全国高职高专计算机系列精品教材

UML统一建模项目教程

UML TONGYI JIANMO
XIANGMU JIAOCHENG

主　编/武雪芳　班娅萌

中国人民大学出版社

全国高职高专计算机系列精品教材

UML 统一建模项目教程

主　编　武雪芳　班娅萌
副主编　王志娟　魏宏昌　李　珩
参　编　王　茜　平金珍　赵月鹏

中国人民大学出版社
·北京·

图书在版编目（CIP）数据

UML 统一建模项目教程/武雪芳，班娅萌主编. —北京：中国人民大学出版社，2011.6
全国高职高专计算机系列精品教材
ISBN 978-7-300-13888-6

Ⅰ.①U… Ⅱ.①武…②班… Ⅲ.①面向对象语言，UML-程序设计-高等职业教育-教材
Ⅳ.①TP312

中国版本图书馆 CIP 数据核字（2011）第 109054 号

全国高职高专计算机系列精品教材

UML 统一建模项目教程

主　编　武雪芳　班娅萌
副主编　王志娟　魏宏昌　李　珩
参　编　王　茜　平金珍　赵月鹏

出版发行　中国人民大学出版社
社　　址　北京中关村大街 31 号　　**邮政编码**　100080
电　　话　010－62511242（总编室）　　010－62511398（质管部）
010－82501766（邮购部）　　010－62514148（门市部）
010－62515195（发行公司）　　010－62515275（盗版举报）
网　　址　http://www.crup.com.cn
http://www.ttrnet.com(人大教研网)
经　　销　新华书店
印　　刷　三河市汇鑫印务有限公司
规　　格　185 mm×260 mm　16 开本　　**版　　次**　2011 年 7 月第 1 版
印　　张　11.75　　**印　　次**　2011 年 7 月第 1 次印刷
字　　数　269 000　　**定　　价**　25.00 元

前　言

本书是关于 UML 统一建模的项目教学教材，介绍了软件工程基础知识和面向对象的分析与设计方法，并对 UML 的体系结构，UML 元素的语义、语法和 UML 应用进行了详细的讲解，重点突出了 UML 语言的表示方法和建模方法，内容精炼，表达简明，实例丰富，便于自学。本书最后应用 UML 图示对一个真实的软件工程项目实例——网络教学系统进行建模示范，介绍了系统开发的各个阶段，以及如何应用 UML 对系统进行建模，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。通过本书的学习，读者可以系统地掌握 UML 语言的阅读方法和建模方法。

在内容及章节编排上，本书充分考虑了高职高专学生的特点和专业需要，突出了以下几个方面的特色：

1. 面向高职高专学生，以够用和实用为度，重点突出了 UML 语言的表示方法和建模方法，表达简明，实例丰富。

2. 突出单元设计、项目教学的思路。每个单元由若干任务组成，每个任务都有具体的描述和最终需要达到的目标。

3. 对 UML 元素的讲解充分体现了文字描述和图形描述的结合，同时采用实例演示了 UML 元素的语义和使用方法，方便学生学以致用。

4. 每单元均设有教学目标、总结提要、概念复习、习题练习与思考等供学生复习和自学使用。

5. 每单元末设有拓展阅读专栏，可以使学生开阔视野，了解一些新的技术发展动态。

本书适于高职高专院校软件技术专业及相关专业使用，也可以作为计算机相关专业的培训教材，同时可供从事与软件建模工作相关的技术人员参考使用。

本书由武雪芳、班娅萌担任主编，王志娟、魏宏昌、李珩任副主编。本书共 13 个单元，武雪芳编写第 1、3、13 单元并负责全书的统稿工作，班娅萌编写第 2、8、9、10 单元，王志娟编写第 4 和第 5 单元，魏宏昌编写第 6、7 单元，李珩编写第 11、12 单元。另外，王茜、平金珍、赵月鹏三位老师也参与了本书的编写工作。

在本书的编写过程中，得到了有关领导和老师的大力支持和帮助，在此谨向他们表示感谢。

由于编者水平有限，书中错漏在所难免，不当之处，恳请读者批评指正。

编者

2011 年 5 月

目　录

第1单元　软件工程概述

单元描述

在近代技术的发展历史上，工程技术的进步一直是产业发展的巨大动力。特别是1946年世界上第一台电子计算机的诞生，标志着人类由工业化社会进入信息化社会，以计算机产业和计算机应用服务业为支柱的信息工业，成了信息化社会的主要基础之一。

20世纪70年代中期到80年代，计算机的应用有了更深入、更普遍的发展，人们对软件的需求量急剧增加。但此时计算机软件的开发技术却远远没有跟上硬件技术的发展，使得软件开发的成本逐年剧增，更为严重的是，软件的质量没有可靠的保证。软件开发的速度与计算机普及的速度不相适应，软件开发技术已经成为影响计算机系统发展的“瓶颈”。

在计算机系统发展的过程中，早期所形成的一些错误概念和做法曾严重地阻碍了计算机软件的开发。导致了20世纪60年代软件危机的发生。60年代后期，西方的计算机科学家开始认真研究解决软件危机的方法，提出借鉴工程界严密完整的工程设计思想来指导软件的开发与维护，并取得了可喜成果，从而诞生一门新的学科——软件工程学。

本单元通过对软件工程理论与方法的学习，深刻理解软件工程的目标与意义，在今后的软件开发中要实际运用软件工程的思想来指导自己，开发出优秀的软件产品。

单元目标

1. 理解软件的概念与特点。
2. 了解软件危机的形成并掌握缓解软件危机的方法与途径。
3. 了解软件工程的概念与基本目标。
4. 掌握软件生存周期的各个阶段需要完成的主要任务。
5. 掌握常用的软件开发模型。
6. 能够使用软件工程的思想来解决软件开发中遇到的问题。

1.1　任务1　软件与软件危机

任务描述

计算机软件的发展与进步，是与计算机硬件的发展和计算机的普及分不开的。同其他

事物的发展规律一样，也经历了从产生、发展到成熟的过程。在这个过程中也经历了软件的危机。为吸取历史经验教训，我们应该认真研究产生软件危机的原因，探讨消除软件危机的途径。

任务目标

首先了解软件的概念与软件的发展情况，并掌握软件的特点。其次，了解软件危机的形成与表现形式，并学会解决软件危机的方法与途径。

知识讲解

1.1.1 软件的发展

在计算机发展的初期，硬件的设计和生产是主要问题，那时的软件就是程序，甚至是机器指令程序，它们处于从属的地位。软件的生产方式是个体手工方式，设计过程是在一个人的头脑中完成的，程序的质量完全取决于个人的编程技巧。

随着计算机技术的发展，人们认识到在计算机上增加软件的功能会使计算机系统的功能得到大大提高，因此在编制大型程序系统时，不但要考虑硬件设备的配置，还要考虑与之相关的软件的功能，这样才能增强整个程序系统的功效。而且随着软件系统规模的逐渐扩大，软件不再是一个人编制完成，而需要多人合作。这时软件的生产方式是互助合作的手工方式，由于有多人合作编程，为了互相读懂程序，就需要增加说明书，所以这个时期软件的含义是“程序＋说明书”。

现代社会对计算机提出了更高的要求，有些大型系统的设计和生产的工作量高达几千人/年，指令达数百万条。软件在计算机系统中的比重越来越大。随着软件规模的增大，它的复杂度也在增加。软件可靠性往往随规模的增长而下降，质量保证也越来越困难，软件的发展速度远不能满足用户的需求，出现了软件危机。

人们感到传统的软件生产方式已不能适应发展的需要。因此，提出把工程学的基本原理和方法引入到软件生产中，像做传统工程那样，把软件生产分成几个阶段，每个阶段都有严格的管理和质量检验，研制软件设计和生产的方法及工具，并用书面文件作为共同遵循的依据，而且大型系统的数据需求量也非常大，所以需要用专门的数据结构来表示和处理。这时软件的含义就成为“程序＋数据＋文档”。

现在对软件的一种公认的解释为：软件是计算机系统中与硬件相互依存的另一部分，是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据为进行通信、解释和处理而使用的信息的形式化表现形式。文档是与程序开发、维护和使用有关的图文材料。

1.1.2 软件的特点

为了能全面、正确地理解计算机和软件，必须了解软件的特点。与硬件相比，软件主要有以下特点。

1. 软件是一种逻辑实体，不是具体的物理实体

硬件是有形有色、看得见摸得着的，而软件是无形无色、看不见摸不着的。软件正确

与否，是好是坏，一直要到程序在机器上运行才能知道，这给设计、生产和管理带来了许多困难。

2. 软件与硬件的生产方式不同

在软件的开发过程中没有明显的制造过程，也不像硬件那样，一旦研制成功就可以重复制造，可在制造过程中进行质量控制以保证产品的质量。软件是通过人们的智力活动，把知识与技术转化成信息的一种产品。一旦某一软件项目研制成功，以后就可以大量地复制同一内容的副本，应用到更多的地方。所以对软件的质量控制，必须着重在软件开发方面下工夫。

3. 软件与硬件的维护不同

硬件是有损耗的，会产生磨损和老化而使故障率增加甚至损坏，其解决的办法是换上一个相同的硬件。而软件不存在磨损和老化的问题，但却存在退化的问题。在软件的生存期中，为了使它能够克服以前没有发现的故障、适应硬件和软件环境的变化以及用户新的要求，必须要多次修改软件，而每次修改都不可避免地引入新的错误，随着一次次地修改，导致软件失效率升高，从而使软件退化。

4. 软件是复杂的

有人认为，人类能够创造的最复杂的产物是计算机软件。软件的复杂性一方面来自它所反映的实际问题的复杂性；另一方面，也来自程序结构的复杂性。软件技术的发展明显落后于复杂的软件需求，并且随着时间的推移，这个差距日益加大。

5. 软件成本相当昂贵

软件的研制工作需要投入大量、复杂、高强度的脑力劳动，研制成本是比较高的。在 20 世纪 50 年代末，软件的开销占总开销的百分之十几，大部分成本花在硬件上；但今天，这个比例完全颠倒过来，软件的开销大大超过硬件的开销。

1.1.3　软件危机

软件危机是指在计算机软件的开发、使用和维护过程中遇到的一系列严重问题。

1. 软件危机的表现

应该说，自计算机诞生以来，软件危机就一直存在。软件危机主要表现在以下几个方面：

（1）人们对软件开发的成本和进度的估计常常不够准确。由于软件的特殊性，不同类型的软件其开发所需的工作量、成本往往差别很大。因此，常出现实际成本比估算成本高出一个数量级，实际进度比计划进度拖延几个月甚至几年的现象，这大大降低了开发商的信誉，也引起了用户的不满。

（2）用户对已完成的软件不满意的现象时有发生。这主要是由于在开发的初期，软件需求不够明确，开发过程中又未能和用户及时交换意见，致使开发出的软件不能满足用户的需求，甚至无法使用。

（3）软件常常是不可维护的。在软件开发过程中，没有统一、公认的方法和规范进行指导，且设计和实现过程的资料很不完整，这使得软件出现问题后很难维护。

（4）软件产品的质量往往不可靠。由于未做好测试阶段的工作，提交给用户的软件质量差，在运行中暴露出大量问题。

(5) 软件开发生产率提高的速度远远跟不上日益增长的软件需求，满足不了社会发展的需要。

2. 缓解软件危机的途径

到 20 世纪 60 年代末期，软件危机已相当严重。要解决软件危机，必须做好以下几方面工作。

(1) 加强软件开发过程的管理，做到组织有序、各类人员协同配合，共同保证工程项目完成，避免软件开发过程中个人单干的现象。

(2) 推广使用开发软件的成功技术和方法，并且不断探索更好的技术和方法，消除一些错误的概念和做法。

(3) 开发和使用好的软件工具，支持软件开发的全过程。

如何做好上述 3 方面的工作来缓解软件危机呢？计算机科学家们经过多年的研究，提出了“软件工程”的概念，即用现代工程的原理、技术和方法进行软件的开发、管理、维护和更新。于是，开创了计算机科学技术的一个新的研究领域——软件工程。几十年来，软件工程在软件开发方法、工具和管理等方面的研究与应用已经取得了很多成果，并已大大缓解了软件危机所带来的影响。

1.2 任务 2 软件工程

任务描述

学习软件工程的基本理论与方法，建立并使用正确的工程方法开发出成本低、可靠性高并能高效运转的软件，从而解决或缓解软件危机。

任务目标

了解软件工程的概念，掌握软件工程的三要素，理解软件工程的基本原理，掌握软件工程的基本目标。学会使用软件工程的理论来解决软件危机的方法。

知识讲解

1.2.1 软件工程的概念

1968 年，北大西洋公约组织在联邦德国召开计算机科学会议，在此次会议上，弗瑞兹·巴尔（Fritz Bauer）首次正式提出了“软件工程”的概念。

软件工程学是一门指导软件开发和维护的工程学科，是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列完善的工程化原则。它应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则、方法来生产软件，以达到提高质量、降低成本的目的。

软件工程包括 3 个要素：方法、工具和过程。

(1) 软件工程方法为软件开发提供了“如何做”的技术，是指导研制软件的某种标准

规范。它包括了多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。软件工程方法常采用某种特殊的语言或图形的表达方法及一套质量保证标准。

（2）软件工具是指软件开发、维护和分析中使用的程序系统，为软件工程方法提供自动或半自动的软件支撑环境。

（3）软件工程的过程是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理及软件开发各个阶段完成的“里程碑”。

1.2.2　软件工程的基本原理

著名的软件工程专家波姆（B. W. Boehm）于 1983 年综合了软件工程专家学者们的意见，并总结了开发软件的经验，提出了软件工程的 7 条基本原理。这 7 条原理被认为是确保软件产品质量和开发效率的原理的最小集合，是相互独立、缺一不可、相当完备的最小集合。

下面简单介绍软件工程的这 7 条基本原理。

1. 用分阶段的生存周期计划严格管理

根据这条基本原理，可以把软件生存周期划分成若干个阶段，并相应地制定出切实可行的计划，然后严格按照计划对软件开发与维护进行管理。需要制定的计划有项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划和运行维护计划等。各级管理人员必须严格按照计划对软件开发和维护工作进行管理。据统计，在不成功的软件项目中，有一半左右是由于计划不周造成的。

2. 坚持进行阶段评审

据统计，在软件生存周期各阶段中，编码阶段之前的错误约占 63%，而编码错误仅约占 37%。另外，错误发现并改正得越晚，所付出的代价越高。坚持在每个阶段结束前进行严格的评审，就可以尽早发现错误。因此，这是一条必须坚持的重要原理。

3. 实行严格的产品控制

由于外部环境的变化，在软件开发的过程中改变需求是难免的，但决不能随意改变需求，只能依靠科学的产品控制技术来顺应用户提出的改变需求的要求。为了保持软件各个配置成分的一致性，必须实行严格的产品控制。其中主要是实行基准配置管理（又称为变动控制），即凡是修改软件的建议，尤其是涉及基本配置的修改建议，都必须按规定进行严格的评审，评审通过后才能实施。这里的“基准配置”是指经过阶段评审后的软件配置成分，即各阶段产生的文档或程序代码等。

4. 采用现代程序设计技术

实践表明，采用先进的程序设计技术既可以提高软件开发与维护的效率，又可以提高软件的质量。多年来，人们一直致力于研究新的“程序设计技术”。例如，20 世纪 60 年代末提出的结构化程序设计技术，以及后来提出的面向对象的分析（OOA）和面向对象的设计（OOD）技术等。

5. 结果应能清楚地审查

软件产品是一种看不见、摸不着的逻辑产品。因此，软件开发小组的工作进展情况可

见性差，难以评价和管理。为了更好地进行评价和管理，应根据软件开发的总目标和完成期限，尽量明确地规定出软件开发小组的责任和产品标准，从而能清楚地审查所得到的结果。

6. 开发小组的人员应少而精

软件开发小组人员的素质和数量是影响软件质量和开发效率的重要因素。实践表明，素质高的人员与素质低的人员相比，其软件开发的效率可能高出几倍至几十倍，而且所开发的软件中的错误也要少得多。另外，开发小组的人数不宜过多，因为随着人数的增加，人员之间交流情况、讨论问题的通信开销将急剧增加，这不但不能提高生产效率，反而由于误解等原因可能会增加出错的概率。

7. 承认不断改进软件工程实践的必要性

遵循上述 6 条基本原理，就能较好地实现软件的工程化生产。但是，软件工程不能停留在已有的技术水平上，应积极主动地采纳或创造新的软件技术，要注意不断总结经验，收集工作量、进度和成本等数据，并进行出错类型和问题报告的统计。这些数据既可用来评估新的软件技术的效果，又可用来指明应优先进行研究的软件工具和技术。

1.2.3 软件工程的基本目标

从技术和管理上采取多项措施以后，组织实施软件工程项目的最终目的是保证项目成功，即达到以下几个主要目标：

- 付出较低的开发成本；
- 达到预期的软件功能；
- 取得较好的软件性能；
- 使软件易于移植；
- 需要较低的维护费用；
- 能按时完成开发工作，及时交付使用。

在项目的实际开发中，使以上几个目标都达到理想的程度往往非常困难，而且上述目标很可能是相互冲突的。图 1—1 表明了软件工程目标之间存在的相互联系。有些目标是互补的，如高可靠性与易于维护之间、低开发成本与按时交付之间。有些目标之间则是互斥的，如低开发成本与高可靠性之间、高性能与高可靠性之间。

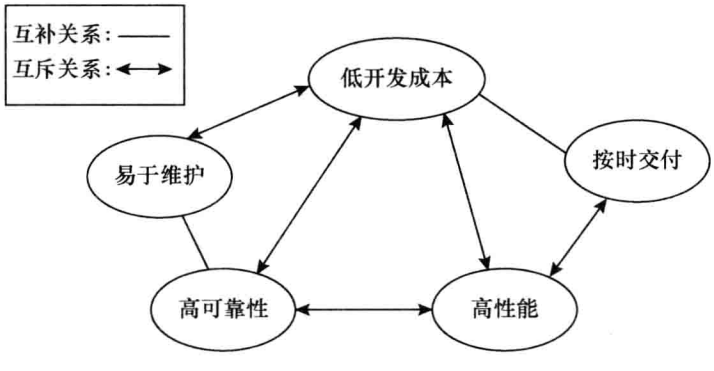

图 1—1 软件工程目标之间的关系

以上几个目标是判断软件开发方法或管理方法优劣的衡量尺度。在一种新的开发方法

提出以后，人们关心的是它对满足哪些目标比现有的方法更为有利。实际上，实施软件项目开发的过程就是在以上目标的冲突中取得一定程度平衡的过程。

1.3　任务 3　软件生存周期

任务描述

利用所学知识进行软件生存周期的各个阶段任务的划分。划分原则是保证各阶段的任务彼此间尽可能相对独立，同一个阶段各项任务的性质尽可能相同，从而降低每个阶段任务的复杂性，简化不同阶段之间的联系，有利于软件开发过程的组织管理。

任务目标

首先了解软件生存周期的概念，其次，熟练掌握软件生存周期各个阶段的工作流程及主要工作任务。

知识讲解

1.3.1　软件生存周期的概念

同任何事物一样，软件也有一个孕育、诞生、成长、成熟、衰亡的生存过程。软件生存周期是指一个计算机软件从功能确定、设计，到开发成功投入使用，并在使用中不断修改、增补和完善，直到停止该软件的使用的全过程。

1.3.2　软件生存周期的主要阶段

软件生存周期主要包括制定计划、需求分析、软件设计、程序编码、软件测试和运行维护等 6 个阶段。以下对这 6 个阶段的工作流程及主要任务做一概括的描述。

1. 制定计划

在软件系统开发之前，首先应当制定项目开发计划，该阶段是软件生存周期的第一阶段。其主要任务如下。

- 确定要开发软件系统的总目标。
- 给出功能、性能、可靠性以及接口等方面的要求。
- 完成该软件任务的可行性研究。
- 估计可利用的资源（硬件、软件和人力等）、成本、效益和开发进度。
- 制定出完成开发任务的实施计划，连同可行性研究报告，提交管理部门审查。

2. 需求分析

当完成计划制定之后，需要对用户的需求去粗取精、去伪存真、正确理解，然后把它用软件工程开发语言表达出来。其主要任务如下。

- 去用户处做需求调研，让用户提出对软件系统的所有需求。
- 对用户提出的需求进行分析、综合，并给出详细的定义。
- 编写软件需求说明书及初步的系统用户手册，提交管理机构评审。