论了进行实时系统开发时需要 UML 进行扩展的技术;《用 UML 构建 Web 应用程序》讨论了运用 UML 进行 Web 应用建模所应该注意的技术与方法;《面向对象系统测试:模型、视图与工具》介绍了将 UML 应用于面向对象的测试领域所应掌握的方法与工具;《对象、构件、框架与 UML 应用》讨论了如何运用 UML 对面向对象的新技术——构件-框架技术建模的方法策略;《UML 与 Visual Basic 应用程序开发》主要讨论了从 UML 模型到 Visual Basic 程序的建模与映射方法;《XML 程序的 UML 建模》讲解了如何将 XML 与 UML 结合,创建动态的 Web 应用程序,实现最优的 B2B 应用集成;《构建可扩展数据库应用程序》介绍了商务模式和数据库模式的建模方法以及集成系统的程序实现;《UML 与并行分布式实时应用程序设计》对 UML 在并行分布式实时系统开发中的应用作了全面而详细的介绍,尤其对面向对象方法解决此类系统特有的问题作了有针对性的讲解;《UML 与 J2EE 企业应用程序开发》系统介绍了使用 J2EE 开发企业级软件工程时,将 UML 建模技术应用到软件开发各个阶段的方法。

介绍面向对象编程技术的有两本书:《COM 高手心经》和《ATL 技术内幕》,深入探讨了面向对象的编程新技术——COM 和 ATL 技术的使用技巧与技术内幕。

还有一本《Executable UML 技术内幕》,这本书介绍了可执行 UML 的理念与其支持技术,使得模型的验证与模拟以及代码的自动生成成为可能,也代表着将来软件开发的一种新的模式。

总之，这套书所涉及的内容包含了对软件生命周期的全过程建模的方法与技术，同时也对近年来的热点领域建模技术、新型编程技术作了深入的介绍，有些内容已经涉及到了前沿领域。可以说，每一本都很经典。

有鉴于此，特向软件领域中不同程度的读者推荐这套书，供大家阅读、学习和研究。

北京大学计算机系 蒋严冰 博士

Foreword

The history of software engineering is, in effect, the history of abstraction. As complexity rises, we respond by raising the level of abstraction in our programming languages and in our methods. Thus, we have seen the move from C to Java, from structured methods to object-oriented design, and from classes to design patterns to architectural frameworks.

J2EE, the Java 2 Platform, Enterprise Edition, is such a framework. J2EE is a comprehensive platform for deploying complex systems. It raises the level of abstraction for the development team by offering a set of mechanisms (JSP, Enterprise JavaBeans, servlets) and services (JDBC, JNDI, JMS, and RMI to name a few), enabling the team to focus on its core business value instead of building infrastructure.

As good as J2EE is, however, there is a large semantic gap between what J2EE provides and what must be crafted for the business. Bridging that gap can only be achieved given a strong, foundational understanding of J2EE together with a sound architecture for the domain-specific system. The Unified Modeling Language (UML) comes into play here, for the UML is essentially the language of blueprints for software. Visualizing, specifying, constructing, and documenting the key elements of a system are essential as complexity rises, and this is the very reason for the UML's existence.

Khawar and Cary bring all of these elements together in this book to help you bridge that semantic gap. Not only do they cover all of the essential pieces of J2EE thus helping you build a foundational understanding, they also explain

how best to use J2EE's mechanisms and services. This book will also guide you in applying the UML to model your systems built upon J2EE, thus enabling you to better reason about and communicate the analysis and design decisions your team must make in building quality software.

The authors have a deep understanding of J2EE and the UML and a strong sense of the best practices that can lead you to the effective use of both. Their experience in building production systems comes through in their writing, and especially in their comprehensive case study.

There is an essential complexity in building enterprise systems; this book will help you master much of that complexity.

—Grady Booch
Chief Scientist
Rational Software Corporation

Preface

Developing complex software requires more than just churning out lines of code. As a software architect or developer involved in an industrial project, you must understand and be able to leverage critical software subdisciplines such as architecture, analysis and design techniques, development processes, visual modeling, and the underlying technology to be successful.

This book brings all these diverse elements together from the Java 2 Platform, Enterprise Edition (J2EE) development perspective to provide a holistic approach for the reader. Specifically, this book tries to answer the following key questions:

- What is the Unified Modeling Language (UML), and how is it relevant to J2EE development?
- How do Java and UML relate to each other?
- What are the key concepts in software architecture?
- How does a software development process fit into the J2EE software development equation?
- How can analysis and design help you to arrive at a better J2EE application design?
- What are the key J2EE technologies, and how do they fit together?
- How can you leverage the UML for J2EE development?

Rather than reinvent the wheel, the approach taken in this book is that of bringing together known works, such as Jim Conallen's Web Modeling Profile and the Sun Java Specification Request-26 for UML/EJB Mapping Specification.

To provide a practical illustration of the topics discussed, this book guides you through a sample J2EE application development project using the Rational Unified Process (RUP) and the UML. A working implementation is provided. Suggestions for further enhancements are also listed to assist you in continuing your exploration of the UML and J2EE technologies.

Intended Audience

This book is suitable for anyone interested in learning about the UML and how it can be applied to J2EE development. Current J2EE application developers will learn how to apply the UML to J2EE application development. UML practitioners will benefit from learning about the J2EE in the context of the UML. And software professionals interested in learning both the UML and J2EE will be able to get to a productive state faster facilitated by the intertwined contextual discussion.

After reading the book, you will

- Be able to effectively utilize the UML for developing J2EE applications
- Learn about the key J2EE technologies (EJB, JSP, and servlets) at a technical level
- Know when to use Model 1 versus Model 2 architecture, and identify situations where patterns such as value object and session bean chaining may be appropriate
- Understand software architecture concepts such as decomposition, layering, components, frameworks, patterns, and tiers
- Be able to apply techniques such as use case analysis, analysis object discovery, and analysis to design transformation to your J2EE project
- Understand the notion of software development processes and the fundamentals of some of the currently popular processes
- Learn how to start using the RUP for your J2EE project

This book only covers the Java language to the extent of providing a mapping of key Java concepts to the UML. Consequently, some familiarity with Java is assumed (knowing C++ or a similar language should be sufficient to get the basics from the examples). Prior knowledge of, or experience with, the UML, J2EE, or enterprise application development is not a prerequisite, but is certainly helpful.

How to Use This Book

If you are new to the UML and J2EE, you will get the most out of this book by reading it completely in a sequential manner.

Those who are comfortable with the UML and are primarily interested in learning about J2EE (or how to apply the UML to J2EE) can jump directly to Chapters 9–16.

On the other hand, if you know J2EE and mostly want to learn about UML, you should concentrate on Chapters 1–8, and then skim through the remaining portions of the book.

You will get the best results if you get your hands on a good modeling tool and try to apply visual modeling to a problem of your own!

Chapter Summaries

Chapter 1: *Introduction to Enterprise Software* provides a high-level overview of enterprise software development and related technologies.

Chapter 2: *Introduction to the J2EE* covers the basics of the Java 2 Platform, Enterprise Edition. It provides an overview of the basic technologies and the APIs, which form the J2EE.

Chapter 3: *Introduction to the UML* provides an overview of the UML and a quick introduction to the UML basics.

Chapter 4: *UML and Java* provides an overview of the Java language's mapping to the UML and covers some of the basic UML constructs.

Chapter 5: *Overview of Activities* introduces the notion of software development processes and outlines the approach taken in the book.

Chapter 6: *Architecture*, which is an important aspect of good software, introduces the notion of software architecture and provides an overview of some of the concepts in software architecture.

Chapter 7: *Analyzing Customer Needs* shows you how to apply UML use cases to better understand customer requirements. No matter how cool the software, if it does not meet the customer's requirements, it is a failure!

Chapter 8: *Creating the Design* focuses on analyzing the requirements further and creating the initial design for the case study. This chapter discusses how to translate the requirements you have gathered into software.

Chapter 9: *Overview of J2EE Technologies* lays the groundwork for the J2EE technologies we discuss in the remaining chapters.

Chapter 10: *Servlets* provides an overview of the Java servlet technology, discusses how they are modeled in the UML, and then shows a representative application of UML and servlets to the case study. Java servlets are ideal for the request-response oriented Web paradigm.

Chapter 11: *JavaServer Pages* teaches you about JSPs, when to use them, and how to use them in the sample project. JavaServer Pages (JSP) combine the power of servlets with the flexibility of HTML pages.

Chapter 12: *Session Beans* discusses how session beans are used in the middle tier and how to best model and utilize them. Session beans are one of the three types of enterprise beans provided in the J2EE. The chapter concludes with the usage of session beans in the context of the case study.

Chapter 13: *Entity Beans* focuses on the entity bean concept, its advantages and issues, and how to effectively model it in the UML. Entity beans provide a convenient way to objectify the stored data.

Chapter 14: *Message-Driven Beans* covers the technology and how to model them in the UML. Message-driven beans are a new addition to the J2EE Enterprise JavaBean specification.

Chapter 15: *Assembly and Deployment* discusses how UML can help assembly and deployment of a distributed application.

Chapter 16: *Case Study* discusses the details of the example used in this book including general requirements, restrictions, and such.

References for further reading include books, articles, and online sources.

A **Glossary** containing specialized terms and their meanings is provided for quick reference. An **Index** is provided for quick lookup and reference.

Conventions

We use several notational conventions throughout this book. A short list is provided for your reference:

- Italicized words are used to highlight key concepts or terminology.
- References to terms such as `javax.servlet.http.HttpServletResponse` are used to identify the exact J2SE or J2EE classes for further details. For example, in the preceding term the user is being referred to the `HttpServletResponse` class, which is found in the `http` package located in the `servlet` package of the `javax` package.

- Boldface text is used to identify keywords and reserved words in the context of Java/J2EE, for example, **ejbCreate**.
- Code samples are shown in a slightly different format to distinguish them from plain text, for example, `public void acceptOrder() {`

Acknowledgments

We would like to acknowledge the contributions of all those who helped make this book possible.

Our sincere thanks to Kirk Knoernschid, Todd Dunnavant, Dave Tropeano, Atma Sutjianto, Kevin Kelly, Terry Quatrani, Carolyn Hakansson-Johnston, Ingrid Subbotin, Jim Conallen, Loïc Julien, Dave Hauck, James Abbott, Simon Johnston, Tommy Fannon, Hassan Issa, and all others who provided direct or indirect input, clarifications, ideas, feedback, guidance, and reviews at various stages, from before inception through to completion. Their help was instrumental in defining and redefining the work, in eliminating inaccuracies, in creating additional material, and in the end, the result is a better product overall.

A special thank you to Todd Dunnavant. He not only reviewed multiple drafts cover to cover, he also generously provided in-depth written explanations, suggestions, and comments on various topics that we were only too happy to incorporate in the book.

Kirk Knoernschid's succinct review was most helpful in getting us to focus and remedy some of the key deficiencies in an earlier, draft version. Thank you for that.

Khawar would like to acknowledge and thank Kevin Kelly for his guidance and mentoring. Kevin's insights and ideas were immensely useful throughout this project.

Dave Tropeano's review of a very early draft directly led to a revectoring of our overall approach and the addition of at least two full chapters. The final version is better because of it, and we have Dave to thank.

Our thanks to Rational Software and its management for fostering a work environment in which such endeavors can be undertaken. We would especially like to thank Steve Rabuchin for his willingness to go the extra mile to help others pursue their ideas and achieve their goals. We would also like to thank Jim McGee, Roger Oberg, Magnus Christerson, John Scumniotales, Matt Halls, and Eric Naiburg. Had it not been for the encouragement and support of these individuals, this book would neither have been conceived nor written.

We are very grateful to the staff at Addison-Wesley for their support throughout this project. We especially thank Paul W. Becker and his assistant Jessica Cirone who assisted, reminded, guided, and prodded us through the publishing process. Many thanks to Anne Marie Walker who, through her thoughtful editing, transformed our semi-coherent passages into readable paragraphs. Thanks also to Kathy Glidden of Stratford Publishing Services, Inc. for her skilled project management in the critical production stage.

We benefited immensely from others who have worked on or written about the UML, J2EE, and related topics. To that end, we would like to thank the various authors whose books, articles, and Web sites are listed in the References section. Their works helped expand our understanding of the subjects.

Last but most importantly, we would like to thank our families for their patience and support throughout these last several months. Khawar would like to thank his wife Heike and his daughter Yasmeen for their cheerful and understanding attitude and for their support during this long commitment. Heike's diligent proofreading and corrections to the draft at various stages were invaluable and resulted in the elimination of numerous late-night typos and incoherent sentences. Cary would like to thank his wife Socorro for all her support and helpfulness during this lengthy project.

—K.Z.A.

—C.E.U.

Contents

	<i>Foreword</i>	<i>xiii</i>
	<i>Preface</i>	<i>xv</i>
	Intended Audience	xvi
	How to Use This Book	xvii
	Chapter Summaries	xvii
	Conventions	xviii
	<i>Acknowledgments</i>	<i>xxi</i>
Chapter 1	Introduction to Enterprise Software	1
	What Is Enterprise Software?	3
	Challenges in Developing Enterprise Software	5
	Evolution of Enterprise Software	6
	Enterprise Software and Component-Based Software	8
	Summary	8
Chapter 2	Introduction to the J2EE	9
	What Is the Java 2 Platform, Enterprise Edition?	11
	A Brief History of J2EE	12
	Why J2EE?	13
	A Brief Overview of J2EE	16
	Technologies	16
	APIs	21
	Other J2EE Technologies and APIs	22
	Summary	23

Chapter 3	Introduction to the UML	25
	UML Overview	28
	Why Use the J2EE and the UML Together?	30
	Challenges in Modeling J2EE in the UML	31
	Extension Mechanisms in the UML	32
	Stereotype	32
	Tagged Value	33
	Constraint	34
	The Approach to J2EE UML Modeling	35
	Summary	36
Chapter 4	UML and Java	37
	Representing Structure	39
	Class	39
	Variable	40
	Method	41
	Object	42
	Interface	43
	Package	44
	Representing Relationships	45
	Inheritance	45
	Realization	45
	Dependency	46
	Association	47
	Aggregation	51
	Composition	53
	Reflexive Relationships	55
	Summary	55
Chapter 5	Overview of Activities	57
	What Is a Software Development Process?	59
	Overview of Popular Approaches to Software Development	59
	The Just-Develop-It Approach	59
	The Waterfall Process	60
	The Iterative Process	60
	The Rational Unified Process	61
	The ICONIX Process	66
	The OPEN Process	66
	Extreme Programming/Feature-Driven Development	66
	Approach Used in This Book	67
	Overview of Major Activities	68
	Chapter 6: Architecture	68
	Chapter 7: Analyzing Customer Needs	69