

# State Modelling

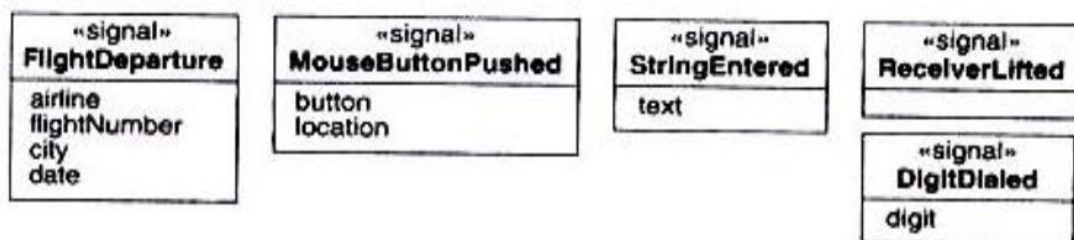
## Event

An event is an occurrence at a point in time, such as user depresses left button or flight 123 departs from Chicago, power turned on, alarm set, paper tray becomes empty, temperature becomes lower than freezing.

Two events that are causally unrelated are said to be concurrent: they have no effect on each other. Flight 123 must depart Chicago before it can arrive in San Francisco; the two events are causally related. Flight 123 may depart before or after flight 456 departs Rome; the two events are causally unrelated.

There are several kinds of events. The most common are **the signal event, the change event, and the time event.**

**Signal Event--** A signal is an explicit one-way transmission of information from one object to another. An object sending a signal to another object may expect a reply, but the reply is a separate signal under the control of the second object, which may or may not choose to send it. We can give each signal class a name to indicate common structure and behavior. For example,



**Figure** Signal classes and attributes. A signal is an explicit one-way transmission of information from one object to another.

## Change Event

A *change event* is an event that is caused by the satisfaction of a Boolean expression.

- when (room temperature < heating set point)
- when (room temperature > cooling set point)
- when (battery power < lower limit)
- when (tire pressure < minimum pressure)

**Change events.** A change event is an event that is caused by the satisfaction of a boolean expression.

- when (date = January 1, 2000)
- after (10 seconds)

**Figure**     **Time events.** A time event is an event caused by the occurrence of an absolute time or the elapse of a time interval.

## Time Event

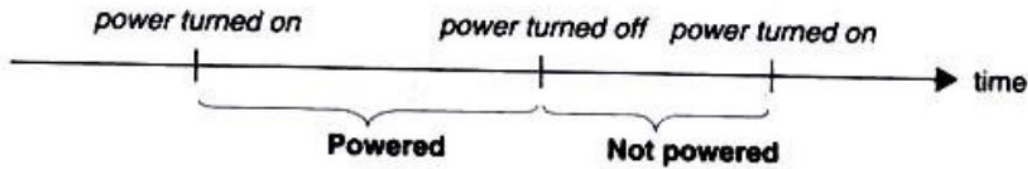
A *time event* is an event caused by the occurrence of an absolute time or the elapse of a time interval.

## States

A *state* is an abstraction of the values and links of an object. Sets of values and links are grouped together into a state according to the gross behaviour of objects. Figure 5.4 shows the UML notation for a state—a rounded box containing an optional state name.



**Figure States.** A state is an abstraction of the values and links of an object.



**Figure Event vs. state.** Events represent points in time; states represent intervals of time.

**Guard--** Transitions from one state to the next are denoted by lines with arrowheads. A transition may have a trigger, a guard and an effect, as below.



"Trigger" is the cause of the transition, which could be a signal, an event, a change in some condition, or the passage of time.

"Guard" is a condition which must be true in order for the trigger to cause the transition.

A *guard condition* is a boolean expression that must be true in order for a transition to occur.

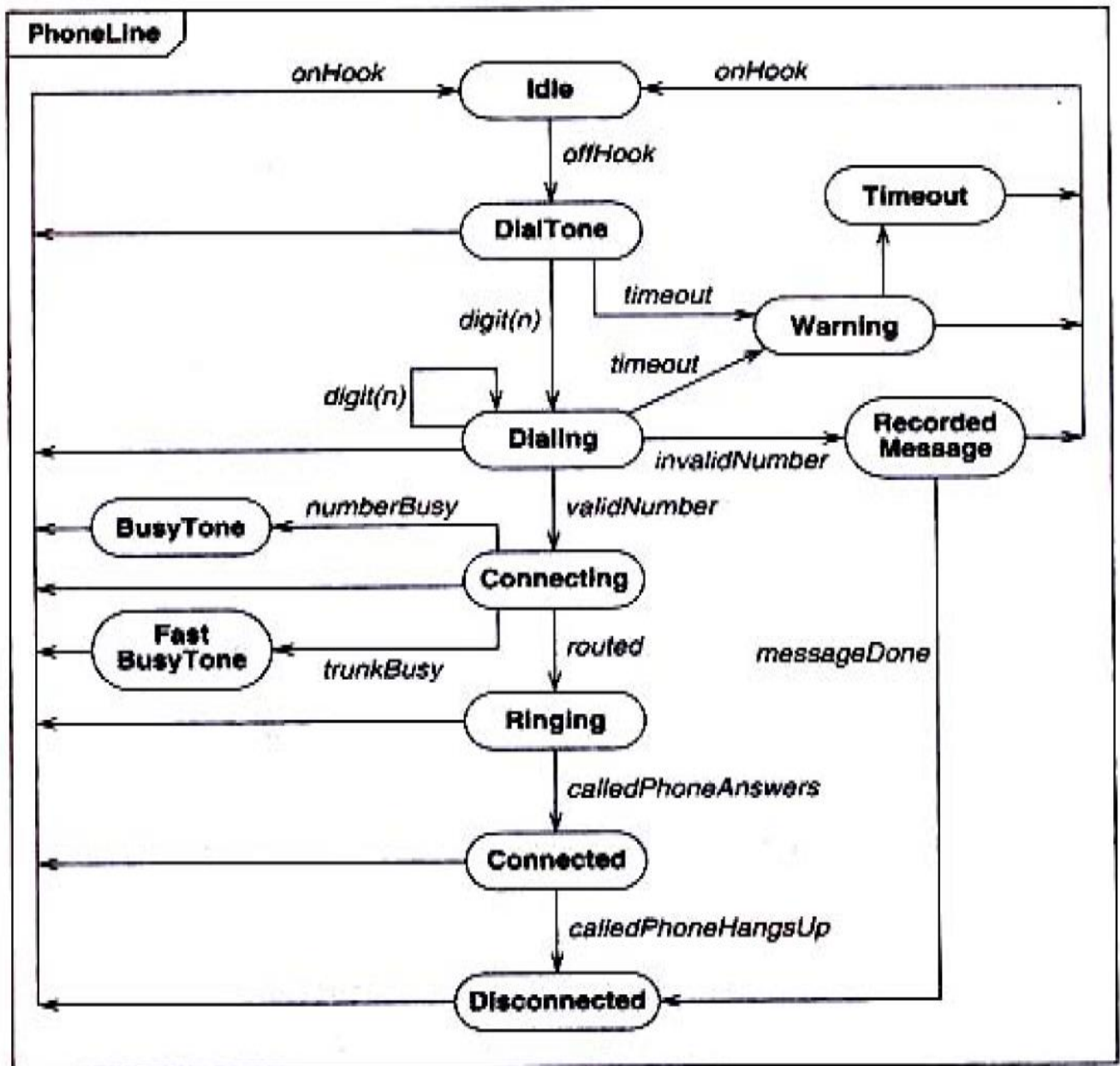
**State Diagrams--** A *state diagram* is a graph whose nodes are states and whose directed arcs are transitions between states. A state diagram specifies the state sequences caused by event sequences.

### Sample State Diagram

The figure in the next page shows a state diagram for a telephone line.

The diagram concerns a phone line and not the caller nor called person.

The diagram contains sequences associated with normal calls as well as some abnormal sequences, such as timing out while dealing or getting busy lines.



**Figure** State diagram for a telephone line. A state diagram specifies the state sequences caused by event sequences.