重庆大学本科课程

面向对象技术与 UML

Object-Oriented Technique and UML

重庆大学软件学院

教材目录 《面向对象技术UML教程》

教材目录 《UML基础、案例与应用（第三版）》

第一部分 基础知识

第二部分 学习案例

第三部分 高级应用

第1章　UML简介

什么是UML： 统一建模语言（Unified Modeling Language)

1.1　在纷繁复杂中寻求解决问题的方法

1.2　UML的诞生

1.3　UML的组成

类图、对象图、用例图、状态图、顺序图、活动图、协作图、构件图、部署图

1.4　其他特征

1.5　UML2.0中的新图

- 为什么需要UML
- UML的诞生
- 如何用图表示UML模型的各个部分
- 为什么使用UML提供的不同类型的图对我们来说很重要

类图

◇ 什么是类（**class**）

类是一类或者一组具有类似属性和共同行为的事物

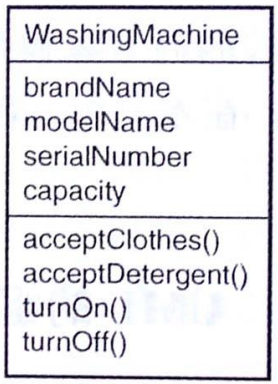


图 1.1　UML 类图标

◇ **UML**类图的特点

* 矩形方框
* 被分为三个区域：类名、类的属性、类的操作
* 类名由多个单词组成；每个单词的首字母要大写，单词之间不用空格
* 属性名和操作名也类似，但首字母不用大写
* 每个操作名的后面都有一对括号

对象图

◇ 什么是对象（object）

对象是一个类的实例，是具有具体属性值的一个具体事物。

◇ UML对象图标的特点

* 矩形方框
* 对象名首字母为小写，对象名下面要带下划线
* 冒号左边为实例名，冒号右边为类名；对象也可以是匿名

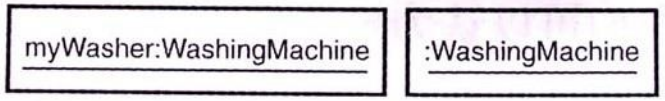


图 1.2　UML 对象图标，左边的图标代表一个具名的对象，右边的图标代表一个匿名的对象

用例图

◇ 什么是用例（**use case**）

用例是从用户的观点对系统行为的一个描述；它是用来从用户的观察角度收集系统需求的主要技术

◇ 用例图的特点

* 直立小人被称为参与者（**actor**）；参与者可以是一个人，也可以是另一个系统
* 椭圆形代表用例
* 矩形代表系统

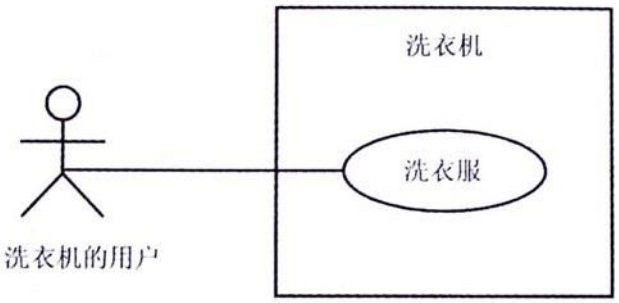


图 1.3　UML 用例图

状态图

◇ 什么是状态（**state**）

在任一给定的时刻，一个对象总是处于某一特定的状态。

◇ **UML**状态图的特点

* 圆角矩形
* 最顶端的符号（实心圆）代表起始状态，

而最底端的符号（眼形圆）表示终止状态

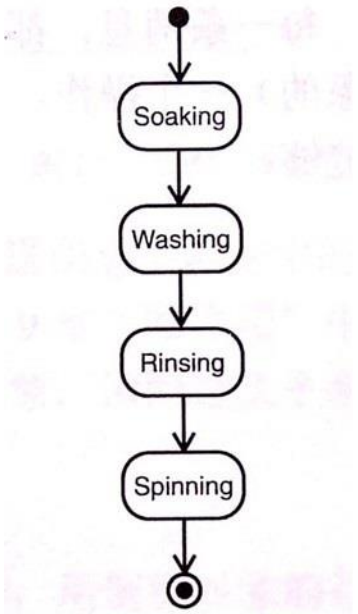


图 1.4　UML 状态图

顺序图

◇ 顺序（sequence）

类图和对象图表达的是系统的静态结构。在一个运行的系统中，对象之间要发生交互，并且这些交互要经历一定的时间。UML顺序图所表达的正是这种基于时间的动态交互。

◇ UML图的特点

* 横坐标为系统中的对象
* 每个对象都有一个或多个操作
* 对象间通过相互传递消息来协同工作
* 纵坐标为时间序列

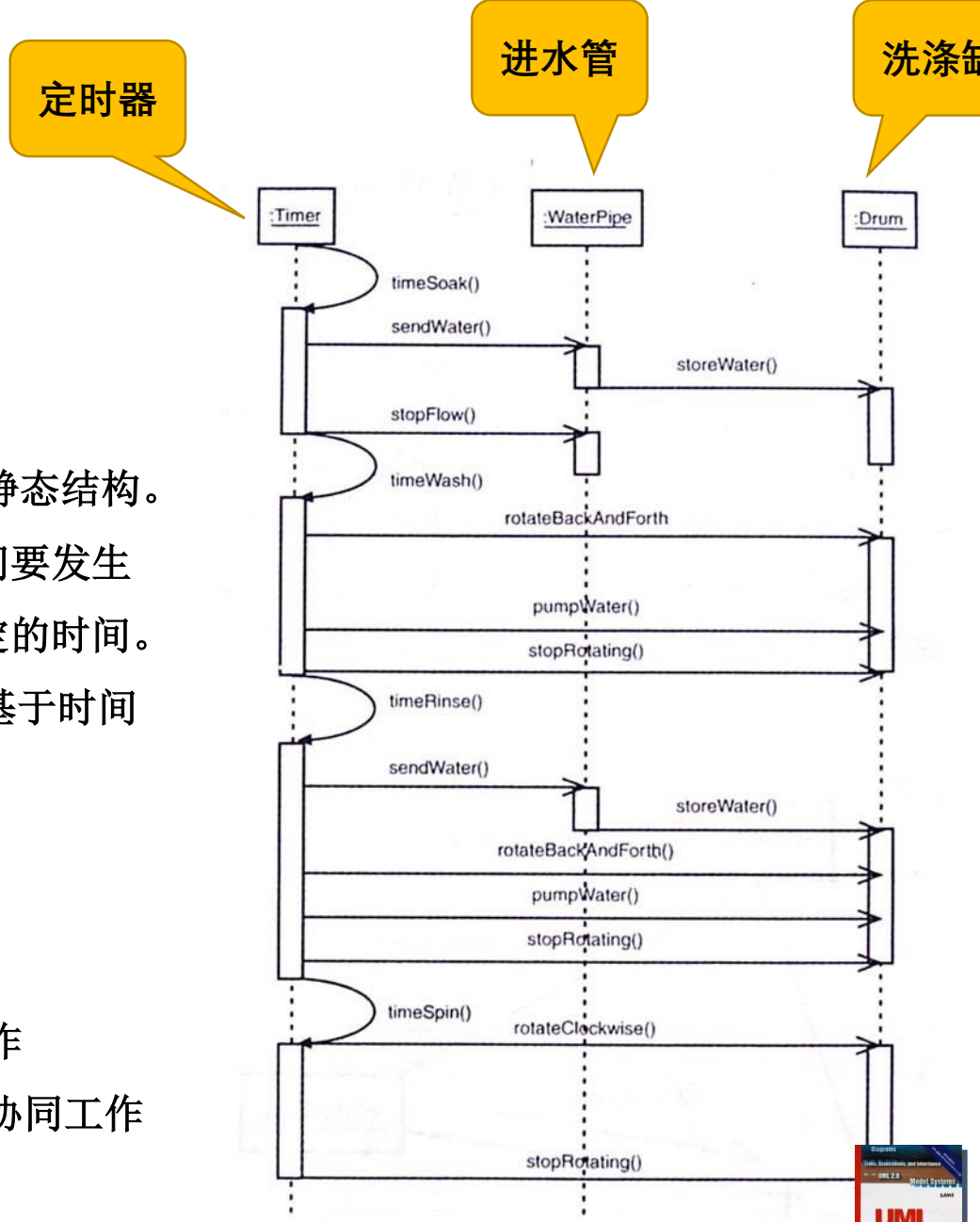


图 1.5 UML 顺序图

活动图

◇ 什么是活动（**active**）

活动即工作步骤

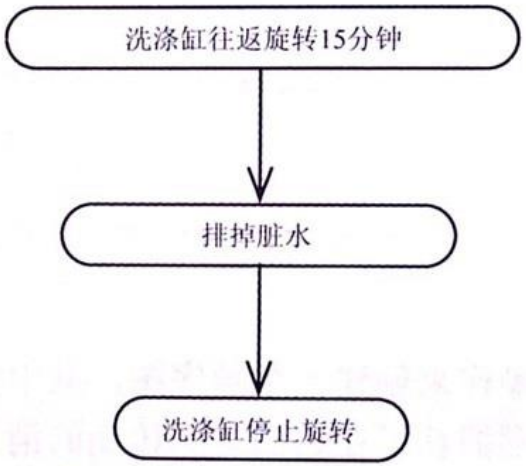


图 1.6　UML 活动图

◇ **UML**活动图的特点

* 和流程图很接近
* 圆角矩形（比状态图更窄，更接近与椭圆）
* 箭头表示活动的转移
* 实心圆代表起点，眼形圆代表终点

协作图

◇ 协作（**collaboration － communication**）图的用途

协作图用于展示对象之间的交互关系。

对象图展示出对象之间的静态关系。协作图是对对象图的扩展。协作图除了展示对象之间的关联，还显示出对象之间的消息传递。

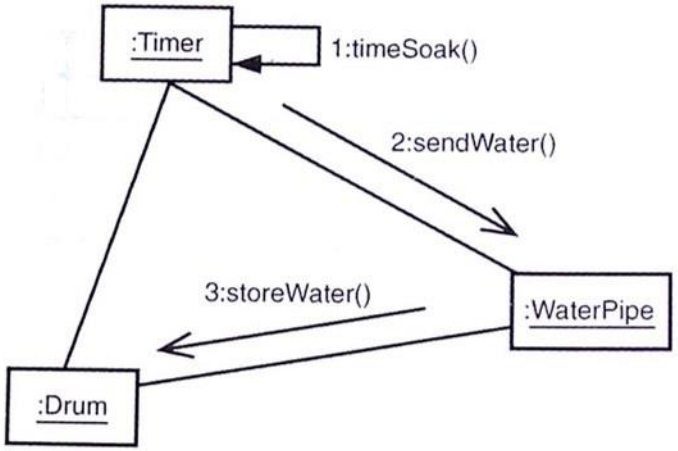


图 1.7　UML 协作图

◇ **UML**类图的特点

＊ 关联线附近的箭头线表示对象之间传递的消息，箭头指向消息接收对象

＊ 消息名称和消息序号附在箭头线附近

＊ 顺序图和协作图之间可以相互转换

构件图

◇ 什么是构件（component）

软件构件是软件系统的一个物理单元。在UML中，数据文件、表格、可执行文件、文档和动态链接库等都被定义为构件。

构件图和部署图与整个计算机系统密切相关。

图 1.8　UML 1.x 版本中的软件构件图标

◇ UML构件图的特点

＊ 一个左侧附有两个小矩形的大矩形框

＊ 也可以用一个顶部带关键字《Component》的矩形表示

图 1.9　UML 2.0 中的软件构件图标记

软件复用和软件构件

软件复用(SoftWare Reuse)是将已有软件的各种有关知识用于建立新的软件,以缩减软件开发和维护的花费。软件复用是提高软件生产力和质量的一种重要技术。早期的软件复用主要是代码级复用，被复用的知识专指程序，后来扩大到包括领域知识、开发经验、设计决定、体系结构、需求、设计、代码和文档等一切有关方面。

构件技术

随着软件复杂度的与日俱增，传统的把整个软件的源程序拿来静态编译的方法显然不适合了。在这个前提下， 产生了软件拼装模式，把软件分成一个个相对独立的目标代码模块， 称之为构件 。

软件开发人员只需要做和自己相关的构件，编译通过，就能够拿来和其他模块组装在一起使用。通过装卸实现某个功能的构件，就可以实现对系统的灵活升级。

如今，已经成熟且广为使用的构件技术有微软的COM(Component Object Model)，OMG 组织的CORBA (Common Object Request Broker Architecture) 等等，用它们生成的构件都是基于二进制目标代码的。现在大行其道的Java 和. NET ，虽然生成的程序都是基于中间代码的，但也处处体现着构件技术的思想。

部署图

◇ 部署（deployment）图的用途

UML部署图显示了基于计算机系统的物理体系结构。

◇ UML部署图的特点

* 立方体图标
* 立方体之间的连线表示体系之间的关系

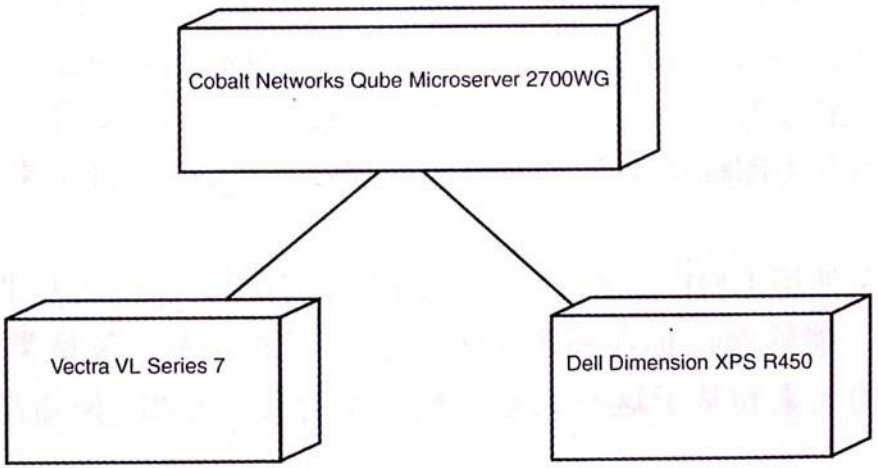


图 1.10　UML 部署图

其他特征

◇ 注释（**note**）

通过附加注释来做解释说明。

◇ 符号特征

＊ 带折角的矩形，矩形中是解释性文字

＊ 注释和被注释的图元素之间用一条虚线连接

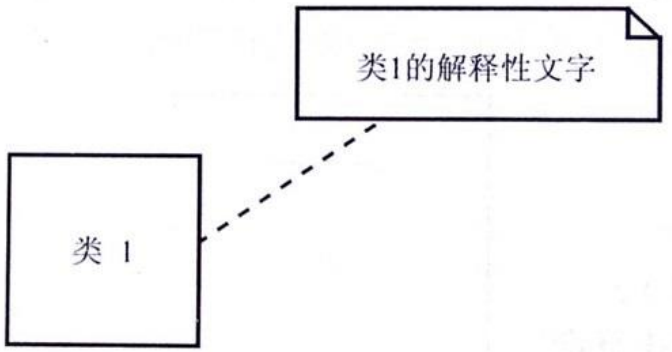


图 1.11　任何图中都可以附加注释来做解释说明

其他特征

◇ 构造型（版型）（**stereotype**）

版型是建模人员在已有的构造块上派生出的新构造块，这些新构造块是和特定问题相关的。

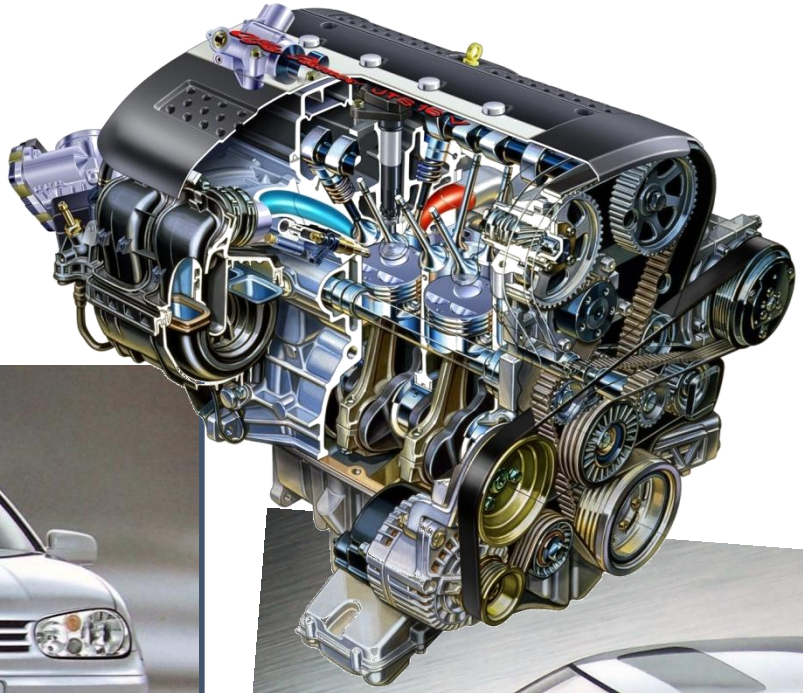
版型可以应用于所有类型的模型元素，包括类、节点、构件、注解、关系、包、操作等。

◇ 关键字（**keyword**）

版型用两对尖括号括起来的一个名称来表示，这个括号叫做双尖括号（**guillemots**）。这个被括起来的名称叫做关键字。

如**<<Interface>>，<<entity>>**

为什么需要这么多种图



为什么需要这么多种图

每一种UML图都提供一种组成特殊视图的方式。

采用多视角的目标是为了能够和每一类风险承担人良好地沟通。

UML是一套表示法系统。

UML由一组图组成，它使得系统分析员可以利用这一标准来建立能够和客户、程序员以及任何参与程序开发的人员理解的多视角的系统蓝图。不同的风险承担人通常使用不同类型的图相互交流。

UML模型只说明一个系统应该做什么，并没有告诉我们系统应该怎么做。

Enterprise Architect 视频教程：

http://www.sparxsystems.cn/resources/demos/index.html

判断题

1、UML中一共有九种图：它们是用例图、类图、对象图、顺序图、协作图、状态图、活动图、构件图、部署图

2、用例图是从程序员角度来描述系统的功能

3、类图是描述系统中类的静态结构，对象图是描述系统中类的动态结构

4、活动图和状态图用来描述系统的动态行为

5、协作图的一个用途是表示一个类操作的实现

选择题

6、请在下面选项目中选出两种可以互相转换的图

(a) 顺序图　(b)协作图 (c) 活动图　(d) 状态图

7、下面哪些图可用于BD阶段

(a)用例图　(b)构件图 (c)类图　(d)顺序图