UML

系统分析设计与应用案例

冀振燕　编著

人民邮电出版社
POSTS & TELECOM PRESS

UML

系统分析设计与应用案例

冀振燕 编著

人民邮电出版社

图书在版编目（CIP）数据

UML 系统分析设计与应用案例/冀振燕编著.—北京：人民邮电出版社，2003.6
ISBN 7-115-10927-3

Ⅰ. U... Ⅱ. 冀... 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 034627 号

内容提要

本书介绍了 UML 语言的基础知识以及 UML 在面向对象的软件系统分析和设计中的应用，并通过实例讲解了系统的面向对象分析与设计过程，以及如何用 UML 语言为系统建模。此外，还介绍了如何使用 Rational Rose 2000 中的前向工程和逆向工程。

本书结合了丰富的实例，通过实例启发读者如何将所学到的面向对象技术应用于软件系统的分析、设计与开发中。

本书是一本内容全面的面向对象技术书籍。可作为软件设计与开发人员的参考手册，也可作为大专院校做面向对象分析与设计课程的教材使用。

UML 系统分析设计与应用案例

◆ 编　　著　冀振燕
责任编辑　屈艳莲　邹文波

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
读者热线　010-67132692
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：20.75
字数：501 千字　　2003 年 6 月第 1 版
印数：1-5 000 册　　2003 年 6 月北京第 1 次印刷

ISBN 7-115-10927-3/TP・3246

定价：29.00 元

前　言

20 世纪 90 年代初，不同的面向对象方法具有不同的建模符号体系，这些不同符号体系极大地妨碍了软件的设计人员、开发人员和用户之间的交流。因此，有必要在分析、比较不同的建模语言以及总结面向对象技术应用实践的基础上，建立一个标准的、统一的建模语言。UML 就是这样的建模语言，UML 1.1 于 1997 年 11 月 17 日被对象管理组织（OMG）采纳成为基于面向对象技术的标准建模语言。

统一建模语言 UML 不仅统一了 Grady Booch、James Rumbaugh 和 Ivar Jacobson 所提出的面向对象方法中的符号表示，而且在其基础上进一步发展，并最终被统一为被大众所接受的标准建模语言。

UML 是可视化（Visualizing）、规范定义（Specifying）、构造（Constructing）和文档化（Documenting）的建模语言。可视化模型的建立为设计人员、开发人员、用户和领域专家之间的交流提供了便利；规范定义意味着用 UML 建立的模型是准确的、无二义的、完整的；构造意味着可以将 UML 模型映射到代码实现；UML 还可以为系统的体系结构以及系统的所有细节建立文档。

UML 语言目前已成为面向对象软件系统分析与设计的必要工具，是软件设计、开发人员的必备知识。

本书的内容由两部分组成：基础篇和实践篇。基础篇包括第 1～11 章，主要介绍了 UML 的基础知识，并简要介绍了 Rational 统一过程和常用的面向对象分析与设计方法。实践篇包括第 12～17 章。通过实例讲解了系统的面向对象分析与设计过程，以及如何用 UML 语言为系统建模。此外还介绍了如何使用 Rational Rose 2000 中的前向工程和逆向工程。

本书最大的特点是结合了丰富的实例，从而使对理论的讲解生动而不抽象。同时，通过实例启发读者如何将所学到的面向对象技术应用于软件系统的分析、设计与开发中。本书是基于作者在国外进行面向对象技术教学的丰富经验精心编写的，兼顾了理论与实际应用，是一本内容全面的面向对象技术书籍。本书不但可作软件设计与开发人员的参考手册，也可作为大专院校做面向对象分析与设计课程的教材使用。

国家开发银行的刘新建参与了本书的写作过程，并提供了不少宝贵的意见，对本书的写作给予了很大的理解和支持，特此感谢。Bertil Andersson，Pär-Ove Forss, Börje Hansson，Birgitta Strömberg，Jenny Bergfors，Mattias Lingesten，Roger Mårtensson，Henrik Björnström 为本书的写作提供了 Rational Rose 等软件支持，张国宝、蒋键、施运梅、朱伟、魏有玉、李晓伟、赵振军、谭燕、高雅洁、赵竞等为书稿绘制了图片，在此致以诚挚的感谢。

作者

2003 年 4 月

目 录

第一部分 基础篇

第二部分 实践篇

第一部分

基础篇

本部分由第 1～11 章组成。

- 第 1 章介绍了 UML 的历史、内容、特点、功能、组成和工具，还对 Rational 统一过程进行了较详细的介绍。
- 第 2 章介绍了常用的面向对象分析与设计方法：OOA/OOD 方法、OMT 方法、Booch 方法、OOSE 方法和 Fusion 方法。
- 第 3 章讲解了 UML 语言的 4 种关系——依赖关系、类属关系、关联关系和实现关系的语义、符号表示、原型和应用。
- 第 4 章讲解了 UML 各种符号的语义、符号表示、应用，还讲解了 UML 的扩充机制。
- 第 5 章介绍了视与图的关系，从而指导读者选择 UML 图为系统 5 个视的静态方面和动态方面建模。
- 第 6～11 章详细描述了 UML 的 9 种图——用例图、类图、对象图、时序图、协作图、活动图、状态图、组件图和配置图，详细介绍了它们的语义、功能和应用。

第1章 绪论

1.1 统一建模语言 UML

面向对象分析与设计（Object-Oriented Analysis and Design，简称 OOA&D 或 OOAD）方法的发展在 20 世纪 80 年代末至 20 世纪 90 年代中出现了一个高潮，UML 就是这个高潮的产物。UML 不仅统一了 Grady Booch、James Rumbaugh 和 Ivar Jacobson 所提出的面向对象方法中的符号表示，而且在其基础上进一步发展，并最终被统一为被开发者所接受的标准建模语言。

1.1.1 UML 的背景

面向对象方法出现于 20 世纪 70 年代中期，从 1989 年到 1994 年，面向对象方法从不到 10 个增加到 50 多个，这些不同的面向对象方法具有不同的建模符号体系，这些建模语言各有优劣，用户很难找到一个完全满足自己要求的模型语言。另外，由于采用不同的建模语言，极大地妨碍了软件设计人员、开发人员和用户之间的彼此交流。因此，有必要在分析、比较不同的建模语言以及总结面向对象技术应用实践的基础上，博采众长，建立一个标准的、统一的建模语言。

20 世纪 90 年代，3 个最流行的面向对象方法是：OMT 方法（由 James Rumbaugh 提出）、Booch 方法（由 Grady Booch 提出）和 OOSE 方法（由 Ivar Jacoboson 提出），每个方法都有自己的价值和重点。分析是 OMT 方法的强项，设计是 OMT 方法的弱项；设计是 Booch 91 方法的强项，分析是 Booch 91 方法的弱项；Jacobson 擅长行为分析，而在其他方面比较弱。

在 20 世纪 90 年代中期，Grady Booch、Ivar Jacobson、James Rumbaugh 开始借鉴彼此的方法，Booch 采用了 James Rumbaugh 和 Ivar Jacobson 所提出的许多很好的分析技术，James Rumbaugh 的 OMT-2 采用了 Booch 所提出的很好的设计方法。但是，不同符号体系的使用给市场带来了混乱，因为同一个符号对于不同的人可能意义不同，而同一个事物可能用不同的符号表示，因此引起了很多混乱，人们用“方法大战”形象地描述了这种混乱局面。

统一建模语言 UML 的诞生结束了符号方面的“方法大战”。UML 统一了 Booch 方法、OMT 方法、OOSE 方法的符号，并采纳了其他面向对象方法的许多好的概念。

1.1.2 UML 的发展

UML 的建立开始于 1994 年 10 月，那时 James Rumbaugh 加入了 Grady Booch 所在的 Rational 公司，他们的 UML 项目主要统一了 Booch 方法和 OMT 方法，并于 1995 年 10 月发布了 Unified Method 0.8（这是 UML 当时的名字）。大约同时，OOSE 的创始人 Ivar Jacobson 加入了 Rational 公司，他们开始在 UML 项目中加入 OOSE 方法，并于 1996 年 6 月将 UM 改名为 UML（Unified Modeling Language），并发布了 UML 0.9。1996 年 10 月，发布了 *UML0.91*。1996 年，一些机构已日趋明显地将 UML 作为其商业策略。UML 的开发者得到了来自用户的

正面反应，倡议成立了 UML 协会，以完善、加强和促进 UML 的规范工作。定义 UML 1.0 时，DEC、HP、I-Logix、Intellicorp、IBM、ICON 计算（ICON Computing）、MCI Systemhouse、Microsoft、Oracle、Rational、Texas 仪器（Texas Instrumnets）、Unisys 等公司都参与了该项工作。此时，UML 在美国获得了工业界和科技界的广泛支持，已有 700 多个公司表示支持采用 UML 作为建模语言。1996 年底，UML 已稳占面向对象技术市场 85%的份额，成为可视化建模语言事实上的工业标准。

UML 1.0 是定义完整、富于表达、功能强大的建模语言，它于 1997 年 1 月被提交给 OMG（Object Management Group，对象管理组织），申请成为标准建模语言。

1997 年 1 月至 1997 年 7 月，Andersen 咨询、Ericsson、ObjectTime 有限公司、Platinum 技术、PTech、Reich 技术、Softteam、Sterling 软件和 Taskon 公司加入了 UML 组。1997 年 7 月，UML 的修正版 UML 1.1 被提交给 OMG。1997 年 11 月 17 日，OMG 采纳了 UML 1.1 作为基于面向对象技术的标准建模语言。1998 年 6 月，OMG RTF（Revision Task Force）发布了 UML 1.2。1998 年秋天，OMG RTF 发布了 UML 1.3。1999 年 8 月发布了 UML 2.0。UML 的发展历程如图 1.1 所示。

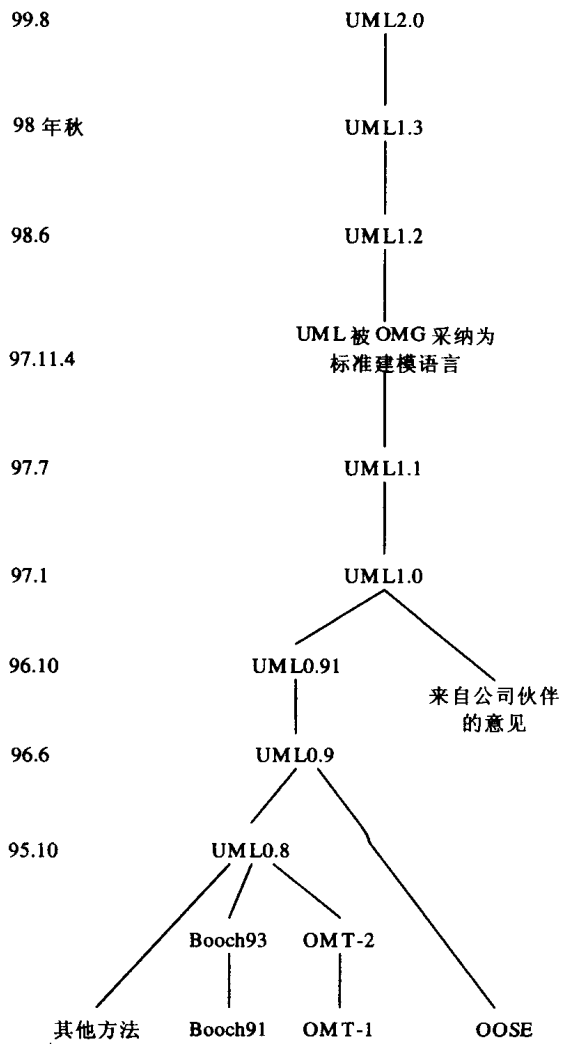

图 1.1 UML 的发展历程

统一了的建模语言 UML 是一种定义良好、富于表达、功能强大且普遍适用的建模语言。它溶入了软件工程领域的新思想、新方法和新技术。它不但支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。

统一建模语言 UML 代表了面向对象软件开发技术的发展方向，具有巨大的市场前景，也具有重大的经济价值。

1.1.3　UML 的内容

作为一种建模语言，UML 的定义包括 UML 语义和 UML 表示法两个部分：

（1）UML 语义

UML 语义给出了基于 UML 的精确的元模型定义。元模型为 UML 的所有元素在语法和语义上提供了简单、一致、通用的定义性说明，使开发者能在语义上取得一致，消除了因人而异的表达方法所造成的影响。此外 UML 还支持对元模型的扩充定义。

（2）UML 表示法

UML 表示法定义了 UML 符号的表示方法，为开发者或开发工具使用这些图形符号和文本语法给系统建模提供了标准。这些图形符号和文字所表达的是应用级的模型，在语义上它是 UML 元模型的实例。

1.1.4　UML 的主要特点

统一建模语言 UML 的主要特点可以归纳如下。

（1）UML 统一了 Booch、OMT、OOSE 和其他面向对象方法的基本概念和符号。同时，UML 汇集了面向对象领域中很多人的思想，如图 1.2 所示。这些思想是 UML 的创始者依据最优秀的面向对象方法和丰富的计算机科学实践经验综合提炼而成的。

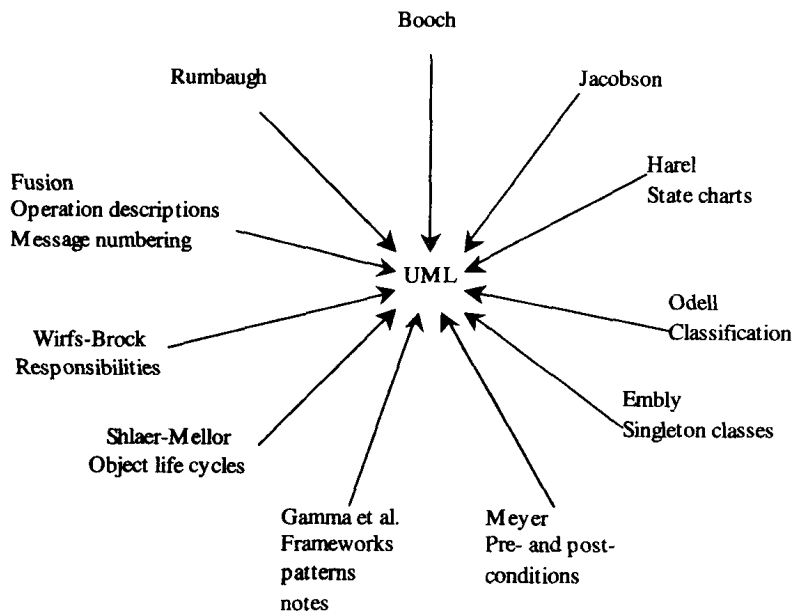

图 1.2　UML 的形成

（2）目前，UML 是一种先进实用的标准建模语言，但其中某些概念尚待实践来验证，UML 的发展存在着一个进化过程。

（3）UML 是一种建模语言而不是一种方法。这是因为 UML 中没有过程的概念，而过程正是方法的一个重要组成部分。UML 本身独立于过程，这意味着用户在使用 UML 进行建模时，可以选用任何适合的过程。过程的选用与软件开发过程的不同因素有关，诸如所开发软件的种类（如实时系统、信息系统和桌面产品）、开发组织的规模（如单人开发、小组开发和团队开发）等。用户将根据不同的需要选用不同的过程。但是，使用 UML 建模仍然有着大致统一的过程框架，该框架包含了 UML 建模过程中的共同要素，同时又为用户选用与其所开发的工程相适合的建模技术提供了很大的自由度。UML 只是一种语言，是独立于过程的，但是最好将它应用于用例驱动的、以体系结构为中心的、迭代的、递增的过程。

1.1.5 UML 的功能

UML 是一种建模语言，该语言具有如下功能。

1. 为软件系统的产出（Artifacts）建立可视化模型

对于大多数程序员来说，在脑海中设想一个软件的实现与用代码来实现这个软件是没有距离的，怎么想，就怎么用代码来实现它。事实上，有些东西最好直接转化为代码，因为文本是写表达式和算法的最直接的方式。但过去即使在这种情况下，程序员仍就要做一些简单的建模工作，例如在白板或纸上画一些简单的模型，这种做法会产生下列问题：

（1）不利于交流。如果公司或项目组使用他们自己的语言来描述模型，这个模型很难为公司或项目组外的人员所理解。

（2）如果不建立模型，软件系统中的有些东西很难用文本的编程语言来表达清楚。

（3）如果程序员在修改代码时，没有将他脑海中的模型记录下来，这个信息可能会永远丢失，不便于软件维护。

而统一建模语言 UML 很好地解决了这些问题。

（1）UML 符号具有定义良好的语义，不会引起歧义。UML 是一个标准的、被广泛采用的建模语言，因此，用 UML 建模有利于交流。

（2）UML 是可视化的建模语言，它为系统提供了图形化的可视模型，使系统的结构变得直观，易于理解。

（3）用 UML 为软件系统建立模型不但有利于交流，还有利于对软件的维护。

2. 规约软件系统的产出（Artifacts）

规约（Specifying）意味着建立的模型是准确的、无歧义的、完整的。UML 定义了在开发软件系统过程中所做的所有重要的分析、设计和实现决策的规格说明。

3. 构造软件系统的产出（Artifacts）

UML 不是可视化的编程语言，但它的模型可以直接对应到各种各样的编程语言，也就是说，可以从 UML 的模型生成 Java、C++、Visual Basic 等语言的代码，甚至还可以生成关系数据库中的表。

从 UML 模型生成编程语言代码的过程被称为前向工程（Forward Engineering），从代码实现生成 UML 模型的过程被称为逆向工程（Reverse Engineering）。目前许多 CASE 工具，如 Rational Rose 和 Prosa 等，都既支持前向工程又支持逆向工程。

4. 为软件系统的产出（Artifacts）建立文档

在软件的开发过程中为软件系统建立清晰、完整、准确的文档是非常重要的。UML 可以为系统的体系结构及其所有细节建立文档。UML 还为描述需求、测试、项目规划活动和软件发布管理活动的建模提供了语言。

软件系统的产出（Artifacts）包括下面这些元素（但不限于这些元素）：

- 需求。
- 体系结构。
- 设计。
- 源代码。
- 项目计划。
- 测试。
- 原型。
- 发布。

1.1.6 UML 的组成

UML 的词汇表包括 3 种构造模块：元素、关系、图。元素是模型中重要的抽象；关系将这些元素连接起来；而图则将元素的集合分组。

1. 元素

UML 中的元素又可分为结构元素、行为元素、分组元素、注释元素 4 种。

- 结构元素

结构元素是 UML 模型中的名词。结构元素是模型中主要的静态部分，代表了概念的或物理的元素。在 UML 中，共有 7 种结构元素：类（Class）、接口（Interface）、协作（Collaboration）、用例（Use Case）、活动类（Active Class）、组件（Component）和节点（Node）。

- 行为元素

行为元素是 UML 模型中的动态部分，它们是模型中的动词，代表了跨越时间和空间的行为。在 UML 中，有两种主要的行为元素：交互作用（Interaction）和状态机（State Machine）。

交互作用是由在特定上下文中为完成特定目的而在对象间交换的消息集组成的行为。交互作用包括许多其他元素：消息、动作序列（由消息激活的行为）、连接（对象间的连接）。

状态机是这样一种行为，这种行为规定了对象在其生命周期为响应事件而经历的状态序列，以及对事件的响应。状态机也包括许多其他元素：状态、跃迁、事件、和活动。

- 分组元素

分组元素是 UML 模型中用来组织元素的元素。在 UML 中，有一种主要的分组元素：包（Package）。