# 7.1状态图

**状态图，用于UML中建立动态模型，主要描述系统随时间变化的行为，这些行为是用从静态视图中抽取的系统的瞬间值的变化来描述的。在对象的生命期建模中，状态图显示了一个状态机，展示的是单个对象内从状态到状态的控制流。状态图通过对类的对象的生存周期建立模型来描述对象随时间变化的动态行为。**

## 7.1.1 状态图概述[1][5]

状态图（又称状态机图）由状态、转换、事件、活动和动作五部分组成，是展示状态与状态转换的图。通常一个状态图依附于一个类，并且描述一个类的实例。

1. 状态：

状态指的是对象在其生命周期中的一种状况，处于某个特定状态中的对象必然会满足某些条件、执行某些动作或者是等待某些事件。一个状态的生命周期是一个有限的时间阶段。

1. 转换：

转换指的是两个不同状态之间的一种关系，表明对象在第一个状态中执行一定的动作，并且在满足某个特定条件下由某个事件触发进入第二个状态。

1. 事件：

事件指的是发生在时间和空间上的对状态机来讲有意义的那些事情。事件通常会引起状态的变迁，促使状态机从一种状态切换到另一种状态，如信号、对象额度创建和销毁等。

1. 活动：

活动指的是状态机中进行的非原子操作。

1. 动作：

动作指的是状态机中可以执行的哪些原子操作。所谓原子操作，指的是他们在运行的过程中不能被其他消息中断，必须一直执行下去，以至最终导致状态的变更或者返回一个值。

举例：图书馆借阅中的图书状态变化。



一个图书对象从它的起始点开始，首先是“新书”状态（new book），然后是“可以借阅”（available）的状态，如果有读者将书借走，则该书的状态为“已借出”状态（borrowed），如果图书被归还图书馆，图书的状态又变为“可以借阅”状态。图书馆如果放弃该图书对象的收藏，则图书对象处于“删除”状态（delete），最后到达“终止”状态。

## 7.1.2 状态图的基本元素[5]

状态之间的过渡事件（event），对应对象的操作。事件有可能在特定的条件下发生，在UML中这样的条件称为警戒条件（guard condition）。发生事件时的处理称为动作（action）。从一个状态到另一个状态之间的连线称为转移（transitions）。状态图通常包含如下内容。

**⑴ 状态**

状态定义对象在其生命周期中的条件或状况。

**⑵ 转换**

对象的状态之间的转移叫转换，它包括事件和动作。

## 7.1.3 状态[1][5]

一个对象的状态可能包含子状态或其它一些更加详细的内容。具体的有以下五个部分组成：名称、 进入/退出动作、 内部转换、子状态和延迟事件。

⑴名称（name）

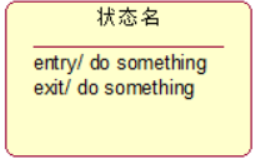
⑵进入/退出动作（entry/exit action）

⑶内部转换（Internal Transition）

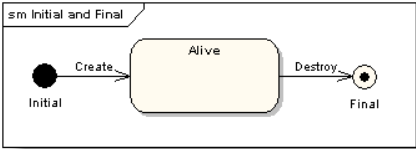
⑷子状态（sub state）

⑸延迟事件（Deferred Event）

状态用圆角矩形表示



初态和终态

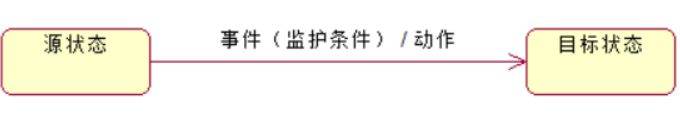


## 7.1.4 转换[1][2][5]

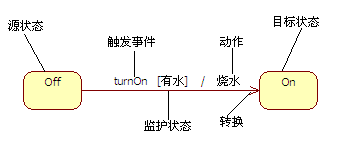
UML状态图中转换是两个状态之间的一种关系，表示对象将在源状态（Source State）或当前状态中执行一定的动作，并在某个特定事件发生而且某个特定的警界条件满足时进入目标状态。

转换是由如下5部分组成：

1. 源状态：指的是激活转换之间对象处于的状态。如果一个一个状态处于源状态，当它接收到转换的触发事件或满足监护条件时，就激活了一个离开的转换
2. 触发事件：指的是引起源状态转换的事件。事件不是持续发生的，它只发生在时间的一点上，对象接收到事件，导致源状态发生变化，激活转换并使监护条件得到满足
3. 监护条件：是一个布尔表达式。当接收到触发事件要触发转换时，要对该表达式求值。如果表达式为真，则激活转换：如果表达式为假，则不激活转换，所接收到的触发事件丢失
4. 动作：是一个可执行的原子计算
5. 目标状态：指的是转换完成后对象所处的状态



**举例：**



在用电磁炉烧开水的过程中，水的状态由源状态“Off”（不沸腾）转换为目标状态的“On”（沸腾）时，水壶中“有水”就是其监护条件，开启电源开关“turnOn”是其触发事件，进行“烧水”是状态转换的动作。

## 7.1.5 状态图的建模技术及应用[2][5]

状态图表示某个类所处的不同状态和该类的状态转换信息。虽然每个类都有状态，但在系统活动期间仅对具有三个或更多潜在状态的类才画一个状态图，进行状态图描述。用状态图对一个对象按事件排序的方法建模，状态图是强调从状态到状态的控制流的状态机的简单表示。

使用状态图的最常见的是对**反应型对象**、尤其是对类、用例或整个系统的实例的行为建模。反应型对象是指这个对象可能处于的稳定状态、从一个状态到另一个状态之间的转换所需的触发事件，以及每个状态改变时发生的动作。反应型对象具有如下的**特点**：

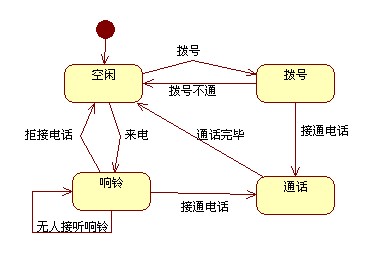
⑴响应外部事件，即来自对象语境外的事件；

⑵具有清晰的生命期，可以被建模为状态、迁徙和事件的演化；

⑶当前行为和过去行为存在着依赖关系；

⑷在对事件做出反应后，它又变回空闲状态，等待下一个事件。

举例：



首先，确定主要的状态，在这里电话开机时，处于空闲状态，当用户拨号呼叫某人时，话机进入拨号状态。如果呼叫成功则电话接通，电话处于通话状态，如果呼叫不成功，拨号失败，这时话机重新回到空闲状态。话机在空闲状态被呼叫，进入响铃状态。如果用户摘机接听电话，话机处于通话状态。完成通话挂机后话机回到空闲状态。如果用户没有摘机则话机处于继续响铃状态。如果用户拒绝来电，话机回到空闲状态。因此，拨打电话的过程可以总结出4个状态，即空闲、拨号、通话和响铃。

## 7.2 活动图

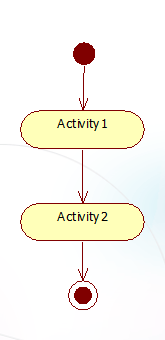
活动图，用于UML中建立动态模型，主要描述系统随时间变化的行为，这些行为是用从静态视图中抽取的系统的瞬间值的变化来描述的。在对象的生命期建模中，活动图是一种特殊形式的状态机，用于对计算机流程和工作流程建模。活动图从本质上说是一个流程图，展现跨过不同的对象从活动到活动的控制流。与传统的流程图不同的是，活动图能够展示并发和控制分支。

## 7.2.1 活动图概述[3][5]

在用例模型中，可以利用**文本**来描述用例的业务流程，但如果业务流程较为复杂的话，则可能会难以阅读和理解，这时需要用更加容易理解的方式（**图形**）来描述业务过程的工作流，在UML中将这类描述活动流程的图形称为活动图（Activity Diagram）。

活动图被设计用于简化描述一个过程或者操作的工作步骤。活动用圆角矩形表示，接近椭圆。一个活动中的处理一旦完成，则自动引起下一个活动的发生。箭头表示从一个活动转移到下一个活动。和状态图类似，活动图中的起点用一个实心圆表示，终点用一个实心圆外加一个圆圈表示。在一个活动图中，只有一个起始状态，可以有零个或多个终止状态。

举例：



## 7.2.2 活动图的基本元素[3][5]

活动图中的基本要素包括状态、转移、分支、分叉和汇合、泳道、对象流等。

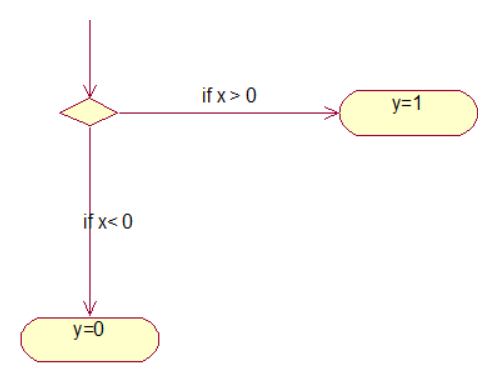
状态：



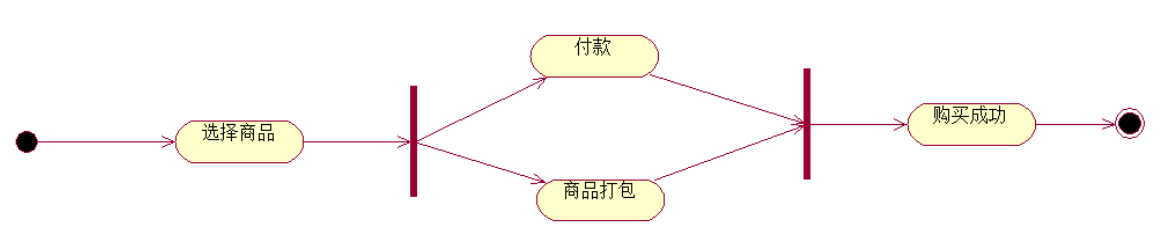
转移：



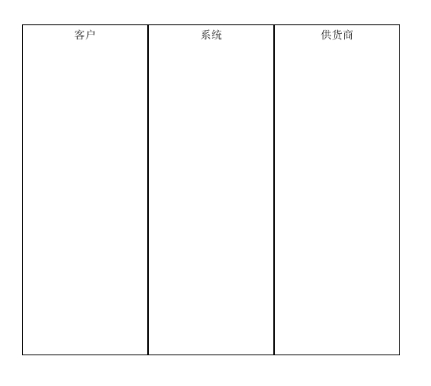
分支：



分叉与汇合：



泳道：



对象流：



## 7.2.3 动作状态[3][5]

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。动作状态有三个特性：

⑴原子性：即不能被分解成更小的部分；

⑵不可中断性：即一旦开始就必须运行到结束；

⑶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。如下图：



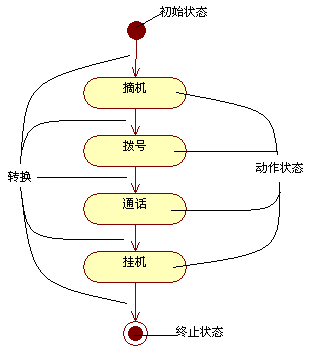
## 7.2.4 活动状态[4][5]

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。特点是：它可以被分解成其他子活动或动作状态，它能够被中断，占有有限的时间。活动状态可以理解为一个组合，它的控制流由其他活动状态或动作状态组成。

在UML中，动作状态和活动状态的图标没有什么区别，都是圆端的方框。只是活动状态可以有附加的部分，如可以指定入口动作、出口动作、动作状态以及内嵌状态机。

## 7.2.5 转移[3][5]

转移也称转换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。在UML中用一条简单的带箭头的直线表示一个转移。箭头上可以带有监护条件表达式。

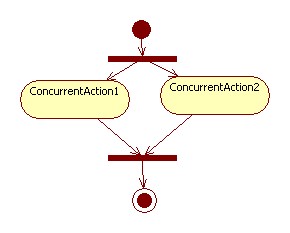


## 7.2.6 分支与汇合[3][5]

对象在运行时可能会存在两个或多个并发运行的控制流，为了对并发的控制流建模，UML中引入了分叉与汇合的概念。分叉用于将动作流分为两个或多个并发运行的分支，而汇合则用于同步这些并发分支，以达到共同完成一项事务的目的。

分叉和汇合在图形上都使用同步条来表示，同步条通常用一条粗的水平线表示。

举例：

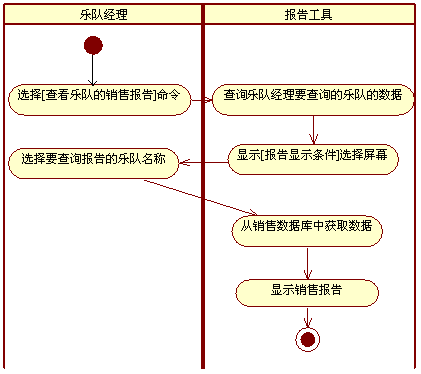


## 7.2.7 泳道[3][5]

泳道将活动图中的**活动划分为若干组**，并把每一组指定给负责这组活动的业务组织，即**对象**。每一组表示一个特定的类、人或部门，他们负责完成组内的活动。在活动图中，泳道**区分了负责活动的对象**，它明确地表示了哪些活动是由哪些对象进行的。在包含泳道的活动图中，每个活动只能明确地属于一个泳道。泳道用于表示实际执行活动的对象。

泳道是用垂直实线绘出，垂直线分隔的区域就是泳道。在泳道的上方可以给出泳道的名字或对象的名字，该对象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，动作流和对象流允许穿越分隔线。

举例：



## 7.2.8 对象流[3][5]

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：

⑴一个对象可以由多个动作操作。

⑵一个动作输出的对象可以作为另一个动作输入的对象。

⑶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。

举例：



## 7.2.9 活动图的建模技术以及应用[5]

活动图用于对系统的动态行为建模，在对一个系统建模时，通常有两种使用活动图的方式：

1. 为工作流建模

对工作流建模**强调与系统进行交互的对象所观察到的活动**。工作流一般处于系统的边界，用于可视化、详述、构造和文档化开发系统所涉及的业务流程。用于业务建模的时候，每一条泳道表示一个职责单位，该图能够有效地体现出所有职责单位之间的工作职责，业务范围及之间的交互关系、信息流程。

1. 为对象的操作建模

在这种情况下活动图**本质上就是流程图**，它描述系统的活动、判定点和分支等部分。因此，在UML中，可以把活动图作为流程图来使用，用于对系统的操作建模。每一个对象占据一个泳道，而活动则是该对象的成员方法。

## 7.2.10 活动图与状态图的比较[4][5]

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**。活动图被设计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，更突出了它的活动**。

## 7.4 参考资料

[1]https://www.cnblogs.com/ywqu/archive/2009/12/17/1626043.html

[2]<https://blog.csdn.net/cold___play/article/details/102770382>

[3] https://blog.csdn.net/Luyanc/article/details/54633436

[4]《UML用户指南》 人民邮电出版社 Grady Booch, James Rumbaugh, Ivar Jacobson著 邵维忠 麻志毅 马浩海 刘辉 译 2013年1月第1版

[5]《UML2基础、建模与设计教程》 清华大学出版社 杨弘平等 2015年10月第1版

## 问题：

下列关于状态图的说法中，正确的是（C ）。

A. 状态图是UML中对系统的静态方面进行建模的五种图之一。

B. 状态图是活动图的一个特例，状态图中的多数状态是活动状态

C. 活动图和状态图是对一个对象的生命周期进行建模，描述对象随时间变化的行为。

D. 状态图强调对有几个对象参与的活动过程建模，而活动图更强调对单个反应型对象建模

对反应型对象建模一般使用（A）。图

A. 状态图 B. 顺序图

C. 活动图 D. 类图

能不能简单说一下状态图和活动图的异同？

答：状态图和活动图都是用于对系统的动态行为建模。状态图强调对象的潜在状态和这些状态的间的转换；活动图强调从活动到活动的控制流。活动图算是状态图的一种拓展方式。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，更突出了它的活动。