Behaviour Specification

A *reactive* system is a system that must react to external events:

- A Calculator must react to the keypresses.
- A Microwave oven must react to keypresses and also to the passage of time.
- An internet Router must react to the arrival of packets.
- A synchronous hardware circuit must react to the clock ticks.

Most systems can be viewed as reactive.
We can specify and/or model a reactive system using state machines.

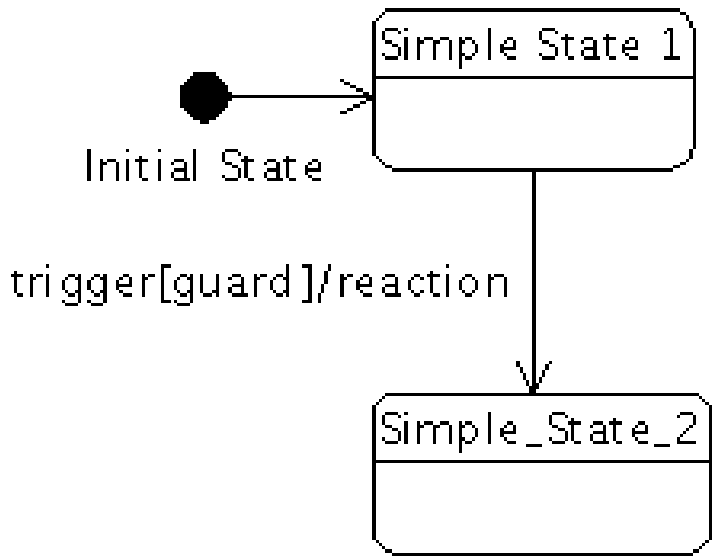
StateCharts

StateCharts is a diagrammatic language for modelling finite state systems.

There are various flavours of StateCharts.

We'll use UML StateCharts.

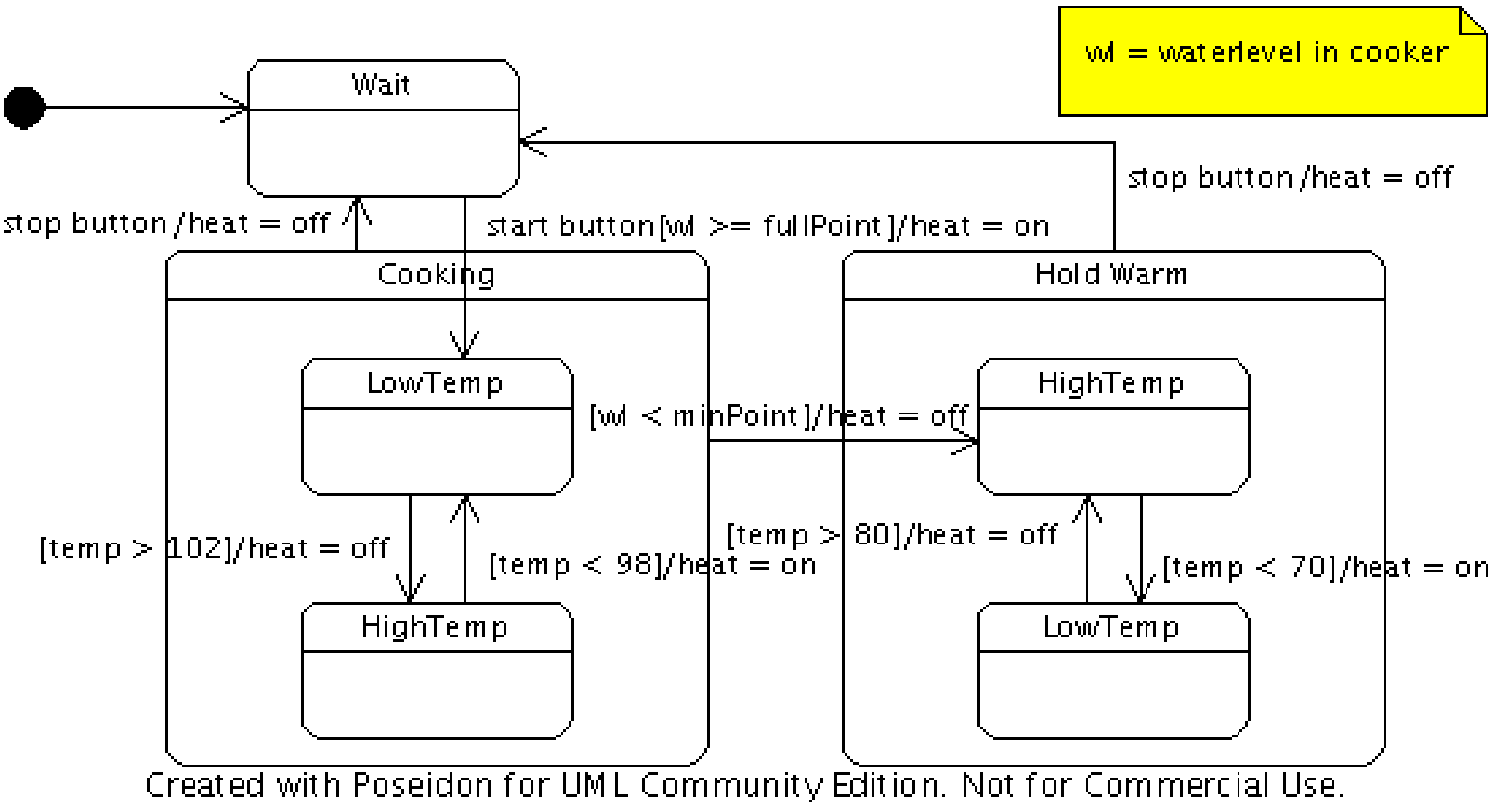
- *trigger* — the event that causes a transition.
- *guard* — a condition that must be true for the transition to occur.
- *reaction* — event or change of system variables at the instant of the transition.
- Time passes in states.
- Transitions are instantaneous.



Statecharts (cont'd)

- All of trigger, guard and reaction are optional.
  - If trigger is omitted then transition occurs as soon as guard is true.
  - If guard is omitted then guard is *true*.
  - If reaction is omitted then there is no reaction, just state change.
- Triggers may be parameterized.
- Conditions can be used to
  - Inhibit transitions — if condition is false then transition can't occur.
  - Select between alternatives — universal & mutually exclusive guards (e.g., $[x < \mathrm{SetPoint}]$, $[x \geq \mathrm{SetPoint}]$).
- Hierarchies of states can be formed:
  - non-orthogonal ("or") states — whenever the super-state is active then exactly one of the sub-states is active.
  - orthogonal ("and") states — concurrent state machines. Whenever the super-state is active one of the states in **each** of the sub-state(machines) is active.

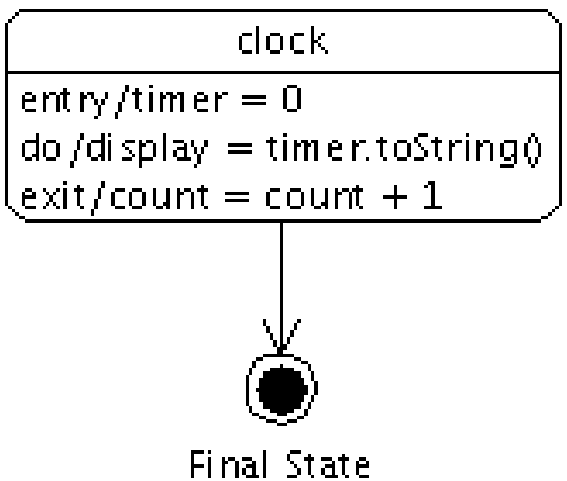
Example: Rice Cooker



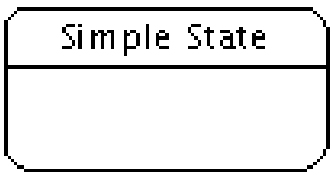
Activities

States can be annotated with *activities*

- *entry* activities occur on entry to the state.
- *do* activities happen continuously when the state is active.
- *exit* actvitites occur on exit from the state.



State Types



Simple state State with no substructure

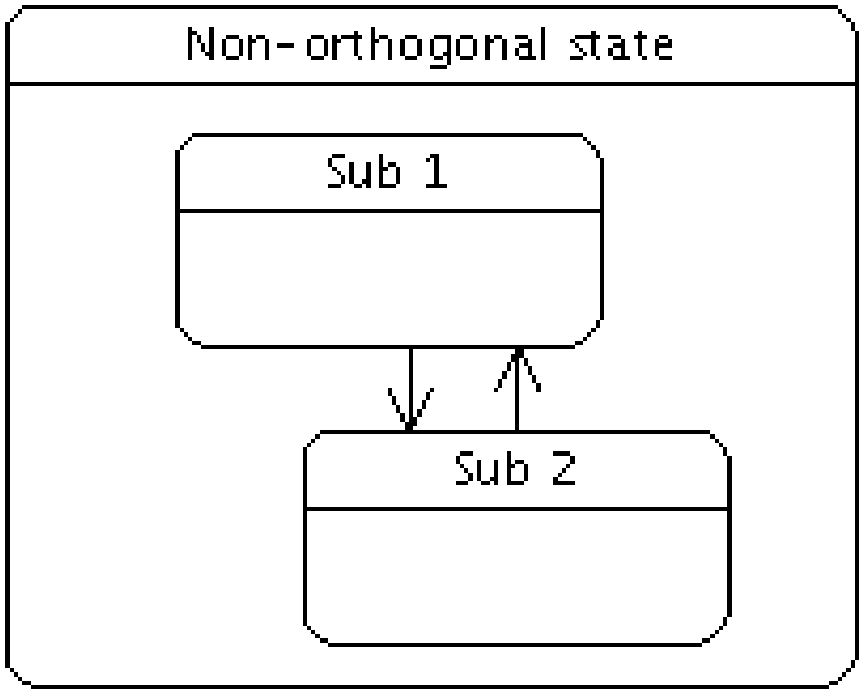
Initial state A pseudostate that indicates the starting state when the enclosing state becomes active.

Final state A special state whose activation indicates the enclosing state has completed activity.

[0] From [1].

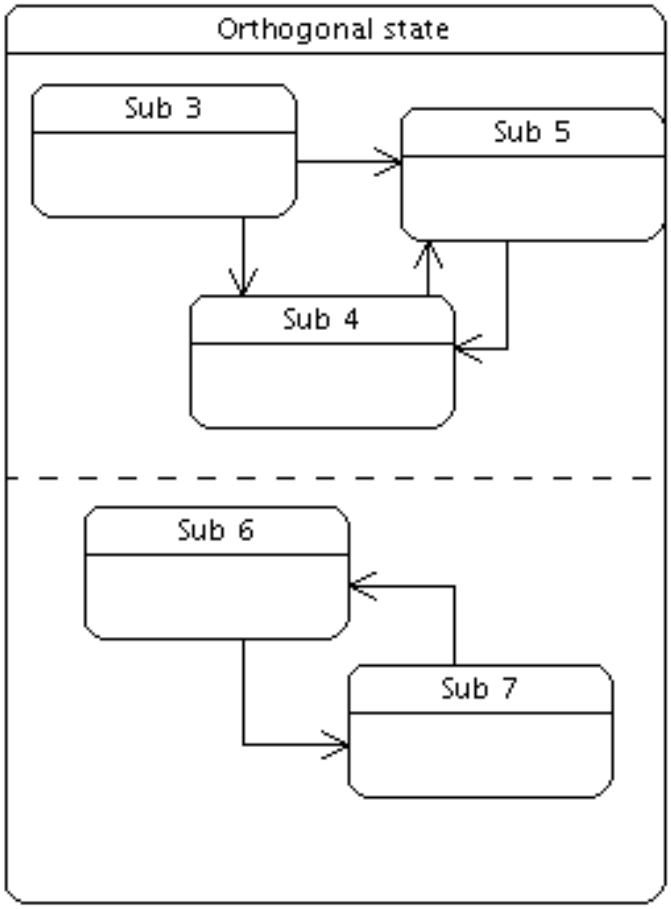
State Types (cont'd): Non-orthogonal state

Composite state that contains one or more direct sub-states, exactly one of which is active at one time when the composite state is active.



State Types (cont'd): Orthogonal state

Divided into two or more regions. One direct substate from each region is concurrently active when the composite state is active.

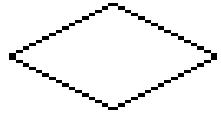


State Types (cont'd): Orthogonal state

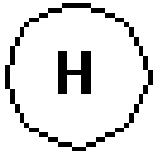
Terminate A special state whose activation terminates execution of the object owning the state machine.

Junction A pseudostate that chains transition segments into a single run-to-completion transition.

Choice A pseudostate that perfoms a dynamic branch within a single run-to-completion transition.

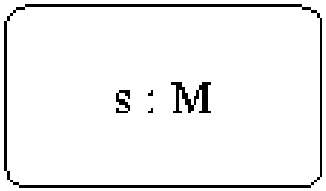


History state A pseudostate whose activation restores the previously active state within a composite state.

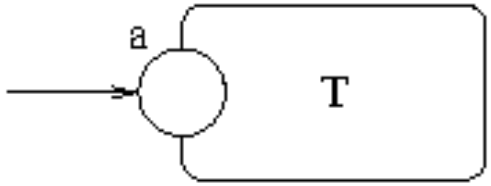


State types (cont'd)

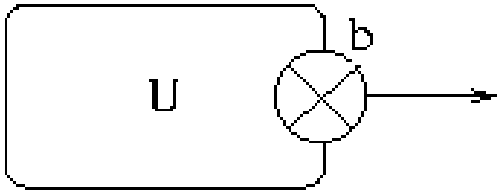
Submachine state State that references a state machine definition, which conceptually replaces the submachine state.



Entry point An externally visible pseudostate within a state machine that identifies an internal state as a target.



Exit point An externally visible pseudostate within a state machine that identifies an internal state as a source.



Transitions

External Changes active state.

Entry Specifies an activity that occurs when a state becomes active.

Exit Specifies an activity that occurs when a state is exited.

Internal Causes execution of an effect but does not cause a change of state or execution of exit or entry activities. (Note: this is different from a self transition, which does invoke exit/entry activities.)

References

[1] James Rumbaugh, Ivar Jacobson, and Grady Booch.
The Unified Modeling Language Reference Manual.
Addison-Wesley, second edition, 2005.