

State diagram

PB007 Software Engineering I

Stanislav Chren

8. 11. 2021

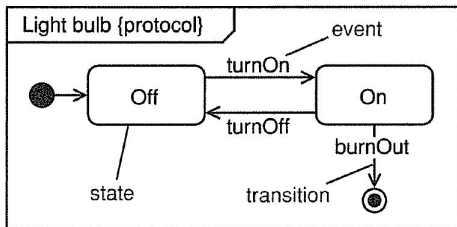
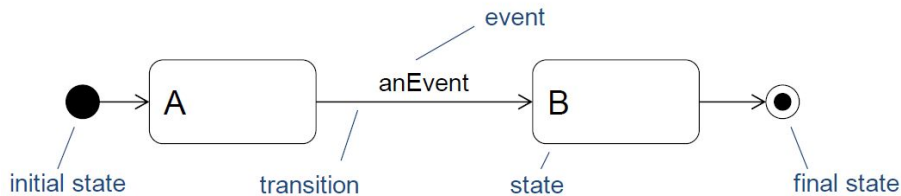


State diagram models dynamic behaviour (lifecycle) of one *reactive object* (class, use case, system, subsystem).

It consists of:

- States
- Transitions
- Events

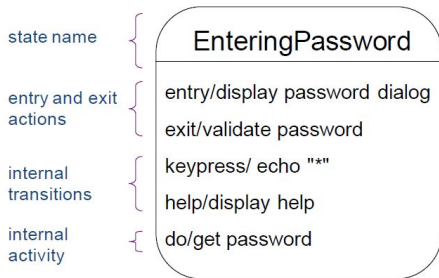
Elements of state diagram



State represents semantically important situation in an object. State of the object is determined by its attribute values by connections to other objects and ongoing activity.

State may include:

- Entry/Exit actions
- Internal actions/transitions
- Internal activity



Transitions

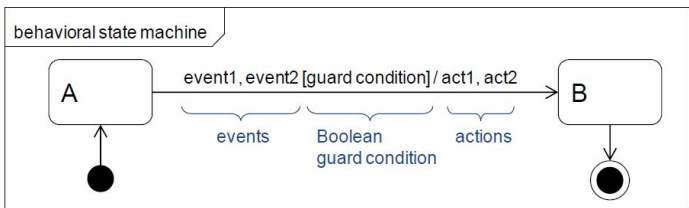
Transitions specify how the states of the object can change.

Transition consists of:

- Events
- Guard Conditions
- Actions

Syntax: *event[guard condition]/action*

Semantics: At the occurrence of the *event*, if the *guard condition* is satisfied, perform the *action* and switch to the new *state*.



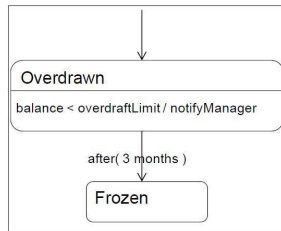
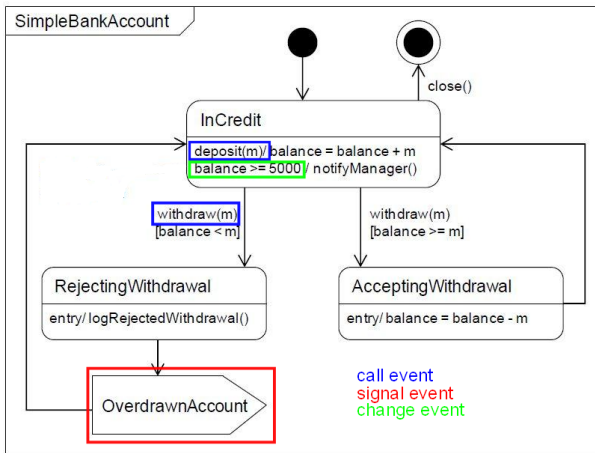
Event is a trigger, to which the object can respond by change of state or by execution of an operation.

There are 4 types of events:

- **Call event** - operation call on the given object.
- **Signal event** - asynchronous sending and reception of a signal from one object to another.
- **Change event** - a boolean expression. The event occurs when the evaluation of the expression changes from false to true.
- **Time event** - the event occurs either at a specified time (*when()*) or after specified duration (*after()*)



Events - example

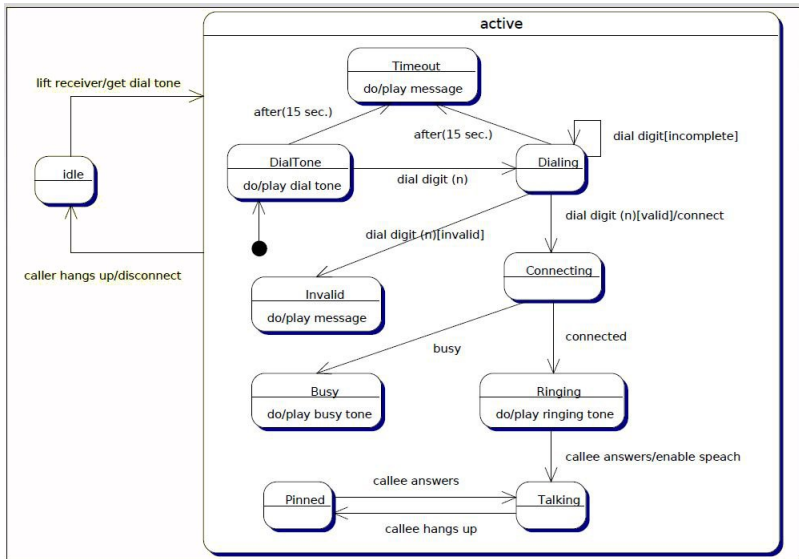


Composite state is a state which contains inner states.

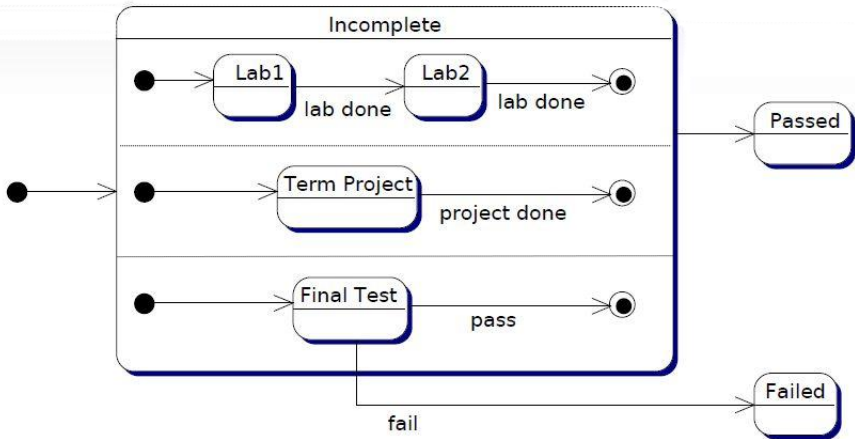
There are two types of composite states:

- **Simple composite states** - they consist of single region. Often used for capturing of "inheritance" between the states.
- **Orthogonal composite states** - they consist of at least two regions, each containing a separate state machine. The inner state machines are executed in parallel.

Composite states - Example I



Composite states - Example II



- <http://www.uml-diagrams.org/state-machine-diagrams.html>
- <http://mpavus.wz.cz/uml/uml-b-state-3-2-2.php>
- <http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm>
- http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_statediagram.html



- Choose one reactive object (class instance) for modelling.
- Make a list of all states that can be distinguished for the chosen object and that are interesting to consider with respect to the requirements and functions of the system.
- Make a list of events that have impact on object states.
- Based on the lists of states and events, construct a state diagram. You can also try to use the composite states.
- Submit the **pdf report** to homework vault(**Seminar 09**).

Deadline:

- Saturday (Groups 03, 04)
- Monday (Group 11)
- Tuesday (Groups 06, 07)
- Wednesday ([06:00 AM] Groups 08, 09)



Rules for report submission

- 1 Submit the PDF report, not the VP source file and not an exported image.
- 2 PDF report must be created using the procedure shown on the seminars including the report settings.
- 3 The name of the PDF report file should be *lastname1-lastname2-lastname3* of the team members.
- 4 PDF report must contain all diagrams modelled until now.
- 5 PDF report must be uploaded to the homework vault by the specified deadline.
- 6 PDF report must be uploaded to the correct homework vault. The name of the homework vault is always specified on the slides.
- 7 Each team uploads only a single PDF report for the whole team.
- 8 Submitted diagrams must be clear and readable.
- 9 Submitted diagrams should not contain serious mistakes. At least, they should not contain mistakes mentioned in the *Catalogue of common mistakes*.



VP report settings

