**Practical No 7**

**Aim: Study of State Transition Diagram**

State transition diagrams have been used right from the beginning in object-oriented modelling. The basic idea is to define a machine that has a number of states. State transition diagrams were around long before object modelling. They give an explicit, even a formal definition of behaviour. A big disadvantage for them is that they mean that you have to define all the possible states of a system. Whilst this is all right for small systems, it soon breaks down in larger systems as there is an exponential growth in the number of states. This state explosion problem leads to state transition diagrams becoming far too complex for much practical use. To combat this state explosion problem, object-oriented methods define separate state-transition diagrams for each class. This pretty much eliminates the explosion problem since each class is simple enough to have a comprehensible state transition diagram.

**Notation For State Transition Diagram:**

**State**: A condition during the life of an object in which it satisfies some condition, performs some action, or waits for some event.

**Event**: An occurrence that may trigger a state transition. Event types include an explicit signal from outside the system, an invocation from inside the system, the passage of a designated period of time, or a designated condition becoming true.

**Guard**: A boolean expression which, if true, enables an event to cause a transition.

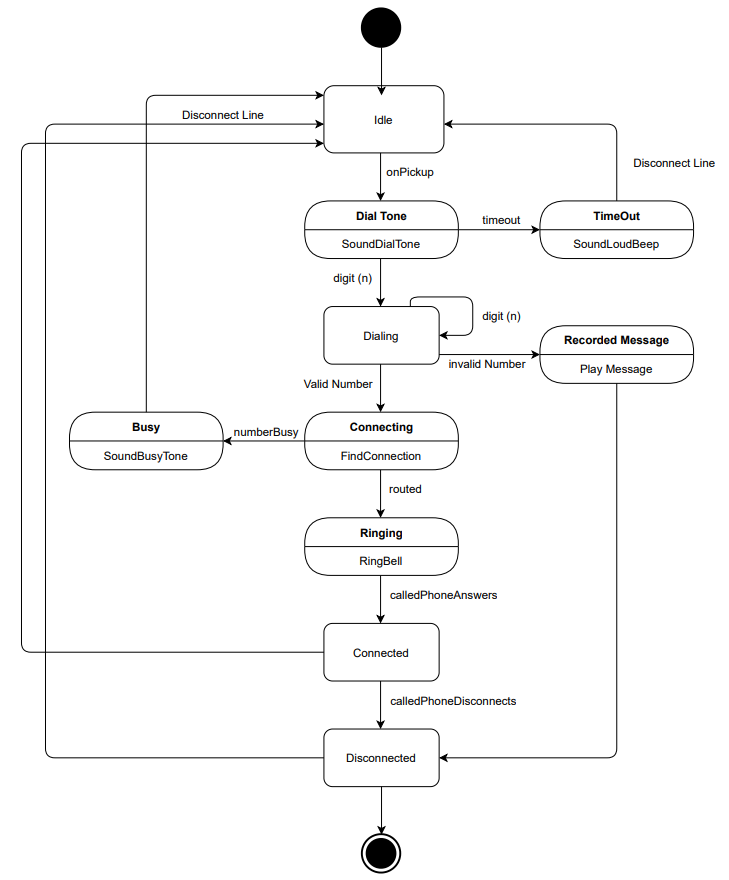
**Transition:** The change of state within an object.

**Action**: One or more actions taken by an object in response to a state change.

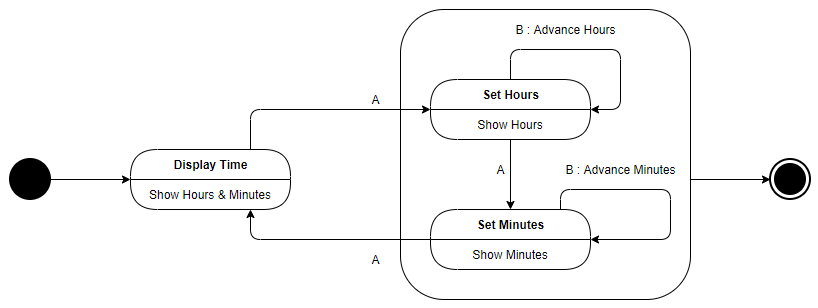
**When to Use**

State models are ideal for describing the behaviour of a single object. They are also formal, so tools can be built which can execute them. Their biggest limitation is that they are not good at describing behaviour that involved several objects, for these cases use an interaction diagram or an activity diagram. People often do not find drawing state diagrams for several objects to be a natural way of describing a process. In these cases, you can try either drawing a single state diagram for the process, or using an activity diagram. This defines the basic behaviour, which you then need to refactor to split it across a number of objects.

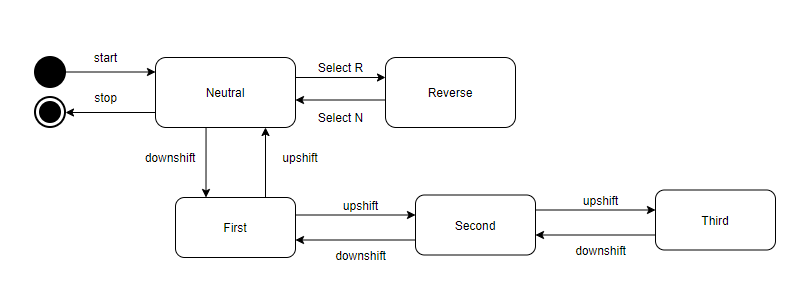
**A. Consider the class for telephone line with following activities and states: As a start of a call, the telephone line is idle. When the phone receiver is picked from hook, it gives a dial tone and can accept the dialling of digits. If after getting dial tone, if the user doesn’t dial number within time interval then time out occurs and phone line gets idle. After dialling a number, if the number is invalid then some recorded message is played. Upon entry of a valid number, the phone system tries to connect a call & routes it to proper destination. If the called person answers the phone, the conversation can occur. When called person hangs up, the phone disconnects and goes to idle state.**



**B. Draw the state transition diagram for above description of telephone line. A simple digital watch has a display and 2 buttons to set it, the A button and the B button. The watch has 2 modes of operations, display time and set time. In the display time mode, hours and minutes are displayed. The set time has 2 sub modes set hours and set minutes. The A button is used to select modes. Each time it is pressed, the mode advance in sequence, display, set hours, set minutes etc. Within the sub modes the B button is used to advance the hours or minutes once each time it is pressed. Buttons must be released before they can generate another event. Prepare a state diagram of watch.**



**C. Draw and explain state diagram for car transmission. The transmission can be either in reverse, forward or neutral state. If it is in the forward state, then it can be in the first, second or third gear. Assume suitable data. Make appropriate.**



**Conclusion: We have studied the details about the state transition diagram.**