

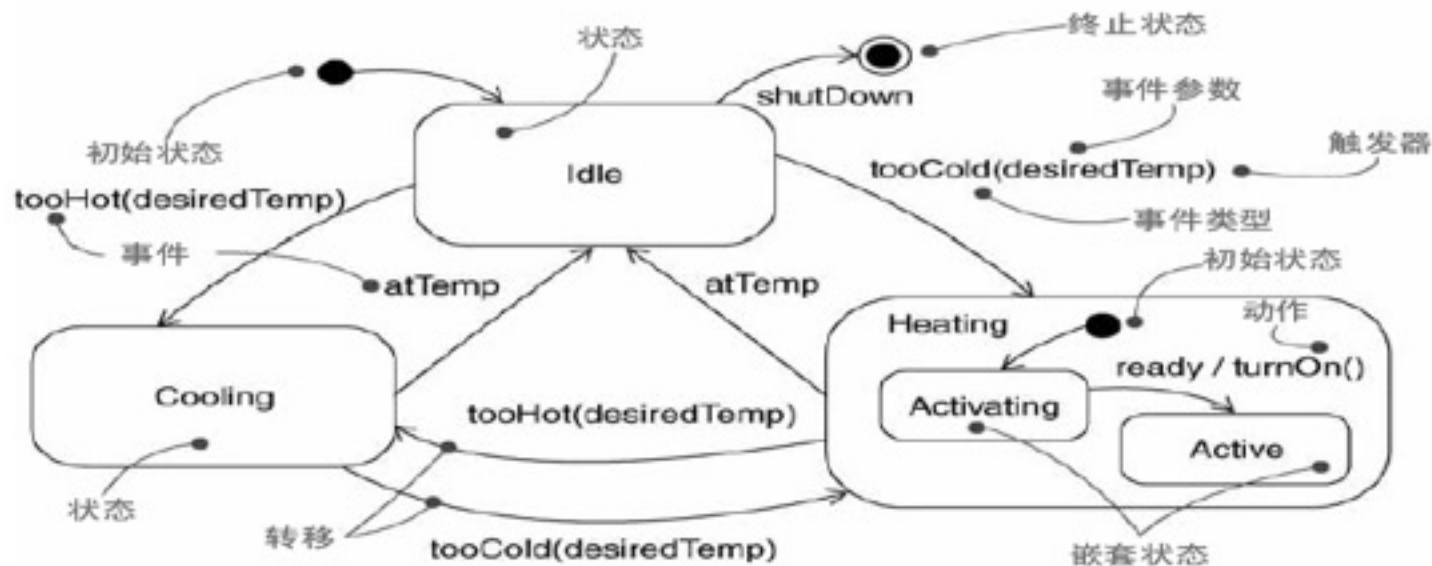
状态图

- 定义
- 内容
- 一般用法

6.3.4 系统行为（生存周期）的建模工具-状态图

6.3.4.1 定义

状态图是显示一个状态机的图，其中强调了从一个状态到另一状态的控制流。



一个状态机是一种行为，规约了一个对象在其生存期间因响应事件并作出响应而经历的状态。



状态图

- 定义
- 内容
- 一般用法



6.3.4.2 状态图的内容

通常包含： ❶ 简单状态和组合状态；

❷ 事件

❸ 转换

像其它图形一样，可以包含注解和约束。

从使用的角度来看，一个状态图可以包含一个状态机中任意的、所有的特征（**features**），即状态图基本上是一个状态机中那些元素(分支、结合、动作状态、活动状态、对象、初始状态、最终状态、历史状态等)的一个投影。

状态图所包含的内容，确定了一个特定的抽象层，该抽象层决定了以状态图所表达的模型之形态。



北京大学

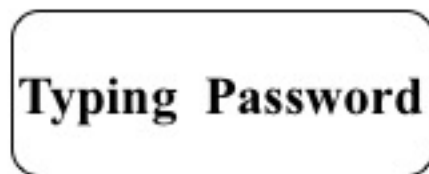
状态图

- 定义
- 内容
- 一般用法

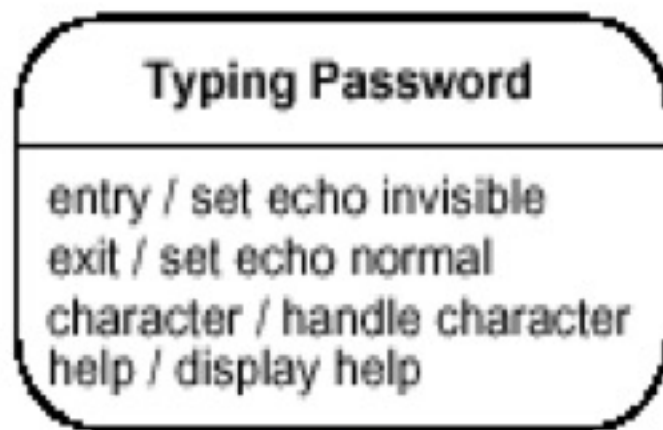
(1) 状态

①**定义**: 一个状态是类目的一个实例（以后简称对象）在其生存期间的一种条件(**condition**)或情况 (**situation**), 该期间该对象满足这一条件, 执行某一活动或等待某一消息。

②表示:



(A)



(B)

一个状态表达了一个对象所处的特定阶段, 所具有的对外呈现 (外征) 以及所能提供的服务。



状态图

- 定义
- 内容
- 一般用法

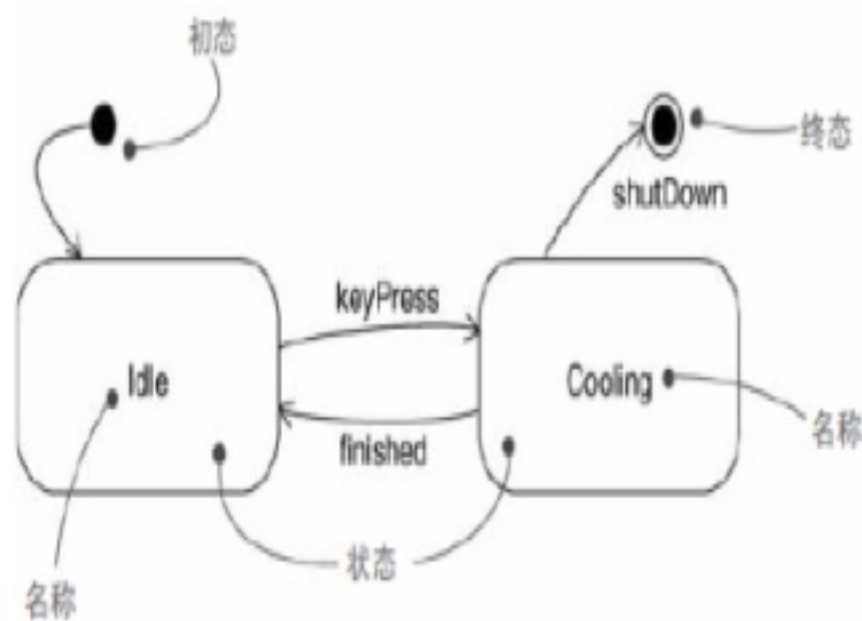
③ 状态分类:

UML把状态分为**初态**、**终态**和**正常状态**:

初态:表达状态机默认的开始位置, 用实心圆来表示;

终态:表达状态机的执行已经完成, 用内含一个实心圆的圆来表示;

正常状态:既不是初态又不是终态的状态,称为正常状态.



初态和终态都是伪状态, 即只有名字。从初态转移到正常状态可以给出一些特征, 例如监护条件和动作。



状态图

- 定义
- 内容
- 一般用法

④状态的规约：

①名字

是一个标识状态的文本串，作为状态名。
也可以有匿名状态—没有给出状态名。

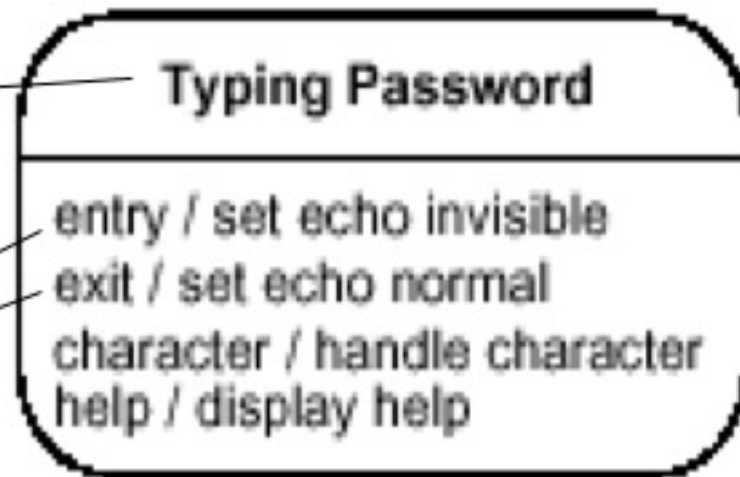
②进入/退出之效应(effect)

是进入或退出该状态时所执行的动作。

为了表达进入/退出之效应，UML给出2个专用的动作标号：

- **entry** 该标号标识在进入该状态时所执行的、由相应动作表达式所规定的动作，简称进入动作。
- **exit** 该标号标识在退出该状态时所执行的、由相应动作表达式所规定的动作，简称退出动作。

一般情况下，进入 / 退出之效应不能有参数或监护条件，但位于类状态机顶层的进入效应可以具有参数，以表示在创建一个对象状态机时所接受的参数。

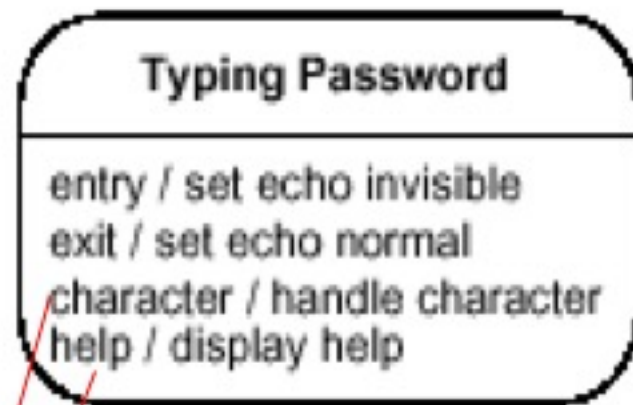


状态图

- 定义
- 内容
- 一般用法

③ 状态内部转移

是指没有导致该状态改变的内部转移。一般情况下，在此给出对象在这个状态中所要执行的内部动作或活动列表。其中表达动作的一般格式为：**动作标号**’/’ **动作表达式**



状态内部转移

动作标号标识了在该环境下所要调用的动作，而该动作是通过’/’之后的**动作表达式**所规约，其中可以使用对象范围内的任何属性和链。若该表达式为空，则可省略斜线分隔符。

为了表达状态内转换中的动作或活动，UML给出了一个专用的动作标号：**do**，该标号标识正在进行由其相应动作表达式所规定的活动，并且只要对象在一个状态中没有完成由该动作表达式所指定的活动，就一直执行之；当动作表达式指定的活动完成时，可能会产生一个完成事件。

注：动作标号“entry”、“exit”和“do”均不能作为事件名。



北京大学

状态图

- 定义
- 内容
- 一般用法

在以上的叙述中，使用了两个词：

动作 (action) 和 **活动** (activity)

一个活动是指状态机中一种可中断的计算，中断处理后仍可继续；

而一个动作是指不可中断的原子计算，它可导致状态的改变或导致一个值的返回。

可见，一个活动往往是有多个动作组成的。



状态图

- 定义
- 内容
- 一般用法

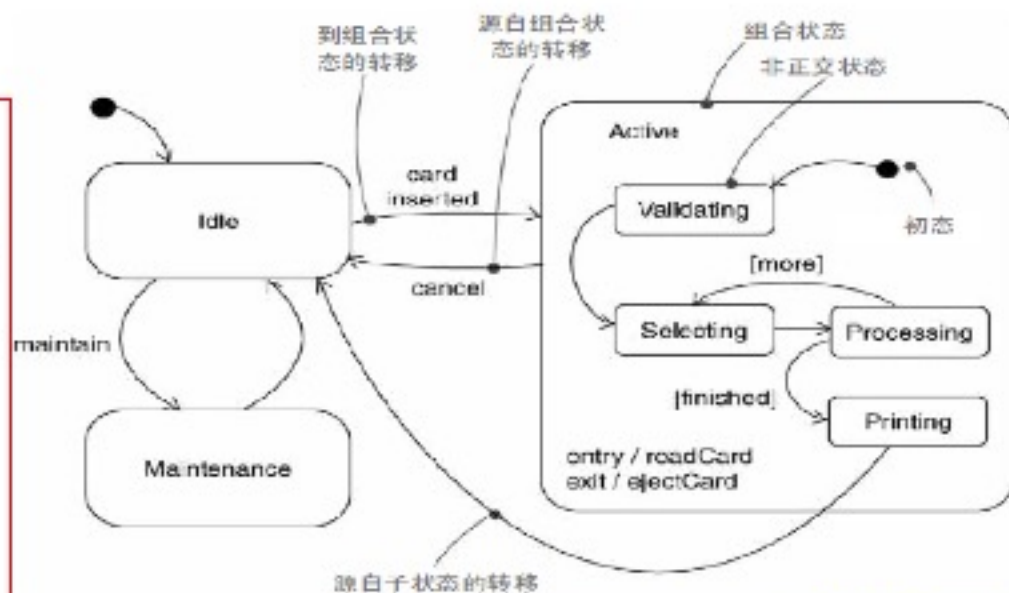
④子状态

如果在一个状态机中引入另一个状态机，那么被引入的状态机称为子状态机。**子状态是被嵌套在另一状态中的状态。**相对地，把没有子状态的状态称为**简单状态**；而把含子状态的状态称为**组合状态**。

子状态机分类：顺序子状态机（非正交）和并发子状态机（正交）

◆非正交子状态机

注意:右边是一个组合状态，包含一些子状态，表达了一个非正交状态机。一个非正交状态机最多有一个子初态和一个子终态。



状态图

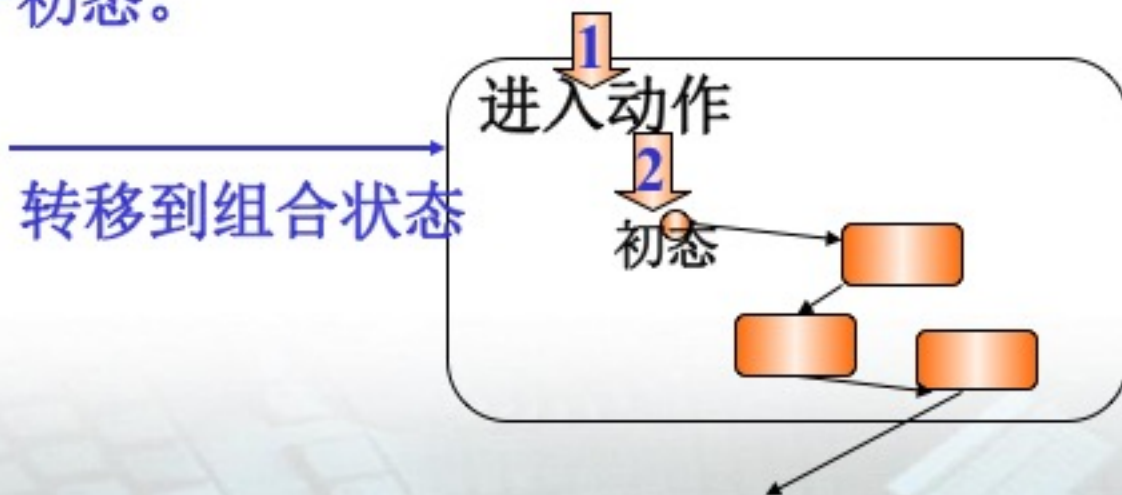
- 定义
- 内容
- 一般用法

关于转入问题:

从组合状态之外的一个源状态（如上图中的“Idle”），

- 可以转移到该组合状态，作为其目标状态；
- 可以转移到组合状态中一个子状态，作为其目标状态。

在第一种情况里，这个被嵌套的子状态机**一定**有一个初态，以便在进入该组合状态并执行其进入动作后，将控制传送给这一初态。



其中 ↓ 表示控制流

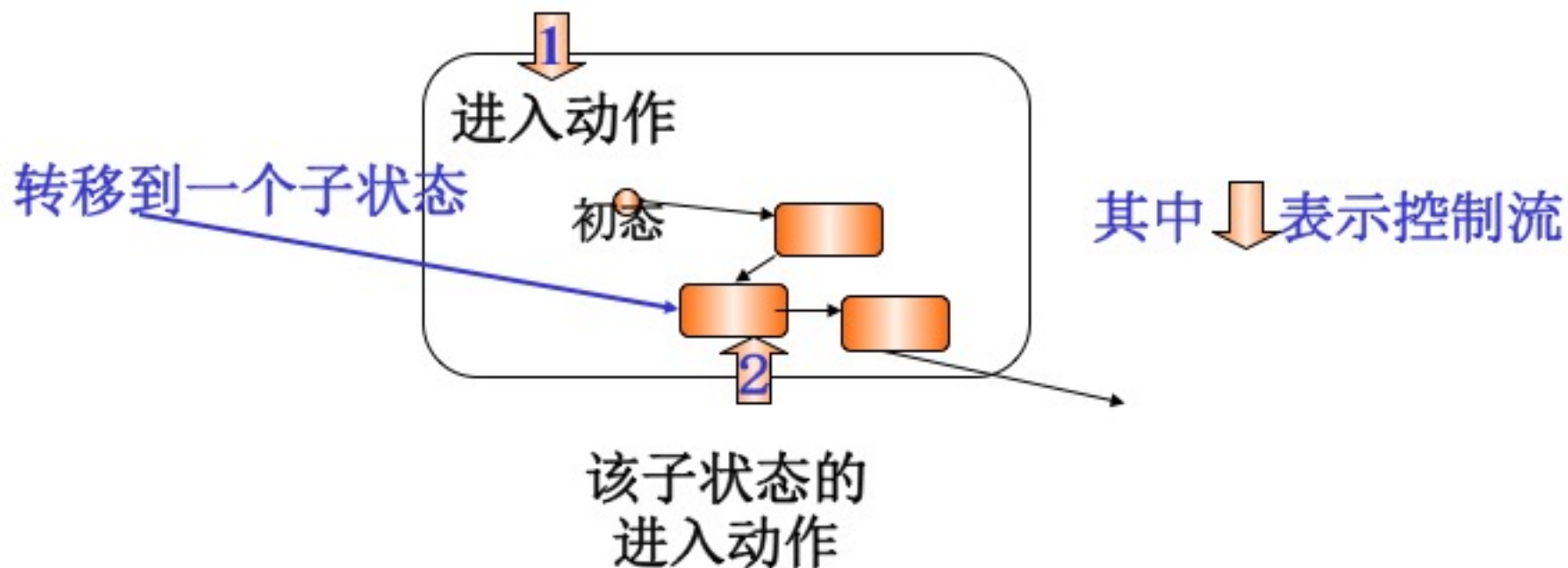


北京大学

状态图

- 定义
- 内容
- 一般用法

在第二种情况里，在执行完该组合状态的进入动作（如有的话）和该子状态的进入动作后，将控制传送给这一子状态。



状态图

- 定义
- 内容
- 一般用法



关于离开问题:

- ① 离开一个组合状态的转移，其源
 - 可以是该组合状态，
 - 可以是该组合状态中的一个子状态。

无论哪种情况，控制都是：

* 首先离开被嵌套的状态，即执行被嵌套状态的退出动作（如有的话）

* 然后离开该组合状态，即执行该组合状态的退出动作（如有的话）。

由此可见，如果一个转移，其源是一个组合状态，那么该转移的本质是终止被嵌套状态机的活动。

② 当控制到达该组合状态的子终态时，就触发一个活动完成的转移。



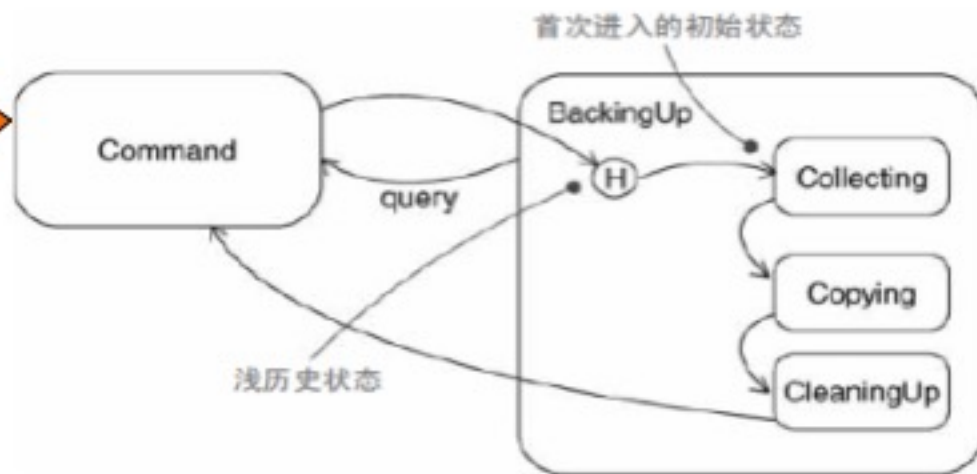
状态图

- 定义
- 内容
- 一般用法

关于离开组合状态之前执行活动相关子状态标识问题

如果需要知道在转移出该组合状态之前所执行的那个活动所在的子状态，UML引入了一个概念-**浅历史状态**，用于指明这样的子状态，并用H来表示之。如下图所示。

右边的图表示:在对一个通过网络进行无人值守的计算机备份的代理的行为建模时,如果它曾被中断(如一个操作员的查询中断),希望它用H(浅历史状态)记住是在该过程的什么地方被中断的.



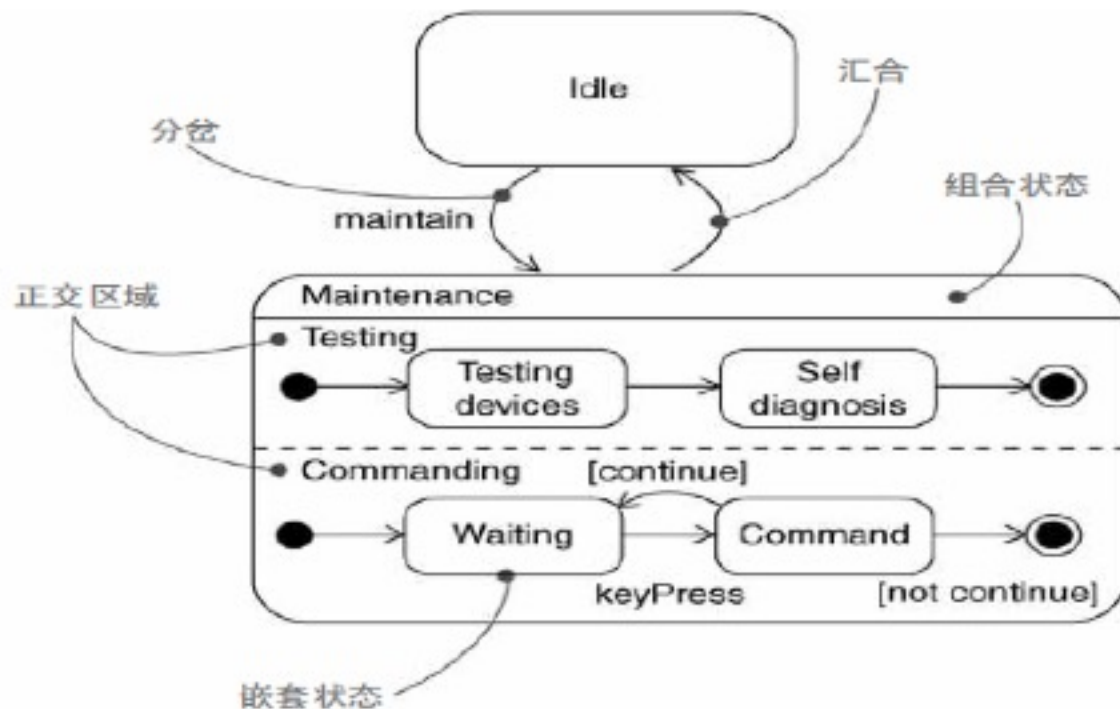
相对于H所表示的浅历史，可用H*来表示深历史，即在任何深度上表示最深的、被嵌套的状态。

状态图

- 定义
- 内容
- 一般用法

◆正交子状态机

正交子状态机是组合状态中一些并发执行的子状态。例如：



其中，使用虚线段形成了两个正交区域（根据需要，可形成多个正交区域），分别以“Testing”和“Commanding”标记之，并且每个区域均有自己的初态和终态。



状态图

- 定义
- 内容
- 一般用法

关于转入问题:分岔

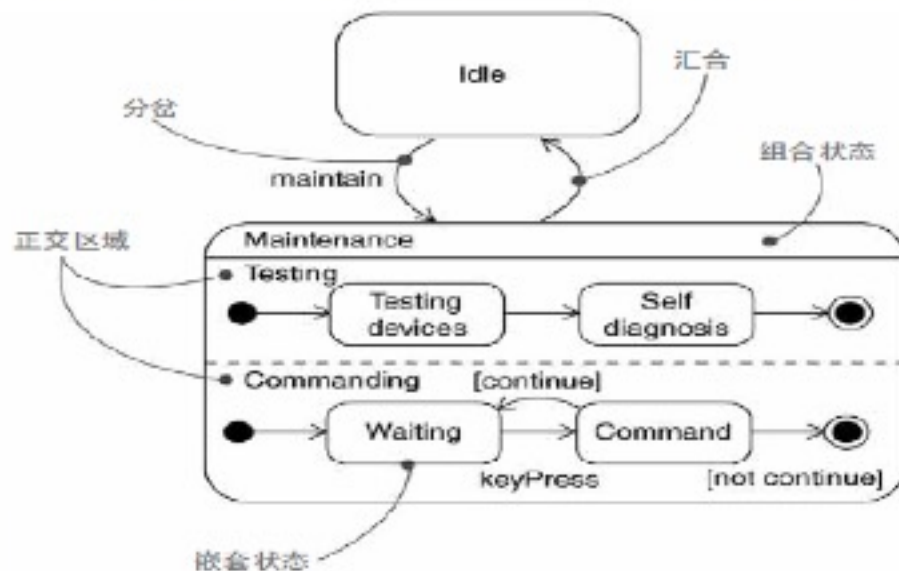
当一个转移到达具有多个正交区域的组合状态时，控制就被分成多个并发流-分岔。正交区域的执行是并行的，相当于存在两个被嵌套的状态机。

关于离开问题:汇合

--如果一个正交区域先于另一个到达它的终态时，那么该区域的控制将在该终态等待，直到另一个区域的控制达到自己的终态时，两个区域的控制才汇合成一个控制流-汇合。

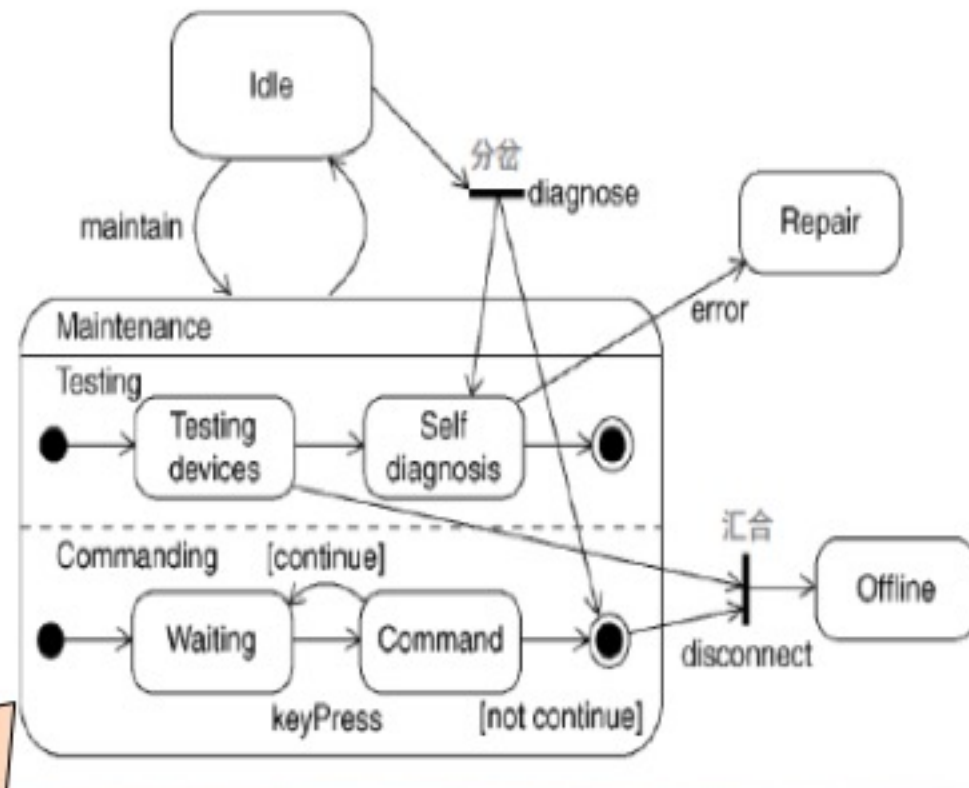
--当一个转移离开这样的组合状态时，控制就汇成一个控制流。

--如果所有正交区域均达到它们的终态，或存在一个指示离开组合状态的转移，那么就汇成一个流。



状态图

- 定义
- 内容
- 一般用法



上图显示了显式的分岔和汇合，以及隐式的分岔和汇合。

（1）进入组合状态Maintenance的转移maintain是一个到所有正交区域的默认初始状态的隐含分岔；（2）从Idle状态到两个嵌套状态，即self diagnose和区域Commanding的终止状态；（3）如果在Self diagnose状态活动时发生了错误，就会激活到Repair的隐式的汇合转移：无论是Self diagnose状态，还是Commanding区域内的任何活动状态都会退出；（4）一个到状态Offline的显式的汇合转移：只有当Testing devices状态和Commanding区域的终止状态是活动的，而且disconnect事件发生，才会激活这个转移；如果两个状态都不活动，则该事件无效。

状态图

- 定义
- 内容
- 一般用法

⑤ 被延迟事件

被延迟事件是那些在一个状态中不予处理的事件列表。往往需要一个队列机制，对这样的事件予以推迟并予排队，以便在该对象的另一状态中予以处理。

小结：状态的规约，包括：

- ◆命名
- ◆进入/退出之效应
- ◆状态内部转移
- ◆子状态与组合状态
- ◆被延迟事件



状态图

- 定义
- 内容
- 一般用法

下面介绍状态图中第二个基本元素-事件。

(2) 事件

一个事件是对一个有意义的发生的规约，该发生有其自己的时空。在状态机的语境下，一个事件是一个激励（stimulus），可引发状态的转换。

事件的种类：

内部事件：是在系统内对象之间传送的事件。例如，溢出异常。

外部事件：是在系统和它的参与者之间传送的事件。例如
按下一个按钮，一个来自传感器的中断。



清华大学

状态图

- 定义
- 内容
- 一般用法



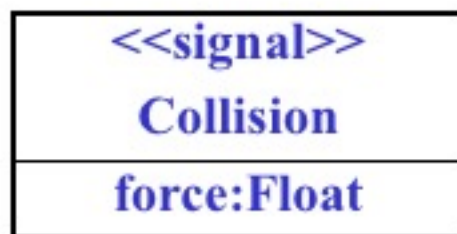
在UML中，可模型化4种事件：

①信号 (signal)

信号是消息的一个类目，是消息类型。

信号有属性和操作，信号之间可以有泛化。

在UML中，可将信号模型化为具有名字<<signal>>衍型类，例如：



可以把衍型看成元类型(一种定义其他类型的类型),因为每一个衍型将创建一个相当于UML元模型中新类的等价物.

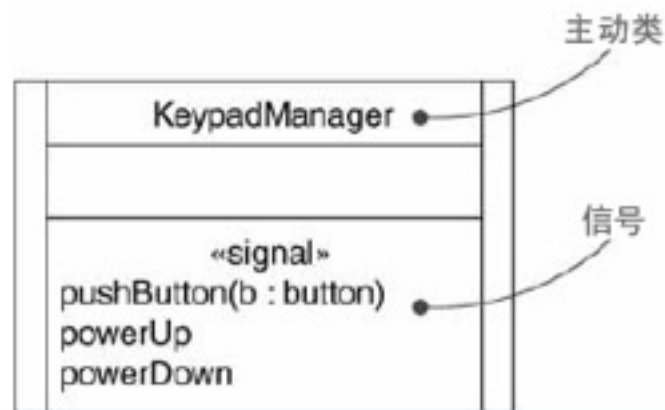
并可用依赖衍型<<send>>来表示一个操作发送了一个特定的信号。



状态图

- 定义
- 内容
- 一般用法

在**UML**中，可以通过在类的附加栏中对信号命名来为对象可能接受的、有名的信号进行模型化。



状态图

- 定义
- 内容
- 一般用法

② 调用 (call)

一个调用事件表示对象接受到一个操作调用的请求。

几点说明：

- 可以使用在类的定义中的操作定义来规约调用事件。
 - 该事件或触发状态机中的一个状态转换，或调用目标对象的一个方法。
- ——“信号”是一种异步事件，而“调用”一般是同步事件，但可以把“调用”规约为异步调用。



在UML中，将一个对象可能接受的调用事件模型化为该对象类的一个操作。



状态图

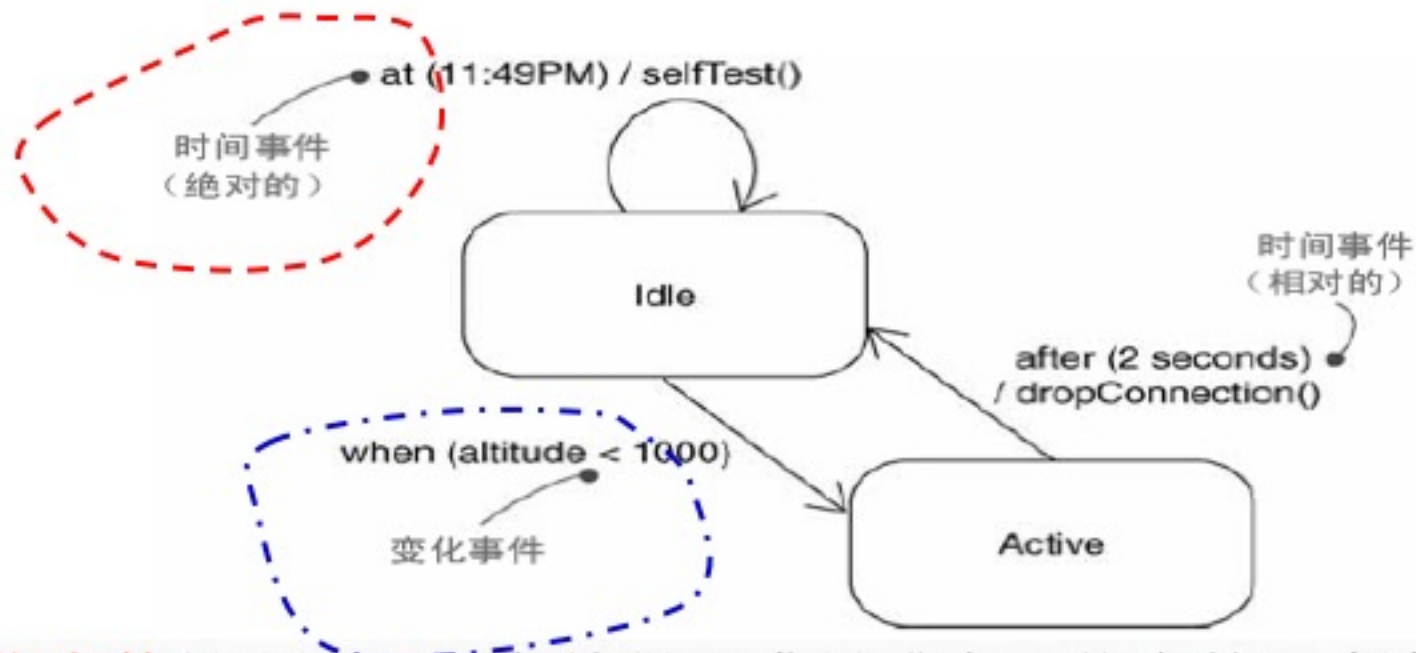
- 定义
- 内容
- 一般用法



③ 时间事件和变化事件

时间事件是表示推移一段时间的事件。

时间表达式可以是复杂的，也可以是简单的，例如：
after 2 seconds, at(1 jan 2007, 12.00)).



变化事件是表示一个条件得到满足或表示状态的一个变化。



状态图

- 定义
- 内容
- 一般用法

④ 发送事件和接受事件

发送事件是表示类的一个实例发送一个调用事件或信号事件。
接受事件是表示类的一个实例接受一个调用事件或信号事件。

- 如果是一个同步调用事件，那么发送者和接受者都处在该操作执行期间的一个汇聚点上，即发送者的控制流一直被挂起，直到该操作执行完成；
- 如果是一个信号事件，那么发送者和接受者并不汇合，即发送者发送出信号后并不等待接受者的响应。
- 在以上两种情况下，事件可能被丢失（如果没有定义对该事件的响应的話），事件的丢失可能触发接受者的状态机（如果有的話）或引起一个常规的方法调用。



状态图

- 定义
- 内容
- 一般用法

(3) 状态转换

① 定义: 一个状态转换是两个状态间的一种关系, 指明: 在第一个状态中的一个对象将执行一些确定的动作, 当规约的事件发生并规约的条件满足时, 进入第二个状态。

② 状态转换的规约:

一般涉及以下5个部分:

① 源状态: 引发该状态转换的那个状态。

② 转换触发器: 在源状态中由对象识别的事件, 并且一旦满足其监护条件, 则使状态发生转换。其中, 在同一个简单状态图中, 如果触发了多个转换, “点火”的是那个优先级最高的转换; 如果这多个转换具有相同的优先级, 那么就随机地选择并“点火”一个转换。

③ 监护 (guard) 条件: 一个布尔表达式, 当某个转换触发器接受一个事件时, 如果该表达式有值为真, 则触发一个转换; 若有值为假, 则不发生状态转换, 并且此时如果没有其它可以被触发的转换, 那么该事件就要丢失。

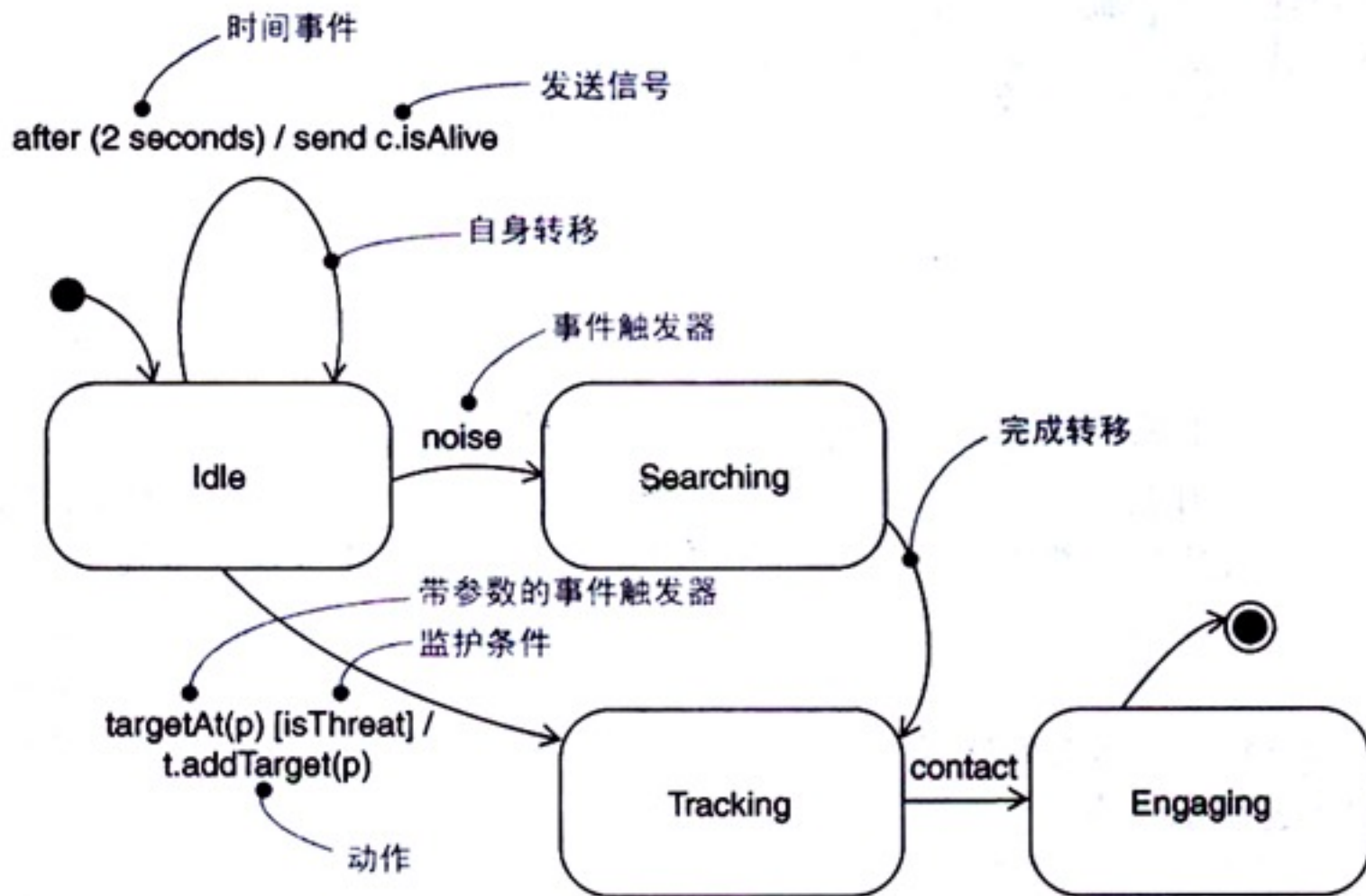


状态图

- 定义
- 内容
- 一般用法

④效应（effect）：一种可执行的行为。例如可作用于对象上的一个动作，或间接地作用于其他对象的动作，但这些对象对那个对象是可见的。

⑤目标状态：转换完成后所处的那个状态。



状态图

- 定义
- 内容
- 一般用法



③ 表达及其格式:

在UML中，把状态转换表示为从源状态出发、并在目标状态上终止的带箭头的实线。转换可以予以标记，其格式为：

转换触发器 ‘[‘监护条件’]’ ‘/’ 动作表达式

其中：

- 转换触发器：描述带参数的事件。其格式为：

事件名 ‘(‘由逗号分隔的参数表‘)’

- 监护条件：通常是一个布尔表达式，其中可以使用事件参数，也可以使用具有这个状态机的对象之属性和链，甚至可在监护条件处直接指定对象可达的某个状态，例如：

“in State1”或“not in State2”。

- 动作表达式：给出触发转换时所执行的动作，其中可以使用对象属性、操作和链以及触发事件的参数，或在其范围内的其它特征。



北京大学

状态图

- 定义
- 内容
- 一般用法

6.3.4.3 状态图的一般用法

①建立一个系统动态方面的模型，这些动态方面包括任意种类对象、任意系统结构（类、接口、构件和节点）视角下以事件定序的行为。

②建立一个场景的模型，其主要途径是针对**use case**给出相应的状态图。

其中，不论是①还是②，通常都是对反应型对象（**reactive object**）的行为进行建模。

反应型对象，或称为事件驱动的对象，其行为特征是响应其外部语境中所出现的事件，并作出相应的反应。



状态图

- 定义
- 内容
- 一般用法

为反应型对象建立一个状态机模型时，其步骤为：

- 选择状态机的语境，即是类、是用况或是子系统。
- 选择其实例（例如类的对象）的初始状态和最终状态，并分别给出初始状态和最终状态的前置条件和后置条件，以便指导以后的建模工作。
- 标识某一可用的时间段，考虑该实例在此期间内存在的条件，以此判断该实例的其它状态。对此应以该实例的高层状态开始，继之再考虑这些状态的可能的子状态。
- 判断该实例在整个生存周期内所有状态的有意义的偏序。
- 判断可以触发状态转换的事件。可以逐一从一个合理定序的状态到另一个状态，来模型化其中的事件。



状态图

- 定义
- 内容
- 一般用法

- 为这些状态转移填加动作，或为这些状态填加动作。
- 使用子状态、分支、合并和历史状态等，考虑简化状态机的方法。
- 检查在某一组合事件下所有状态的可达性。
- 检查是否存在死状态，即不存在任何组合事件可以使该实例从这一状态转换到任一其它状态。
- 跟踪整个状态机，检查是否符合所期望的事件次序和响应。



状态图

- 定义
- 内容
- 一般用法

应用实例：创建一个控制器状态机的状态图，其中该控制器负责对一些传感器进行监视。

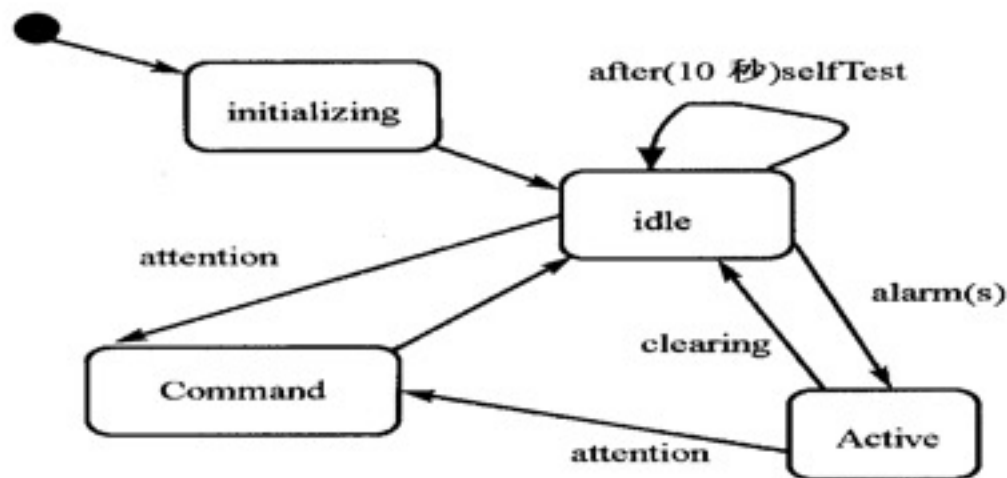
- 当创建这种控制器的一个对象时，由于要做一些初始化工作，因此该对象首先进入“初始化”（initializing）状态；
- 一旦完成初始化活动后，则自然进入“休眠”（Idle）状态；
- 在休眠状态中，按一定时间间隔，接受传感器的alarm事件（具有参数S，表示相关的传感器），一旦当接收到一个alarm事件，或接受到用户发送的attention信号，控制就从休眠状态转移到“活化”（Active）状态，或转移到“命令”（Command）状态；
- 在活化状态中，仅当发生clearing事件或需要发送attention信号时，分别进入休眠状态或命令状态。



状态图

- 定义
- 内容
- 一般用法

在嵌入式系统中，以上是一种常见的控制行为，其状态图如下所示：



其中:◆在该状态图的休眠状态上，有一个由时间事件触发的自转移，意指每隔10秒，接受传感器的selfTest事件。
◆该状态图没有终止状态，意指该控制器不间断地运行。

