



21 世纪高职高专规划教材

计算机系列

UML 及其 建模工具的使用

主 编 殷兆麟

副主编 王潜平 张永平 周智仁

主 审 袁迎菊



清华大学出版社

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



北京交通大学出版社

<http://press.bjtu.edu.cn>

21 世纪高职高专规划教材·计算机系列

UML 及其建模工具的使用

主 编	殷兆麟		
副主编	王潜平	张永平	周智仁
主 审	袁迎菊		
参 编	徐月美	戴 磊	
	谢轶才	孙锦程	

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本教材按照国家程序员、高级程序员和程序设计师职业标准要求编写。

UML 及其建模工具是用于为面向对象开发的软件系统进行可视化描述、建造和文档编制的建模语言。全书分为 10 章。第 1 章简明介绍传统软件工程的概念、技术、方法；第 2 章介绍 UML 的基本知识；第 3 章介绍如何利用 UML 为系统建模；第 4 章至第 7 章介绍如何使用 Rational Rose 建模工具建立系统用例视图、逻辑视图、组件视图和部署视图；第 8 章介绍如何由 Rose 模型生成 Java 代码及其逆向工程；第 9 章介绍如何由 Rose 模型生成 VB 代码及其逆向工程；第 10 章介绍 Rational 统一过程和成熟度模型 CMM。

本教材以超市进销存系统为示例贯穿全书主要的教学内容，力求概念清楚、简明实用。可以作为高职高专计算机应用、网络专业、电子商务、多媒体专业的现代软件工程或 UML 及其建模工具等方面课程的教材。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目 (CIP) 数据

UML 及其建模工具的使用 / 殷兆麟主编. — 北京：清华大学出版社；北京交通大学出版社，2004.3

(21 世纪高职高专规划教材·计算机系列)

ISBN 7-81082-254-3

I. U… II. 殷… III. 面向对象语言, UML—程序设计—高等学校：技术学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 122988 号

责任编辑：谭文芳 特邀编辑：刘卫华

出版者：清华大学出版社 邮编：100084 电话：010-62776969

北京交通大学出版社 邮编：100044 电话：010-51686045, 62237564

印刷者：北京瑞达方舟印务有限公司

发行者：新华书店总店北京发行所

开本：787×1092 1/16 印张：12.5 字数：316 千字

版次：2004 年 3 月第 1 版 2004 年 3 月第 1 次印刷

印数：1~4 000 册 定价：19.00 元

21 世纪高职高专规划教材·计算机系列

编审委员会成员名单

主任委员 李兰友 边奠英

副主任委员 周学毛 崔世钢 王学彬 丁桂芝 赵 伟
韩瑞功 汪志达

委 员 (按姓名笔画排序)

马 辉	万志平	万振凯	王永平	王建明
尤晓晔	丰继林	左文忠	叶 华	叶 伟
付晓光	付慧生	冯平安	江 中	佟立本
刘 炜	刘建民	刘 晶	曲建民	孙培民
邢素萍	华铨平	吕新平	陈小东	陈月波
李长明	李 可	李志奎	李 琳	李源生
李群明	李静东	邱希春	沈才梁	宋维堂
汪 繁	张文明	张权范	张宝忠	张家超
张 琦	金忠伟	林长春	林文信	罗春红
苗长云	竺士蒙	周智仁	孟德欣	柏万里
宫国顺	柳 炜	钮 静	胡敬佩	姚 策
赵英杰	高福成	贾建军	徐建俊	殷兆麟
唐 健	黄 斌	章春军	曹豫莪	程 琪
韩广峰	韩其睿	韩 劼	裘旭光	童爱红
谢 婷	曾瑶辉	管致锦	熊锡义	潘玫玫
薛永三	操静涛	鞠洪尧		

出版说明

高职高专教育是我国高等教育的重要组成部分，它的根本任务是培养生产、建设、管理和服务第一线需要的德、智、体、美全面发展的高等技术应用型专门人才，所培养的学生在掌握必要的基础理论和专业知识的基础上，应重点掌握从事本专业领域实际工作的基本知识和职业技能，因而与其对应的教材也必须有自己的体系和特色。

为了适应我国高职高专教育发展及其对教学改革和教材建设的需要，在教育部的指导下，我们在全国范围内组织并成立了“21 世纪高职高专教育教材研究与编审委员会”（以下简称“教材研究与编审委员会”）。“教材研究与编审委员会”的成员单位皆为教学改革成效较大、办学特色鲜明、办学实力强的高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院，其中一些学校是国家重点建设的示范性职业技术学院。

为了保证规划教材的出版质量，“教材研究与编审委员会”在全国范围内选聘“21 世纪高职高专规划教材编审委员会”（以下简称“教材编审委员会”）成员和征集教材，并要求“教材编审委员会”成员和规划教材的编著者必须是从事高职高专教学第一线的优秀教师或生产第一线的专家。“教材编审委员会”组织各专业的专家、教授对所征集的教材进行评选，对列选教材进行审定。

目前，“教材研究与编审委员会”计划用 2~3 年的时间出版各类高职高专教材 200 种，范围覆盖计算机应用、电子电气、财会与管理、商务英语等专业的主要课程。此次规划教材全部按教育部制定的“高职高专教育基础课程教学基本要求”编写，其中部分教材是教育部《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》的研究成果。此次规划教材编写按照突出应用性、实践性和针对性的原则编写并重组系列课程教材结构，力求反映高职高专课程和教学内容体系改革方向；反映当前教学的新内容，突出基础理论知识的应用和实践技能的培养；适应“实践的要求和岗位的需要”，不依照“学科”体系，即贴近岗位群，淡化学科；在兼顾理论和实践内容的同时，避免“全”而“深”的面面俱到，基础理论以应用为目的，以必要、够用为度；尽量体现新知识、新技术、新工艺、新方法，以利于学生综合素质的形成和科学思维方式与创新能力的培养。

此外，为了使规划教材更具广泛性、科学性、先进性和代表性，我们希望全国从事高职高专教育的院校能够积极加入到“教材研究与编审委员会”中来，推荐“教材编审委员会”成员和有特色、有创新的教材。同时，希望将教学实践中的意见与建议及时反馈给我们，以便对已出版的教材不断修订、完善，不断提高教材质量，完善教材体系，为社会奉献更多更新的与高职高专教育配套的高质量教材。

此次所有规划教材由全国重点大学出版社——清华大学出版社与北方交通大学出版社联合出版。适合于各类高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院使用。

21 世纪高职高专教育教材研究与编审委员会
2003 年 9 月

前 言

使用 UML 及其建模工具, 替代传统软件工程技术得到越来越多 IT 专家的认同。

学习、掌握 UML 及其建模工具, 具有为中小软件系统建模的能力, 必须具备三方面的知识: 一是对传统软件工程有一定的了解; 二是至少掌握了一门面向对象的语言, 有一定的面向对象的编程能力; 三是有基本的数据库知识和使用能力。

本书按上述思想组织教材内容, 以不另外开设传统“软件工程”为前提。第 1 章简明地介绍传统软件的工程方法, 分析该方法存在的问题, 介绍面向对象的方法及面向对象方法的建模。第 2 章至第 9 章介绍 UML 语言及 UML 建模工具 Rational Rose 的使用方法。第 10 章总结介绍 Rational 统一过程和成熟度模型。

教材概念清楚、简明实用, 可以作为计算机应用、网络专业、电子商务、多媒体专业(高职、高专) 软件工程或 UML 及其建模工具等方面的教材。

本教材的组织编写得到江苏省教委“计算机网络人才培养模式的研究”项目资助; 得到江苏省教委百门优秀课程教材基金资助; 配套的 CAI 课件、学生实习多媒体课件得到我校教改基金资助; 系列教材建设列为中国矿业大学新世纪教材工程项目。在此表示衷心地感谢。

本书有配套的电子教材、CAI 课件。课件包括 Rational Rose 建立的完整的超市进销存模型及生成的 VB 和 Java 应用系统框架, 可运行的演示系统。需要课件的读者可以与主编联系。主编电子邮箱地址为: zhlyin@cumt.edu.cn。

由于时间仓促, 水平有限, 书中难免有不妥和错误之处, 恳请广大读者指正。

殷兆麟

2004 年 1 月于中国徐州

目 录

第 1 章 软件工程方法学	(1)
1.1 概述	(1)
1.2 软件生命周期方法学	(2)
1.2.1 问题定义	(2)
1.2.2 可行性分析	(2)
1.2.3 需求分析	(3)
1.2.4 概要设计与结构化设计方法	(8)
1.2.5 详细设计	(12)
1.2.6 编码	(12)
1.2.7 软件测试与维护	(14)
1.3 面向对象的方法	(15)
1.3.1 传统软件工程方法的评述	(15)
1.3.2 提高软件质量的新途径	(17)
1.4 计算机软件系统的建模及建模原则	(22)
1.4.1 计算机软件系统具有“产品”的特征	(22)
1.4.2 软件系统必须进行建模	(22)
小结	(23)
习题	(24)
第 2 章 建模语言 UML	(25)
2.1 UML 目标	(25)
2.2 UML 语言概述	(25)
2.2.1 UML 的基本图素	(25)
2.2.2 UML 模型图	(31)
2.2.3 公用机制	(42)
2.2.4 建模规则	(46)
2.3 UML 建模	(47)
2.3.1 用例视图	(47)
2.3.2 逻辑视图	(48)
2.3.3 组件视图	(48)
2.3.4 部署视图	(50)
小结	(51)
习题	(51)
第 3 章 UML 实例分析	(53)
3.1 软件系统需求分析	(53)

3.1.1	概述	(53)
3.1.2	UML 系统需求分析的内容	(54)
3.2	建立系统用例视图	(58)
3.2.1	超市进销存系统用例	(58)
3.2.2	系统动态行为分析	(64)
3.2.3	用例测试	(66)
3.3	系统逻辑视图	(67)
3.3.1	系统类分析	(67)
3.3.2	系统类说明	(69)
3.4	数据库设计	(70)
小结		(70)
习题		(71)
第 4 章	Rational Rose 环境介绍	(72)
4.1	Rational Rose 简介	(72)
4.2	Rose 开发环境	(72)
4.2.1	启动 Rose	(72)
4.2.2	Rose 图形界面构成	(74)
4.2.3	Rose 的四种视图	(77)
4.2.4	Rose 的模型图	(79)
小结		(79)
习题		(80)
第 5 章	Rational Rose 用例模型分析	(82)
5.1	业务用例模型 (Business Model)	(82)
5.1.1	业务用例模型的基本图素	(82)
5.1.2	业务用例模型的创建	(83)
5.2	系统用例模型	(91)
5.2.1	系统用例模型基本图素	(91)
5.2.2	系统用例模型的建立	(92)
小结		(95)
习题		(95)
第 6 章	Rational Rose 逻辑模型分析	(97)
6.1	业务对象模型	(97)
6.1.1	业务对象模型分析	(97)
6.1.2	业务对象模型的创建	(98)
6.2	分析模型	(101)
6.2.1	分析模型基本概念	(101)
6.2.2	分析模型的创建	(103)
6.2.3	创建用例实现	(104)
6.3	设计模型	(111)

6.3.1	设计模型基本概念	(111)
6.3.2	建立设计模型	(112)
小结		(119)
习题		(119)
第 7 章	Rose 组件视图和部署视图	(120)
7.1	Rose 组件视图	(120)
7.1.1	组件、组件图和组件视图	(120)
7.1.2	Rose 组件图的基本图素	(120)
7.1.3	建立组件图	(121)
7.1.4	将类赋予组件	(122)
7.2	Rose 部署视图	(123)
7.2.1	打开部署图	(123)
7.2.2	增加和删除节点	(123)
7.2.3	设置处理机描述信息	(124)
7.2.4	建立节点间的连接	(125)
小结		(126)
习题		(126)
第 8 章	模型生成 Java 代码及其逆向工程	(127)
8.1	模型生成 Java 代码概述	(127)
8.1.1	模型生成代码的步骤	(127)
8.1.2	检查模型	(128)
8.2	Java 语法单元与 Rose 模型图素的对应	(128)
8.3	与项目有关的 Java 代码生成控制属性	(129)
8.3.1	选择模型的 Java 框架	(129)
8.3.2	设置与项目有关的 Java 代码生成的控制属性	(129)
8.4	与类有关的 Java 代码生成控制属性	(132)
8.5	类特征生成 Java 类代码的有关属性	(133)
8.6	类操作生成代码的有关属性	(135)
8.7	关联作用生成代码的有关属性	(138)
8.7.1	关联作用生成代码的有关属性	(138)
8.7.2	生成 Java 代码的步骤	(138)
8.8	Rose 图素生成的 Java 代码	(139)
8.8.1	类 Java 代码	(139)
8.8.2	类特征 Java 代码	(140)
8.8.3	类方法 Java 代码	(141)
8.8.4	一对一单向关联	(143)
8.8.5	一对多的单向关联	(144)
8.8.6	双向关联	(144)
8.8.7	依赖性关系	(145)
8.8.8	泛化关系生成的 Java 代码	(146)

8.8.9 接口	(146)
8.8.10 模块属性	(148)
8.9 逆向工程	(149)
小结	(151)
习题	(151)
第9章 模型生成 VB 代码	(152)
9.1 VB 代码生成概述	(152)
9.2 创建组件及代码生成	(152)
9.2.1 组件的创建	(152)
9.2.2 代码生成	(153)
9.3 模型成分与对应的 VB 语法成分之间的对应	(157)
9.3.1 类	(157)
9.3.2 模型成分与 VB 语法成分的对应	(158)
9.4 逆向输出 Rose 模型	(166)
小结	(169)
习题	(170)
第10章 Rational 统一过程	(171)
10.1 软件工程的过程概述	(171)
10.2 Rational 统一过程	(172)
10.3 Rational 统一过程的结构	(173)
10.3.1 Rational 统一过程的动态结构	(173)
10.3.2 Rational 统一过程的静态结构	(176)
10.4 如何在过程中使用 UML	(179)
10.5 评价软件过程成熟度的标准	(180)
10.5.1 能力成熟度模型概述	(180)
10.5.2 CMM 的 5 个等级	(182)
10.5.3 正确理解 CMM 内涵	(185)
小结	(185)
习题	(185)
参考文献	(186)

第 1 章 软件工程方法学

1.1 概述

软件工程是从管理和技术两个方面研究如何更好地开发和维护计算机软件的一门学科。当前公认的软件开发方法主要有以下两种。

(1) 面向过程的方法

面向过程的方法人们习惯上称为软件生命周期法，或传统软件工程方法。它的基本思想是以功能为中心，自上而下分解。这种方法适应于功能需求稳定的系统，对诸如商业系统这样的用户需求多变的系统是不太合适的。目前软件生命周期方法在我国还很流行。

(2) 面向对象的方法

面向对象的方法要点归结于四点：类、对象、消息和继承。这在面向对象的程序设计中已经介绍得很多了。这种方法适应于快速原型、反馈、迭代的开发方式，适应于功能需求稳定或多变的系统。

网络技术的普及促进了面向对象的技术与分布式计算技术结合。组件技术是这种结合的产物，与面向对象的技术不同，它不强调网络对象的继承，而强调组件界面接口的规范、组件的功能及其重用。J2EE，.Net，DCOM 和 CORBA 都属于当代有名的分布式组件开发技术规范。

不管面向过程的方法，还是面向对象的方法，它们都在按自己的理论、技术、方法描述软件系统，描述系统也叫系统建模。长期以来，人们在寻求适合软件系统建模的技术、方法。面向对象的方法经过 20 多年的实践总结，可视化建模、特别是 UML 可视化建模得到认可、普及。

UML (Unified Modeling Language) 是一种用来为面向对象开发的软件系统进行可视化描述、建造和文档编制的建模语言。UML 取代目前软件业众多的分析和设计方法，成为一种标准，这使信息产业第一次有了一个统一的建模语言。

要想成功地使用 UML，科学的过程是完全必要的，尤其在一些需要团队合作的大型系统中，此时必须协调所有的工作，确保开发团队向同一方向努力。科学的过程能够有效地计划工作进程，控制和提高工作效率，保证软件的质量和软件的重用。

Rational 统一过程是当今比较流行的几种重要的科学的软件开发过程之一。Rational 统一过程与 UML 能够最好地结合。Rational 统一过程将系统的开发划分成不同的阶段和一定的工作流程。其核心体现了当今软件开发的规律：迭代地开发软件、以需求为中心、使用基于组件的软件架构、为软件可视化建模、验证软件质量、控制对软件的变更；为软件开发团队提供指南、文档模板和工具。

CMM 是能力成熟度模型(Capability Maturity Model)的缩写，是一种软件质量管理系统的软件开发过程的框架。这里，“成熟度”意味着 IT 企业管理环境的好坏。CMM 用其大型软件开发过程的框架模型，评估企业的软件产品开发的可预见性程度和预测风险性程度的相对水平。

1.2 软件生命周期方法学

一个软件从计划开始，到废弃不用为止，要经历一个漫长的时期，通常把软件经历的这个时期称为软件生命周期。软件生命周期通常被划分为若干个阶段，每个阶段的任务比较简单、相对独立、方便审查和管理。这样，保证了软件的质量，提高了软件的可维护性。

通常，软件生命周期由软件定义、软件开发和软件维护三个时期组成，每个时期又进一步划分为若干阶段。软件定义包括问题定义、可行性研究和编制项目开发计划；软件开发包括需求分析、概要设计、详细设计、编码和测试；软件维护包括运行维护。表 1-1 简明地描述了软件生命周期各阶段的任务及文档。

表 1-1 软件生命周期各阶段的任务及文档

阶 段	任 务	文 档
问题定义	明确要解决什么问题	系统目标与范围说明书
可行性研究	确定问题能否解决	项目计划任务书
需求分析	确定系统的功能	需求规格说明书
概要设计	建立软件结构	概要设计说明书
详细设计	确定模块的过程结构	详细设计说明书
编码	编写程序	程序清单
测试	发现并排除错误	测试分析报告
运行维护	运行和管理	改进的软件

软件生命周期方法学是建立在软件生命周期理论基础上的软件开发方法，它特别强调预先定义需求的重要性，通过需求分析确定系统需要的功能，然后如上述一个阶段一个阶段地按功能实现这些需求。软件生命周期方法使用的主要技术是结构化分析和设计，其本质是功能分解。

1.2.1 问题定义

问题定义阶段的任务是搞清楚“要解决的问题是什么”。通过对系统用户的访问调查，系统分析员要简明地阐述他对问题的理解，形成书面文档，并正式地与用户认真地讨论，改正文档中存在的问题。文档的内容包括问题的性质、项目的目标和规模等。

1.2.2 可行性分析

1. 目的

明确所定义的项目是不是能够实现和值得开发。

2. 研究内容

(1) 技术可行性：现有技术能否实现这一新系统？技术难点何在？建议采用的技术先进程度怎样？

(2) 经济可行性：评估系统的经济效益。

(3) 运行可行性：新系统的运行方式用户是否可以接受？包括体制的改变和人员的变动等。

(4) 法律可行性：新系统的开发是否引起其他责任问题，如保密、拷贝和转让的限制等。

3. 可行性研究的步骤

(1) 复查问题定义，进一步澄清系统的范围和目标，准确地把握要解决的问题。

(2) 研究当前正在使用的系统，建立当前系统的物理模型。

(3) 抽象出当前系统高层逻辑模型，计划目标系统的逻辑模型和主要功能。

(4) 对计划的新系统重新定义。步骤(1)~(4)是个反复循环的过程。

(5) 导出供选择的几种方案，进行可行性论证。

(6) 推荐行动方案。

(7) 草拟开发计划。

(8) 书写文档：可行性论证报告、项目开发计划，提交审查。

4. 编制项目开发计划

经过可行性分析，如果软件项目值得开发，需要编制项目开发计划。开发计划主要包括如下内容。

(1) 项目概述：工作内容、条件与限制、产品、运行环境、服务和验收标准。

(2) 实施计划：任务分解、进度、预算和关键问题。

(3) 人员组织与分工。

(4) 交付日期。

(5) 专题计划要点。

1.2.3 需求分析

需求分析是软件定义时期最为重要的阶段。用户了解他们所面对的问题，知道要做什么，但是通常不能完全准确地表达出他们的要求；软件开发人员对特定用户的具体要求并不完全清楚，虽然，他们知道怎样用软件实现用户的要求。因此，系统分析员在需求分析阶段必须与用户密切配合，充分交流信息，以得出满足用户需求的系统逻辑模型。需求分析的任务是明确“目标系统究竟做什么”，步骤如下。

1. 确定系统的综合要求

(1) 功能要求：系统必须完成的所有功能。

(2) 性能要求：系统的处理速度，存储容量，响应时间，安全限制等。

(3) 运行要求：系统运行时所需要的环境要求，如主机、软件、外设等。

(4) 扩充要求：为系统将来扩展提出的要求。

2. 分析系统的数据要求

系统的数据要求主要包括：系统需要有哪些数据、数据之间有什么联系、数据本身性质、数据的结构，等等。为了清晰明确表达数据要求，需要建立一个概念性的数据模型，DFD (Data

Flow Diagram, 数据流图), DD (Data Directory, 数据字典), ER (Entity Relation, 实体联系) 模型是最常用的数据模型。

3. 结构化分析方法

结构化分析方法的分析结果由以下三部分组成。

- ✎ 一套分层的数据流图：用来描述数据流从输入到输出的变换流程。
- ✎ 一组数据词典：用来描述 DFD 中的每个数据流、文件及组成数据流或文件的数据项。
- ✎ 加工逻辑说明：用来描述每个基本加工（即不再分解的加工）的加工逻辑。

其中，DFD 的基本成分如下。

- ✎ 数据流：由一组固定成分的数据组成。
- ✎ 加工：描述输入数据流到输出数据流的变换。
- ✎ 文件：表示暂时存储的数据。
- ✎ 源/宿 (Source/Sink)：表示存在于软件系统之外的人员或组织。

(1) 数据流图

① 数据流图基本成分

数据流图描述系统的逻辑模型，即信息在系统中的流动和处理情况。DFD 的基本成分及其图形表示方式如图 1-1 所示。

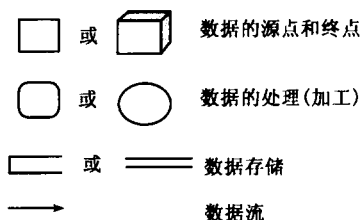


图 1-1 数据流图常用符号

✎ 数据流。数据流由一组固定成分的数据组成，表示数据的流向，它可以从加工流向加工，从数据源流向加工，从加工流向数据宿。除流向文件或从文件流出的数据流不必命名（因为这种数据流的组成成分就是相应的文件组成部分）外，每个数据流都必须有一个合适的命名，以反映该数据流的含义。要说明的是 DFD 中描述的是数据流，而不是控制流。

✎ 加工。加工描述输入数据流到输出数据流之间的变换，也就是输入数据流做了什么处理后变成了输出数据流。每个加工有一个名字和一个编号。编号能反映出该加工位于分层 DFD 中的哪个层次和哪张图中，也能看出它是哪个加工分解出来的子加工。

✎ 文件。文件用来表示暂时存储的数据，每个文件都有文件名。流向文件的数据流表示写文件，流出文件的数据流表示读文件，双向箭头表示对文件既读又写。

✎ 数据源/数据宿（简称源/宿）。源和宿通常是指存在于软件系统之外的人员或组织，它指出系统所需数据的发源地和系统所产生数据的归宿地，在一个软件系统中，有些源和宿可以是同一个人或组织，这时，源和宿使用同一个图形符号。

② 分层数据流图的画法

✎ 画系统的输入和输出。把整个软件系统看做一个大的加工，然后根据系统从外界的哪

些源接受哪些数据流, 以及系统的哪些数据流送到外界的哪些宿, 就可画出系统的输入和输出图。这张图称为顶层图。

☞ 画系统的内部。将顶层图中的加工分解成若干个子加工, 并用数据流将这些加工连接起来, 使得顶层图中的输入数据流经一连串的加工处理后变换成顶层图的输出数据流。这张图, 称为 0 层图。从一个加工画了一张数据图的过程实际上就是对这个加工的分解过程。

可用下述的方法来确定加工。在数据流的组成或值发生变化的地方, 应画一个加工。这个加工的功能就是实现这一变化, 也可根据系统的功能确定加工。

确定数据的方法可以是: 当用户把若干个数据看做一个单位来处理(这些数据一起到达, 一起加工)时, 可把这些数据看成一个数据流, 通常可以把实际工作中的单据(如报名单)作为一个数据流。

对于以后某个时间要使用的一些数据可组织一个文件存储。

☞ 画加工的内部。每个加工看做一个小系统, 把该加工的输入输出数据流看成小系统的输入输出数据流, 于是, 可以用画 0 层的方法同样地画出每个加工的 DFD 子图。

☞ 对第③步分解出来的 DFD 子图中的每个加工重复第③步的分解, 直至图中尚未分解的加工都足够简单(也就是说这种加工不必再分解)为止, 到此得到了一套分层数据流图。

③ 父图与子图的管理

对于一个软件系统, 其数据流图可能有许多层, 每一层又有许多张图, 为了区分不同的加工和不同的 DFD 子图, 应该对每张图和每个加工进行编号, 以便于管理。

☞ 父图与子图

假定分层数据流图里的某张图 A, 其中的某个加工可用另一张图(记为图 B)来分解, 则图 A 即为图 B 的父图, 图 B 为图 A 的子图。在一张图中, 有些加工需要进一步分解, 有些加工则不必分解。因此, 如果父图中有 n 个加工, 那么它可以有 $0 \sim n$ 张子图(这些子图位于同一层), 但每张子图都只对应于一张父图。

☞ 编号

顶层图只有一张, 图中的加工也只有一个, 所以不必编号。0 层图只有一张, 图中的加工号可分别是 0.1, 0.2 等, 或者是子图号就是父图中被分解的加工号。

子图中的加工号由子图号、圆点和序号组成。

例如, 某图的加工号为 2.4.3, 这个加工分解出来的子图号就是 2.4.3, 该子图的加工号分别是 2.4.3.1, 2.4.3.2, ...

④ 简单示例

求一元二次方程根是一个简单的问题。这里用一个简单的问题展示如何用数据流图描述它的需求分析。实际上对不复杂的问题只要建立程序流程图就可以了。

图 1-2 为求一元二次方程根系统的顶层数据流图, 它从总的方面描述系统的功能。图 1-3 为求一元二次方程根系统在其 0 层数据流图分解为三个子功能。

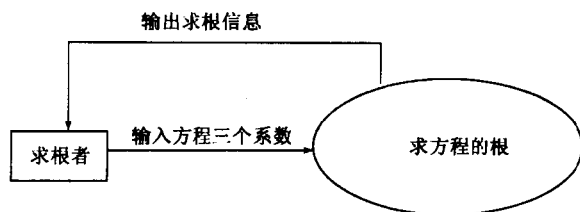


图 1-2 求一元二次方程根数据流图（顶层）

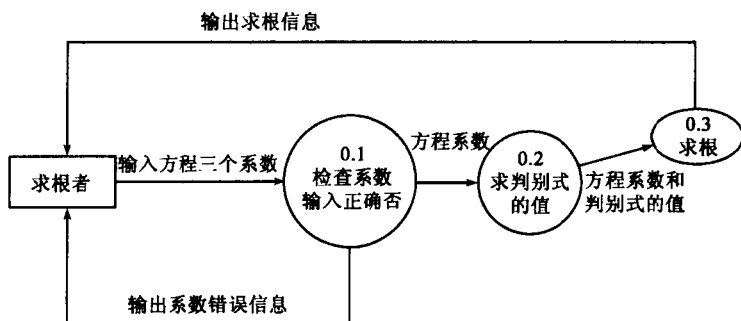


图 1-3 求一元二次方程根数据流图（0层）

（2）数据词典

数据流图描述了系统的分解，但没有对图中各成分进行说明。数据词典就是为了对数据流图中的每个数据流、文件、加工及组成数据流或文件的数据项作出说明，其中对加工的描述称为“加工逻辑说明”，也叫“小说明”。

① 词典条目

☞ 数据流条目

一个数据流条目可以有以下内容。

数据流名：数据流的名称。

编号：用于词典管理，如排序、查找。

简述：简述该数据流的组成。

来源：描述该数据流来源哪个加工或源。

去向：描述该数据流流向哪个加工或宿。

数据量：描述在软件系统中该数据流的数量（也可以是某段时间里的数据量、如每天多少）。

峰值：描述在某一时间范围内要处理的数据流的最大数量。

在这些内容中数据流名和组成是必不可少的，数据量和峰值一般也是需要的，其他内容则是可有可无的。数据流的组成使用以下符号：

$a+b$ 表示 a 与 b ；

$[a|b]$ 表示 a 或 b ；

$\{a\}$ 表示 a 重复出现多次， $\{a\}_m^n$ 表示 a 最少重复出现 m 次，最多重复出现 n 次；

(a) 表示 a 是任选的，即 a 可出现 0 次或 1 次。

◇ 文件条目

一个文件条目可以有如下内容。

文件名：文件的名称。

编号：用于词典管理。

简述：简述该文件的含义。

组成：描述该文件由哪些数据项组成。

文件组织：描述该文件的类型，按什么关键字排序等。

读文件：描述哪些加工需要读该文件。

写文件：描述哪些加工需要写该文件。

数据量：描述该文件的记录个数。

注解。

其中，文件名和组成是必须的，文件组织和数据量通常也是需要的，其他内容则是可有可无的。

◇ 数据项条目

一个数据项条目可以有如下内容。

数据项名：数据项的名称。

编号：用于词典管理。

简述：简述该数据项的含义。

单位：该数据的计量单位。

类型：描述该数据项的数据类型。

值域：描述该数据项的值范围。

编辑方式：描述数据项在输出时间时的编辑要求，如每三位数字前加一个逗号。

与其他数据项的关系：它有利于数据的合法性检查，如，供电量=发电量-线损。

注解。

其中数据项名、类型是必须的，单位、值域、编辑方式、与其他数据项目的关系通常也是需要的，其他内容则是可有可无的。

② 词典管理

词典管理主要是把词典条目按某种格式组织后储存到词典中，并提供排序、查找和统计等功能。如果数据流条目包含来源和去向，文件条目中包含读文件和写文件，则还可以检查数据词典与数据流图的一致性。

(3) 逻辑加工说明

在数据流图中，有些加工分解成一张子图，这种加工的逻辑功能反映在这张子图中，对那些不再分解的加工，称之为基本加工。一个基本加工的逻辑加工说明（也称为加工条目）可以有以下内容。

加工名：加工的名称。

编号：用于加工条目的管理。

加工号：显示它在数据图中的位置。

简述：对该加工简要的说明。

输入数据流：描述该加工有哪些输入数据流。