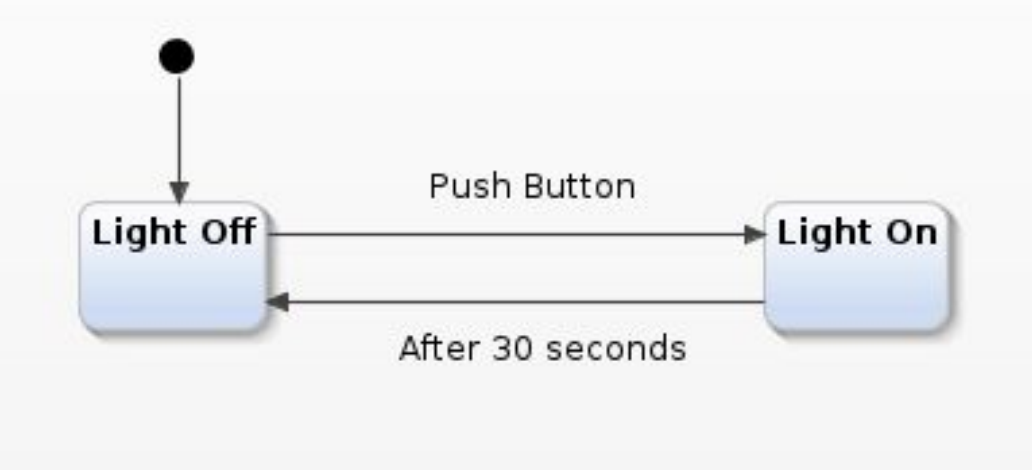
Object Oriented Analysis

Lecture 6 : State-machine Diagram

State-machine diagram

- UML proposes to model an object's life by using the state-machine diagram (finite-state machine).
- It describes the states and state transitions of an object as a result of interactions with other objects.

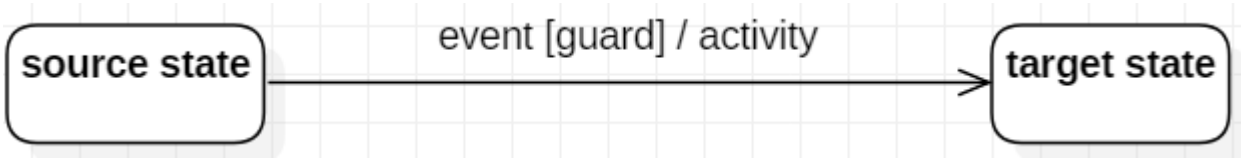


State

- A state is a constraint or a situation in the life cycle of an object, in which a constraint holds, the object executes an activity or waits for an event.
- There are 2 particular states :
  - initial state 
  - final state 
- There are :
  - simple state
  - composite state

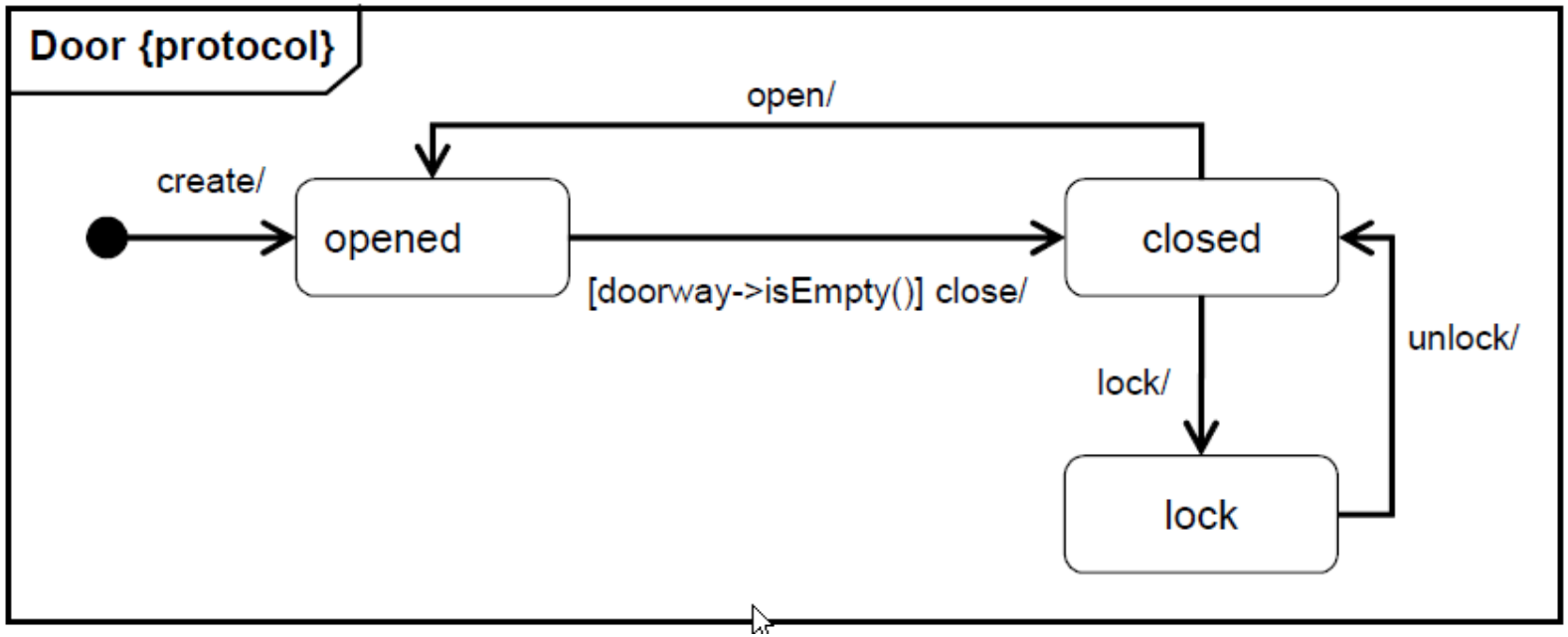
Transition

- A transition depicts the movement from one state to another



- A transition occurs following an event
- A transition can be conditioned using a guard (Boolean expression expressed in natural language)
- If a trigger event occurs and the guard condition is true, the object changes from the source state to the target state and executes an activity (transition effect)

Example

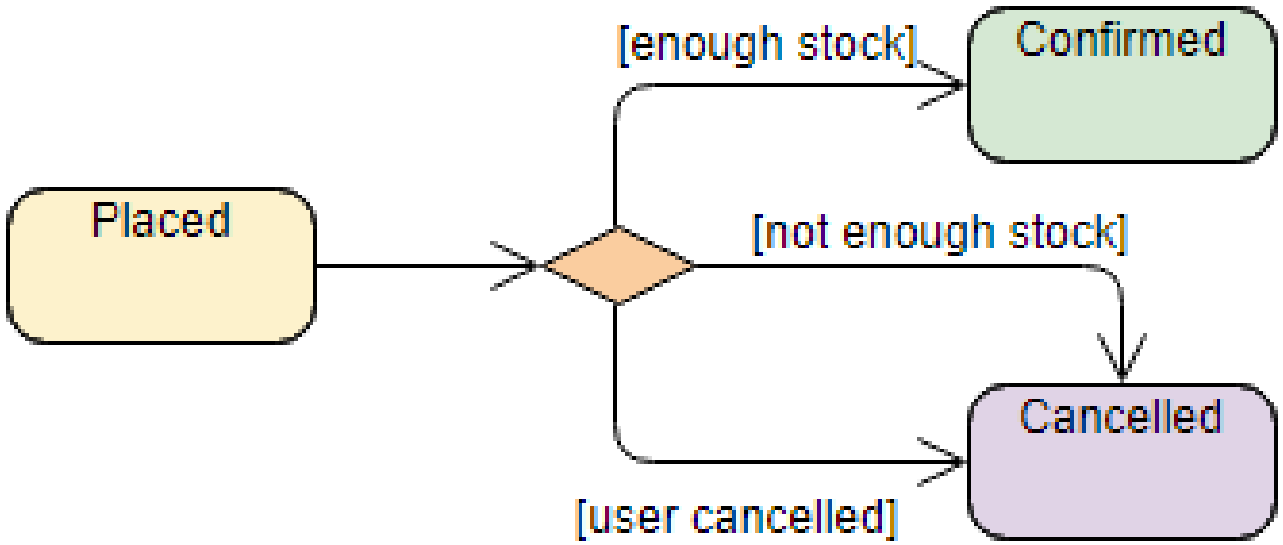

Door {protocol}
open/
create/
opened
closed
[doorway->isEmpty()] close/
unlock/
lock/
lock

Types of events

- Signal event
  - corresponding to the arrival of an asynchronous message or signal
- Call event
  - corresponding to the arrival of a call to an operation (a declared method in the class diagram)
- Time event
  - a time event occurs after a specified time has elapsed
  - after(duration)
  - when(date = <date>)
- Change event
  - a change event occurs whenever a specified condition is met
  - when(expression).

Choice/Decision Node

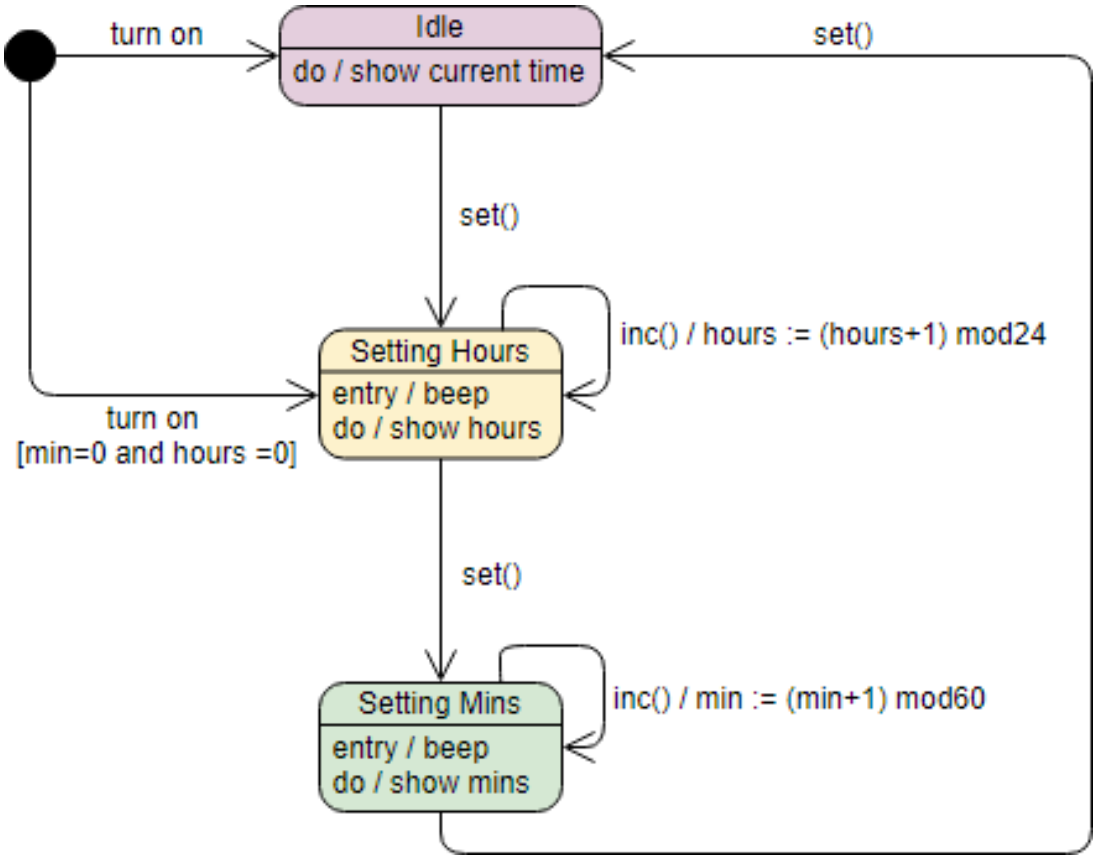
- **Choice** is a pseudo state which allows splitting of transitions into multiple outgoing paths such that the decision on which path to take.



https://online.visual-paradigm.com/diagrams/tutorials/state-machine-diagram-tutorial/

Entry/Do/Exit Action

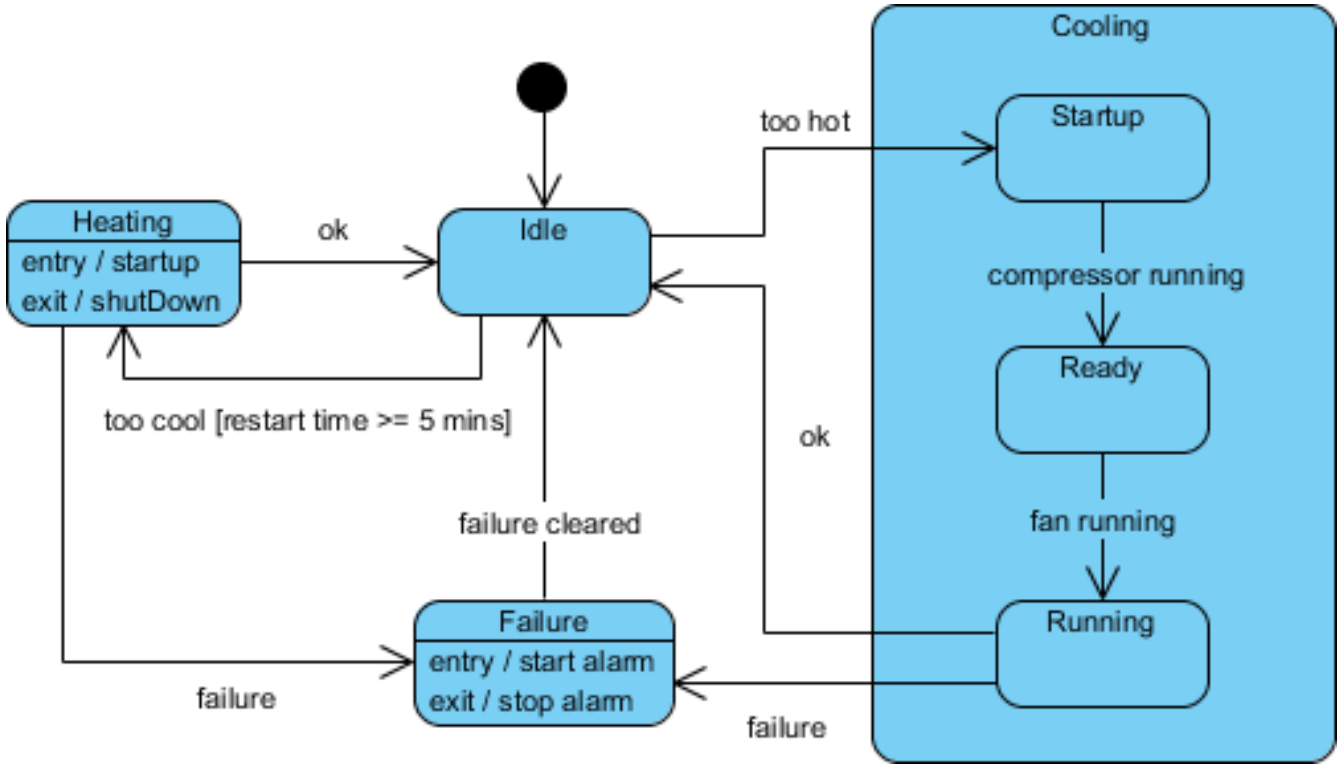
- Entry : action performed on entry to state
- Do : action performed during state
- Exit : action performed on leaving state



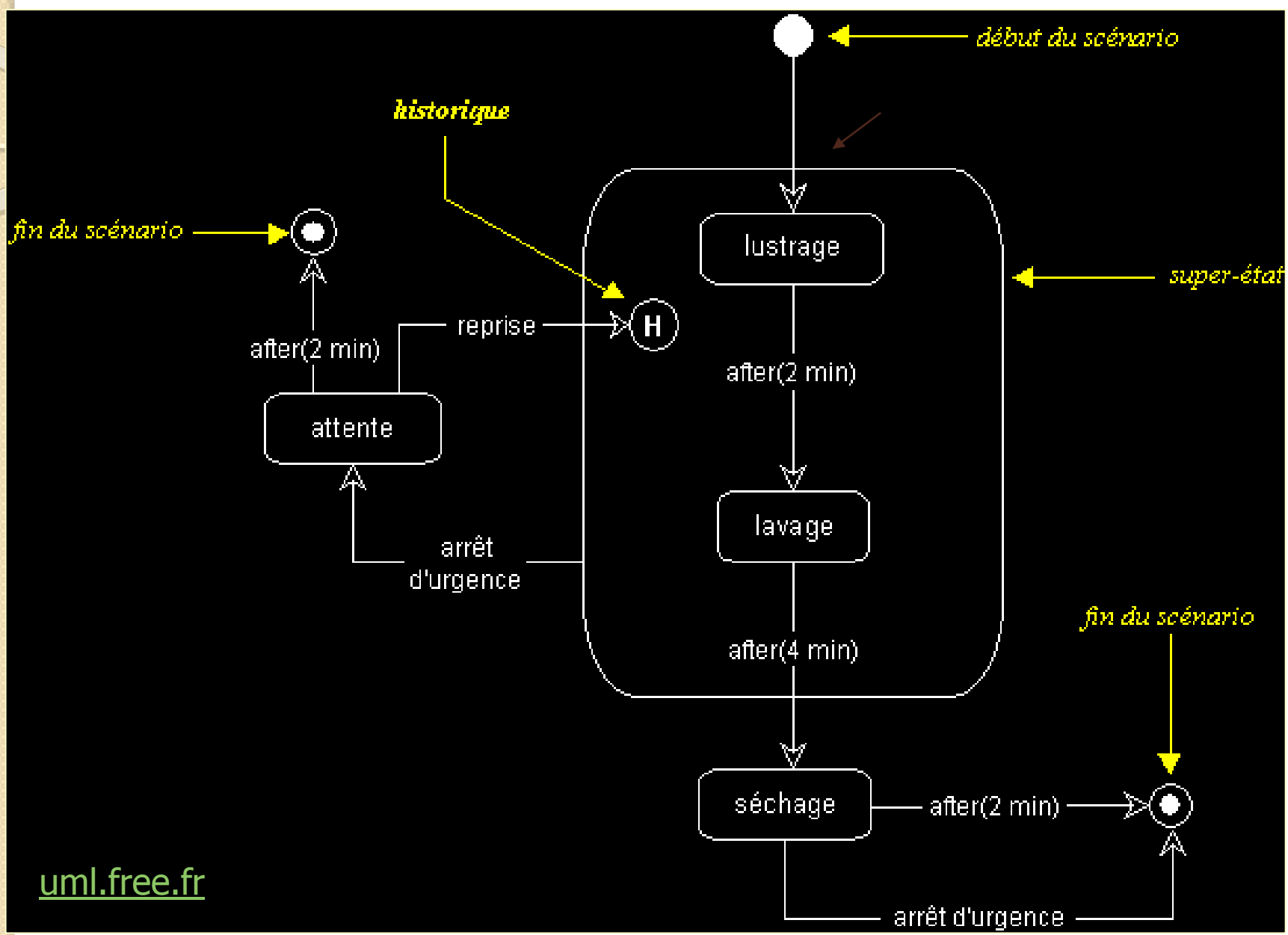
https://online.visual-paradigm.com/diagrams/tutorials/state-machine-diagram-tutorial/

Composite state

- A state which has substates (nested states) is called a composite state. A nested state machine may have at most one initial state and one final state. It contains a history that allows to remember the latest active substate.



Composite State with History



State-machine with Arduino

- https://www.hackerspacetech.com/state-machine-with-arduino/