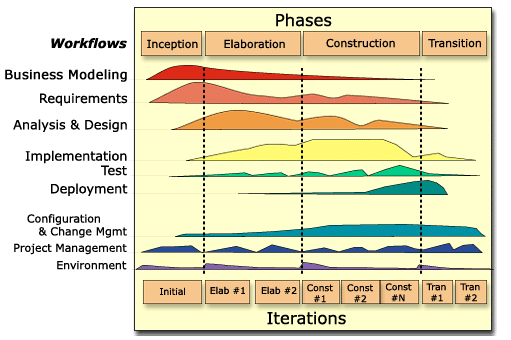
UML(Unified Modeling Language)统一建模语言的概念已经出现了近20年，虽然并不是所有的概念都非常有实践意义，但常见的用例图、类图、序列图和状态图却实实在在非常有效，是项目中不同成员间沟通的有效载体，在此，将主要介绍之前提到的几种最常见的图示方法。

Tip:

RUP软件过程是Ration Unified Processs的简称，是一个不错的软件过程体系，其最佳的开发实践包括：迭代式开发，管理需求、使用基于构件爱的体系架构、可视化软件建模、验证软件质量和控制软件变更。其包括9个核心工作流：业务建模、需求、分析与设计、实现、测试、部署、配置与变更管理、项目管理环境，4个主要阶段：初始阶段、细化阶段、构造阶段和移交阶段。



此外，RUP通过如下的”4+1”视图模型来描述软件系统的体系结构，当然这部分内容的实践意义不是特别大，但可以作为一个很好的方法论参考。



用例是一个活动者使用系统的一项功能时所进行的交互过程的一个文字描述序列，用例分析是一种功能分解的技术，在实际中，就是将业务需求转化为系统的需求的过程。

用例图由参与者、用例、系统边界和箭头组成，有时还涉及用例描述。用例间的关系包括：包含，扩展、泛化等，如下表所示。

|  |  |  |
| --- | --- | --- |
| 关系类型 | 说明 | 表示符号 |
| 关联(Association) | Actor和use case之间的关系，表示参与者与用例之间的通信，箭头指向消息接收方 |  |
| 泛化(Generalization) | Actor之间或use case之间的关系，等价于继承关系，表示更特别的行为 |  |
| 包含(include) | use case之间的关系，用于把一个复杂用例表示的功能分解为较小的步骤 |  |
| 扩展(extend) | use case之间的关系，指用例功能的延伸，相当于给基础用例提供一个附加功能 |  |

附上一个简单的示例，其中inherits等价于泛化。



类图是UML最主要的静态图，通过它可以完成软件系统基础骨架的建立，类的属性包括：可见性、属性名、类型、多重性和初始值等；类之间的关系比较多，如下表所示。此外由于类图太过常见，省略其示例图。

|  |  |  |
| --- | --- | --- |
| 关系类型 | 说明 | 表示符号 |
| 继承关系 | 类之间的继承关系，箭头指向父类 |  |
| 实现关系 | 类对于接口的实现， |  |
| 聚合关系 | 类之间的较弱耦合关系，类似鸟群有0只或多只鸟 |  |
| 组合关系 | 类之间的强耦合关系，类似鸟有两个翅膀 |  |
| 依赖关系 | 两个或多个类之间的依存关系，例如鱼依赖水 |  |
| 关联关系 | 类之间相互影响的关系，如气候和森林相互影响 |  |

补充：OO设计原则

开闭原则(Open/Closed Principle, OCP):一个模块对扩展开放，对修改关闭。

里氏代换原则(Liskov Substitution Principle, LSP):类可以替换父类，并出现在父类出现的任何位置。

依赖倒置原则(Dependency Inversion Principle, DIP):高层模块依赖于低层模块；具体依赖于抽象。

接口分离原则(Interface Segregation Principle, ISP):在设计时采用多个与特定客户有关的接口比采用一个通用的接口要好。

在UML中，序列图（顺序图）和协作图统一称为交互图，序列图较为常见，可以说是动态图中最重要的类型了。序列图的建模元素包括：对象、生命线、控制焦点、消息等。对象可以包含类名和对象名，也可以包含部分；生命线是表示从对象图标向下延伸的一条虚线，表示对象存在的时间；控制焦点是表示时间段的符号，对象在该时间段内，执行相应操作，为生命线上的小矩形；消息类型包括同步消息、异步消息、返回消息、阻止消息和超时消息。一个简单的序列图示例如下所示。



**活动图**很像流程图的规范形式，也非常的常用，其基本概念包括：活动、泳道、分支、分叉和汇合、对象流。其中泳道swimlane表示职责的区域划分；分支会根据不同条件跳转到不同的活动；分叉与汇合用于表述并发行为，常见活动图如下所示。



**状态图**主要用于描述一个对象在其生存期间的动态行为，表现一个对象所经历的状态序列，引起状态转移的事件，以及因状态转移而伴随的动作。常用在状态比较复杂的场景下，通过状态机描述状态之间的控制流，若下图所示。



在状态机中，动作既可以与状态相关也可以和转移相关。如果动作是与状态相关，则对象在进入一个状态时将触发某一动作，而不管是从哪个状态转入这个状态的；如果动作是和转移相关的，则对象在不同的状态之间转移时，将触发对应的动作（理解还不够深入）。对于一个状态机，如果其中所有动作都是与状态相关的，则称这个状态机是Moore机；如果其中所用的动作都是与转移相关的，则成为Mealy机。理论上，这两种方式的表示能力是一致的，一般状态图中都会混用，其包含以下几个基本概念。

**状态**：对象生命周期中的某个条件或状况，在此期间对象将满足某些条件、执行某些动作或等待某些事件。

子状态与组合状态：多个子状态通过or关系或and关系组合成组合状态。

历史状态：是一个伪状态（pseudostate），其目的是记住从组合状态中退出时所处的子状态，当再次进入组合状态时，可以直接进入这个子状态而不是初态。

**转移**：是两个状态之间的一种关系，表示对象将在第一个状态中执行一定的动作，并在某个特定事件发生后进入第二个状态。

事件：对一个在时间和空间上占有一定位置的有意义的事情的详细说明，可以分为4类，调用事件、变化事件、时间事件和信号事件。

动作：可执行的原子计算。

参考资料

1. 王少锋. 面向对象技术UML教程[M]. 北京:清华大学出版社, 2004.