

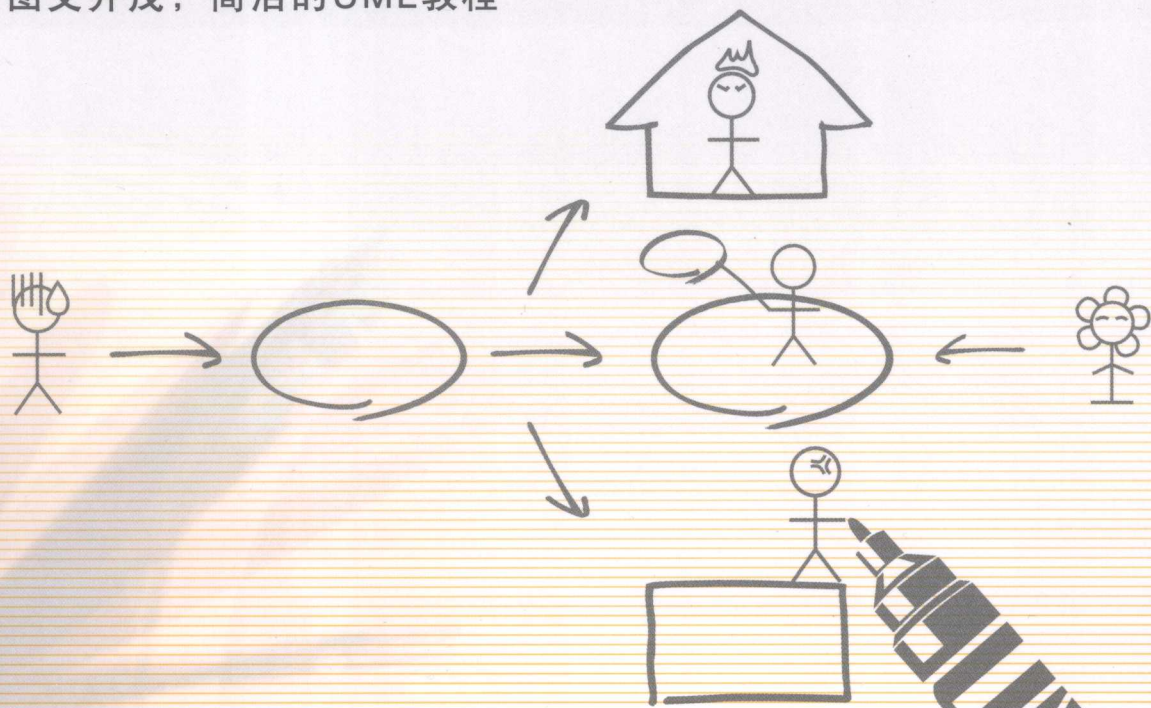
提供
全部源代码
开发视频下载

UML

周力 何雪飞 编著

建模图解教程

图文并茂，简洁的UML教程



- 200 个设计流程图和Rose建模图
- 4 个综合性的建模案例，贯穿UML基础知识
- 附赠 全书全部源代码，可以直接用于工程实践
- 附赠 视频下载，手把手教你Rose建模



人民邮电出版社
POSTS & TELECOM PRESS

UML

建模图解教程

周力 何雪飞 编著

人民邮电出版社

北京

图书在版编目(CIP)数据

UML建模图解教程 / 周力, 何雪飞编著. —北京: 人民邮电出版社, 2009.6
ISBN 978-7-115-20469-1

I. U… II. ①周…②何… III. 面向对象语言, UML—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2009)第026786号

内 容 提 要

本书是一本针对 Java 学员的 UML 建模教程。它以图文并茂的形式, 精炼而全面地讲解了 UML 各个组成部分, 描述了使用 UML 进行开发的过程, 旨在让读者掌握 UML 的术语、规则和语言特点, 以及如何有效地使用 Rational Rose 工具进行 UML 建模, 知道如何应用 UML 解决一些 Java 开发过程中常见的建模问题。

本书文字朴实简洁, 并辅之以大量表格和代码示例, 对于希望快速进入项目开发的 Java 学员具有自学和指导作用, 既可作为大专院校相关专业的教材, 又可作为软件开发人员的技术参考手册。

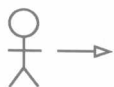
UML 建模图解教程

-
- ◆ 编 著 周 力 何雪飞
责任编辑 屈艳莲
执行编辑 蒋 佳
 - ◆ 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>
北京顺义振华印刷厂印刷
 - ◆ 开本: 800×1000 1/16
印张: 16.5
字数: 363 千字
印数: 1—4 000 册
- 2009 年 6 月第 1 版
2009 年 6 月北京第 1 次印刷

ISBN 978-7-115-20469-1/TP

定价: 35.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154



前言

UML 是用来对软件系统进行描述、构造、可视化和文档编制的一种语言，它统一了 Booch、Rumbaugh 和 Jacobson 的表示方法。UML 的出现满足了行业对于一种用来讨论信息技术的通用语言的需求。事实证明，UML 是非常成功的，现在 UML 已经是面向对象开发的标准，而且最新的 UML 2.0 也正在新领域中寻求更多的应用。

本书的特点

- 内容全面。书中详细地介绍了 UML 的基础知识，例如视图、图、模型元素和通用机制等，同时结合具体案例，给出了相关理论的建模实践。书中还介绍了正向工程和逆向工程在 Rational Rose 中的具体实现方法。本书还包括了最新标准 UML 2.0 的内容。
- 案例丰富。本书提供了 4 个综合性的建模案例，即新闻发布系统、书店借书系统、销售管理系统和在线银行系统。其中，书店借书系统的建模案例贯穿于 UML 基础知识的相关章节，有助于读者边学习、边思考、边实践。
- 图文并茂。书中的各个章节都配有大量的设计流程图和 Rose 建模图，有助于读者更加直观地理解 UML 的理论知识，并在实际的学习和工作中学以致用。
- 通俗易懂。本书在介绍每个章节、知识点和案例时，语言通俗易懂，并且都给出了全面的、详细的步骤说明。

读者对象

本书适用于从事面向对象软件开发的软件工程人员，例如系统分析人员、系统设计人员、项目经理和实际开发者等，也适用于高等院校计算机、软件工程等相关专业的广大师生。

本书概览

本书从 UML 的整体概述开始，然后用一些具体的章节来阐述相关内容，最后以案例学习来结束。本书将重点放在了成功的项目中是如何应用 UML 的。

第 1 章回答了“什么是 UML？”这个问题。该章解释了 UML 的基本概念、UML 的组



成、建模技术和 UML 的建模工具。

第 2 章主要介绍了面向对象机制。包括面向对象的基本要素和面向对象的基本原则。本章的最后介绍了基于 UML 的面向对象的分析和设计。

第 3 章介绍如何使用 UML 进行 Java 项目开发,通过一个新闻发布系统中的几个简单范例,示范 UML 如何应用于 Java 的实际项目中。

第 4 章剖析了用例图。用例图主要作用是向用户展示系统的功能。好的用例会将着眼点放在系统需要发布的功能上。用例可以提供系统的一个组织单元,以便能够在静态和动态模型中对该系统进行更加详细的分析。

第 5 章讲述了静态模型图,即类图、对象图和包图。类定义了一个系统中的事物。严格来说,类只是代表了系统中的事物类型,如果要表示一个特定的事物,UML 则使用对象来作为类的特定实例。类之间的关系提供了有关某个系统的实质性信息。一些类是按照等级结构来组织的,而另一些类则需要依赖其他元素才能正常工作。

第 6 章讨论了动态模型图。这些图用来揭示系统中的动态流程,展示特定对象是如何交互并产生某种结果的。活动图通常着眼于高级的系统流程。状态机可以显示出一个类的生命周期。交互图则可以显示出系统中各对象协作时的消息。

第 7 章讨论了实现与部署模型图,涉及逻辑架构和物理架构。重点讨论了组件图和部署图,说明了如何将系统表示成实际的制品,这些制品最终驻留在物理处理器中。

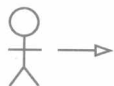
第 8 章讨论了 UML 与统一开发过程。统一过程使用统一建模语言 UML 来制定软件系统的所有蓝图。UML 是整个统一过程的一个完整部分,是与 RUP 共同发展起来的。RUP 是一个通用的软件流程框架,以架构为中心、用例为驱动的迭代化软件开发流程。

第 9 章介绍了正向和逆向工程。无论是从模型生成代码还是从代码回到模型,都是一项非常复杂的工作。正向和逆向工程这两方面结合在一起,定义为双向工程。双向工程提供了一种在描述系统的架构或设计和代码的模型之间进行双向交换的机制。

第 10 章和第 11 章提供了两个实际的 UML 示例,即销售管理系统和在线银行系统。阐述了在实际项目中如何应用 UML 进行设计,以及相对应的 Java 实现源码。

附录 A 展示了如何扩展 UML 的特性以用于特定的环境或领域,UML 中包括构造型、标记值和约束等扩展机制。附录 B 主要阐述了如何使用 OCL 来描述约束以实现 UML 和扩展。附录 C 主要介绍了 UML 的体系结构并讲解了 UML 是如何建造的,对于只是使用建模工具来绘制基本类图的建模人员而言不必了解。

本书主要由周力、何雪飞编写,同时参与资料搜集和编写工作的还有梁建全、卞征辉、许丽艳、王志强、邹小梅、石增娟、胡明地、胡玉芳、江海洋、江小红、毕连雪、陈景燕、丁庆丰、董丽丽、方基成、何立君、刘洪敏、石龙、王常友、郑展鸿、薛立滨、段世林、郭光、刘玖增、孙蓓蓓、谭翔天、王紫、杨惠如、袁翠、张庆丰、张亚东等,在此向他们表示衷心的感谢。我们在本书的编写过程中,力求精益求精,但由于水平有限,疏漏之处在所难免,如果读者在阅读本书时发现差错或者遇到问题,可以发送 E-mail 到 zhouli1234@163.com 与我们联系。



目 录

第 1 章 UML 概述	1
1.1 统一建模语言	1
1.1.1 什么是 UML	1
1.1.2 UML 的发展	1
1.2 UML 静态模型图	2
1.2.1 类图	2
1.2.2 对象图	3
1.2.3 组件图	3
1.2.4 部署图	3
1.3 UML 动态模型图	4
1.3.1 用例图	4
1.3.2 序列图	4
1.3.3 协作图	5
1.3.4 状态图	6
1.3.5 活动图	6
1.4 UML 2.0 中的新图	7
1.4.1 组成结构图	7
1.4.2 交互纵览图	7
1.4.3 计时图	8
1.4.4 包图	8
1.5 建模技术	8
1.5.1 为什么需要建模	8
1.5.2 什么是软件建模	9
1.5.3 建模的基本原则	9
1.5.4 软件建模的实现过程	10



1.5.5 UML 建模的基本过程	10
1.6 UML 建模工具	11
1.6.1 Rational Rose	11
1.6.2 Microsoft Visio	17
1.6.3 MagicDraw UML	18
1.7 常见问题与解答	18
1.8 本章小结	22
第 2 章 面向对象的分析与设计方法	23
2.1 面向对象机制	23
2.1.1 面向对象的要素	23
2.1.2 主要原则	25
2.2 基于 UML 的 OOA 和 OOD	27
2.3 常见问题与解答	27
2.4 本章小结	30
第 3 章 UML 与 Java 项目开发	31
3.1 基于 UML 开发项目的基本过程	31
3.2 Java EE 典型项目开发概述	31
3.2.1 访客登录	32
3.2.2 编辑添加文章	34
3.2.3 访客查看文章列表	37
3.3 常见问题与解答	39
3.4 本章小结	40
第 4 章 用例图	41
4.1 用例图概念	41
4.1.1 概述	41
4.1.2 定义用例图	41
4.1.3 主要组件	42
4.2 识别参与者	44
4.2.1 参与者的类型	44
4.2.2 捕获需求	44
4.2.3 识别参与者的方法	44
4.3 识别用例	46
4.3.1 识别用例的方法	46
4.3.2 用例的命名	46
4.4 用例之间的关系	47
4.4.1 泛化关系	47

4.4.2 包含关系	48
4.4.3 扩展关系	49
4.4.4 用例之间的关系比较	49
4.5 用例文档	50
4.6 UML 2.0 中的相关变化	51
4.7 Rational Rose 建模实例	51
4.7.1 使用 Rose 绘制用例图的步骤	53
4.7.2 书店借书系统的用例图	55
4.8 常见问题与解答	56
4.9 本章小结	63
第 5 章 静态模型图	64
5.1 基本概念	64
5.2 类图	64
5.2.1 概述	64
5.2.2 类	65
5.2.3 接口	67
5.2.4 关系	69
5.3 对象图	76
5.3.1 概念	76
5.3.2 对象图与类图的区别	77
5.4 包图	77
5.4.1 概述	77
5.4.2 包	77
5.4.3 包之间的关系	78
5.4.4 合并包	79
5.5 UML 2.0 中的相关变化	80
5.6 Rational Rose 建模实例	80
5.6.1 使用 Rose 绘制类图的步骤	80
5.6.2 书店借书系统的类图	85
5.6.3 使用 Rose 绘制包图的步骤	86
5.6.4 书店借书系统的包图	89
5.7 常见问题与解答	89
5.8 本章小结	94
第 6 章 动态模型图	95
6.1 概述	95
6.2 活动图	95



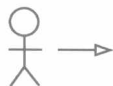
6.2.1	常见的主要概念	96
6.2.2	UML 2.0 中的新概念	99
6.2.3	创建活动图的步骤	101
6.2.4	Rational Rose 建模实例	101
6.3	顺序图	106
6.3.1	概念	107
6.3.2	对象的创建和销毁	109
6.3.3	UML 2.0 中的新概念	110
6.3.4	顺序图的用途	113
6.3.5	Rational Rose 建模实例	115
6.4	协作图	118
6.4.1	定义协作图	118
6.4.2	协作图与顺序图	118
6.4.3	UML 2.0 中的协作图	119
6.4.4	Rational Rose 建模实例	119
6.5	状态图	122
6.5.1	状态机	122
6.5.2	状态图	123
6.5.3	UML 2.0 中的新概念	126
6.5.4	状态图应用	127
6.5.5	Rational Rose 建模实例	130
6.6	UML 2.0 中的交互纵览图	134
6.7	UML 中的图及相互关系	136
6.8	常见问题与解答	136
6.9	本章小结	147
第 7 章	实现与部署模型图	148
7.1	概述	148
7.2	组件图	148
7.2.1	概念	148
7.2.2	关键技术	149
7.2.3	组件图与类图	151
7.2.4	UML 2.0 中的组件图	151
7.2.5	组件图的建模技术	152
7.2.6	Rational Rose 建模实例	152
7.3	部署图	154
7.3.1	概念	154

7.3.2	部署图的建模步骤	156
7.3.3	组件图与部署图	156
7.3.4	Rational Rose 建模实例	157
7.4	常见问题与解答	160
7.5	本章小结	163
第 8 章	UML 与统一开发过程	164
8.1	统一开发过程	164
8.1.1	定义	164
8.1.2	发展史	164
8.2	RUP	165
8.2.1	概述	165
8.2.2	二维开发模型	165
8.2.3	开发的 4 个阶段	166
8.2.4	开发的核心工作流	167
8.2.5	开发的资源分配	169
8.2.6	迭代策略	170
8.3	UML 与 RUP	171
8.3.1	概述	171
8.3.2	业务模型	171
8.3.3	需求捕获阶段	172
8.3.4	分析阶段	172
8.3.5	设计阶段	174
8.3.6	实现阶段	176
8.4	常见问题与解答	177
8.5	本章小结	179
第 9 章	双向工程	180
9.1	概述	180
9.2	正向工程	180
9.2.1	定义	180
9.2.2	生成代码的步骤	180
9.3	逆向工程	185
9.3.1	定义	185
9.3.2	逆向工程的步骤	185
9.4	应用实例	189
9.5	常见问题与解答	194
9.6	本章小结	194



第 10 章 在线销售系统	195
10.1 系统概述	195
10.2 需求分析	195
10.3 UML 建模	196
10.3.1 用例图	196
10.3.2 静态建模	197
10.3.3 动态模型	200
10.3.4 实现与部署	201
10.4 系统 Java 实现	203
10.5 常见问题与解答	209
10.6 本章小结	211
第 11 章 在线银行系统	212
11.1 系统概述	212
11.2 需求分析	212
11.2.1 功能需求	212
11.2.2 系统体系结构	212
11.3 UML 建模	213
11.3.1 用例图	213
11.3.2 静态建模	214
11.3.3 动态模型	216
11.3.4 实现与部署	218
11.4 系统的 Java 实现	219
11.5 常见问题与解答	229
11.6 本章小结	231
附录 A 扩展 UML	232
A.1 概述	232
A.2 标准扩展	232
A.3 标记值	232
A.3.1 利用用户配置文件定义标记值	233
A.3.2 定义自己的标记值	233
A.3.3 标准标记值	234
A.4 构造型	234
A.4.1 构造型的表示法	235
A.4.2 标准构造型	235
A.5 约束	236
A.6 本章小结	237

附录 B 对象约束语言	238
B.1 概述	238
B.2 OCL 元模型	238
B.3 标准 OCL 类型	239
B.3.1 基本类型	240
B.3.2 集合类型	241
B.4 OCL 表达式	241
B.4.1 OCL 表达式的语法	241
B.4.2 操作符的优先级	242
B.5 对象性质约束	243
B.5.1 不变量	243
B.5.2 前置条件与后置条件	243
B.5.3 let 表达式	243
B.6 OCL 中的关键字	244
B.7 本章小结	244
附录 C UML 的体系结构	245
C.1 概述	245
C.2 高级语言结构	246
C.3 元模型	247
C.4 审查 UML 内核	248
C.5 本章小结	250



第 1 章

UML 概述

1.1 统一建模语言

UML 是当今世界上面向对象系统开发领域中最激动人心的工具之一。在软件开发的每个步骤，从需求分析到技术规范，再到结构设计及配置要求，UML 都提供了模型化和可视化的支持。

1.1.1 什么是 UML

统一建模语言（UML，Unified Modeling Language）是一种可视化的建模语言，能让系统构造者用标准的、易于理解的方式表达出系统蓝图，便于用户、开发者之间进行交流设计结果。

UML 是一种基于面向对象的可视化建模语言，不同于其他程序设计语言。UML 模型使用代码生成器，可以将 UML 模型转换为程序源代码，使用逆向生成器也可以将程序源代码转换为 UML 模型。

1.1.2 UML 的发展

在 UML 产生之前，主流的面向对象分析和设计方法主要有 Booch、Jacobson（OOSE）和 Rumbaugh（OMT），每一种方法都各有优缺点及其表示符号，没有共同的标准，因此常常给软件开发工作者带来困惑。

1995 年 Grady Booch、James Rumbaugh 和 Ivar Jacobson 等人在总结了前期优秀成果的基础上创建了 UML。1995 年 10 月发布了第 1 个公开版本，称之为统一方法 UM 0.8（Unitied Method）。经过不断地补充和完善，于 1996 年 6 月和 10 月分别发布了两个新的版本，即 UML 0.9 和 UM 0.91，并将 UM 重新命名为 UML（Unified Modeling Language）。

在 1996 年成立了 UML 成员协会，主要是完善、加强和促进 UML 的定义工作。该机构对 UML 1.0（1997 年 1 月）及 UML 1.1（1997 年 10 月）的定义和发布起了重要的促进作用。

1997 年初，一些合作机构根据 OMG（Object Management Group，对象管理组）的提议，建立了一种通用的元模型来描述密集型软件系统。1997 年 11 月，UML 被 OMG 全体成员一致通过，并被采纳为标准。OMG 承担了进一步完善 UML 标准的工作。

2003 年发布了 UML 1.5，2005 年发布了 UML 2.0。

UML 的发展历程如图 1-1 所示。

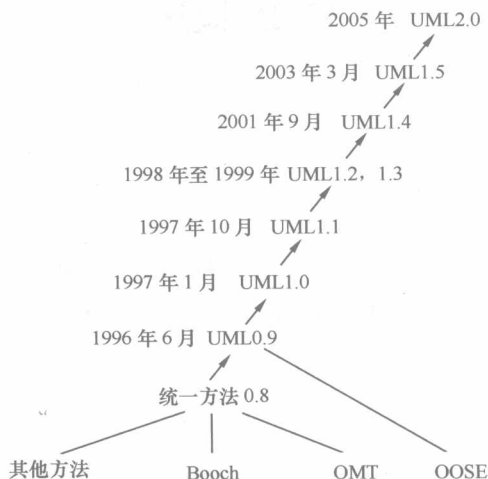


图 1-1 UML 的发展历程

1.2 UML 静态模型图

静态模型图描述系统应用的静态结构，包括类图、对象图、包图、组件图和部署图。

1.2.1 类图

一个系统可以看作是由一些不同类型的对象组成的，对象类之间的各种关系反映了系统内部各种成分之间的静态结构。类图主要用来描述系统中各种类之间的静态结构。类图适用于软件开发的各个阶段。

类图的标识符是一个长方形，其中包括类的名称、属性和操作，放在 3 个不同的区域中，如图 1-2 所示。

图 1-2 所示的类图中，顶部区域是类的名字 Account（账户）。类的属性放在中间区域，Account 类的属性包括 accountId（账户 ID）、cId（客户 ID）、type（账户类型）和 balance（账户余额）。底部区域列出的是类的操作，Account 类的操作包括 saving（存款）、expend（取款）和 update（更新账户）。

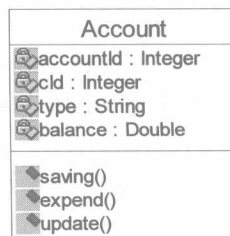


图 1-2 一个简单的类图

1.2.2 对象图

对象图是类图的一个实例，用于显示系统执行时的一个可能的快照，即在某一时间点上系统可能呈现出的样子。对象图用带下划线的对象名称来表示对象，如图 1-3 所示。

图 1-3 建立了一个洗衣机（WashingMachine）类的对象，对象名为 myWasher。该对象具有以下属性值。

- 品牌名称：Laundatorium。
- 型号名称：Washmeister。
- 产品编号：GL57774。
- 容量：16。

<u>myWasher : WashingMachine</u>
brandName : string = Laundatorium
modelName : string = Washmeister
serialNumber : string = GL57774
capacity : short = 16

图 1-3 一个简单的对象图

对象图并不像类图那样具有重要的地位，但是利用它可以帮助分析人员通过具体的实例进行分析，更加直观地了解类图所表达的丰富内涵。对象图还常被用作协作图的一部分，用以展示一组对象实例之间的动态协作关系。

1.2.3 组件图

组件图描述代码组件的物理结构及各组件之间的依赖关系。组件可以是可执行程序、库、表、文件和文档等，包含了逻辑类或者逻辑类的实现信息。

组件图中也可以包括包或者子系统，它们都用于将模型元素组织成较大的组件，如图 1-4 所示。

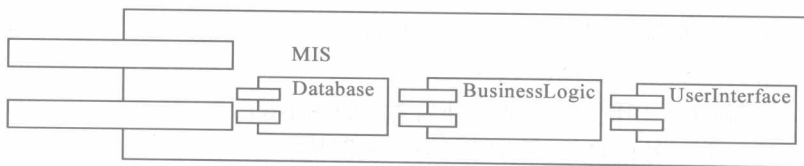


图 1-4 组件图

图 1-4 表示 MIS 组件包含了 Database 组件、BusinessLogic 组件和 UserInterface 组件。

1.2.4 部署图

部署图展示了运行时处理节点及其组件的部署。它描述系统硬件的物理拓扑结构（包括网络布局和构件在网络上的位置）以及在此结构上执行的软件（即运行时软件在节点中的分布情况）。它说明系统结构的静态部署视图，即说明了分布、交付和安装的物理系统。

图 1-5 描述的是一个 B/S 系统的部署图。该系统包含 Web 浏览器、Apache 服务器、MySQL 数据库服务器以及 Printer 4 个节点。

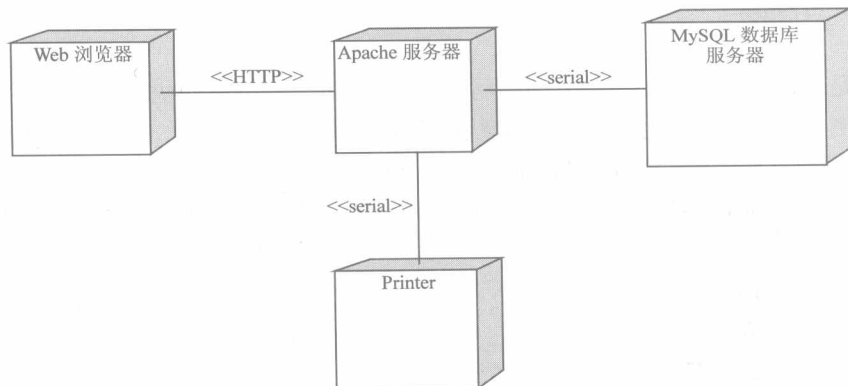


图 1-5 某个 B/S 系统的部署图

1.3 UML 动态模型图

动态模型图描述了系统动态行为的各个方面，包括用例图、序列图、协作图、活动图和状态图。

1.3.1 用例图

用例图描述系统外部的执行者与系统提供的用例之间的某种联系。所谓用例是指对系统提供的功能的一种描述。执行者是使用用例的人或外部系统，二者之间的联系描述了“谁使用哪个用例”。用例图着重于从系统外部执行者的角度来描述系统需要提供哪些功能，并指明该执行者是谁。

在用例图中，椭圆表示用例，小人表示执行者，如图 1-6 所示。

图 1-6 所示是一个家教信息系统的用例图。该图描述了当一个学生进入家教信息系统时，“学生”是该系统的执行者，学生具有“注册信息”、“修改资料”、“查看家教信息”和“搜索家教信息”4 个用例。

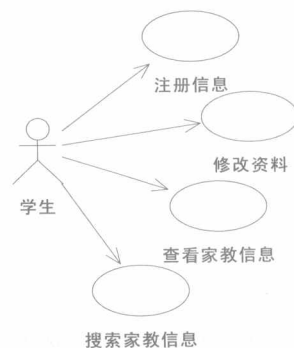


图 1-6 简单用例图

1.3.2 序列图

序列图又被称为时序图、顺序图，用于描述几个对象之间的动态协作关系。一个资金转账的序列图，如图 1-7 所示。序列图侧重于描述对象之间传送消息的时间顺序，反映了对象之间的一次特定的交互过程。

序列图由一组对象构成，每个对象分别带有一条竖线，代表时间轴，称为生命线。从一个对象的生命线指向另一个对象的生命线的水平箭头表示消息。序列图就是描述这些对象随着时间的推移相互之间交换消息的过程。

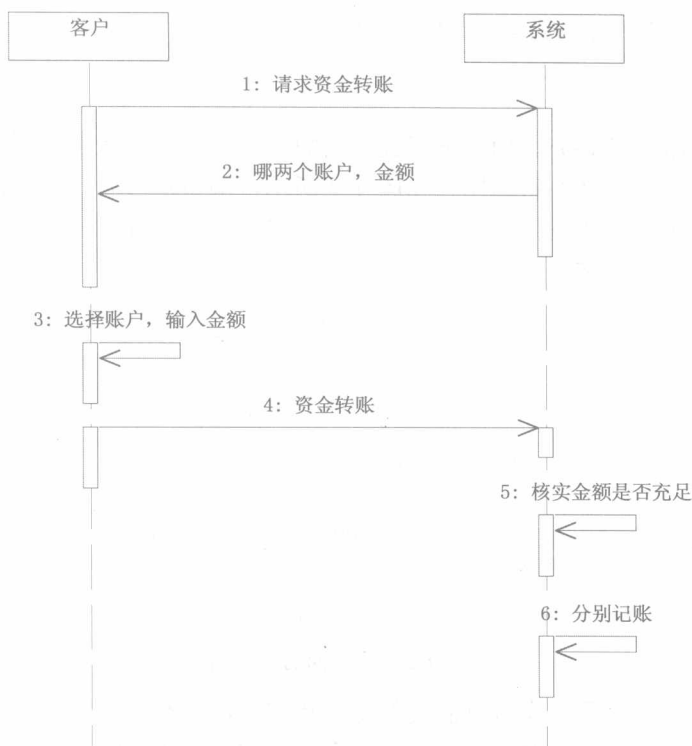


图 1-7 资金转账的序列图

图 1-7 所示是一个资金转账的序列图,描述了当客户向系统请求资金转账时的交互过程。通过该模型图的描述,可以很清晰地了解客户与系统之间的交互过程及其交互顺序。

1.3.3 协作图

与序列图相同,协作图也是用来描述系统中对象之间的动态协作关系。协作图侧重于描述各对象之间存在的消息收发关系,而不专门突出这些消息发送的时间顺序。图 1-8 就是图 1-7 对应的协作图,强调了客户与系统之间消息收发的过程。

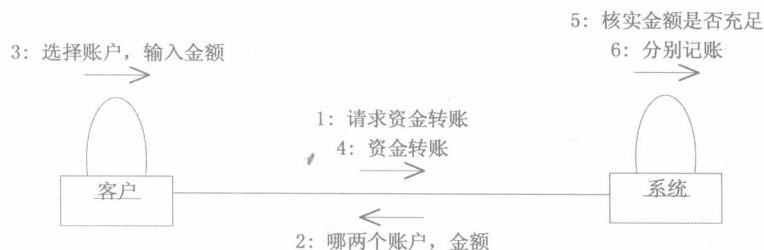


图 1-8 资金转账的协作图

协作图是对象图的另一种表达方式。在图 1-8 中的对象之间的消息箭头表示它们之间的