



第3章 软件体系结构描述语言



目录

3.1 软件体系结构描述语言

3.2 形式化软件体系结构描述语言-C2SADL

3.3 非形式化软件体系结构描述语言- UML



目标

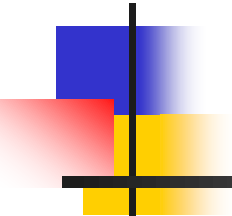
- 了解软件体系结构描述语言-**C2**
- 掌握软件体系结构描述语言-**UML**



3.1 软件体系结构描述语言

3.1.1 软件体系结构描述语言

- 为了支持软件体系结构开发，需要**形式化建模符号**和分析开发工具。
- ADL (Architecture Description Language)和对应的工具包就能够解决这一问题。
 - **形式化建模语言**
 - **非形式化的建模语言**



3.1.1 软件体系结构描述语言

ADL的三个基本元素是：

- 构件
- 连接件
- 体系结构配置



3.1.1 软件体系结构描述语言

典型的软件体系结构描述语言（形式化）

ADL

研发组织

Unicon

Carnegie Mellon大学

Wright

Carnegie Mellon大学

C2

Southern California大学

Rapide

Stanford大学

SADL

SRI

Aesop

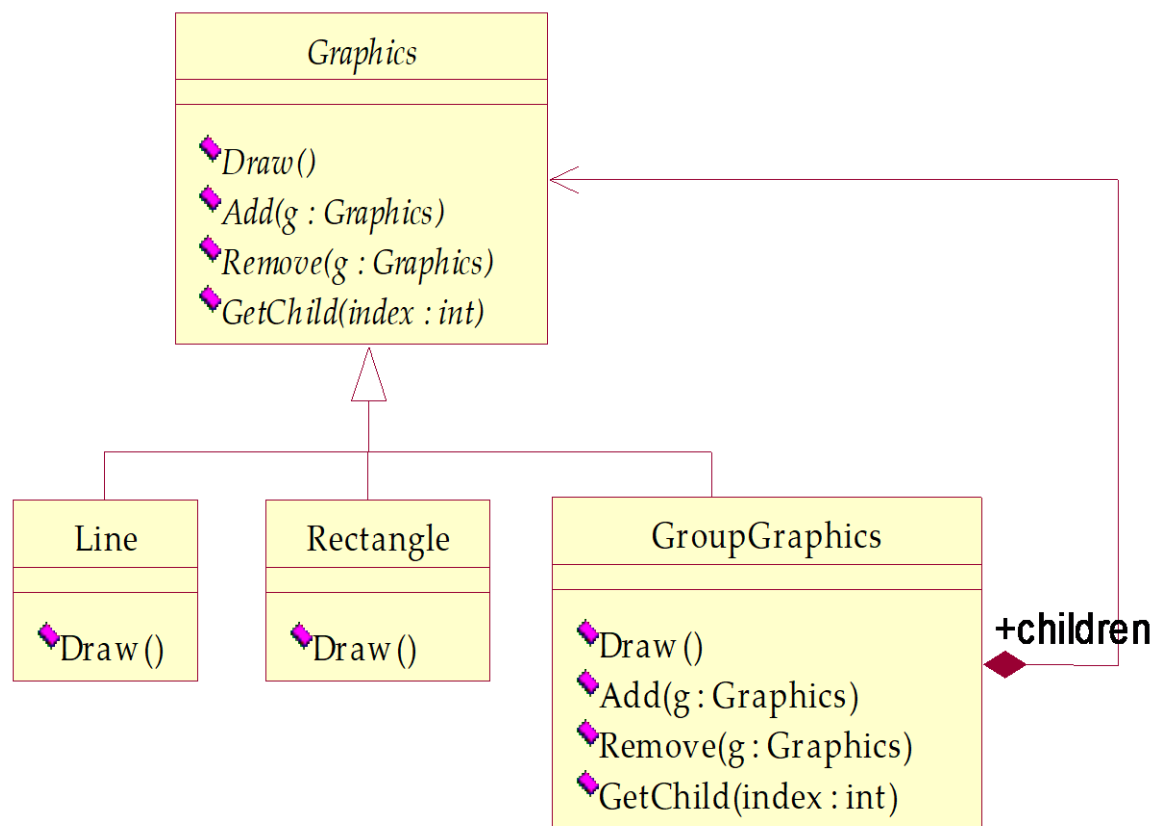
Carnegie Mellon 大学

ACME

Carnegie Mellon 大学

3.1.1 软件体系结构描述语言

典型的软件体系结构描述语言（非形式化）



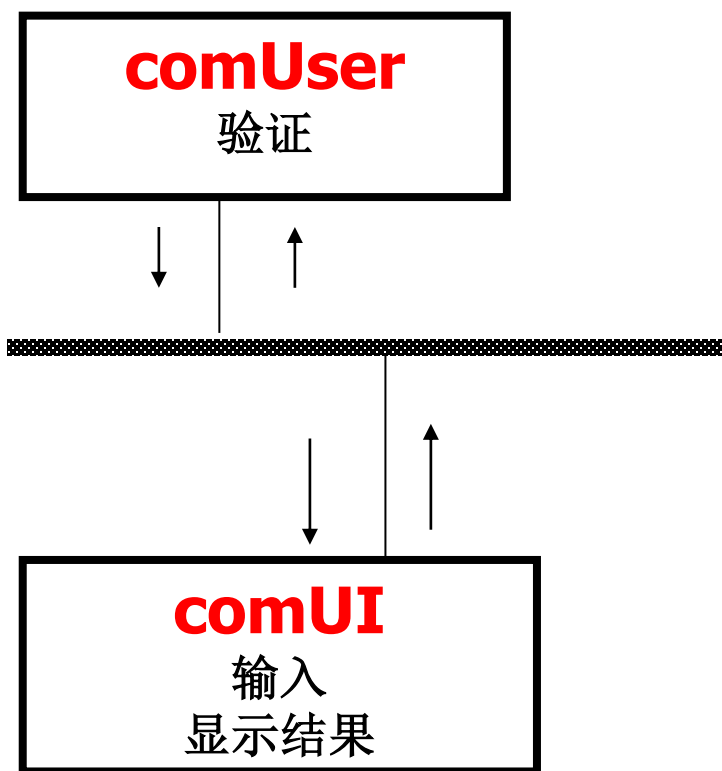


3.2 形式化软件体系结构描述语言-C2SADL

- C2是一种常见(有图形化用户界面)的软件体系结构风格
- C2SADL(Software Architecture Description Language)是用于描述C2风格的体系结构的ADL。
- 常常用C2这一简称来指代C2风格和C2 SADL的结合

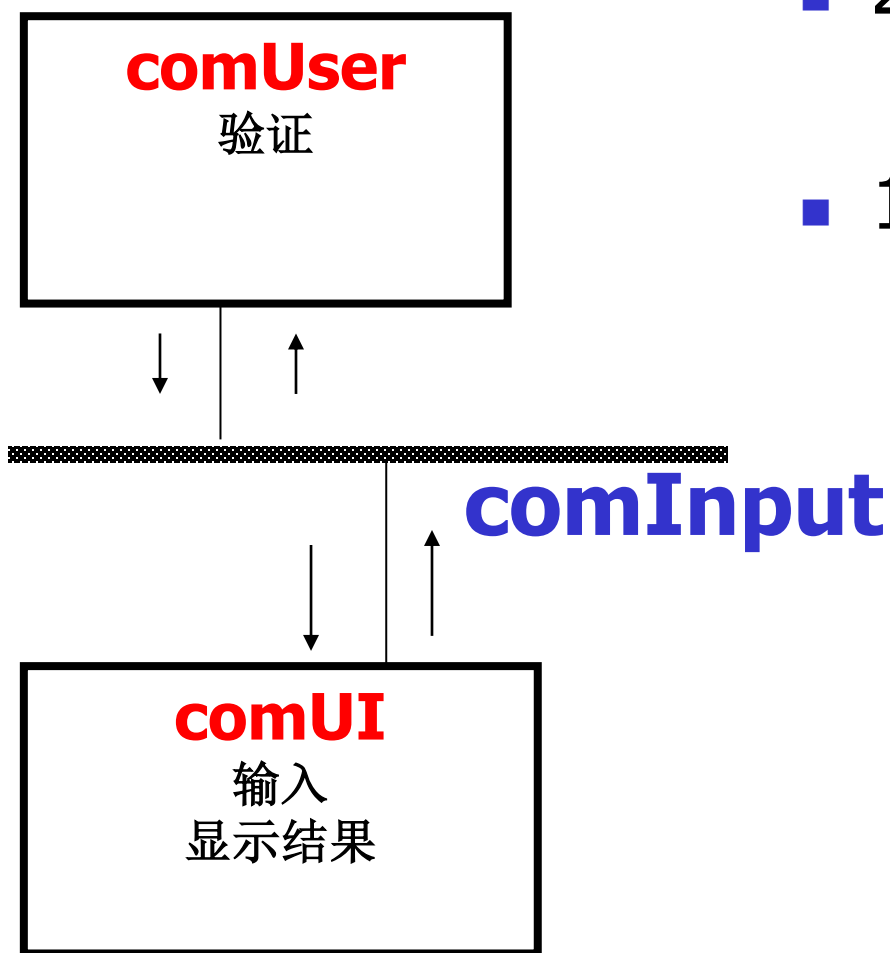
3.2.1 C2风格

- C2底层向上层索取数据，递交请求
- C2上层为下层提供数据、反馈结果，不能递交请求



3.2.2 C2风格实例

■ 用户验证



- 2个构件:
 - comUI ,comUser
- 1个连接件:
 - comInput



3.2.3 C2描述语法

■ C2对构件的描述

```
component <identifier> is (构件名)
interface
  top_domain is(面向top的接口)
  bottom_domain is(面向bottom的接口)
  out
    ...//请求、输出方法体
  in
    ...//内部、输入方法体
  [behavior]
  [context component_context]
end <identifier>
```



3.2.3 C2描述语法

■ C2对连接件的描述

connector <identifier> conections

top_ports top_component1; top_component2...

bottom_ports bottom_component



3.2.3 C2描述语法

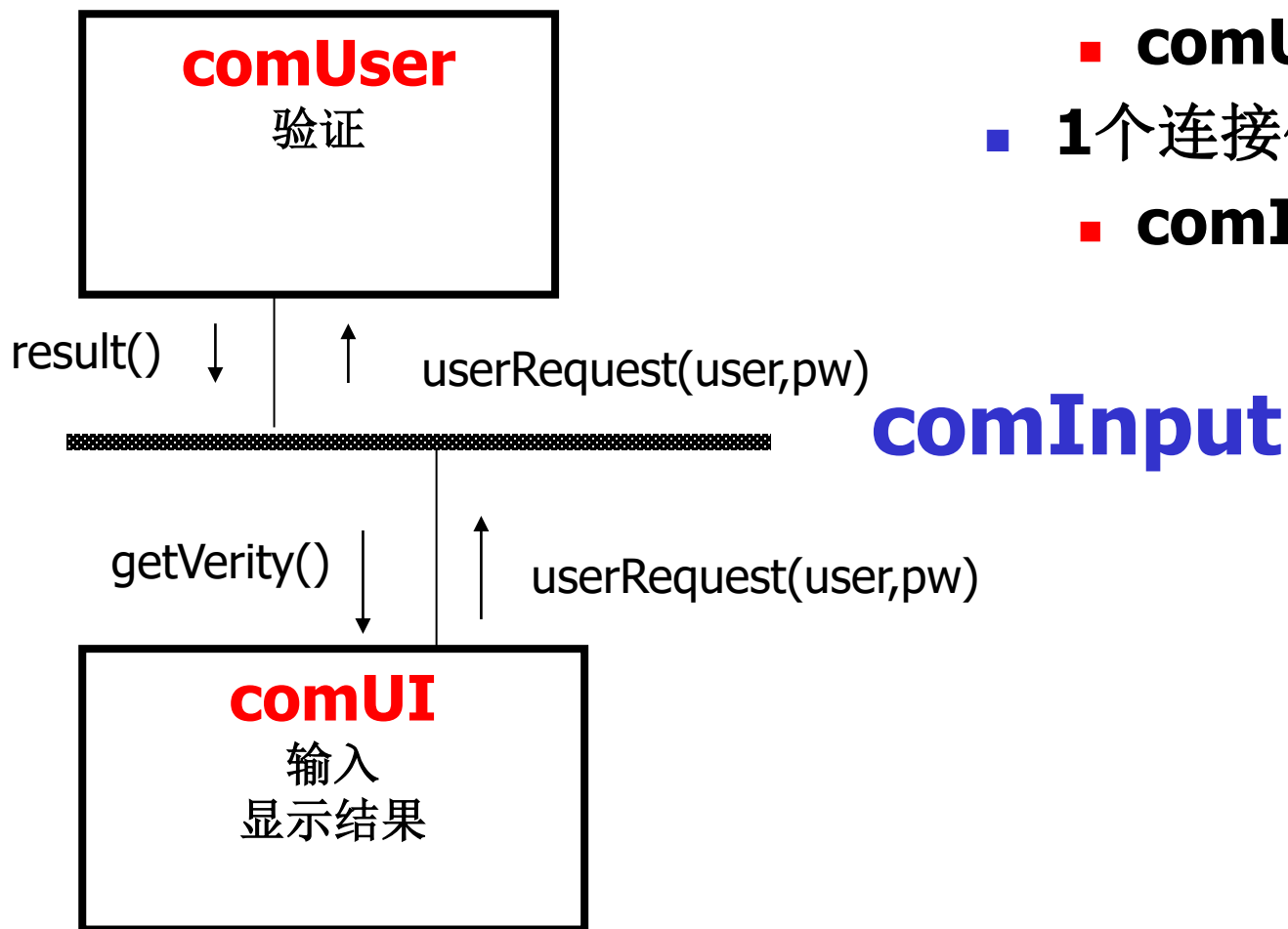
■ C2对体系结构（系统）的描述

语法：

```
system 系统名_1 is
    architecture 系统名 with
        instance (实例)
        instance
end 系统名_1
```

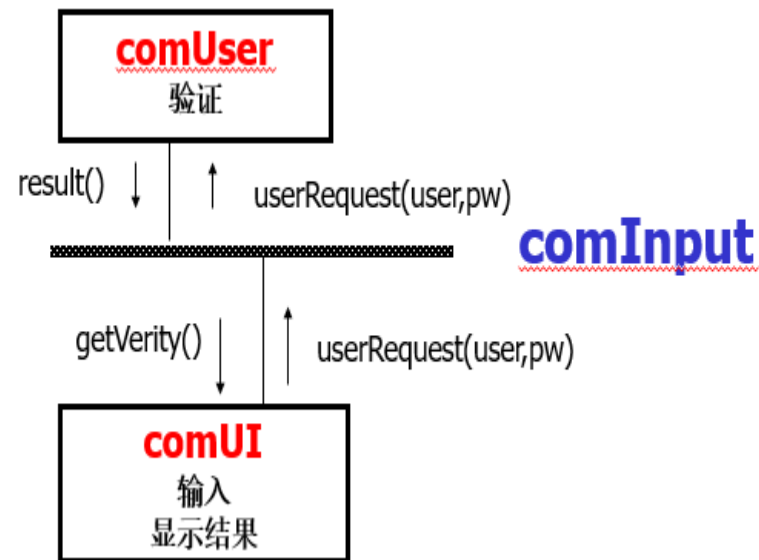
3.2.4 C2风格描述实例1-用户验证

- 2个构件：
 - **comUI ,comUser**
- 1个连接件：
 - **comInput**



3.2.4 C2风格描述实例1-用户验证 comInput连接件的描述

connector comInput connections
top_ports comUser
bottom_ports comUI

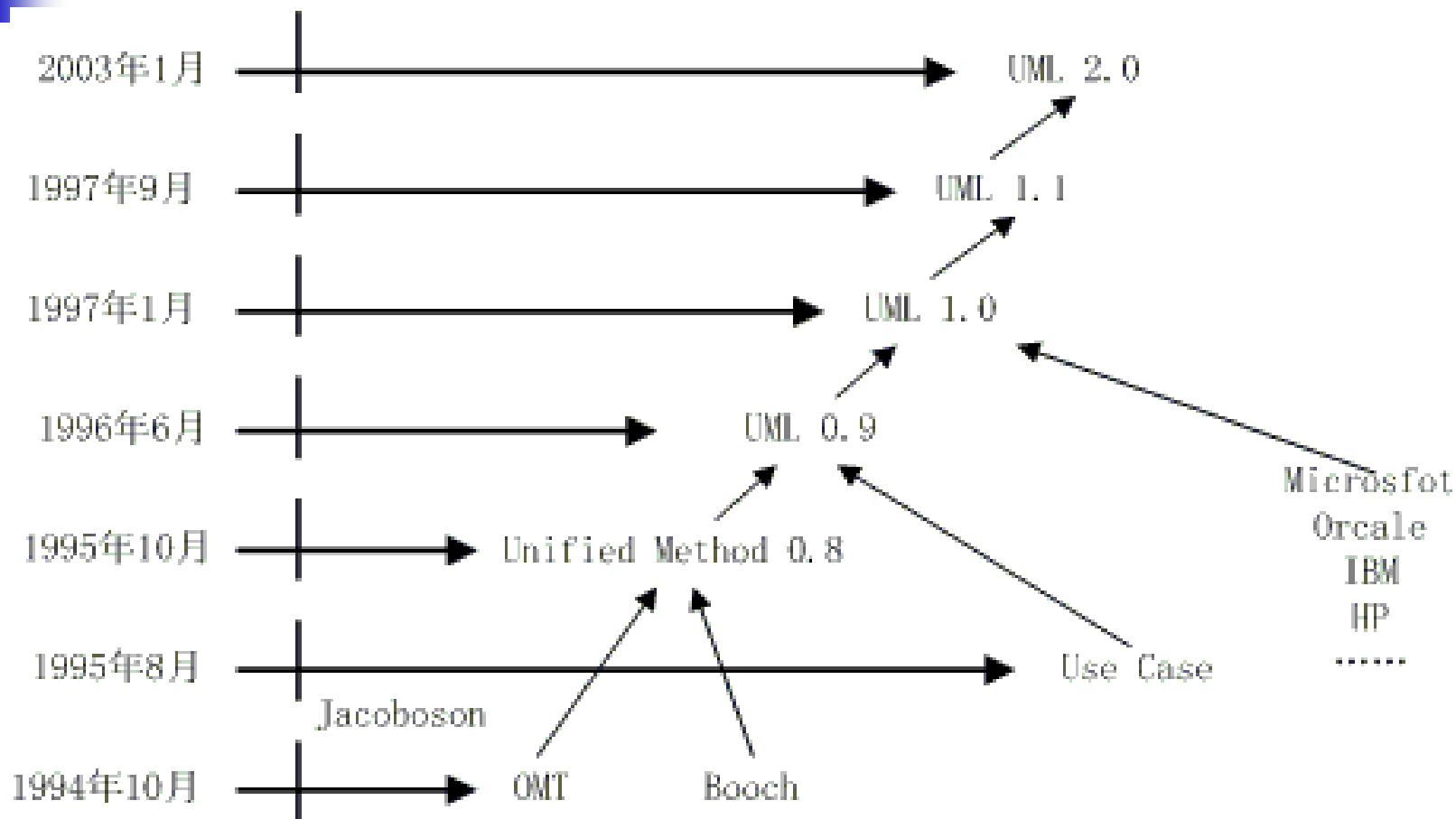




3.3 非形式化软件体系结构描述语言- UML

- UML(Unified Modeling Language)统一建模语言，可视化建模语言
- 支持面向对象开发过程

UML历史



3.3.1 UML概述

(1) UML简介

- UML (Unified Modeling Language)为面向对象软件设计提供统一的、标准的、可视化的建模语言。
- 适用于描述以用例为驱动，以体系结构为中心的软件设计的全过程
- UML的定义包括UML语义和UML表示法两个部分。
 - ① **UML表示法**：UML表示法定义UML符号的表示法，为开发者或开发工具使用这些图形符号和文本语法为系统建模提供了标准。
 - ② **UML语义**：UML对语义的描述使开发者能在语义上取得一致认识，消除了因人而异的表达方法所造成的影响



3.3.1 UML概述

(2) UML模型图的构成

- 事物(Things): UML模型中最基本的构成元素, 是具有代表性的成分的抽象
- 关系(Relationships): 关系把事物紧密联系在一起
- 图(Diagrams): 图是事物和关系的可视化表示

3.3.2 UML事物

- (1) UML包含4种事物：

构件事物、行为事物、分组事物、注释事物

- (2) 构件事物：UML模型的静态部分，描述概念或物理元素
它包括以下几种：

类：具有相同属性相同操作 相同关系相同语义的对象的描述

接口：描述元素的外部可见行为，即服务集合的定义说明

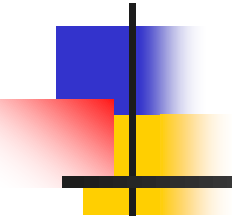
协作：描述了一组事物间的相互作用的集合

用例：代表一个系统或系统的一部分行为，是一组动作序列的集合

构件：系统中物理存在，可替换的部件

节点：运行时存在的物理元素

- 另外，参与者、信号应用、文档库、页表等都是上述基本事物的变体

- 
- (3) 行为事物：UML模型图的动态部分，描述跨越空间和时间的行为

交互：实现某功能的一组构件事物之间的消息的集合，涉及消息、动作序列、链接

状态机：描述事物或交互在生命周期内响应事件所经历的状态序列

- (4) 分组事物：UML模型图的组织部分，描述事物的组织结构

包：把元素组织成组的机制

- (5) 注释事物：UML模型的解释部分，用来对模型中的元素进行说明，解释

注解：对元素进行约束或解释的简单符号

3.3.3 UML关系

■ (1) 依赖

依赖(**dependency**)是两个事物之间的语义关系，其中一个事物(独立事物)发生变化，会影响到另一个事物(依赖事物)的语义

■ (2) 关联

关联(**association**)是一种结构关系，它指明一个事物的对象与另一个事物的对象间的联系

■ (3) 泛化

泛化(**generalization**)是一种特殊/一般的关系。也可以看作是常说的继承关系

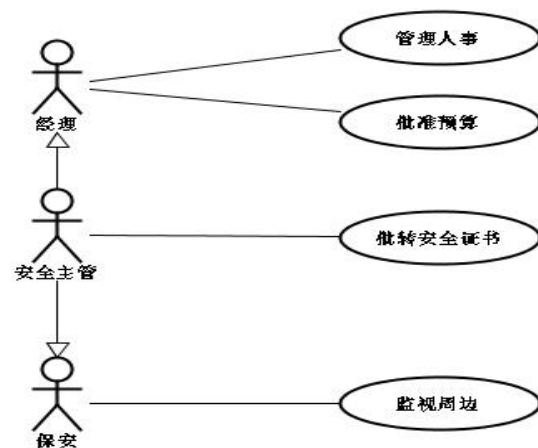
■ (4) 实现

实现(**realization**)是类元之间的语义关系，其中的一个类元指定了由另一个类元保证执行的契约

3.3.4 各UML图及特征

■ (1) 用例图(Use Case Diagram)

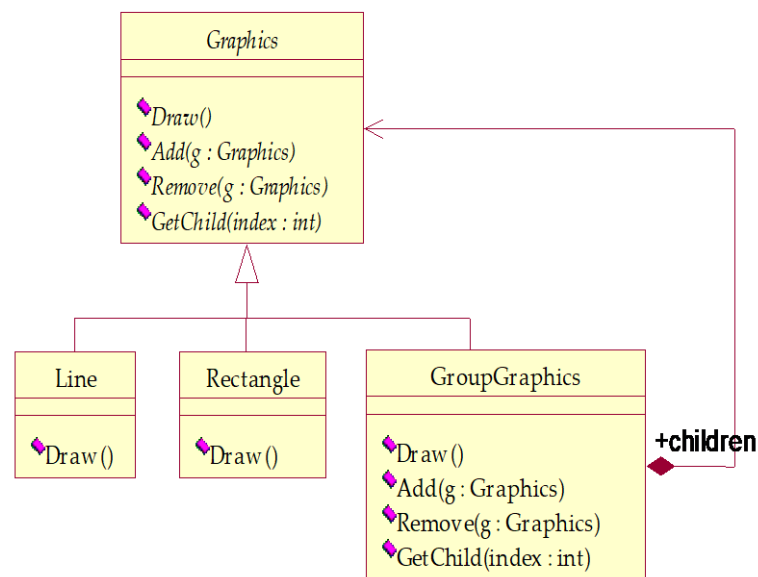
※用例图是从用户角度描述系统功能，是用户所能观察到的系统功能的模型图，用例是系统中的一个功能单元



■ (2) 类图(Class Diagram)

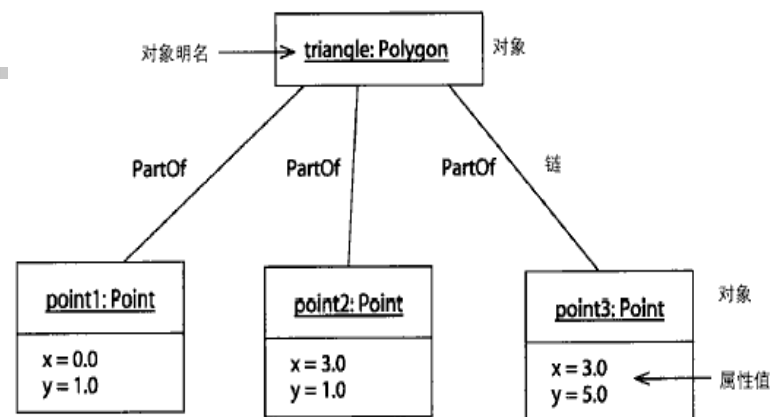
※类图描述系统中类的静态结构。不仅定义系统中的类，表示类之间的联系如关联、依赖、聚合等，也包括类的内部结构(类的属性和操作)

※类图是以类为中心来组织的，类图中的其他元素或属于某个类或与类相关联



■ (3) 对象图(Object Diagram)

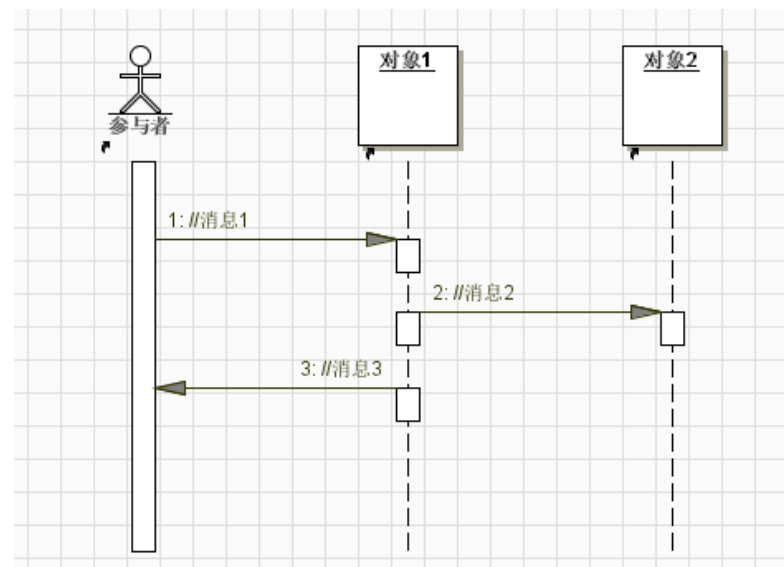
※ 对象图是类图的实例，几乎使用与类图完全相同的标识。他们的不同点在于对象图显示类的多个对象实例，而不是实际的类



■ (4) 顺序图(Sequence Diagram)

※ 顺序图显示对象之间的动态合作关系，它强调对象之间消息发送的顺序，同时显示对象之间的交互

※ 顺序图的一个用途是用来表示用例中的行为顺序。当执行一个用例行为时，顺序图中的每条消息对应了一个类操作或引起状态转换的触发事件

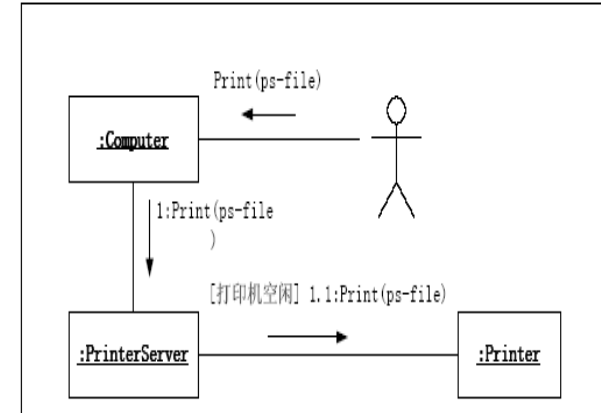


■ (5) 通信图(Communication Diagrams) uml2.0

协作图(Collaboration Diagram) uml1.0

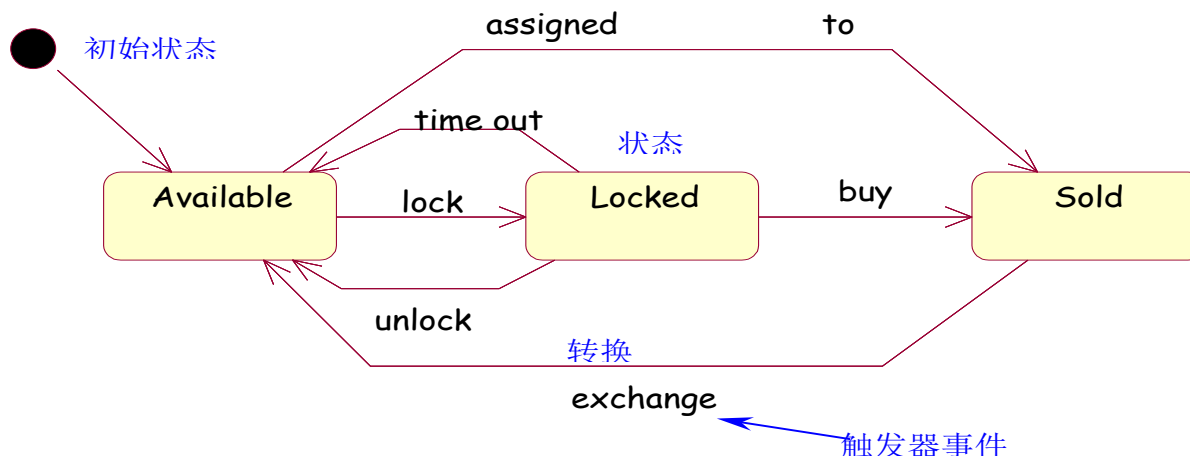
※通信图描述对象间的协作关系，协作图跟顺序图相似，显示对象间的动态合作关系。除显示信息交换外，协作图还显示对象以及它们之间的关系。

※通信图的一个用途是表示一个类操作的实现



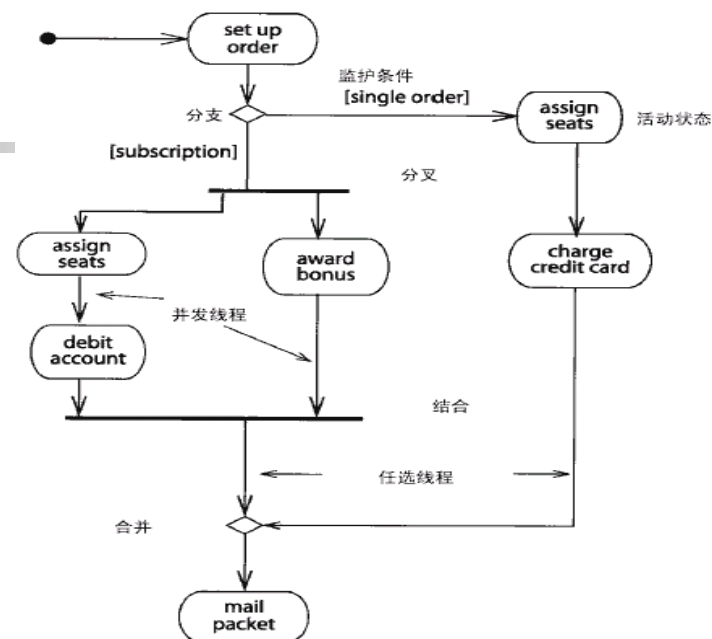
■ (6) 状态图(State Chart Diagram)

※状态图是一个类对象所可能经历的所有历程的模型图。状态图由对象的各个状态和连接这些状态的转换组成



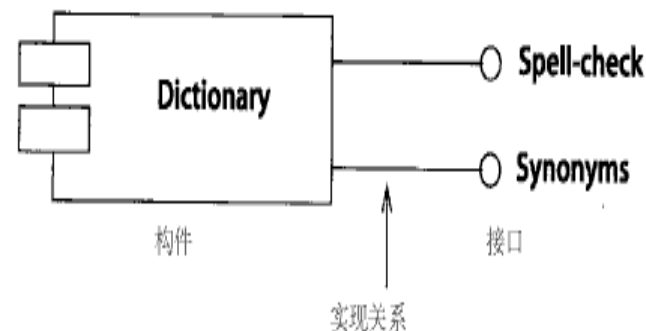
■ (7) 活动图(Activity Diagram)

- ※ 活动图是状态图的一个变体，用来描述执行算法的工作流程中涉及的活动
- ※ 活动图描述了一组顺序的或并发的活动



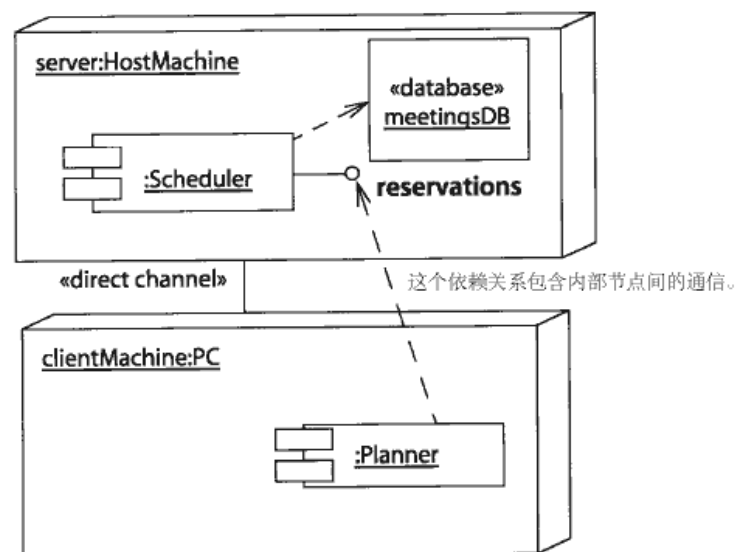
■ (8) 构件图(Component Diagram)

- ※ 构件图为系统的构件建模——构件即构造应用的软件单元——还包括各构件之间的依赖关系，以便通过这些依赖关系来估计对系统构件的修改给系统可能带来的影响

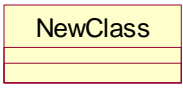
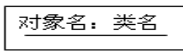
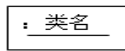




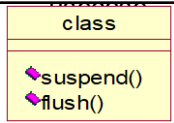
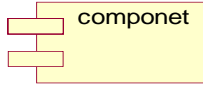




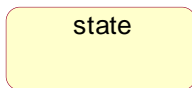
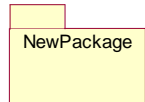

■ (9) 部署图(Deployment Diagram)

部署视图描述位于节点实例上的运行构件实例的安排。节点是一组运行资源，如计算机、设备或存储器。这个视图允许评估分配结果和资源分配



3.3.5 UML语法描述

类	是对一组具有相同属性、相同操作、相同关系和相同语义的对象的描述	
对象	  	
接口	是描述了一个类或构件的一个服务的操作集	
协作	定义了一个交互，它是由一组共同工作以提供某种协作行为的角色和其他元素构成的一个群体	
用例	是对一组动作序列的描述	
主动类	对象至少拥有一个进程或线程的类	
构件	是系统中物理的、可替代的部件	
参与者	在系统外部与系统直接交互的人或事物	

节点	是在运行时存在的物理元素	
交互	它由在特定语境中共同完成一定任务的一组对象间交换的消息组成	
状态机	它描述了一个对象或一个交互在生命期内响应事件所经历的状态序列	
包	把元素组织成组的机制	
注释事物	是UML模型的解释部分	
依赖	一条可能有方向的虚线	
关联	一条实线，可能有方向	
泛化	一条带有空心箭头的实线	
实现	一条带有空心箭头的虚线	