# 7 状态机图与活动图

### 7.1 状态机图

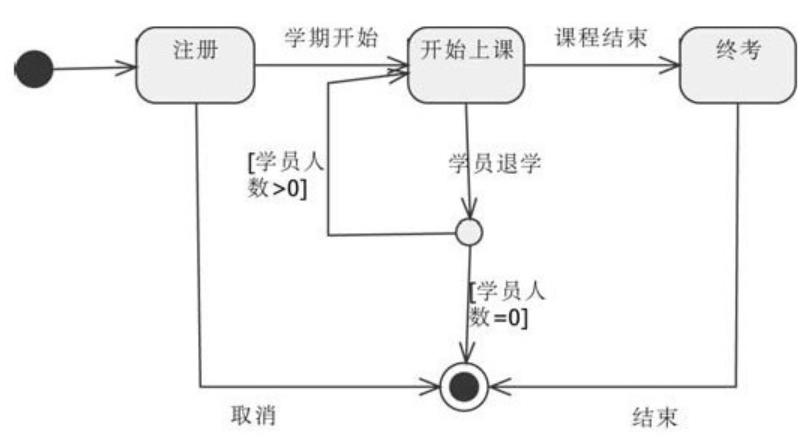
##### 7.1.1 状态图概述

状态图（State Diagram）主要用于描述单个对象在其生命周期内的状态变化，状态变化所需的条件，及状态变化时伴随的动作。

一个对象总是处于某一特定的状态，如洗衣机可以处于浸泡、洗涤、漂洗、脱水或者关机等状态，而状态之间的转换又需要按下按钮等事件

简单状态图示意：

1. 状态 - 圆角矩阵
2. 转移 – 箭头
3. 初态（Initial States）- 实心圆点
4. 终态（Final States）- 原型内嵌原点

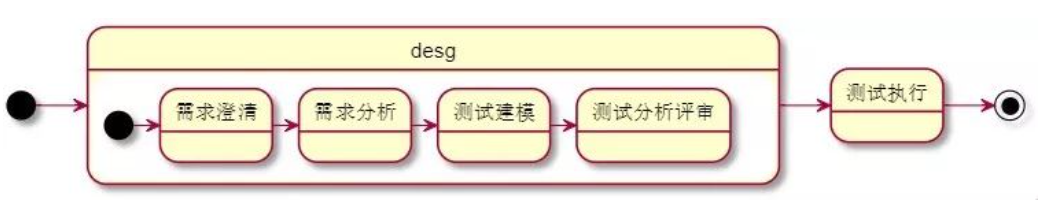


##### 7.1.2 状态图的基本元素

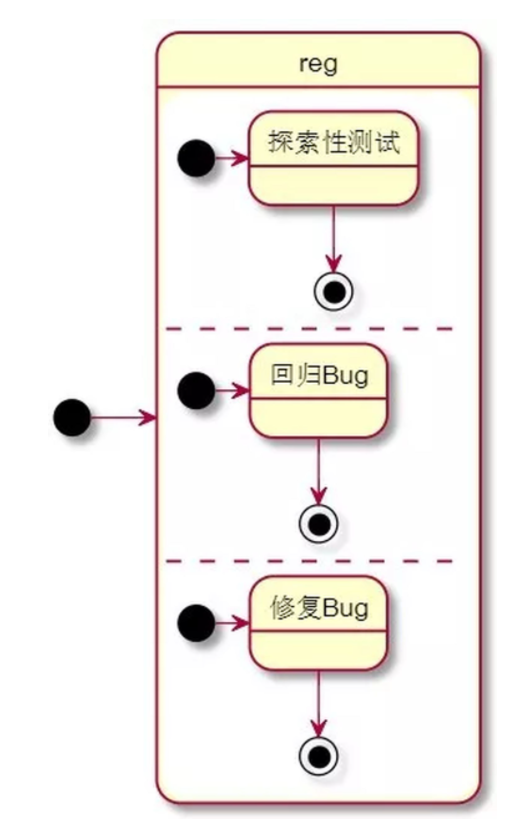
状态图的基本要素可以归类为状态与转换

7.1.2.1 状态

1. 进入/退出动作
   1. 即进入/退出一个状态所执行的动作，体现在状态机图上则是与一个状态相连的状态
2. 组合状态与子状态
   1. 一个状态内部也可能有其他子状态图组成，含有子状态的状态被称为组合状态
   2. 当状态机转移到一个组合状态时，会将控制权交给它的子状态
   3. 为什么需要子状态？有时候入口/出口动作是多余的。例如，某状态的入口/出口分别是打开/关闭某个文件，但如果用户仅仅是想更改该文件的文件名，那么，这里所定义的入口/出口动作就会显得多余。而子状态就可以省去这多余的动作



1. 并发状态
   1. 在一个状态下可能有多个子状态并行。比如一个网页在“工作”状态下，可能既要监听用户输出，还要定时刷新页面



7.1.2.2 转换

转换对应对一个对象的操作，即面向对象思想中的方法。

1. 源状态

转换是指状态机从一个状态到另外一个状态的转换，这种转换要接受触发事件或满足监护条件才能完成。对象在被激发前所处的状态就是转换的源状态。

1. 触发时间

转换的触发事件就是引起转变的事件，是转移的诱因，可以带有参数，参数值可以由监护条件和动作的表达式的转换得到。

1. 监护条件

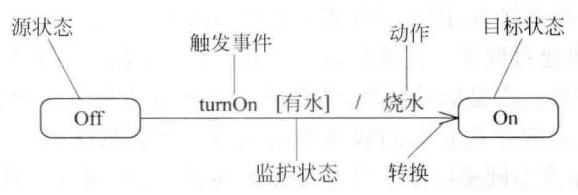
当转移的触发事件发生时，将对监护条件求值。监护条件是一个用方括号括起来的布尔表达式，它位于触发条件的后面。在事件发生时，为转移进行一次监护条件求值。如果值是真，则触发事件使转移有效。如果值是假，则不会引起转移。该布尔表达式可能会引用对象的状态。

1. 动作

当转换发生时，执行它对应的动作。动作是一个可执行的原子操作，即不可中断的，且其执行时间是可忽略不计的。动作包括操作调用、向一个对象发送信号和另外一个对象的创建或撤销，也可以是包含一系列简单动作的动作序列。

1. 目标状态

与源状态对应。在图形中，源状态是指表示转换的箭头的起始位置的状态，目标状态是指表示转换的箭头所指的那个状态



### 7.2活动图

##### 7.2.1 活动图概述

活动图（Activity Diagram），描述了活动的顺序，展现从一个活动到另一个活动的控制流，即活动图是一种流程图。活动图描述了业务实现用例的工作流程。活动图主要由活动和动作构成，也可以支持分支、迭代、并行。

在UML中，活动图主要用于计算性和组织性过程（即工作流）建模。

实际上，在活动图引入到UML之初，是存在巨大争议的，因为活动图实际上描述的是业务流程，是一种过程化的分析方法，这让人担心面向过程的活动图对面向对象的类产生混乱。

但是，UML引入活动图是必须的，活动图的引入解决了业务目标过程化的描述。

活动最终可以细分成多个动作，动作是不可再分的步骤。

##### 7.2.2 活动图元素

7.2.2.1 动作状态

对象的动作状态是活动图中最小单位的构造块，表示原子动作。动作状态有3个特性。

* 原子性：即不能被分解成更小的部分。
* 不可中断性：即一旦开始就必须运行到结束。
* 瞬时性：即动作状态所占用的处理时间通常是极短的，甚至是可以被忽略的动作状态，使用带圆端的方框表示

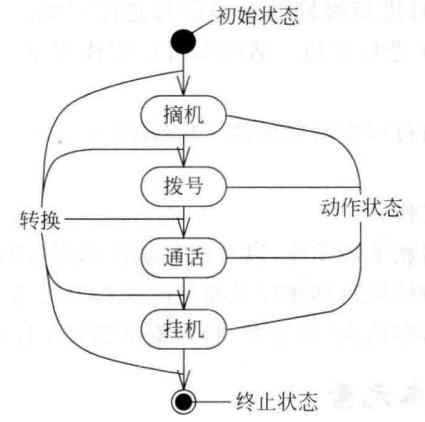


7.2.2.2 活动状态

动作状态表示的是不可分割的原子动作，而活动状态则不同，它表示的是可以分割的动作。其特点是它可以被分解成其他子活动或动作状态，还可以被中断，占有有限的时间。可以把活动状态理解为一个组合，它的控制流由其他活动状态或动作状态组成。

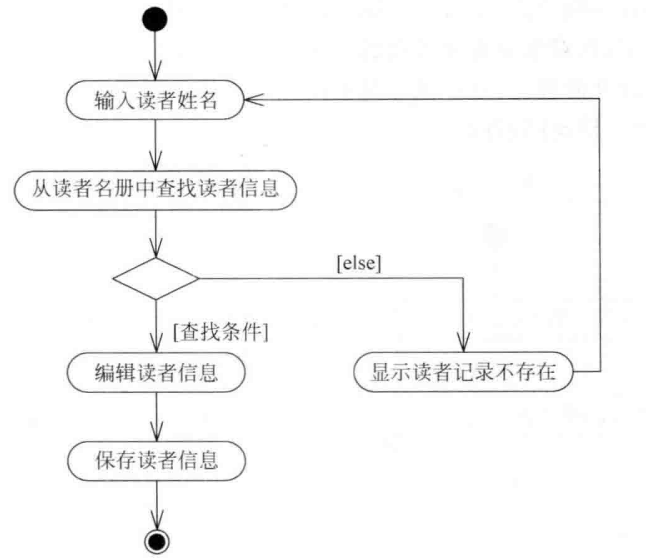
7.2.2.3 转移

转移是两个状态间的一种关系，表示对象将在当前状态中执行动作，并在某个特定事件发生或某个特定的条件满足时进入后继状态。



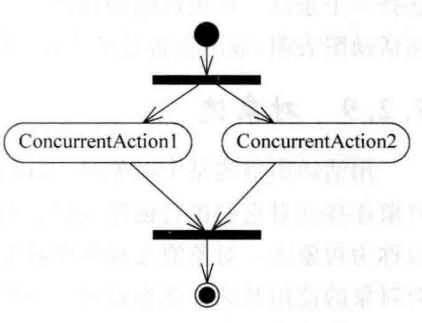
7.2.2.4 分支与合并

分支用于描述基于某个条件的可选择路径。一个分支可以有一个进入转移和两个或多个输出转移。在每条输出转移上都有监护条件表达式保护，当且仅当监护条件表达式为真时，该输出路径才有效。在所有输出转移中，其监护条件不能重叠，而且它们应该覆盖所有的可能性。UML活动图中的分支用菱形表示。



7.2.2.5 分叉与汇合

分叉可以用来描述并发线程。汇合代表两个或多个并发控制流同步发生，当所有的控制流都达到汇合点后，控制才能继续往下进行。

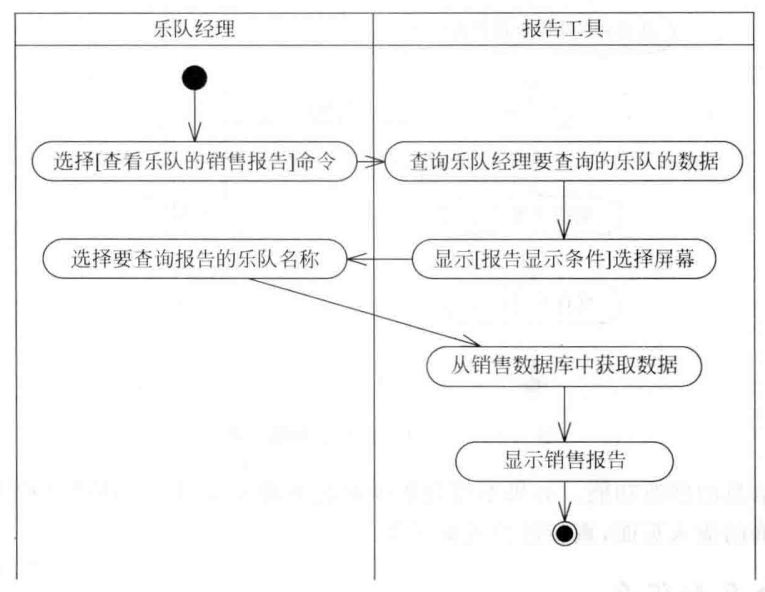


7.2.2.6 泳道

泳道将活动图中的活动分成了若干小组，并把每一组制定给负责这种活动的业务组织，通常为对象。泳道区分了负责活动的对象，明确地表示哪些活动是由哪些对象触发的。

每个活动只能明确地属于一个泳道，并且泳道要负责泳道内的全部活动。

泳道没有顺序，不同泳道中的活动既可以顺序进行，也可以并发进行，动作流和对象流都可以穿越分割线。



7.2.2.7 对象流

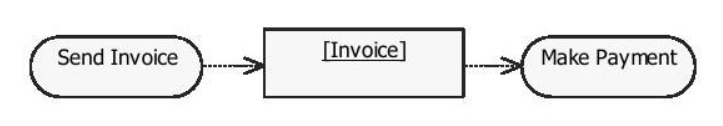
用活动图描述某个对象时，可以把所涉及的对象放置在活动图上，并用一个依赖将这些对象连接到对它们进行创建、撤销和修改的活动转移上。这种包括依赖关系和对象的应用被称为对象流。对象流是动作和对象间的关联。对象流可用于对下列关系建模：动作状态对对象的使用以及动作状态对对象的影响。

对象流中的对象有以下特点。

* 一个对象可以由多个动作操作。
* 一个动作输出的对象可以作为另一个动作输入的对象。
* 在活动图中，同一个对象可以多次出现，它的每一次出现表明该对象正处于对象生存期的不同时间点。

对象流用带有箭头的虚线表示。如果箭头是从动作状态出发指向对象，则表示动作对对象施加了一定的影响。施加的影响包括创建、修改和撤销等。如果箭头从对象指向动作状态，则表示该动作使用对象流所指向的对象。状态图中的对象用矩形表示，矩形内是该对象的名称，名称下的方括号表明对象此时的状态。

一个支付账单的对象流：



### 7.3 状态图与活动图的比较

状态图和活动图都是用于对系统的动态行为建模。状态机是展示状态与状态转换的图，通常一个状态机依附于一个类，并且描述这个类实例对接收到的事物的反应。状态机有两种可视化方式，分别是状态图和活动图。如果强调对象的潜在状态和这些状态的间的转换，一般使用状态图；如果强调从活动到活动的控制流，一般使用活动图。活动图用于描述一个过程或操作的工作步骤，从这方面讲，它可以算是状态的一种扩展方式，状态图描述一个对象的状态以及状态的改变，而活动图除了描述对象状态外，更突出它的活动。

### 7.4 提问

1. 在绘画状态图时，使用什么来区分若干个并行状态？

虚线

1. 下面哪一个不是活动图中的基本元素（D）
2. 状态、分支
3. 转移、汇合
4. 泳道、转移
5. 用例、状态
6. 活动图使用什么来描述并发线程？

分叉与汇合

### 7.5 参考资料

1. 杨弘平，《UML2基础、建模与设计教程》
2. 知乎，《软件工程专题：UML状态图》，<https://zhuanlan.zhihu.com/p/78051066>
3. 知乎，《软件工程专题：UML活动图》，<https://zhuanlan.zhihu.com/p/77840996>