状态机描述的状态有两种大类：

第一类型是任务运行似的逻辑。这种状态图的状态通常是任务在等待某个条件达到，也就是任务处于“等待状态”。通常任务本身是直接推动状态机运作的发动机，任务本身检查状态机的事件达到情况并进行分发。任务状态的变迁由某个事件的达成触发，可能要求任务间同步机制 ITC 具有回调能力，以便在条件成立时向状态机事件队列中压入事件。这种状态机明显有异步逻辑特征，称为“异步类状态机”。

第二类型是算法逻辑，利用状态机逻辑清晰的特性描述复杂的算法。该状态机通常嵌入在 某个算法函数类部，帮助该函数完成功能。函数入口后，模拟任务等具有主动性的实体，向 状态机分发事件。这种状态机明显有同步函数调用特征，称为“同步类状态机”。同步状态机的一次事件过程就是一次入口函数调用过程。入口函数调用返回时，状态机必须处于一个明确定义的状态，不能是不确定的中间状态。无论异步还是同步，特别是对异步状态机而言，状态机内部不能在某个状态发生等待、阻塞等情况，而应该是标记状态后返回，把执行的机会让给其他任务或逻辑体（状态机的驱动实体）。之后在条件达到的情况下，通过事件分发机构调动状态机的运行。

初始事件：初始转换时，如果该状态没有历史，则应该通过初始转换事件进入设计的默

认状态，否则进入历史状态。初始事件只对逻辑目标和其后的设计目标进行状态机结构上的初始化，中间状态设计上不推荐初始事件，即便有,也会被忽略。因此，初始事件一 定不能有除了指定默认状态之外的任何其它动作，否则容易产生极难调试的 bug。单纯只有指定默认状态情况下，中间初始事件被忽略掉是完全可以理解的。

退出事件：清理状态，但是要保留历史数据。退出事件并不表示要进入上层超态，退出

是状态转换的中间过程，中间没有转换到任何状态，仅在结束时转换到目标态，并可能由目标态初始事件引发转换到进一步的子态。退出仅仅是退出，一直退到 LCA 为止， 之后开始进入。退出也不引起转换源状态的变化。转换源状态仅由事件的第一处理状态决定。

正交状态：不采用正交状态。如果需要，多个状态机来设计正交状态机。

执行环境：执行环境的概念实际上是为了事件处理函数能够方面地通过状态参数获得当

前的内部数据。特别是经典的用类继承实现状态机的实现方式，执行环境的概念很重要。