

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
软件工程

统一建模语言UML

袁涛 孔蕾蕾 编著

清华大学出版社



高等学校教材
软件工程

统一建模语言UML

袁涛 孔蕾蕾 编著

清华大学出版社
北京

内 容 简 介

本书是学习面向软件分析与设计的和应用 UML 2.0 的教材。它不仅详细阐述了 UML 在建模活动中的基本应用方法,而且对 UML 的建模图示在软件生命周期中的应用进行了分类。在介绍 UML 10 种最为重要的语言图示时,强调该 UML 图示在软件建模活动中的建模目的,并根据 UML 各种图示的语法结构详细解释该图示在实际建模中的不同表示形式和语法,最后应用 UML 图示对一个贯穿全书的真实软件工程项目实例进行建模示范,使读者在深入理解 UML 语义、语法和图示法的同时,能牢牢把握住学习该 UML 图示的目的和意义。本书可作为高等学校计算机、电子、通信等专业高年级学生及研究生课程教学用书,同时对软件研究者和开发人员亦颇具参考价值。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

统一建模语言 UML/袁涛,孔蕾蕾编著. —北京:清华大学出版社,2009.5

(高等学校教材·软件工程)

ISBN 978-7-302-19438-5

I. 统… II. ①袁… ②孔… III. 面向对象语言,UML—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 015843 号

责任编辑:丁 岭 李玮琪

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京嘉实印刷有限公司

装 订 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:7.75 字 数:181 千字

版 次:2009 年 5 月第 1 版 印 次:2009 年 5 月第 1 次印刷

印 数:1~3000

定 价:20.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系
调换。联系电话:(010)62770177 转 3103 产品编号:031917-01

献给：

Derek, Karen, 我的妻子和我的父母

——袁涛

改改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学

科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

- (1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 高等学校教材·信息管理与信息系统。
- (6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过 20 多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

在回国的这几年里,我一直在努力使用 UML 与软件项目开发者、我的学生以及同事进行软件的分析 and 设计方面的交流。但是,我发现周围还有许多软件设计和开发人员并不熟悉 UML 这个在软件工程领域已经成熟应用十几年的建模工具。在软件工程实践中,甚至有相当一部分软件设计人员、程序员和学生还在为是否学习和使用 UML 而困惑。我认为造成这种困惑的主要原因是对于 UML 的应用目的和它的建模对象不十分清楚,这就使 UML 使用者或初学者无法有效地把 UML 中的建模语言与实际软件开发中的问题建立起关联。因此,本书不仅在 UML 语法方面给予详细的描述,而且在每种 UML 语言图示中着重阐述了图示的产生环境、使用目的和应用对象。为了更好地理解本书的组织结构和目的,我们在以下 3 个方面进行了论述和规范。

1. 关于 UML 建模图示的应用分类问题

对 UML 中诸多的建模图示,人们对它有着不同的分类和建模理解,例如一种很常见的分类是把 UML 的建模图示分为需求、静态、行为、交互和实现等几个不同领域的建模工具,但是,上述分类方式很容易给 UML 学习和使用带来困惑,因为上述的几个领域在软件工程中几乎是完全交叉的,并不能帮助 UML 使用者明确 UML 建模如何与实际开发相关联。

本书在第 1 章引言中,较为系统地论述了 UML 建模图示的分类方法及其在软件开发和运行中的固有特征。根据建模工具的特点和软件固有特征,对 UML 的 14 种建模图示在软件生命周期中的应用进行了分类。我们建议本书的读者首先要理解引言中关于 UML 建模工具分类的方法和目的,然后,再以该分类方法为知识框架,进一步学习 UML 每一种具体的建模方法。

2. 关于面向对象分析和设计中术语使用问题

建模的过程就是对一个事物的一个抽象化和准确化的过程。在面向对象分析和设计中,对各种各样建模对象的描述必须规范化。注意,这里提出的术语规范化并不是 UML 语言,而是被 UML 建模语言描述的面向对象分析和设计中描述软件结构

和行为的语言。

例如,在软件工程中经常遇到的术语:软件、系统、类、对象、实例、方法、属性、操作、行为、状态、成员变量、消息、静态、动态、运行和执行等。其中,比较容易被混淆通用的如:方法、属性、操作、行为、状态、成员变量、消息、静态、动态和运行等。应用 UML 的建模语言建模时,这种没有严格定义指导下的术语混用,很难明确 UML 建模的目的和对象,这使得在使用模型进行交流时给人们带来极大的不准确性,从而造成在不同类型的模型中,或在同类模型中由于所被描述对象的术语混用而出现理解上的差异。本书为了使读者准确理解书中强调的知识体系结构,特在此建立本书范围内的面向对象分析和设计术语的应用规范。

(1) 与软件系统相关的术语

软件系统是指一个具有整体功能的软件,它与构件和类相区别。在软件系统的概念下,有两种状态:非执行状态(或静止状态)、执行状态(或运行状态)。本书不使用“动态”这个词。在本书中,软件系统只有在运行状态下才有行为可言,但是,无论在静止还是运行状态下,软件系统均有各自特殊的结构形式。

(2) 与类相关的术语

类是指软件在非运行状态下的基本结构单位,它与对象相区别。在类的概念范围内,本书使用描述类的术语有:属性(Attribute)和成员变量(Member Variable),这两个词基本可以相互代替使用;方法(Method)和操作(Operation)也可以相互代替,但是本书只用方法(Method)这个词汇来描述类。

(3) 与对象相关的术语

对象是类在系统执行状态下的存在形式。它与类相区别。在对象的概念范围内,本书使用描述对象的术语有:属性(Attribute)和状态(State),这两个词在本书中可以代替使用;行为(Behavior)和消息(Message),在本书中这两个词可以互相代替使用描述对象。

3. 本书章节的组织 and 内容特点

在本书中,每章的第一节讲述的是 UML 图示的目的和意义,这样安排的目的是让读者在学习某种具体 UML 图示建模之前,了解该图示的应用领域和建模对象,以便在进一步学习图示语法时,有助于更好地了解图示中建模方法的设计理念,以便读者能有的放矢地学习该建模工具。在学习 UML 时,学生经常提出一些典型问题,例如,顺序图与通信图的区别;在类图中,关联(Association)和依赖(Dependency)的实践差别问题;类图中 xor 关联的实现问题等等。针对这些问题,本书均给出详尽解释,另外也提供了一些 UML 建模中的实施技巧。总之,本书不仅对 UML 语法进行解释,而且在各个章节中尽量加入我们平时应用 UML 时积累的经验和方法,这更有助于读者快速理解和应用 UML 建模。

在描述 UML 语法过程中,本书针对每种 UML 建模图示都以公式的形式把该建模图示最为重要的组成元素列出,然后,根据公式中列出的每个元素做出详细解释,这样可以使读者在纷乱的 UML 图示元素符号中把握其知识体系结构。

本书是以 UML 2.0 为基础阐述其建模语言的,没有关于与 UML 2.0 以前版本的比较。所以,书中提到的 UML,指的是 UML 2.0 版。另外,根据 UML 的各种不同建模语言应用的普遍性,本书没有对 UML 2.0 新引进的时间配置图、综合互交图和复合结构图进行专门阐述。

袁涛负责全书所有章节内容的组织,并完成第 1、3、4、7、8、10 章的内容撰写;孔蕾蕾负责第 2、5、6、9、11 章,以及附录 A、B 和术语对照表的撰写。

最后我要强调的是,这本书能够问世还要特别感谢哈尔滨商业大学校长曲振涛博士的支持,同时也感谢我的好友穆业伟先生对本书出版的关心。

袁 涛

2008 年 11 月

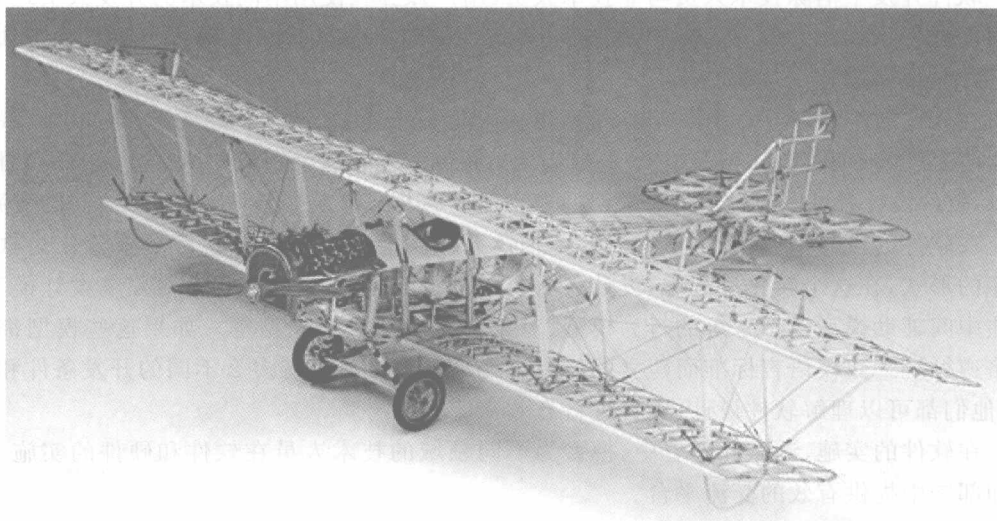
第 1 章 导言	1
1.1 模型	1
1.2 开发软件为什么需要模型	2
1.3 什么是统一建模语言	2
1.4 UML 的发展史	4
1.5 UML 的建模对象	5
1.5.1 UML 的结构模型	5
1.5.2 UML 的行为模型	6
1.6 总结	7
第 2 章 用例图	8
2.1 基于用例的系统行为建模	8
2.2 用例图	9
2.3 用例图的表示方法	9
2.3.1 参与者	9
2.3.2 用例	9
2.3.3 用例之间的关系	12
2.4 总结	15
第 3 章 对象图	17
3.1 基于对象的系统瞬间状态建模	17
3.2 对象图	18
3.3 对象图的表示方法	18
3.3.1 对象	18
3.3.2 链	19
3.4 总结	19

第4章 顺序图	20
4.1 基于交互的对象行为建模：交互时的行为顺序	20
4.2 顺序图	21
4.3 顺序图的表示方法	21
4.3.1 生命线	21
4.3.2 活动条	22
4.3.3 消息	23
4.3.4 交互框	28
4.4 案例分析	31
4.5 总结	33
第5章 通信图	34
5.1 基于交互的对象行为建模：交互时的对象结构	34
5.2 通信图	35
5.3 通信图的表示方法	35
5.3.1 交互的参与者	35
5.3.2 链接	36
5.3.3 消息	36
5.4 案例分析	38
5.5 总结	38
第6章 类图	40
6.1 基于类的系统结构建模	40
6.2 类图	41
6.3 类图的表示方法	41
6.3.1 表示类	41
6.3.2 类的关系	47
6.4 总结	57
第7章 状态图	58
7.1 基于状态的对象行为建模	58
7.2 状态图	59
7.3 状态图的表示方法	59
7.3.1 状态	59
7.3.2 迁移	61
7.4 案例分析	63

7.5 总结	64
第 8 章 活动图	65
8.1 基于活动的系统行为建模	65
8.2 活动图	66
8.3 活动图的表示方法	66
8.3.1 活动和动作	66
8.3.2 活动边	67
8.3.3 活动节点	68
8.3.4 活动划分或泳道	72
8.3.5 调用其他活动	73
8.4 案例分析	73
8.5 总结	74
第 9 章 包图	76
9.1 基于包的系统静止状态下的结构建模	76
9.2 包图	77
9.3 包图的表示方法	77
9.3.1 包	77
9.3.2 包中元素的可见性	78
9.3.3 包之间的关系	78
9.4 总结	81
第 10 章 构件图	83
10.1 基于构件的系统静止状态下的结构建模	83
10.2 构件和构件图	84
10.2.1 构件	84
10.2.2 构件图	84
10.3 构件图的表示方法	84
10.3.1 构件	84
10.3.2 供接口和需接口	85
10.3.3 构件间的关系	86
10.3.4 实现构件的类	87
10.3.5 外部接口——端口	87
10.3.6 连接器	87
10.3.7 显示构件的内部结构	88
10.4 总结	89

第 11 章 部署图	90
11.1 基于物理环境的系统执行状态下的结构建模	90
11.2 部署图	91
11.3 部署图的表示方法	91
11.3.1 制品	91
11.3.2 节点	92
11.3.3 部署	93
11.3.4 部署规约	93
11.3.5 通信路径	94
11.4 总结与强调	96
附录 A UML 的扩展机制	97
附录 B PPS 项目的部分主要用例的用例规约	99
术语英汉对照表	104
参考文献	108

导 言



1.1 模 型

为了更好地了解一个过程或事物,人们通常根据所研究对象的某些特征(形状、结构或行为等)建立相关的模型(Model)。模型是从一个特定视点对系统进行的抽象,它可以是实物模型,例如建筑模型、教学模型、玩具等,也可以是抽象数字或图示模型,例如数学公式或图形等。模型建立的目的是不是复制真实的原物,而是帮助人们更好地理解复杂事物的本质,反映过程或事物内部各种因素之间的相互关系。所以,模型是对复杂事物进行的有目的的简化和抽象。在开发软件的过程中同样需要建立各种各样的软件模型。

1.2 开发软件为什么需要模型

在开发软件的过程中,开发者在动手编写程序之前需要研究和分析软件的诸多复杂和纷乱的问题。例如,用户需求的准确描述问题、功能与功能之间的关系问题、软件的质量和性能问题、软件的结构组成问题、建立几十个甚至几百个程序或组件之间的关联问题等。所以,软件系统的开发是一个非常复杂的过程,它们的复杂程度不比任何一项大型的复杂土木建设工程逊色。但是,在这个复杂的开发过程中,我们最关注的还是开发者之间的交流问题。

软件开发中消除技术人员与非技术人员(用户)之间、使用不同技术的开发人员之间、不同功能使用者之间等交流障碍是软件开发成功的关键。直观的软件模型将有助于软件工程师与他们进行有效地交流。

在软件的需求分析中,用户和系统所属领域的专家更熟悉将要构建的系统的功能,我们称他们为领域专家(Domain Expert)。他们提出软件系统在这个领域中所需要具有的功能。所以,软件设计者可以通过建立需求模型来实现技术人员与非技术人员(用户)之间的交流。

在软件的设计中,设计人员首先要把描述系统功能需求的自然语言形式转化为软件程序的形式,在这个转化过程中,设计人员要借助许多模型来完成最终的程序设计模型。这些中间辅助模型包括系统的行为模型、对象的状态和行为模型等。如果这些模型都是严格遵循统一建模语言标准而建立的,那么,无论开发人员具有多么不同的开发条件和技能,他们都可以理解软件设计,并且进行有效交流。

在软件的实施、测试和部署中,模型为不同领域的技术人员在软件和硬件的实施、测试和部署中提供有效的交流平台。

最后,要强调的是,在各种各样的软件中,软件模型是最有效的软件文档保存形式,软件模型在开发团队人员的培训、学习和知识的传递与传播等方面起着非常重要的作用。

所以,软件开发中需要建立需求(Requirement)模型、问题域(Domain)模型、设计(Design)模型、实施(Implementation)模型、测试(Test)模型和部署(Deployment)模型。可见,在系统开发生命周期中,需要从多角度来建立模型才能全面、准确地分析和设计软件系统。

1.3 什么是统一建模语言

软件模型有多种表达方式或语言。但是,开发者们经过多年的实践发现,相对于以数学为基础的体系结构描述语言(Architecture Description Language, ADL)来说,以图形符号(Graphical Notation)为基础的统一建模语言(Unified Modeling Language, UML)描述软件模型既简单又清晰。

统一建模语言是由一系列标准的图形符号组成的建模语言,它用于描述软件系统分析、设计和实施中的各种模型。UML的定义有两个主要组成部分:语义和表示法。

UML 的语义用自然语言和对象约束语言(Object Constraint Language,OCL)描述,UML 的表示法定义了 UML 的可视化标准表示符号,这决定了 UML 是一种可视化的建模语言。这些图形符号和文字用于建立应用级的模型,在语义上,模型是元模型(Metamodel)的实例。元模型是定义表达模型所用语言的模型,它定义了 UML 模型的结构。此外,UML 的定义还给出了语法结构的精确规约(Specification)。对于一般建模者,应重点掌握基本的概念与表示法,并熟练运用它们,这正是本书的目的。

图 1-1 给出了 UML 中各种图的分类。

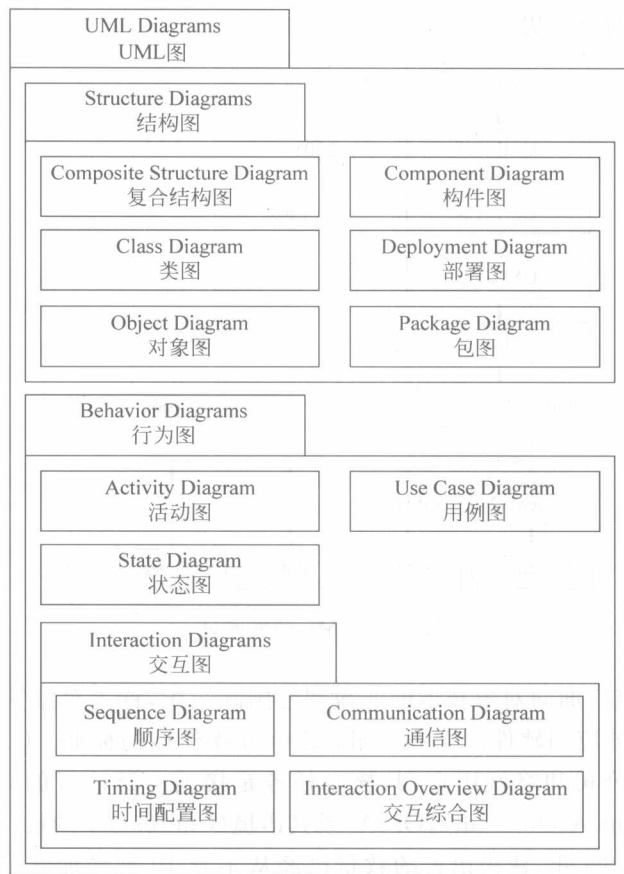


图 1-1 UML 图的分类

图 1-1 展示了 UML 的图示建模工具,被分为两大类共 13 种图形。

第一类是结构图。在结构图中,UML 2.0 有 6 种图示建模工具:类图(Class Diagram)、构件图(Component Diagram)、对象图(Object Diagram)、复合结构图(Composite Structure Diagram)、部署图(Deployment Diagram)和包图(Package Diagram)。

第二类是行为图。在行为图中,UML 2.0 有 7 种图示建模工具:活动图(Activity Diagram)、用例图(Use Case Diagram)、状态图(State Diagram)以及 4 种交互图

(Interaction Diagram)——顺序图 (Sequence Diagram)、通信图 (Communication Diagram)、交互综合图 (Interaction Overview Diagram) 和时间配置图 (Timing Diagram)。

1.4 UML 的发展史

自从对象管理组织 (Object Management Group, OMG) 采纳 UML 作为其标准建模语言以来, UML 得到了广泛的应用, 并在世界范围内成为了事实上的建模规范。图 1-2 简要描述了 UML 的发展史。

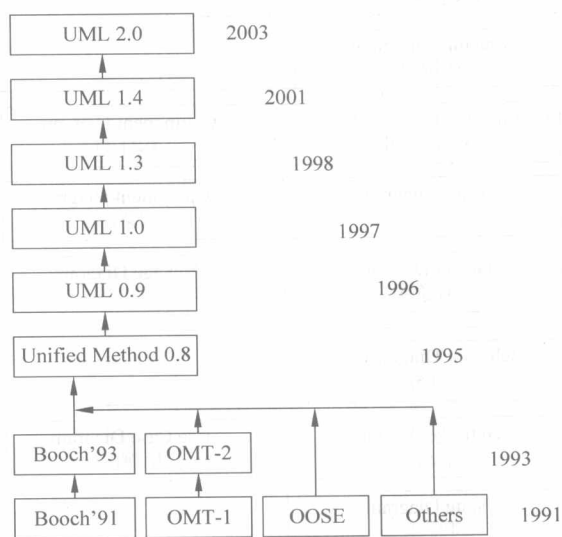


图 1-2 UML 的发展史

OMG 是一个推广面向对象技术的非盈利性国际组织, 称为对象管理组织, 目标是通过网络为独立开发的应用软件建立一个相互之间互操作性的标准。OMG 的成员已包括绝大多数信息技术公司和终端用户, 其核心任务是接纳广泛认可的对象管理体系结构 (Object Management Architecture, OMA) 或其语境 (Context) 中的接口和规程的规范。

从 1989 年到 1994 年, 建模语言的数量已经从不到 10 种增加到了 50 多种, 90 年代中期出现了一批新方法, 其中最引人注目的是 Booch 提出的 Booch'93、Jacobson 提出的 OOSE 方法和 Rumbaugh 等人提出的 OMT 方法。1995 年, Booch 和 Rumbaugh 决定将他们各自的方法结合起来成为一种方法。1995 年 10 月发布了第 1 个版本, 被称作“统一方法” (Unified Method 0.8)。一年之后 Jacobson 加入其中, 结合了 Booch、OMT 和 Jacobson 方法的优点, 统一了符号体系, 并从其他的方法和工程实践中吸收了许多经过实际检验的概念和技术, 于 1996 年形成了 UML 0.9。由于统一的符号体系只是一种建模语言, 而不是一种建模方法, 所以自 0.9 版起, 改称“统一建模语言” (Unified Modeling Language)。在此过程中, 由 Rational 公司发起成立了 UML 伙伴组织。开始时有 12 家公司加入, 共同推出了 UML 1.0 版, 并于 1997 年 1 月提交到对象管理组织 (OMG), 申请