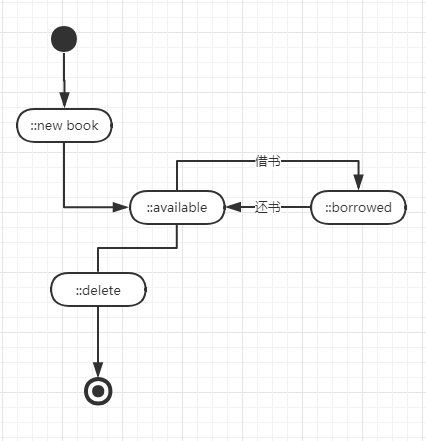
**7.1状态图概述**

状态图是系统分析的常用工具之一，它通过建立类对象的生存周期模型来描述对象随时间变化的动态行为。

状态图由状态、转换、事件、活动和动作5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，并且描述一类的实例。状态图包含一个类的对象在其生命周期期间的所有状态的序列以及对象对接收到的事件所产生的反应。它是状态节点通过转移连接的图，描述了一个特定对象的所有可能状态，以及由于各种事件的发生而引起状态之间的转移。大多数面向对象技术都使用状态图来描述一个对象在其生命周期中的行为。

状态图用初始状态表示对象创建时的状态，每一个状态图只有一个初始状态，用实心圆点表示。每一个状态图可能有多个终止状态，用一个实心圆外加一个圆圈表示。



图书馆书籍的状态图

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，图书的状态又变成“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。

**7.2状态图的基本元素**

**P119，看下，我不确定状态图的基本元素。**

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。

1. 状态

状态定义对象在其生命周期中的条件或状况。

1. 转换

对象的状态之间的转移叫转换，它包括事件和动作。

**7.3 状态**

状态定义对象在其生命周期中的条件或状况。

一个对象的状态可能包含子状态或其他一些更加详细的内容。具体由以下5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。

1. 名称

名称（name）是将一个状态与其他状态区分开来的文本字符串；状态也可能是匿名的，这表示它没有名称。

2.进入/退出动作

进入/退出动作（entry/exit action）表示进入/退出这个状态所执行的动作。入口动作的语法是entry/执行的动作；出口动作的语法是exit/执行的动作。每当进入或退出状态时，进入和退出操作将分别允许发出同一操作。这可以通过进入和退出操作来顺利地完成，而不必明确地将操作放在每个输入或输出转移上。动作与一个转移相关联，在较少的时间内完成，其操作具有原子性，也可以是动作序列，通常发生于状态的初始化、进入和退出时。

进入和退出操作时。进入和退出操作可能没有实参或警戒条件。位于模型元素的状态机顶层的进入操作可能具有特定的参数，这些参数代表了在创建该模型元素时状态机所接收到的实参。

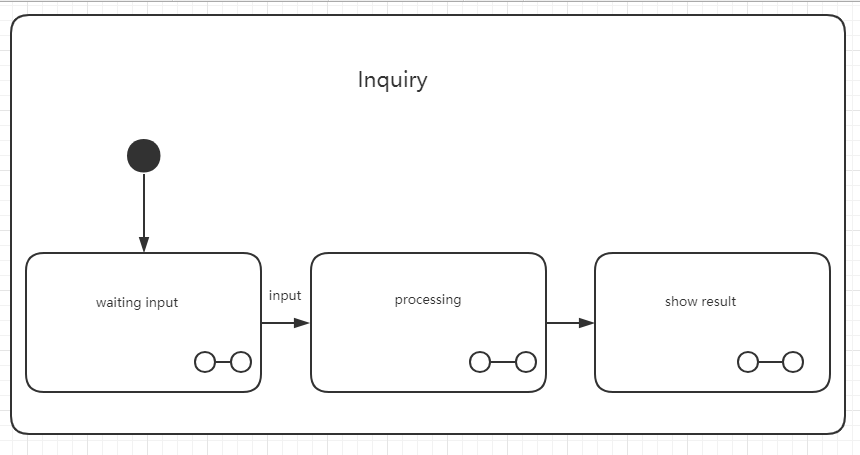
3.内部转换

内部转换（Internal Transition）使事件可以在不退出状态的情况下在状态内得到处理，从而可避免出触发进入或退出操作。定义内部转换的原因是有时候入口/出口动作显得是多余的。例如，某状态的入口/出口分别是打开/关闭某文件，但如果用户仅仅是想更改该文件的文件名，那么，这里定义的入口/出口动作显得多余，这时就可以使用内部转换，而不触发入口/出口动作的执行。内部转移可能会有带参数和警戒条件的事件，它们所代表的基本上是中断处理程序。

4.子状态

UML状态图中嵌套在另外一个状态中的状态称为子状态（Sub State），简单状态是没有子结构的状态。具有子状态（嵌套状态）的状态被称为组合状态。子状态可能被嵌套到任意级别。嵌套的状态机最多可能有一个初始状态和一个终止状态。

**P120 下面三段看看要不要加**



图书馆信息系统的图书查询的子状态

当人们通过计算机网络查询图书馆中某图书的有无时，需要在查询的客户端输入查询条件，客户端的查询状态又可以分成三个子状态：发送查询条件信息给主机服务器、等待查询结果和显示查询结果。

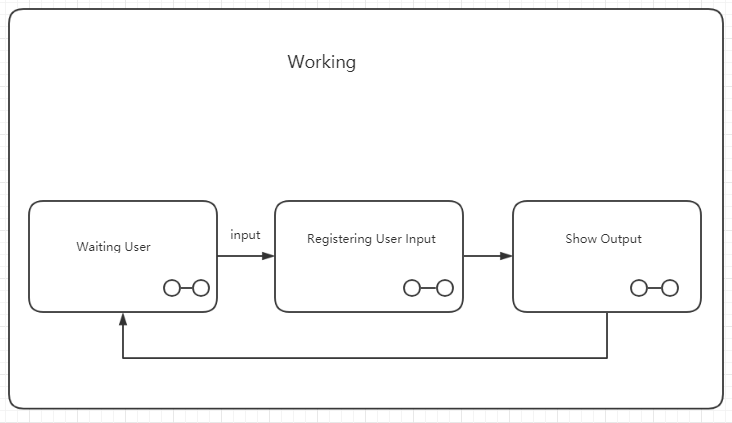
1. 顺序子状态

顺序子状态（Sequence Substate）顾名思义是按照顺序一个接一个出现。如果一个复合状态的子状态对应的对象在其生命期内任何时刻都只能处于一个子状态，即不会有多个子状态同时发生的情况，这个子状态就叫顺序子状态。

分析子状态机图形用户界面（GUI）的“工作”状态 （Working），可以得到以下状态序列。

1. 等待用户输入，简单记作“等待”状态（Awaiting User Input）
2. 登记用户输入，简单记作“注册”状态（Registing User Input）
3. 显示用户输入，简单记作“显示”状态

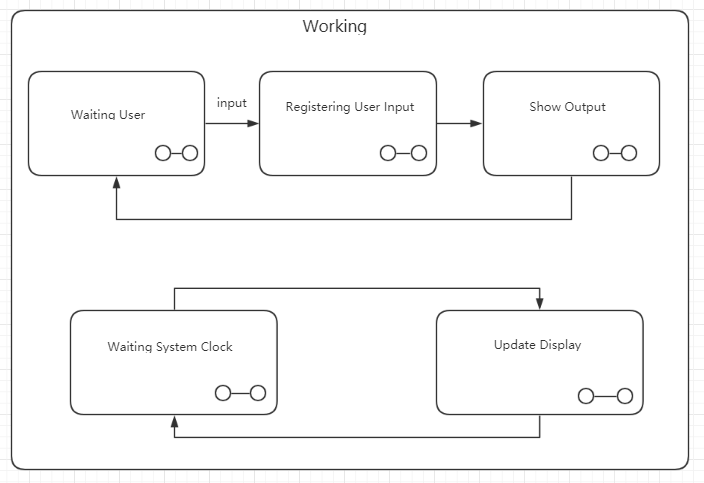
用户输入触发了从“等待”状态到“注册”状态的转移。“注册”状态内的活动引起了图形用户界面（GUI）到“显示”状态的转移。



GUI的Working状态中的顺序子状态

1. 并发子状态

所有这些与前面的顺序子状态的转移同时进行。尽管每个状态序列是一组顺序子状态，但是两个状态序列之间是并发关系。并发子状态（Concurrent Substate）之间用虚线隔开，表示状态序列之间是并发关系。



GUI的并发子状态

图显示了图形用户界面（GUI）处于“工作”状态时的子状态

在处于“工作”状态时，图形用户界面（GUI）并不是仅等待用户输入。它还要监视系统的时钟或者有时定期更新应用程序的界面显示。例如，一个应用程序可能包括一个屏幕时钟，它的图形用户界面GUI需要定期被更新。

5.延迟事件

延迟事件（Deferred Event）是其处理过程被推迟的事件，它们的处理过程要到事件不被延迟的状态被激活时才会执行。当该状态被激活时，将触发该事件，同时可能导致转移（好像该事件刚刚发生）。要实践延迟的事件，需要有事件的内部队列，也就是延迟事件的一个列表，如果事件已发生但被列为延迟，它就会被添加到队列中。队列中的事件当前状态下不会处理。当对象进入了不会使事件延迟的状态时，将立即从该队列中取出这些事件。对于这些被延迟的事件，可以使用状态的延迟事件来建模。

**7.4 转换**

对象的状态之间的转移叫转换，它包括事件和动作。

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，并在特定事件发生而且某个特定的警戒条件满足时进入目标状态。

转换是由如下5部分组成：源状态、触发事件、监护条件、动作和目标状态。

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。
2. 触发事件

转换的触发事件就是引起转变的事件，是转移的诱因，可以是一个信号、事件、条件变化和时间表达式。一个信号或调用可以带有参数，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，将对监护条件进行求值。监护条件是一个方括号括起来的布尔表达式，它被放在触发条件的后面。只要监护条件不重叠，就只可能会有来自同一源状态并具有同一事件触发器的多个转移。当事件发生时，只为转移进行一次监护条件求值。如果是“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，它对应的动作被执行。它是一个可执行的原子操作，也就是说动作是不可中断的，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。它可以是包含一系列简单动作的动作序列。

1. 目标状态

转换使对象从一个状态转换到另一个状态。当转换完成后，对象的状态发生了变化，这时所处的状态就是转换的目标状态。在图形上，源状态和目标状态不同于初始状态和终止状态。源状态位于表示转换的箭头的起始位置的状态，目标状态位于表示转换的箭头所指的那个状态。

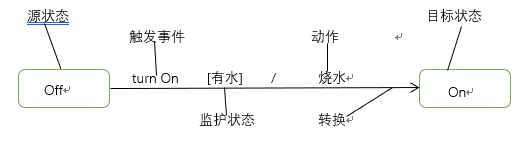


图7.8 状态图转换的元素

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”就是其触发事件，进行“烧水”是状态转换的动作。