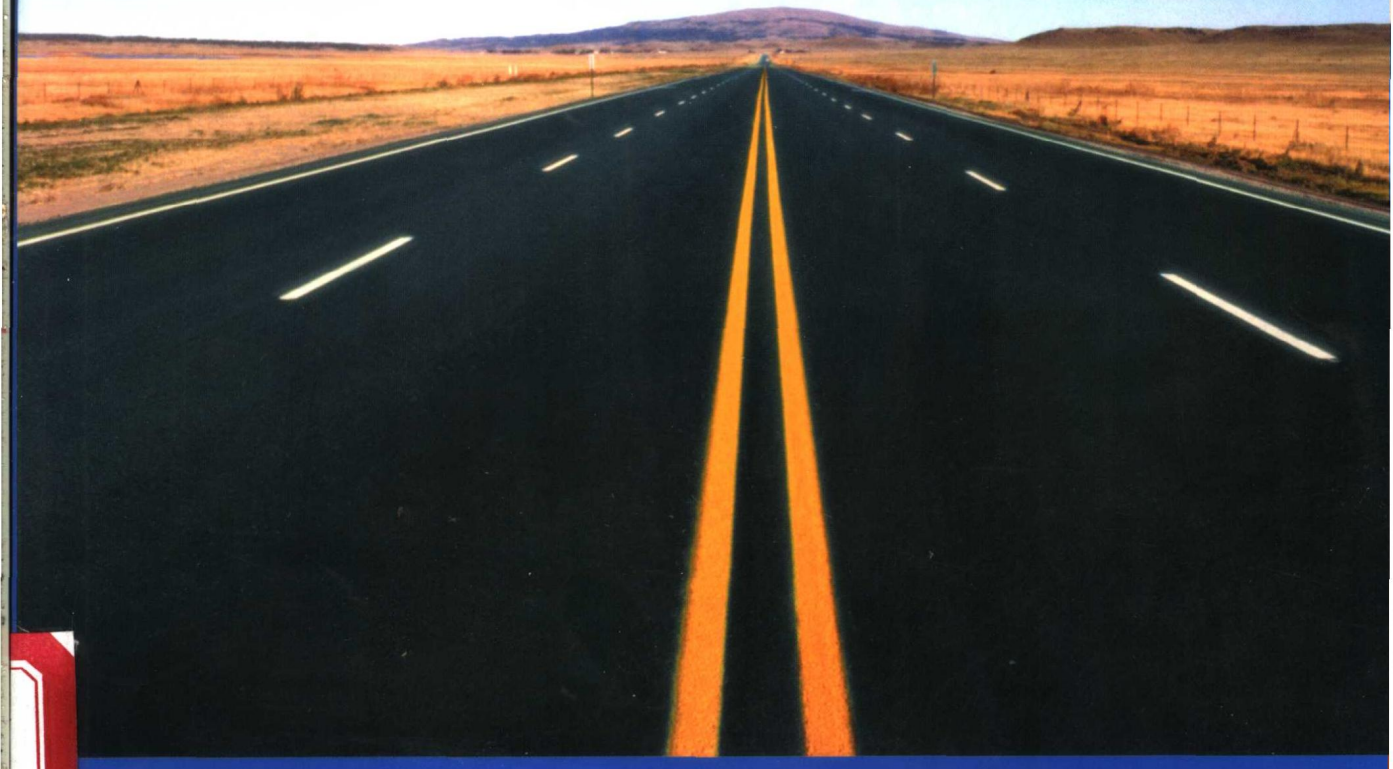
UML
面向对象结构设计与应用

施昊华　张朝辉　编著



国防工业出版社
http://www.ndip.cn

UML 面向对象结构设计与应用

施昊华　张朝辉　编著

国防工业出版社

·北京·

内容简介

统一建模语言(UML)是一种通用的可视化建模语言,用于对软件进行描述、可视化处理、构造和建立软件系统制品的文档。它记录了对必须构造的系统的决定和理解,可用于对系统的理解、设计、浏览、配置、维护和信息控制。UML适用于各种软件开发方法、软件生命周期的各个阶段、各种应用领域以及各种开发工具,是一种总结了以往建模技术的经验并吸收当今优秀成果的标准建模方法。

本书按照从初级到高级、从基本概念到应用实例的顺序,逐渐由浅入深地对UML进行了详细地讲述,具有层次清楚、思路清晰、讲解透彻、实例丰富的特点。适合于广大软件开发人员、系统分析人员、开发人员以及相关专业的科技人员自学使用,同时也适合于广大高等院校的师生作为教材或自学参考使用。

图书在版编目(CIP)数据

UML面向对象结构设计与应用/施昊华,张朝辉编著.
北京:国防工业出版社,2003.9
ISBN 7-118-03156-9

Ⅰ.U... Ⅱ.①施... ②张... Ⅲ.面向对象语言,UML-程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字(2003)第040925号

国防工业出版社出版发行
(北京市海淀区紫竹院南路23号)
(邮政编码 100044)
北京奥隆印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 23 522千字
2003年9月第1版 2003年9月北京第1次印刷
印数:1—3500册 定价:32.00元

前　言

统一建模语言（Unified Modeling Language，UML）由 Booch、Rumbaugh、Jacobson 三位专家联手提出，宗旨是使 UML 成为一套用来构建系统模型的一般性语言，这意味着它可以表达不同种类及用途的模型，恰似一种程序语言或是一种自然语言能够被使用在各个方面。这一宗旨，简单且有力地表达了 UML 所具备的语言特性，可以用在软件开发的各个阶段中，与使用者及工作中的各种角色（Worker）进行有效的沟通。

UML 的出现可谓是划时代的。在此之前，人们一直在为建模烦恼。虽然已经有很多种不同的建模语言，但由于它们互相之间不兼容，而且各自专注于某一领域，使得人们很难完成各领域之间的对话。统一建模语言的出现彻底改变了这一切，使人们可以真正实现软件的流水线生产。

著名的程序设计专家 Rhein Frank 说过这样一句话："设计语言是我们如何创造与世界上的事物互动的基础，模块语言和乐符皆属于设计语言的一种，不同的仅在于乐符是用来创作音乐，而让作曲者、演奏者、指挥者甚而听众间实现互动。而模式语言则用在软件创作，目的是让系统分析、设计、程序设计、稽核乃至于使用者之间实现沟通与互动。统一建模语言就像目前世界通用的乐符一样，没有它将会怎样，我想我们只要想一想如果没有通用乐符世界将会怎样就会非常容易理解这个问题。"

UML 是一个通用的可视化建模语言，用于对软件进行描述、可视化处理、构造和建立软件系统制品的文档。它记录了对必须构造的系统的决定和理解，可用于对系统的理解、设计、浏览、配置、维护和信息控制。UML 适用于各种软件开发方法、软件生命周期的各个阶段、各种应用领域以及各种开发工具，是一种总结了以往建模技术的经验并吸收当今优秀成果的标准建模方法。UML 包括概念的语义、表示法和说明，提供了静态、动态、系统环境及组织结构的模型。它可以被交互的可视化建模工具所支持，这些工具提供了代码生成器和报表生成器。UML 标准并没有定义一种标准的开发过程，但它适用于迭代式的开发。它是为支持大部分现存的面向对象开发过程而设计的。

UML 描述了一个系统的静态结构和动态行为，是将系统描述为一些离散的相互作用的对象并最终为外部用户提供一定功能的模型结构。静态结构定义了系统中重要对象的属性和操作以及这些对象之间的相互关系。动态行为定义了对象的时间特性和对象为完成目标而相互进行通信的机制。从不同但相互联系的角度对系统建立的模型可用于不同的目的。

UML 还包括可将模型分解成包的结构组件，以便于软件小组将大的系统分解成易于处理的块结构，并理解和控制各个包之间的依赖关系，在复杂的开发环境中管理模型单元。它还包括用于显示系统实现和组织运行的组件。

UML 不是一门程序设计语言，但可以使用代码生成器工具将 UML 模型转换为多种程序设计语言代码，或使用反向生成器工具将程序源代码转换为 UML。UML 不是一种可用于定理证明的高度形式化的语言，这样的语言有很多种，但它们通用性较差，不易理解和使用。UML 是一种通用建模语言。对于一些专门领域，例如用户图形界面（GUI）设计、超大规模集成电路（VLSI）设计、基于规则的人工智能领域，使用专门的语言和工具可能会更适合些。UML 是一种离散的建模语言，不适合对诸如工程和物理学领域中的连续系统建模。它是一种综合的通用建模语言，适合对诸如由计算机软件、硬件或数字逻辑构成的离散系统建模。

本书分为 4 个部分，共 17 章。第 1 部分“基础知识”，是对 UML 本身的介绍，包括第 1～4 章。第 1 章“UML 基础”，介绍了 UML 的基本组成、发展历史、目标、概念范围、应用领域等，此外还介绍了到目前为止 UML 未涉及的领域，最后将 UML 与 Booch、OMT、OOSE 等建模方法作了比较。通过这一章的学习读者将会对 UML 有一个总的了解。第 2 章“面向对象的技术”，主要介绍了面向对象的知识。之所以把面向对象单独作为一章来论述，是因为 UML 是基于面向对象进行设计的。在这一章中讨论了面向对象程序设计中范型的概念、面向对象的抽象原理、对象和类、对象的生命周期、面向对象的核心特征以及面向对象的优点等。第 3 章“运用面向对象的思想”，主要介绍了实际中如何定义一个类、类的属性、类的操作、重载操作、属性和操作的可视性、类属性和类操作等。第 4 章“UML 语言”，介绍了 UML 语言中的标准元素、图形和符号等，同时还介绍了 UML 中模型的概念、通用机制和扩展机制等。

第 2 部分“UML 静态建模”，主要介绍了有关静态建模的知识，包括第 5～8 章。第 5 章“UML 用例图”，介绍了执行者、用例以及如何使用用例捕获用户需求的知识；第 6 章“类图”，介绍了类图的一般化、类图关联结构等；第 7 章“包图”，讲解了 UML 中包的基本含义、包的依赖、包的模型和子模型等；第 8 章“构件图和配置图”，分别介绍了 UML 构件图和配置图的知识。

第 3 部分“UML 动态建模”，主要讨论有关 UML 动态建模的知识，这部分包括第 9～12 章。第 9 章“消息”，主要介绍了 UML 中消息的定义、消息结构、消息参数、对象角色以及消息的类型等；第 10 章“交互图”，主要讲解了 UML 中协作和交互的概念，以及以这些概念为基础的顺序图和协作图的知识；第 11 章“活动图”，讲解了活动图的组成及其应用；第 12 章“状态图”，介绍了状态和状态机、事件和转换等知识。

第 4 部分“UML 高级应用”，主要介绍了 UML 中一些高级建模的知识，此外还包括几个例子，这部分包括第 13～17 章。其中第 13 章论述了 UML 建模的过程，介绍了 UML 建模的核心工作流程等；第 14 章“UML 的设计模式”，介绍了 UML 中模式的概念，讨论了模式的类别、组成元素、质量等概念；第 15 章“类型一致性”，主要讲解了 UML 中类型一致性的知识；第 16 章“UML 实例”，主要通过一个例子来系统阐明 UML 建模的过程；第 17 章“使用 UML 设计数据库应用”，通过一个例子来说明 UML 在数据库领域的广泛应用。

本书按从初级到高级、从基本概念到应用实例的顺序，逐渐进地对 UML 进行了详细地讲述，具有层次清楚，思路清晰，讲解透彻，实例丰富的特点，适合于广大软件开发

人员、系统分析人员、开发人员、科技人员自学使用，同时也适合于广大高等院校的师生选作教材或自学参考使用。我们相信本书一定会给您带来莫大的收获，这也是我们最大的欣慰。

编　者

2003.8

目　录

第 1 部分　基础知识

第 2 部分 UML 静态建模

第3部分 UML动态建模

第 4 部分 UML 高级应用

第 1 部分　基础知识

第 1 章　UML 基础

1.1　什么是 UML

统一建模语言（UML）是一种直观化、明确化、构建和文档化软件系统产物的通用可视化建模语言。它记录了被构建系统的有关决定和理解，可用于对系统的理解、设计、浏览、配置、维护以及信息控制。UML 可以与所有的开发方法、生命阶段、应用领域和媒介同时使用。它意图统一过去建模技术的经验，将当前软件最佳实践合并至标准的方法。UML 包括语义概念、标记符号和指南，具有静态、动态、环境上的和组织性的部分。它可以被具有代码产生和报表生成的交互式可视建模工具所支持。UML 规范没有定义标准过程，但可用于迭代的开发过程，并支持现有的大多数面向对象的开发过程。

UML 描述了一个系统的静态结构和动态行为。UML 将系统描述为一些离散的相互作用的对象，并最终为外部用户提供一定功能的模型结构。静态结构定义了系统中的重要对象的属性和操作以及这些对象之间的相互关系。动态行为定义了对象的时间特性和对象为完成目标而相互进行通信的机制。从不同但相互联系的角度对系统建立的模型可用于不同的目的。

UML 还包括用包来分解模型的组织性结构，它允许软件团队将系统分解为可工作的单元，对包之间的依赖进行理解和在复杂的开发环境中管理模型单元的版本。它包含了表达实现上的决策和用构件来组织运行时元素的结构。

UML 不是一门程序设计语言。但它可以使用代码生成器工具将 UML 模型转换为多种程序设计语言代码，以及使用反向生成器工具将现有的程序源代码转换成 UML，从而实现逆向构筑模型。UML 不是用于定理证明的高度正式的语言。实际上有很多正式的语言，但它们不易理解或不适用于多种用途。UML 是通用性的建模语言。对于一些专门的领域，例如用户图形界面（GUI）设计、超大规模集成电路（VLSI）设计或基于规则的人工智能领域，使用专门的语言和工具可能更加合适，UML 是离散的建模语言，它不打算对如工程和物理的连续系统建模。UML 是对诸如软件、硬件或数字逻辑的离散系统建模的通用语言。

1.2　UML 的基本组成

UML 由图和元模型共同组成。其中图是 UML 的语法，而元模型则是给出的图的意思，它是 UML 的语义。UML 的语义是定义在一个 4 层抽象级建模概念框架中的，这样

它便形成一个 4 层结构，这 4 层分别是：

(1) 元元模型（meta-metamodel）层。UML 元元模型描述基本的元元类型、元元属性、元元关系，这些都用于定义 UML 元模型。元元模型实现了下述要求的一个基本设计：强调使用少数功能较强的建模成分，而这些成分易于组合起来表达复杂的语义。尽管在本文档中元元模型的提出和定义 UML 元模型有关，但它被设计在一个方法和技术都相对独立的抽象层次上。因此，它可以用于其他用途，例如定义库 repositories 或者模型转换格式。

(2) 元模型（metamodel）层。该层组成了 UML 的基本元素，包括面向对象和面向组件的概念。这一层的每个概念都是元元模型中“事物”概念的实例。

(3) 模型（model）层。该层组成了 UML 的模型，这一层中的每个概念都是元模型层中概念的一个实例，这一层的模型通常叫做类模型（class model）或类型模型（type model）。

(4) 用户模型（user model）层。该层中的所有元素都是 UML 模型的例子。这一层中的每个概念都是模型层的一个实例，也是元模型层的一个实例。这一层的模型通常叫做对象模型（object model）或实例模型（instance model）。

表 1-1 总结了这 4 层的基本情况。

表 1-1 UML 语义的 4 层结构

层	说 明	例 子
元元模型	元建模体系结构的基础构造。定义了描述元模型的语义	元类、元属性、元操作
元模型	元元模型的实例。定义了描述模型的语义	类、属性、操作、构件
模型	元模型的实例。定义了描述信息论域的语义	StockShare,askPrice, sellLimitOrder,StockQuoteServer
用户模型（对象模型）	模型的实例。定义了一个特定的信息论域	<Acme_Software_Share 98789>, 654.56,sell_limit_order, <Stock_Quote_Svr 32123>

UML 使用模型来描述系统的结构或静态特征，以及行为或动态特征。它从不同的角度为系统的架构建模，并形成系统的不同视图（view），这些视图包括如下几种：

(1) 用例视图（use case view）。该视图强调以从用户的角度所看到的或需要的系统功能为出发点建模。这种视图有时也被称为用户模型视图（user model view）或想象视图（scenario view）。

(2) 逻辑视图（logical view）。该视图用于展现系统的静态或结构组成及其特征，它也被称为结构模型视图（structural model view）或静态视图（static view）。

(3) 并发视图（concurrent view）。该视图体现了系统的动态或者行为特征，它也被称为行为模型视图（behavioral model view）、过程视图（process view）、协作视图（collaborative view）或者动态视图（dynamic view）。

(4) 组件视图（component view）。该视图体现了系统实现的结构和行为特征，它有