Object Oriented System Design

- Marc Conrad
  - D104 (Park Square Building)
  - Email: Marc.Conrad@beds.ac.uk
  - WWW: http://perisic.com/marc
- This week:
  - State Diagrams

Static Models and Dynamic Models

- Class diagrams model the *static behaviour* of objects, i.e.
  - Attributes of objects
  - Operation of objects
  - Relationships between objects.
- Statechart diagrams model the dynamic behaviour of objects.

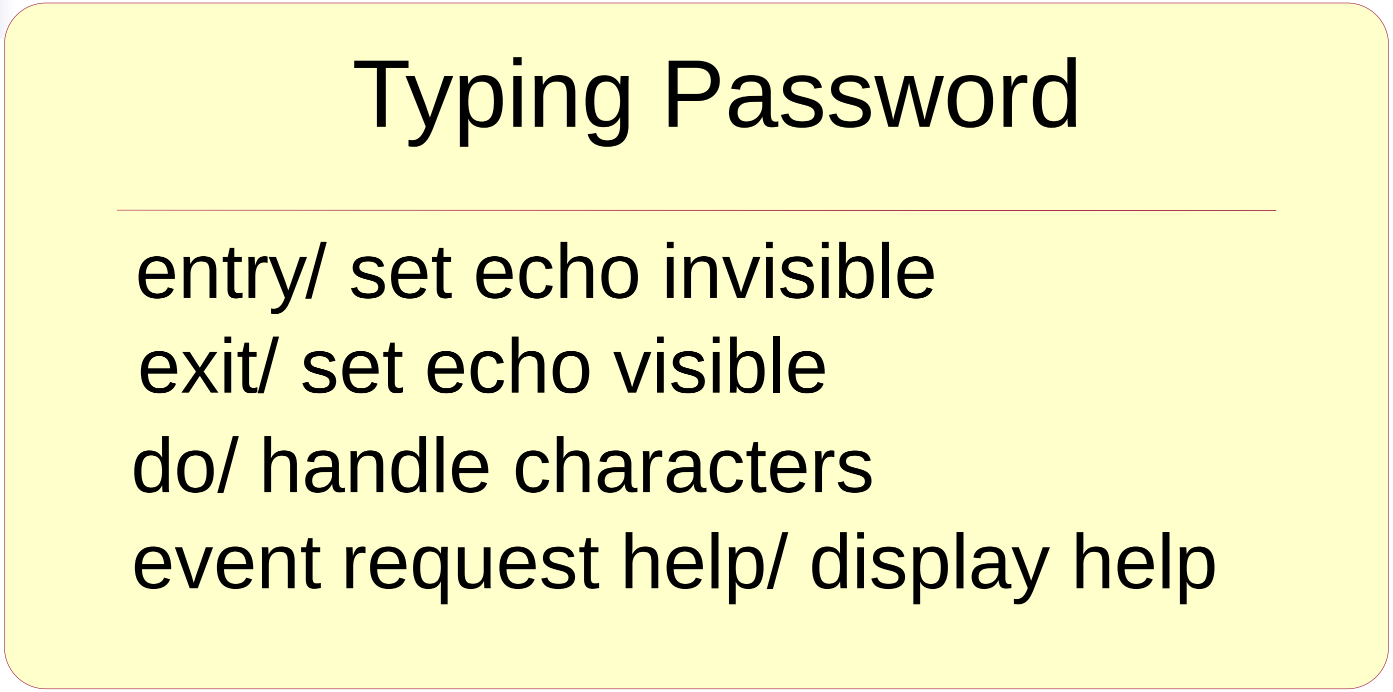
Statechart diagrams

- State diagrams illustrate the dynamical behaviour of an object.
- State diagrams are about *events* and *transitions.*
  - Events trigger a state change.
  - Transitions show the change from one state to another.
- See google (images, statechart) for real-life examples.

States (definition)

- A state is a condition in which an object can be at some point during its lifetime, for some finite amount of time.
- An object can
  - perform an activity
  - wait for an event

Example of a state



Typing Password

entry/ set echo invisible

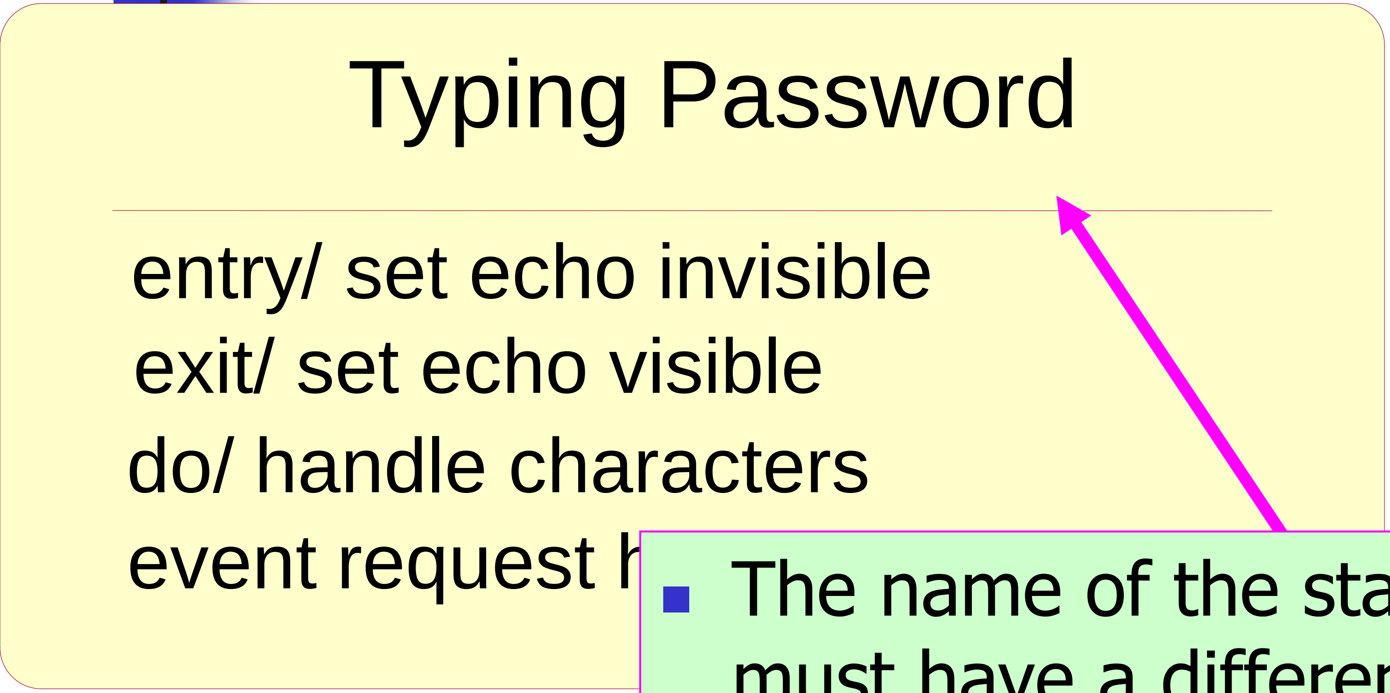
exit/ set echo visible

do/ handle characters

event request help/ display help

- The UML notation for a state is a rectangle with rounded corners.

Example of a state



Typing Password

entry/ set echo invisible

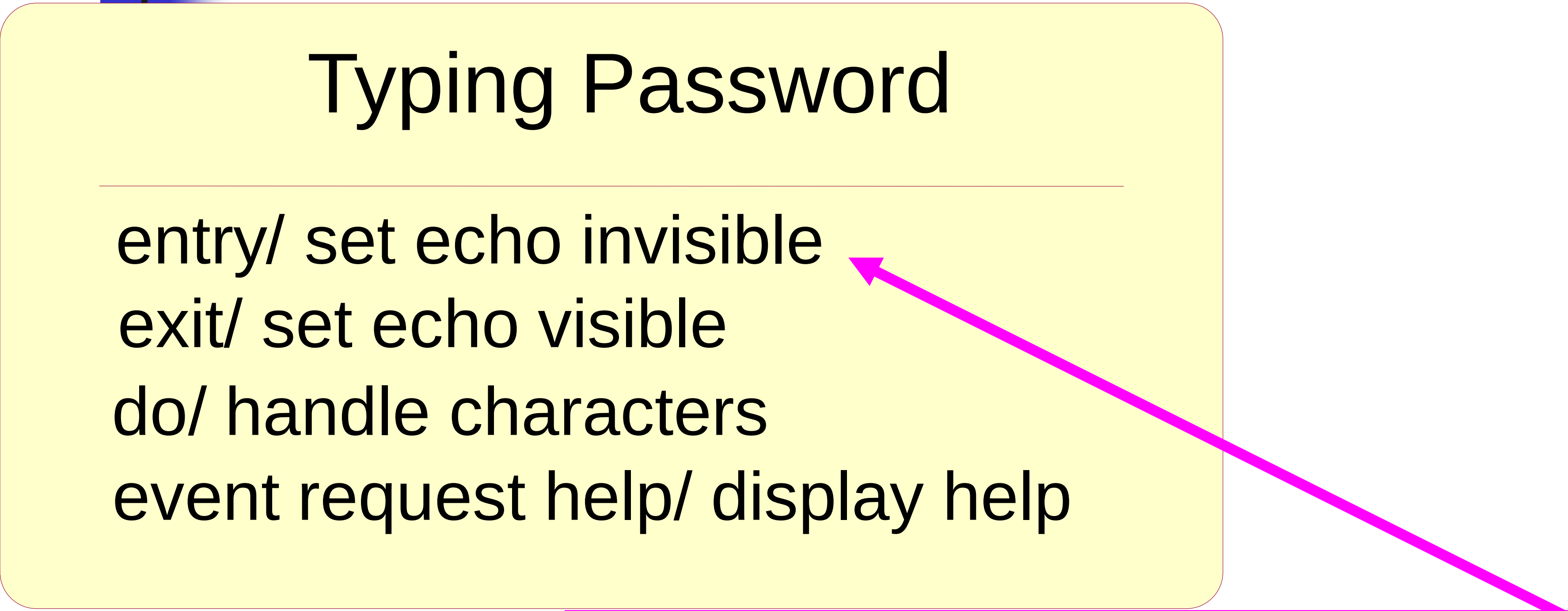
exit/ set echo visible

do/ handle characters

event request h

- The name of the state. Each state must have a different name.
- Nameless states are also allowed and are considered as being different.

Example of a state



Typing Password

entry/ set echo invisible

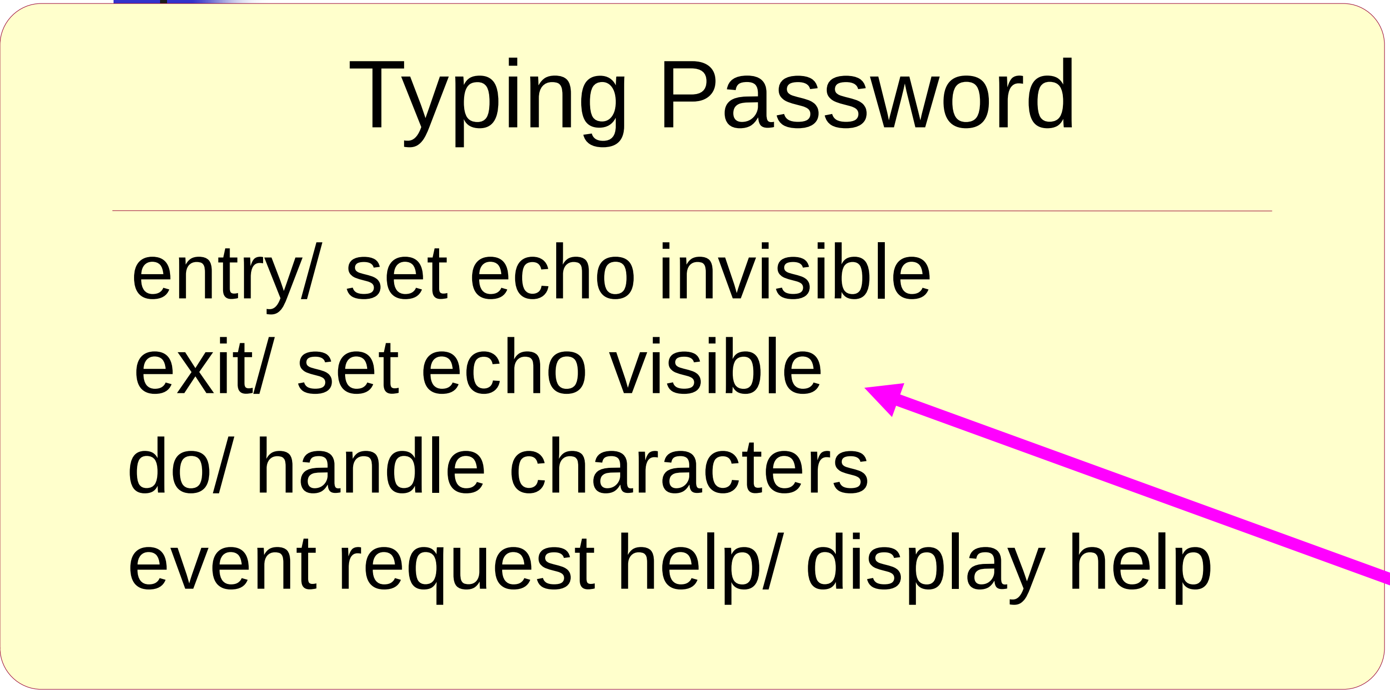
exit/ set echo visible

do/ handle characters

event request help/ display help

- An action which is performed when the state is entered.
- entry/ is a keyword in states

Example of a state



Typing Password

entry/ set echo invisible

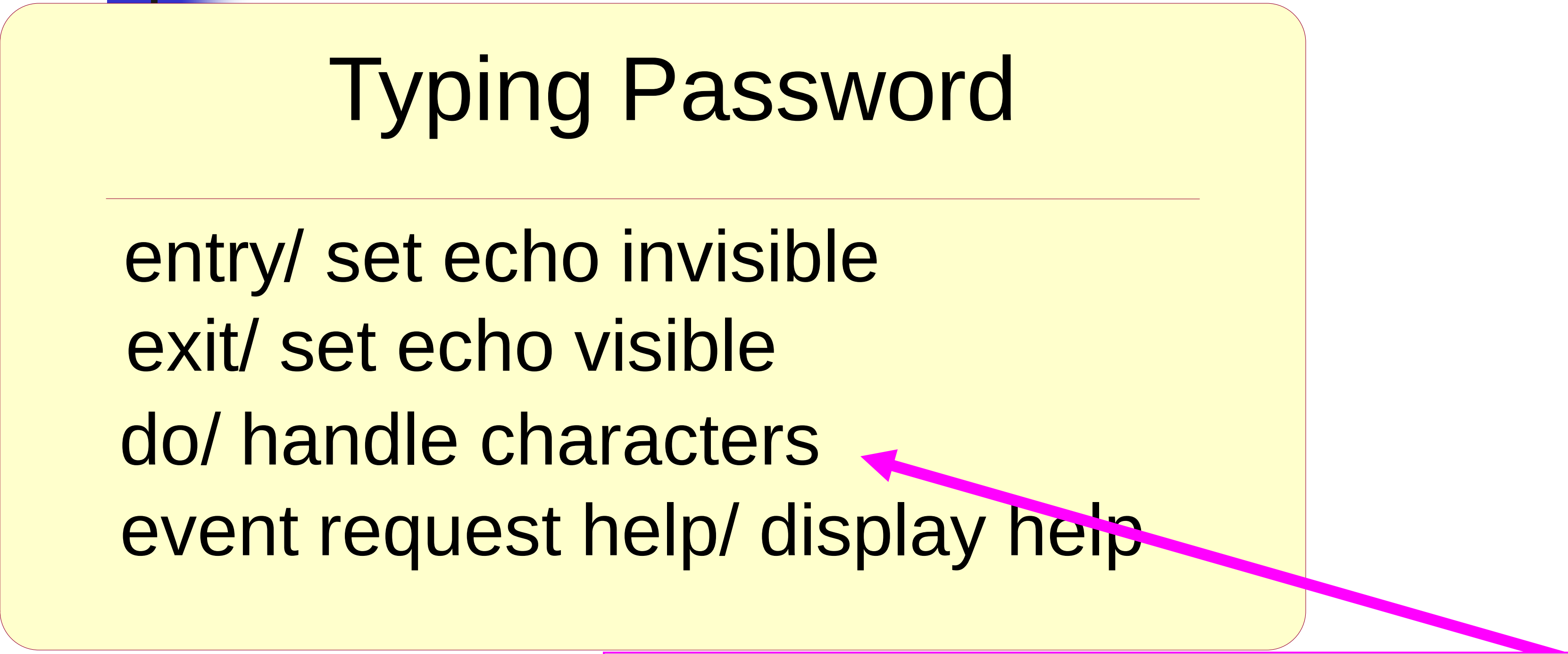
exit/ set echo visible

do/ handle characters

event request help/ display help

- An action which is performed on exit from the state.
- exit/ is a keyword in states

Example of a state

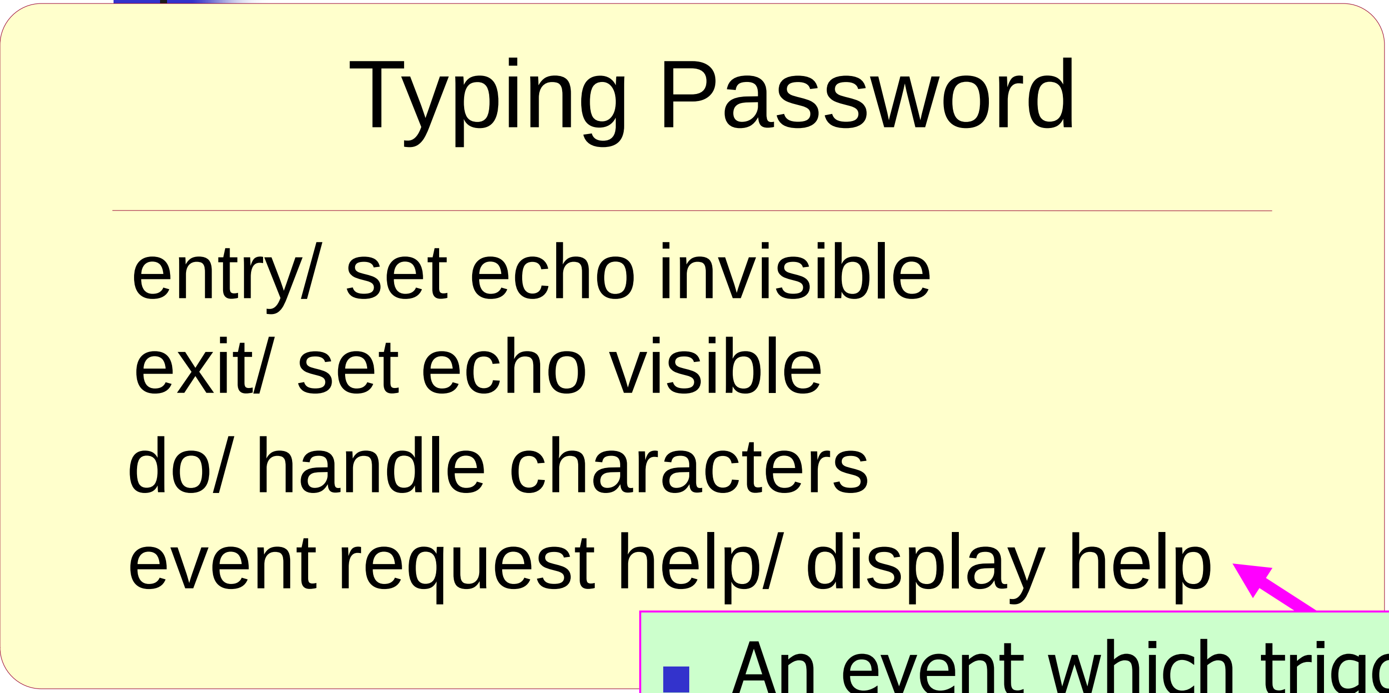


Typing Password

entry/ set echo invisible
exit/ set echo visible
do/ handle characters
event request help/ display help

- An action which is performed while the machine is in this state.
- do/ is a keyword.

Example of a state



Typing Password

entry/ set echo invisible

exit/ set echo visible

do/ handle characters

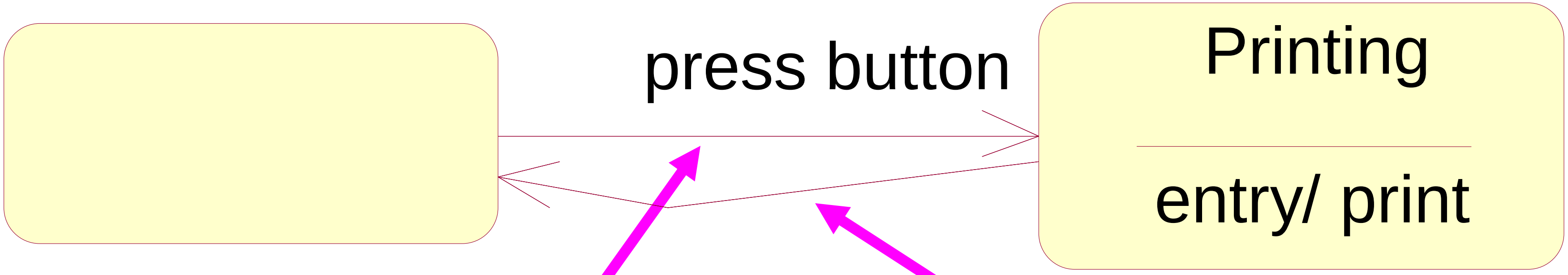
event request help/ display help

- An event which triggers an *internal* transition. The object is not leaving its state while the event is dealt with.

Transitions

- A transition is a change of an object from one state (the *source state*) to another (the *target state*).
- A transition is triggered when an event of interest of the given object occurs.
- Alternatively, a transition may be executed unconditionally when the activity associated with the source state is complete (*triggerless transition)* .

Transitions

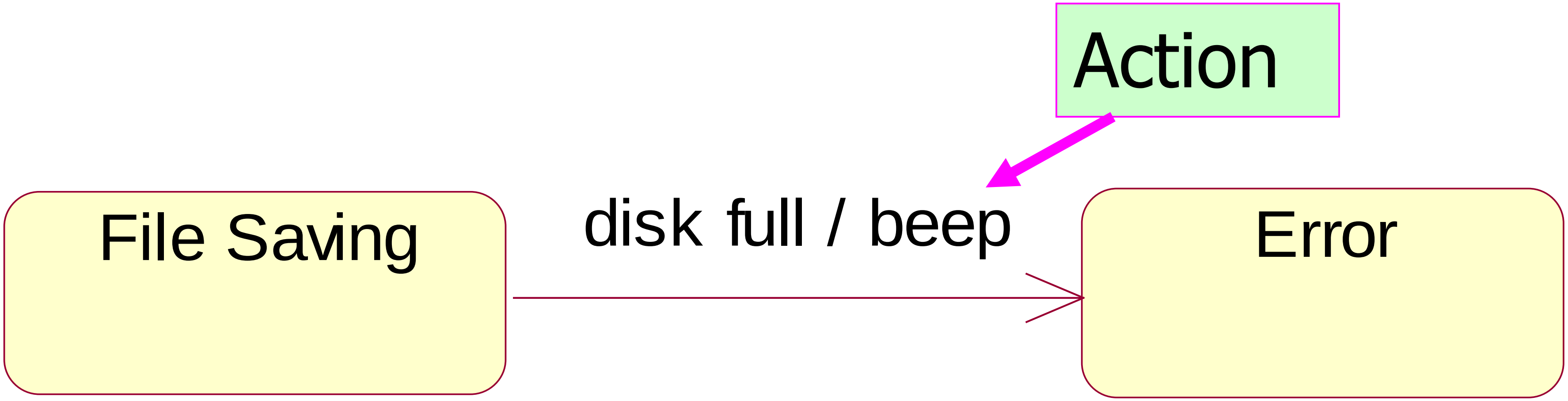


- Transition triggered by the press button event.

- Triggerless transition (executed when the receipt is printed)

Transitions and Actions

- There may be an *action* associated with a triggered transition. This action executes before the object enters the target state.



Action

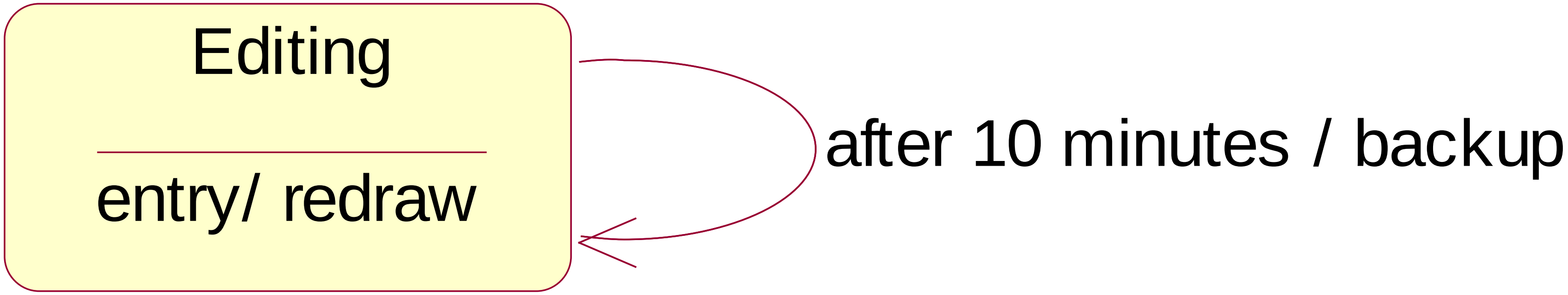
File Saving

disk full / beep

Error

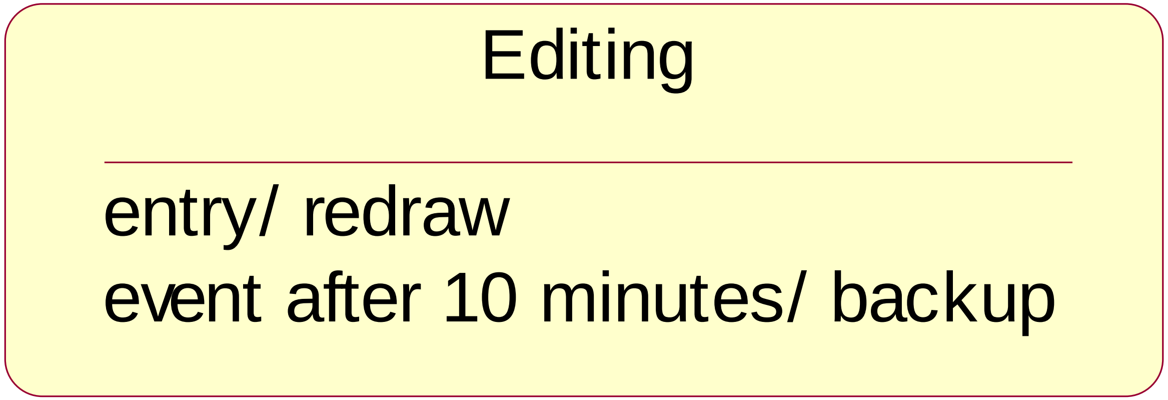
Self-transition

- A self-transition is a transition whose source state and target state are the same.
- Note that entry and exit actions are executed at a self-transition.



Example: redraw is executed each time at backup.

Example: not a self-transition



Editing

entry/ redraw

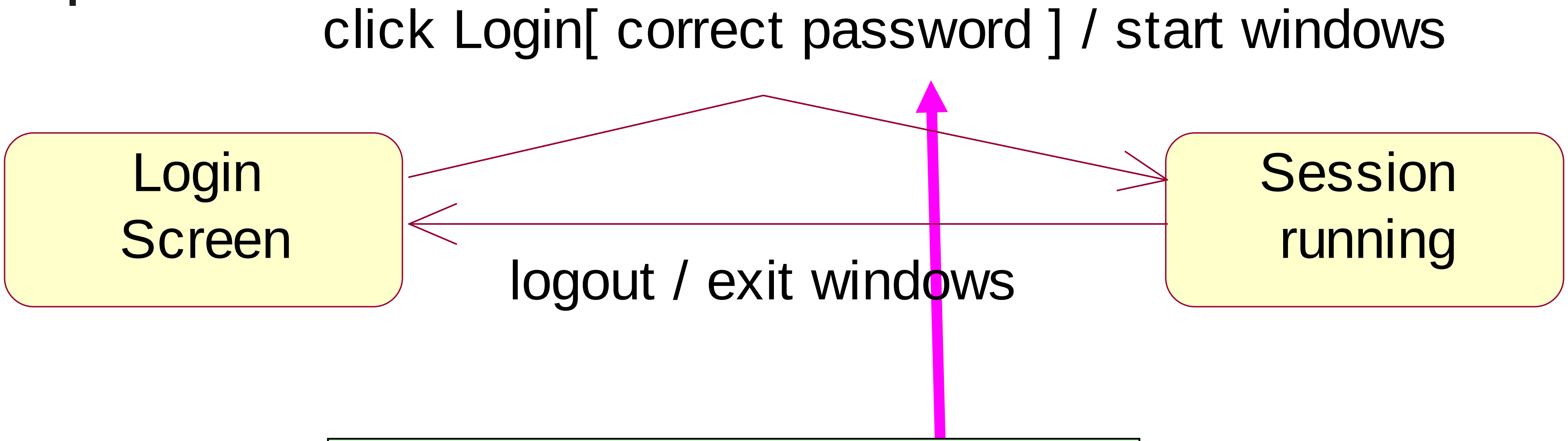
event after 10 minutes/ backup

redraw is not executed when the backup is made.

Guard conditions

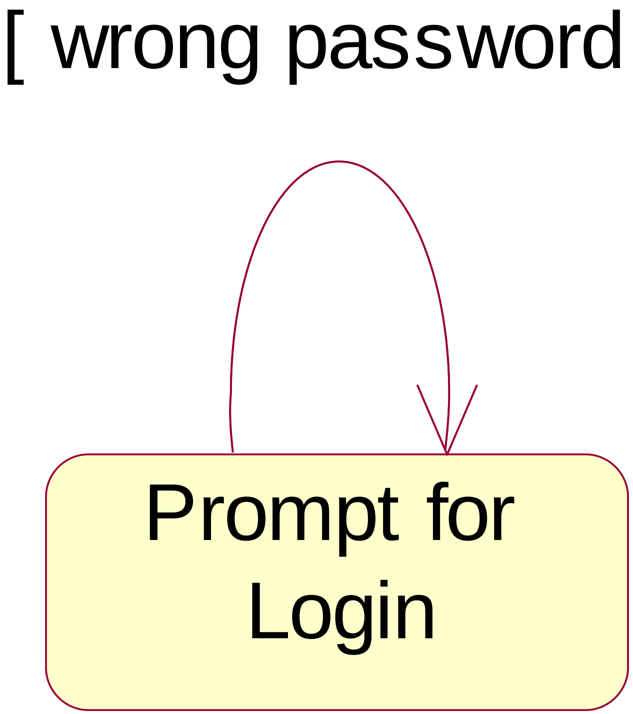
- A guard condition is a condition which must be true before a given transition is triggered.
- Notation of event with guard:
  - *eventName*[*guard condition*]
- Event, guard, and action:
  - *eventName*[*guard condition*]/action
- Guard only:
  - [guard condition]

Example for a guard condition



- Transition is triggered only when password is correct.

Example for an eventless guard condition

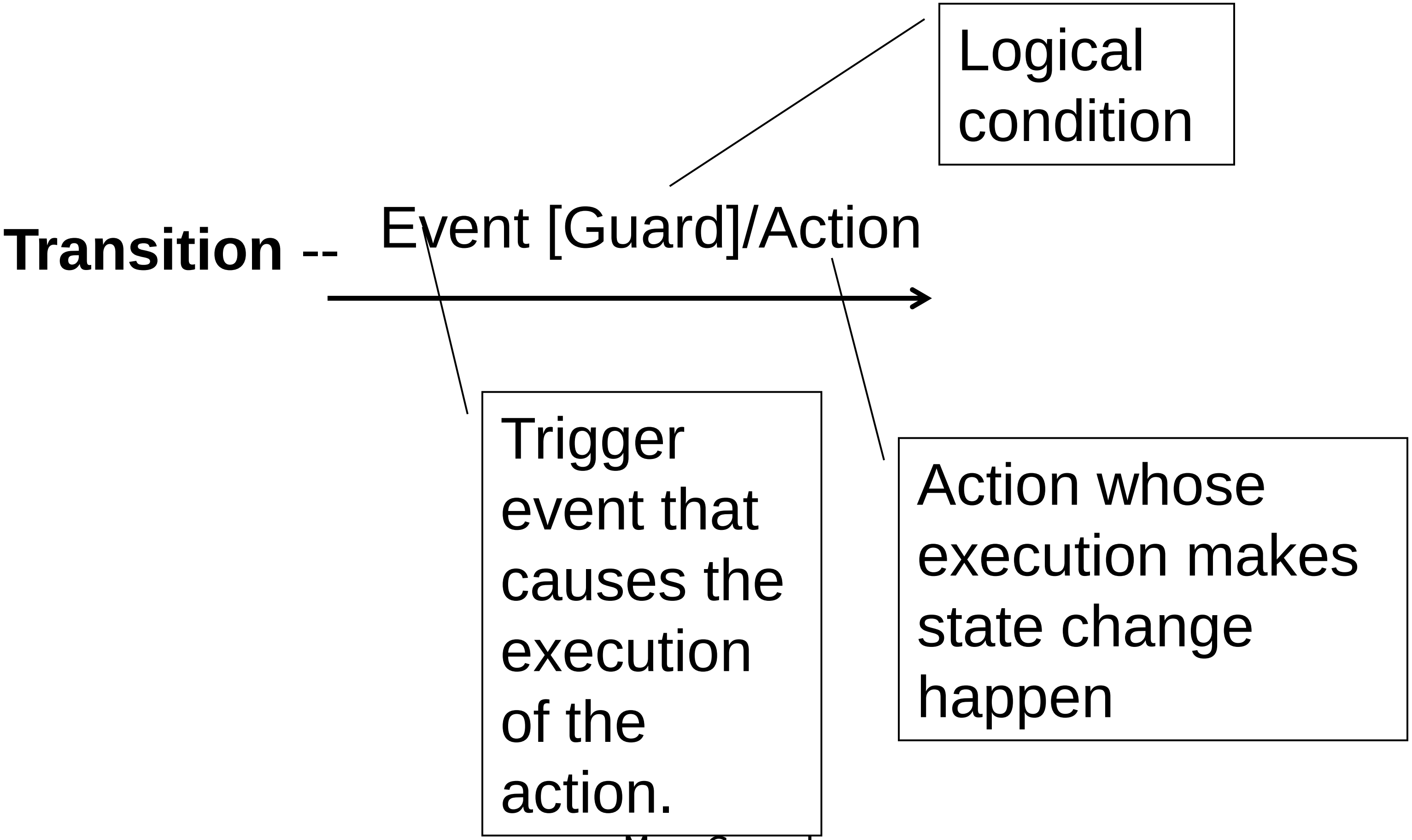


[wrong password]

Prompt for Login

- Guard condition with self transition.

Transition (Summary)



Transition --

Event [Guard]/Action

Logical condition

Trigger event that causes the execution of the action.

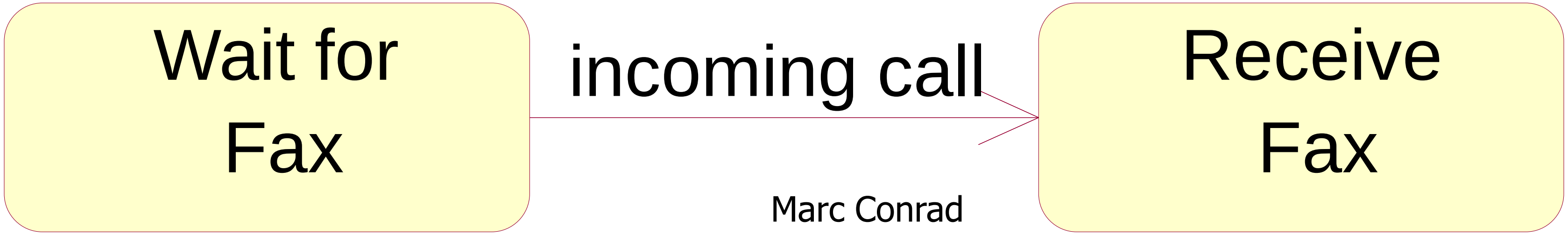
Action whose execution makes state change happen

What is an event?

- The following occurrences are considered as events in UML:
  - A *signal*
  - A *call event*
  - A *time event*
  - A *change event*

What is an event? SignalEvent

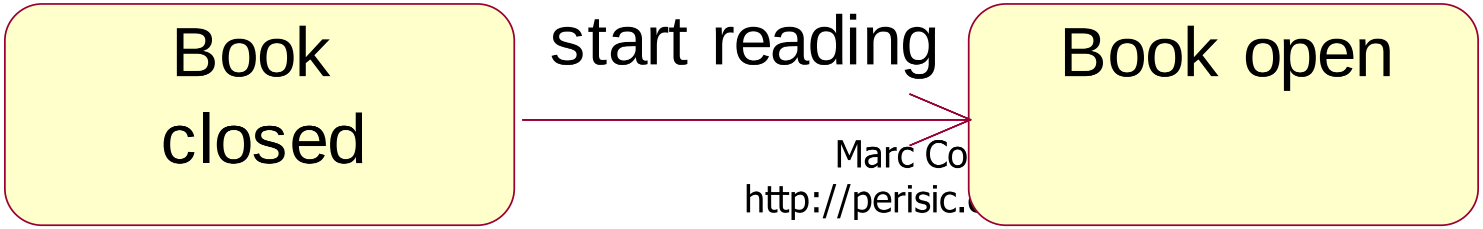
- A *signal*
  - An asynchoronous communication between objects.
- A *call event*
- A *time event*
- A *change event*



What is an event? Call Events

- A *signal*
- A *call event*
  - A synchronous communication where an object sends a message to another object.
- A *time event*
- A *change event*

- There is no notational difference between Signal Events and Call Events in Statechart diagrams.

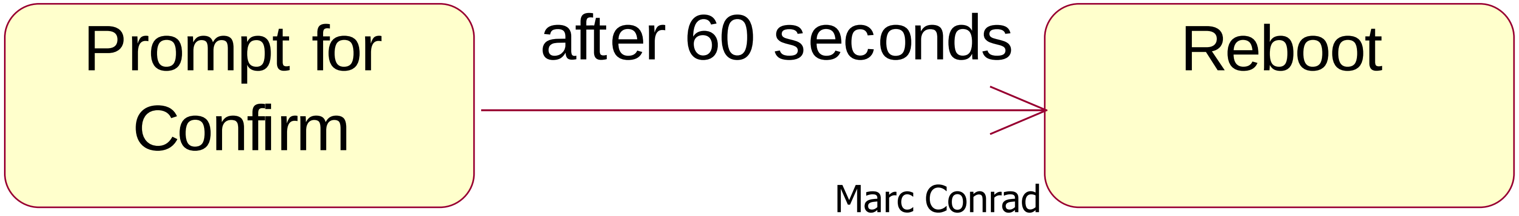

Book closed

start reading

Book open

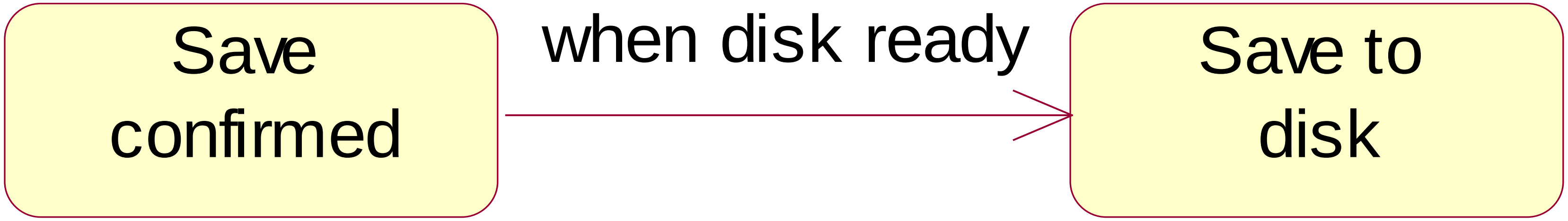
What is an event? Time Events

- A *signal*
- A *call event*
- A *time event*
  - An event that occurs after a specified period of time. A time event is expressed using the word “after” followed by a time expression.
- A *change event*



What is an event? Change Events

- A *signal*
- A *call event*
- A *time event*
- A *change event*
  - An event that occurs when some condition is satisfied. A change event is expressed using the keyword "when".

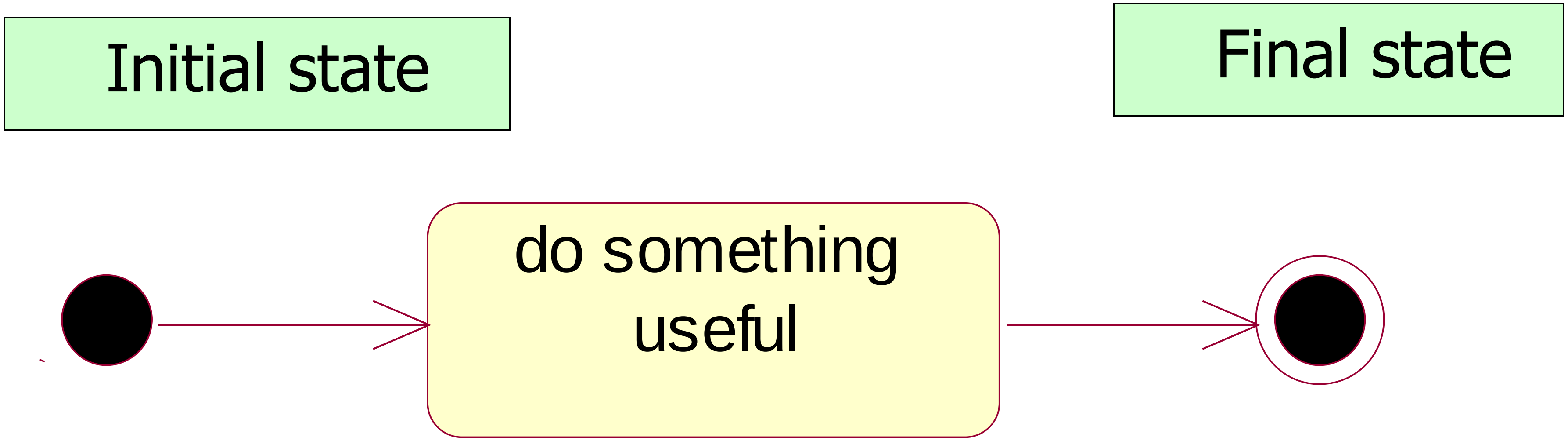


What is an action?

- An action is a procedural expression that is executed when the transition fires.
- The action must be executed entirely before any other actions are considered. It is not interruptible.

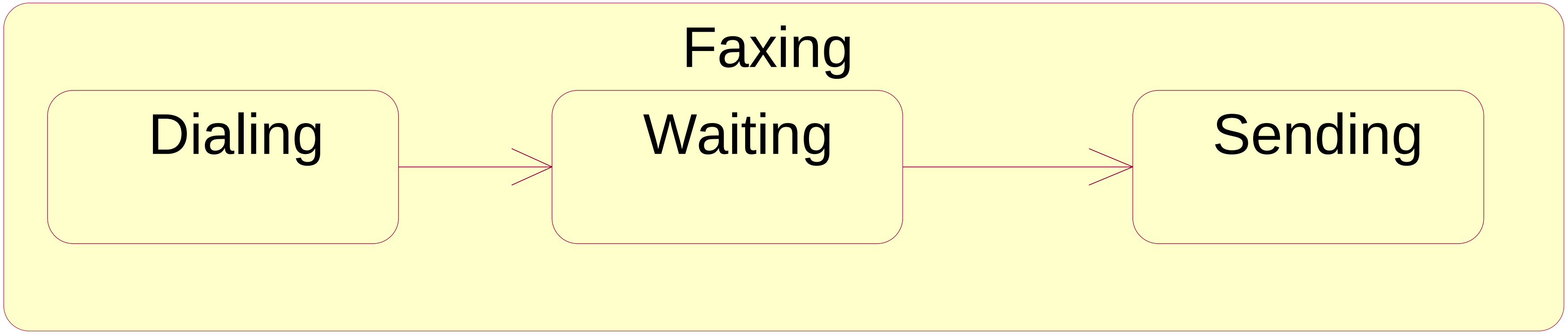
Initial state and final state

- There are special symbols for a state where the flow of control starts and a symbol for a final state.



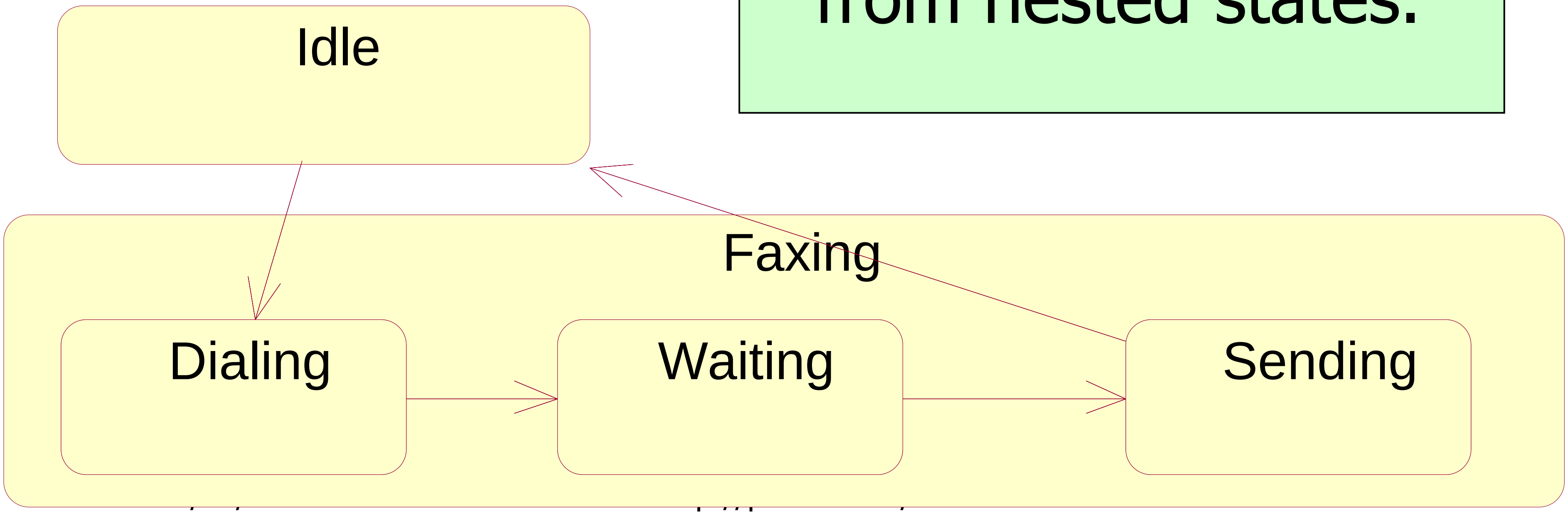
Substates

- A state diagram can be refined further by dividing a state into several substates. This will give much richer information about the system to be developed.



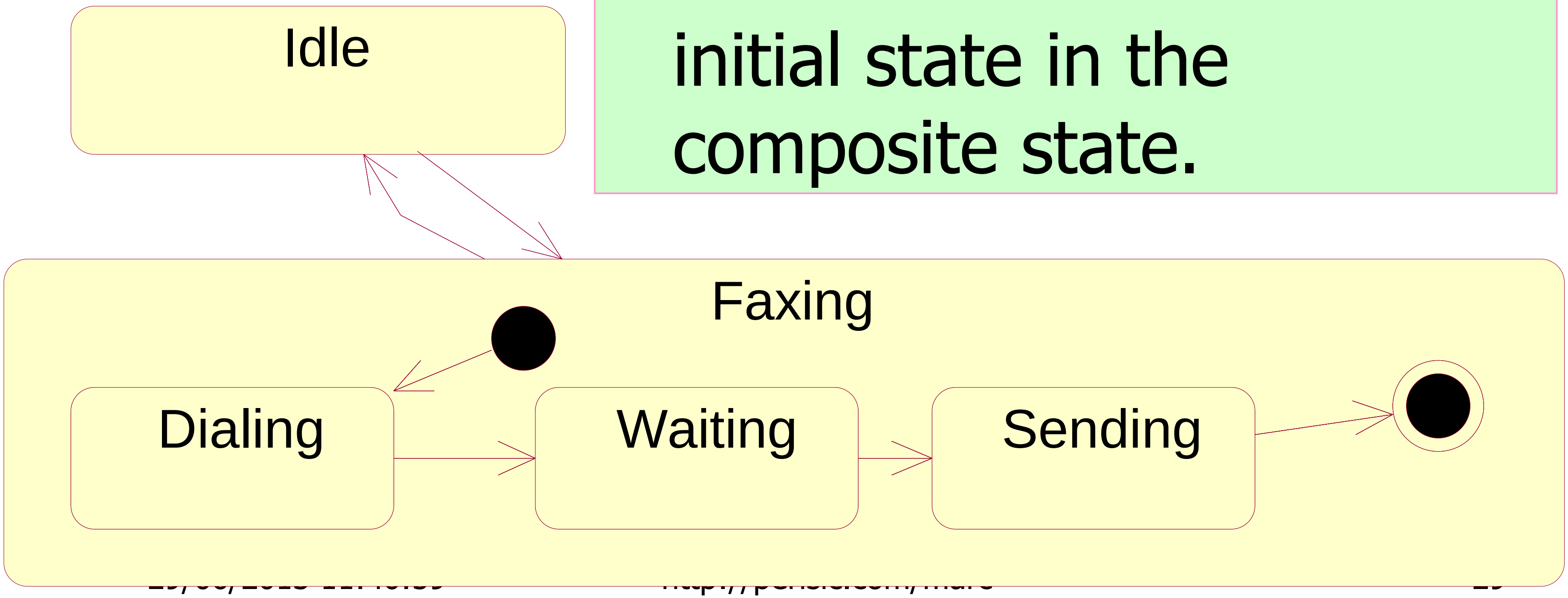
Substates and transitions (1)

- Transitions can be drawn directly to and from nested states.



Substates and transitions (2)

- When a transition ends in the composite state the flow continues with the initial state in the composite state.



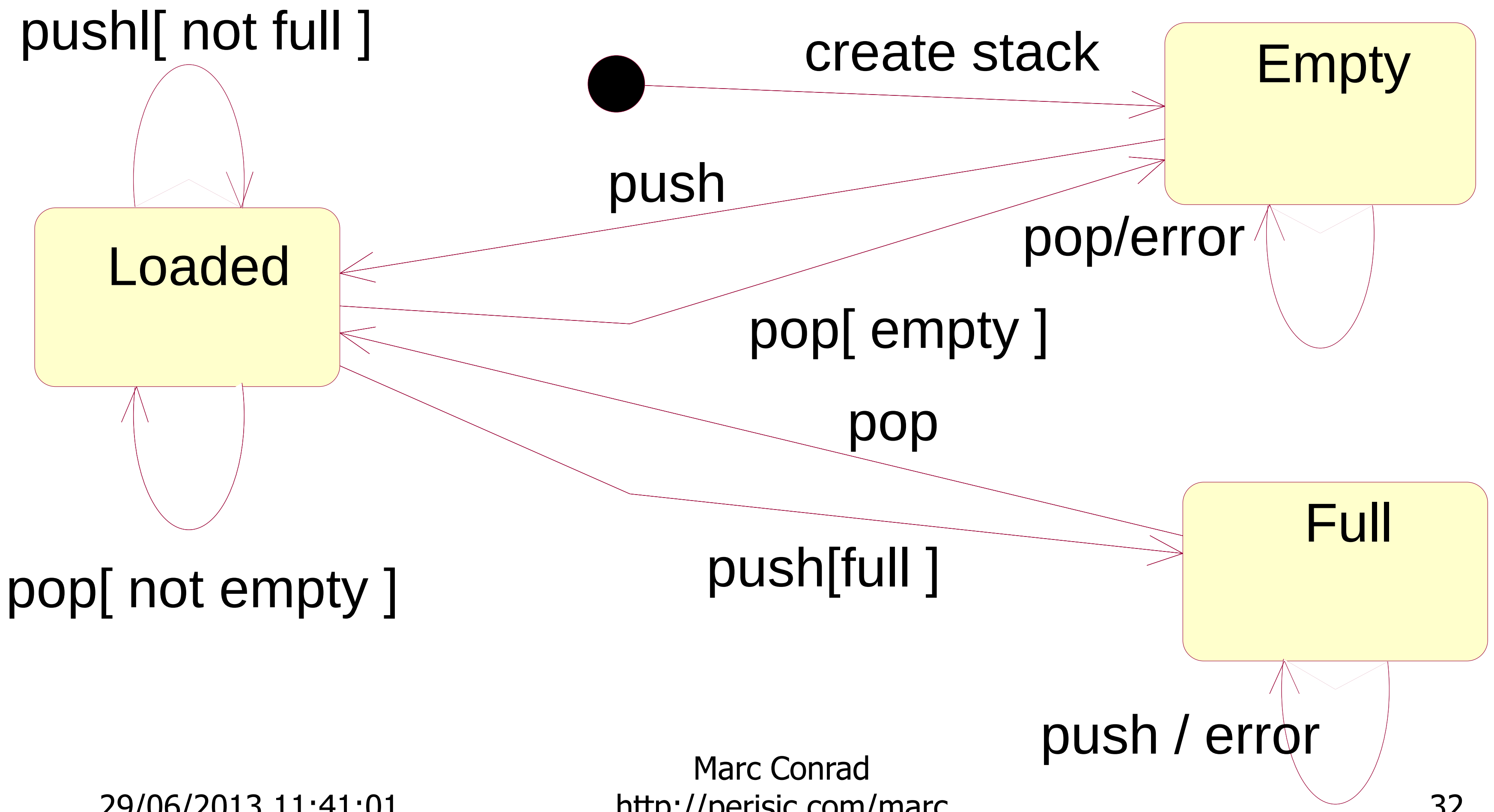
State Machines

- A statechart diagram shows an object’s state machine:
    - The states that an object can assume during its life.
    - The events to which that object can respond.
    - The possible responses the object can make to those events.
    - The transitions that occur between the object’s states.

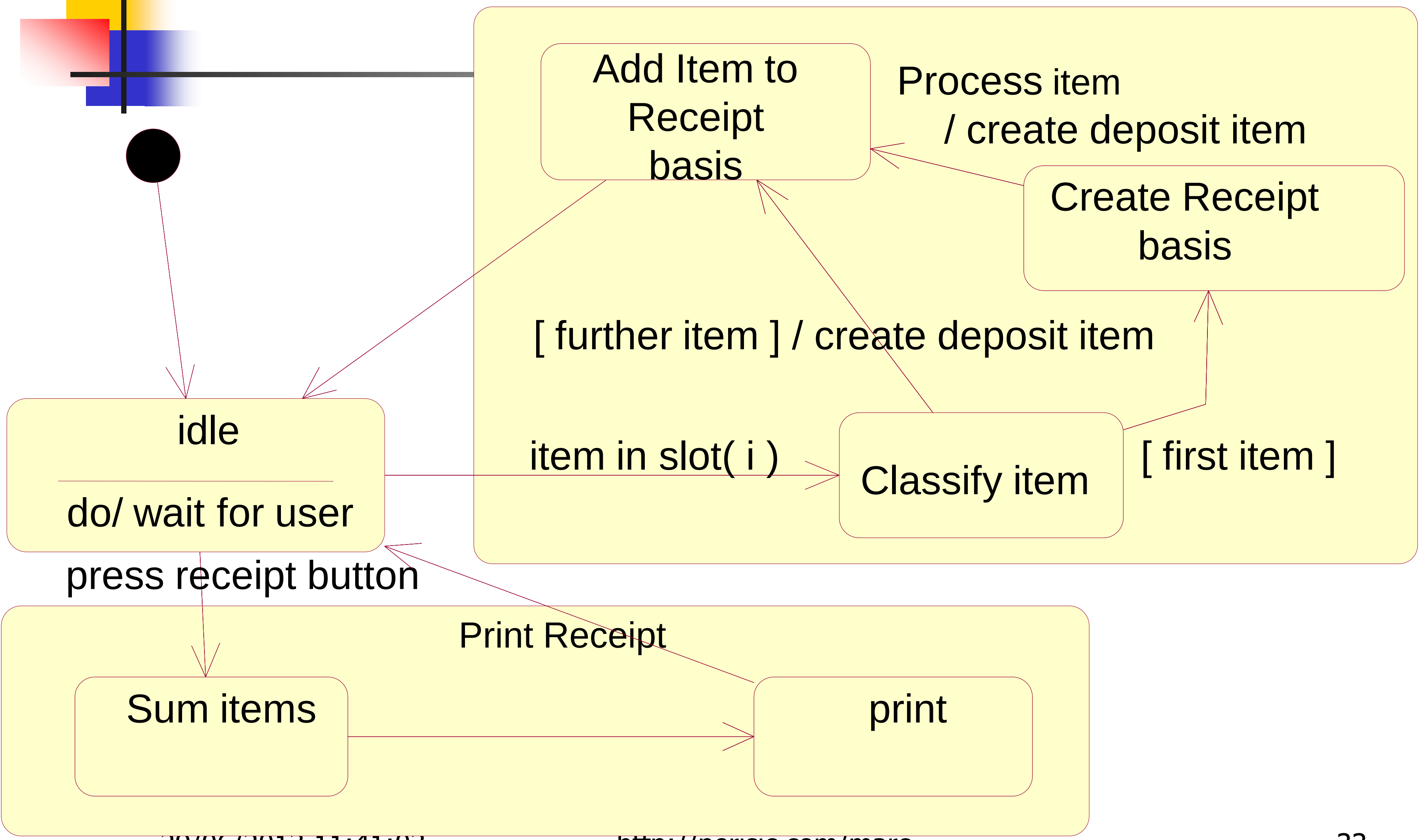
Statechart diagrams and state machines

- **Semantics**
  - A state machine is a graph of states and transitions that describes the response of an object of a given class to the receipt of outside stimuli. A state machine is attached to a class or a method
- **Notation**
  - A statechart diagram represents a state machine. The states are represented by state symbols and the transitions are represented by arrows connecting the state symbols. States may also contain subdiagrams by physical containment and tiling.

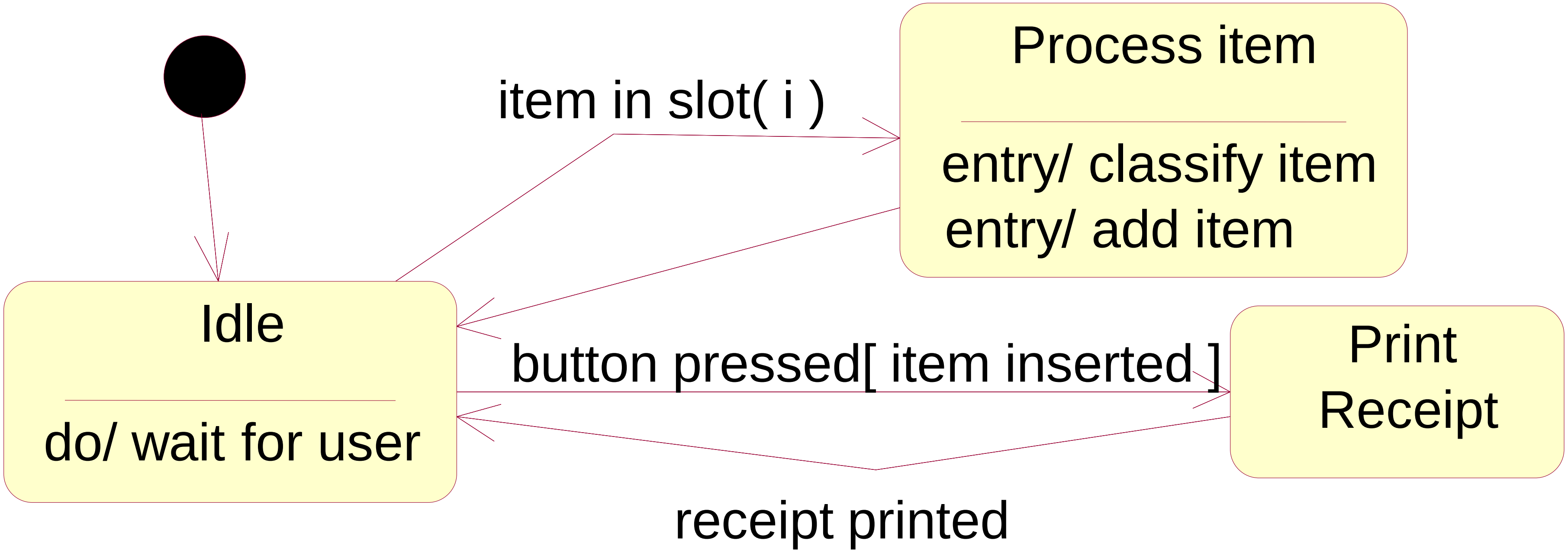
Example: The statechart diagram of a stack



33 Detailed Statechart diagram for the Deposit Item Receiver object, 1st version.



Statechart diagram for the Deposit Item Receiver object, 2nd version


item in slot(i)
Process item
entry/ classify item
entry/ add item
Idle
do/ wait for user
button pressed[item inserted]
Print Receipt
receipt printed