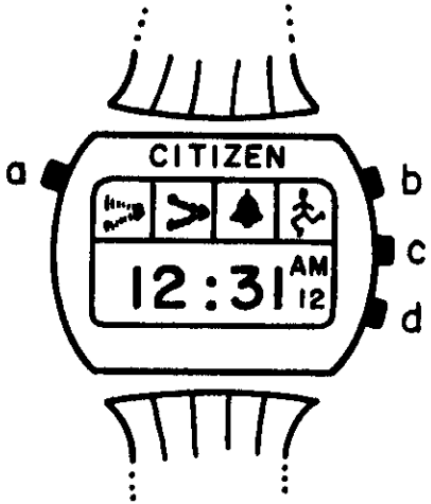
Statecharts

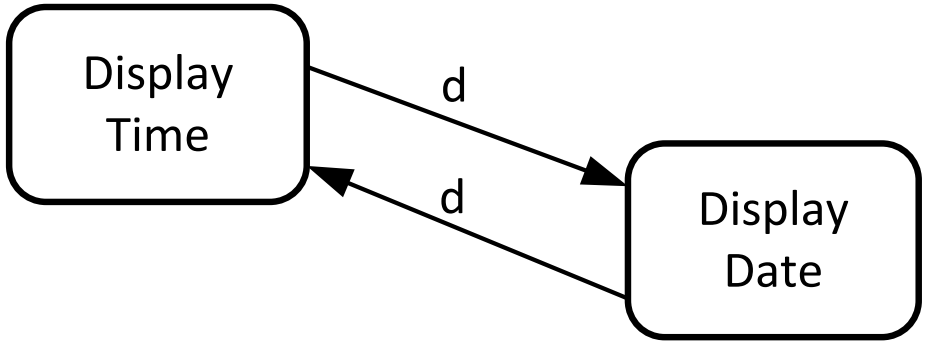
Agenda

- Introduction
- Concepts
  - States
  - Transitions
    - » Events
    - » Conditions
  - Hierarchy
  - Default state
  - History
  - Orthogonality
  - Conditional entrance
  - Actions
- Practice

What is it good for?

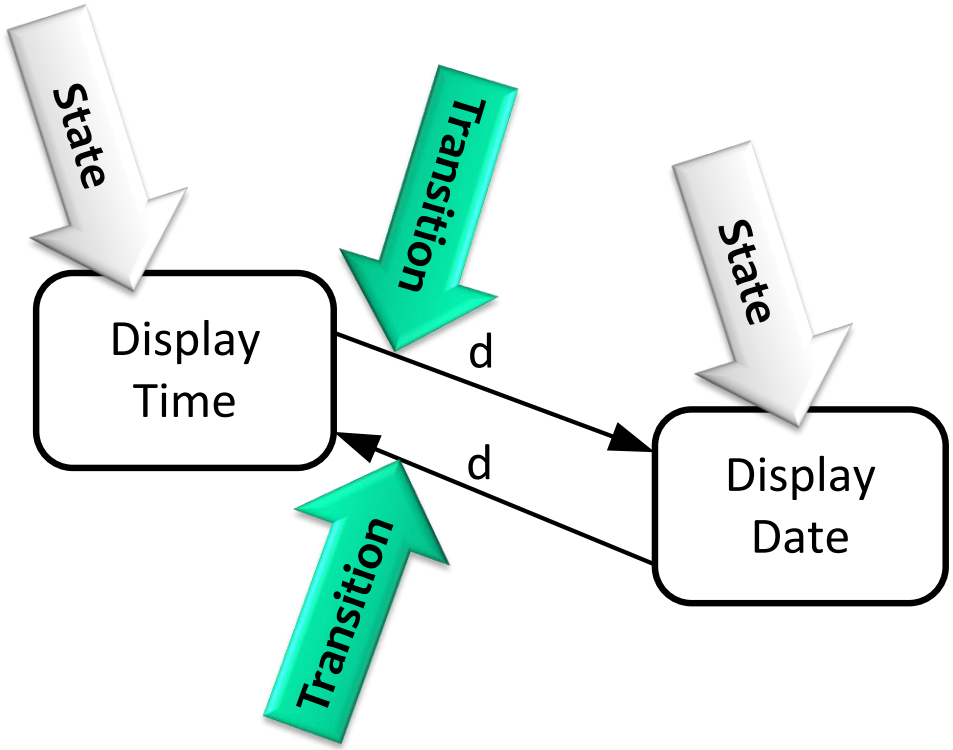
- State diagrams express **behavior**
  - What happens in a system, in a given condition





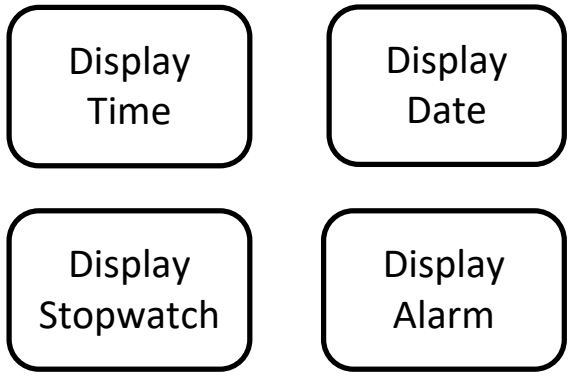
What is it good for?

- State diagrams express **behavior**
  - What happens in a system, in a given condition



State

- A configuration of the system (or entity)
- Each possible assignment of values to attributes is a “state”
- A recognizable situation



States

- Exists over an interval of time
- Represents an interval between successive events
- Can be a Macro state or a Micro state
- A Macro State is defined by another FSM containing Macro and Micro states
- A Micro state is a primitive state not defined any Further

State examples

- Room
  - Available
  - In use
  - Under renovation
- Student
  - Approved
  - Not approved
- Printer
  - Printing
  - Idle
  - Waiting for paper
- Traffic Light
  - Off
  - Green
  - Yellow
  - Red
- Telephone
  - Idle
  - Busy
  - Dial tone
  - Ringing
  - Connected

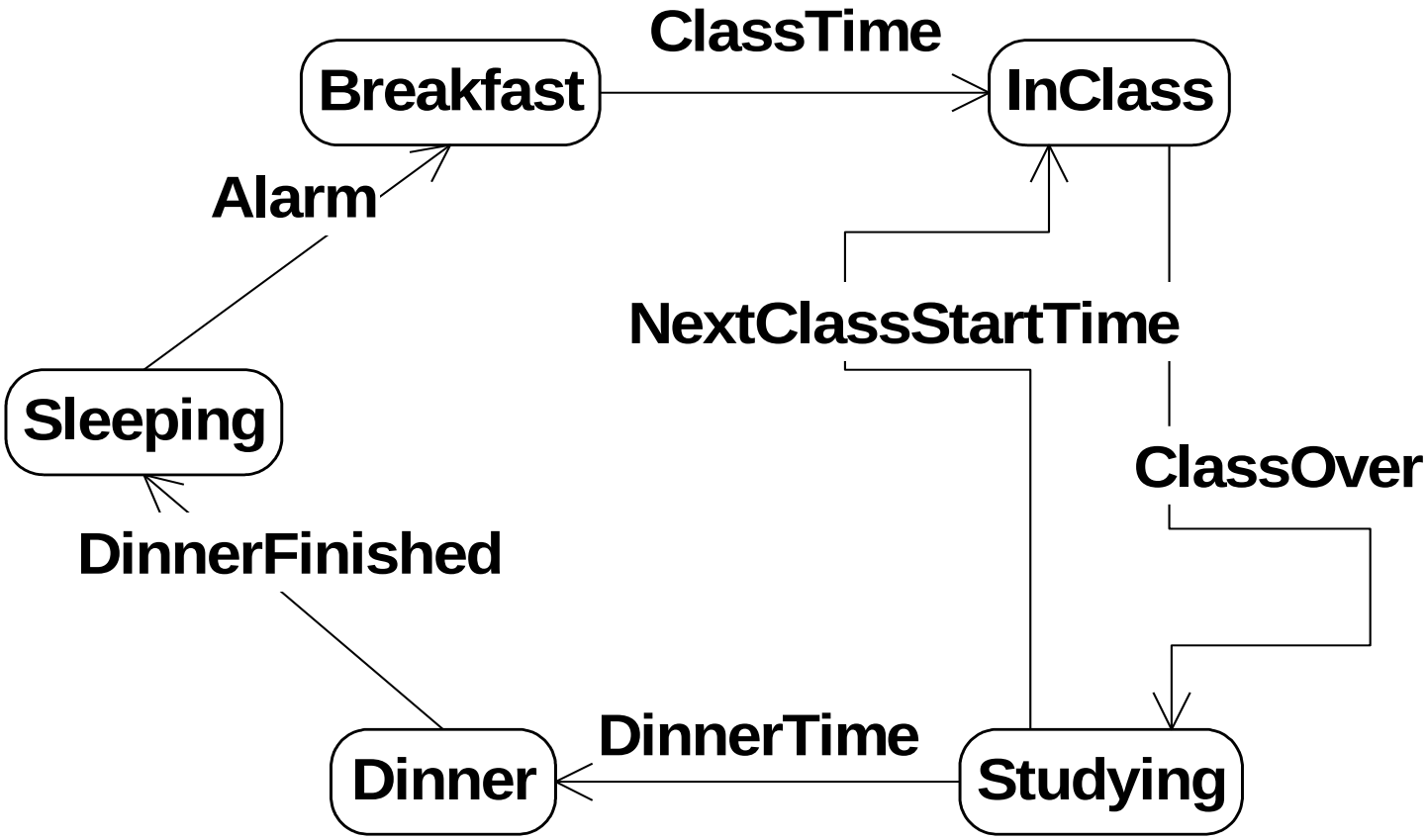
Transition

- Move from a state to another
- Event + Condition + Action
- Event
  - Something that happens
  - External or internal stimuli
  - May trigger a transition
- Condition
  - Something that must be true, or else the transition will not be performed
  - Also known as “guard”
- when event *a* occurs in state A, if condition C is true at the time, the system transfers to state B

Events

- Event–A discrete signal that happens at a point in time
  - –Also known as a stimulus
  - –Has no duration
- Two events
  - –May logically depend on each other
  - –E.g, ATM Card inserted before Pin # entered
- Two events
  - –May be independent of each other (they can occur independently)
  - –E.g., Cancel

Student



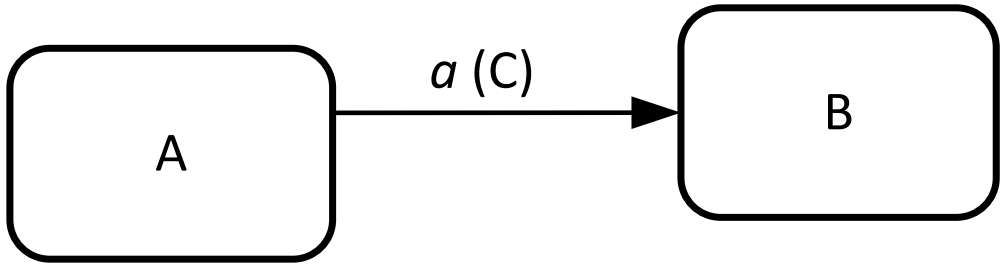
10

Events and Conditions

- State transition label
  - Event [Condition]
- Condition is a Boolean function
  - Conditions are optional on statecharts
  - Condition is true for finite period of time
- When event occurs, condition must be *true* for state transition to occur.

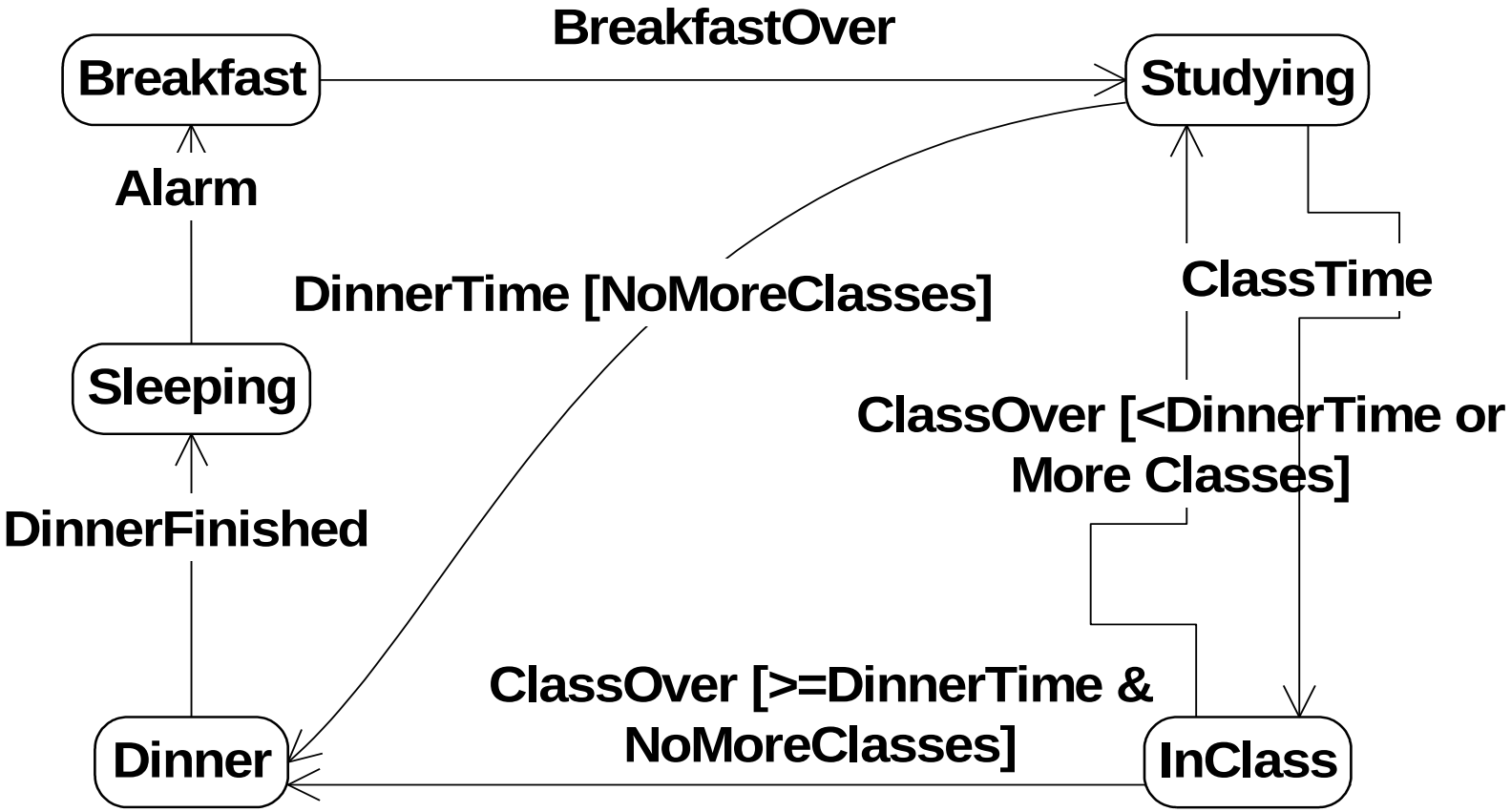
Transition

- Move from a state to another
- Event + Condition



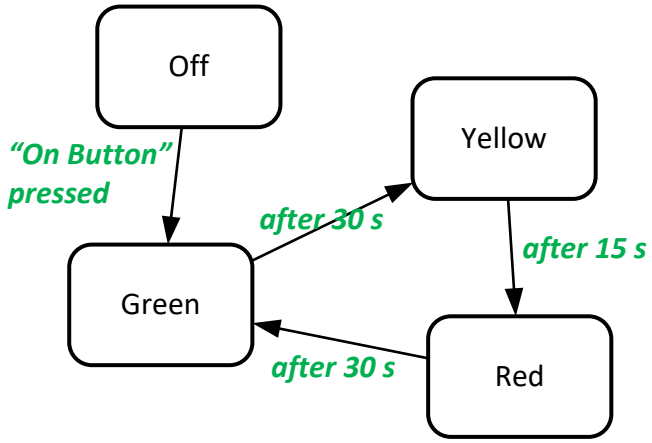
- when event *a* occurs in state A, if condition C is true at the time, the system transfers to state B

Student Example 2



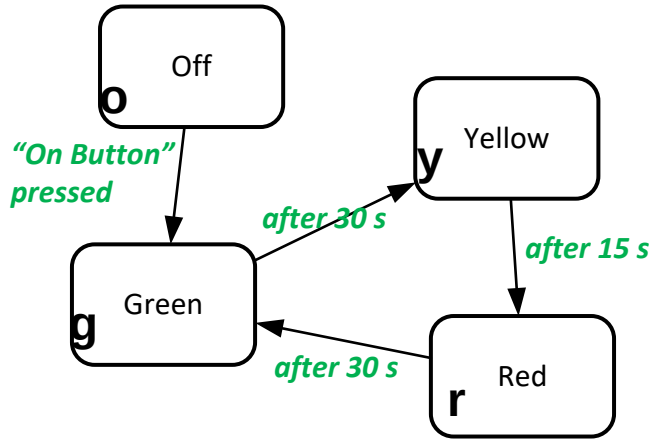
13

Transition example



Traffic Light

Tracing



Traffic Light

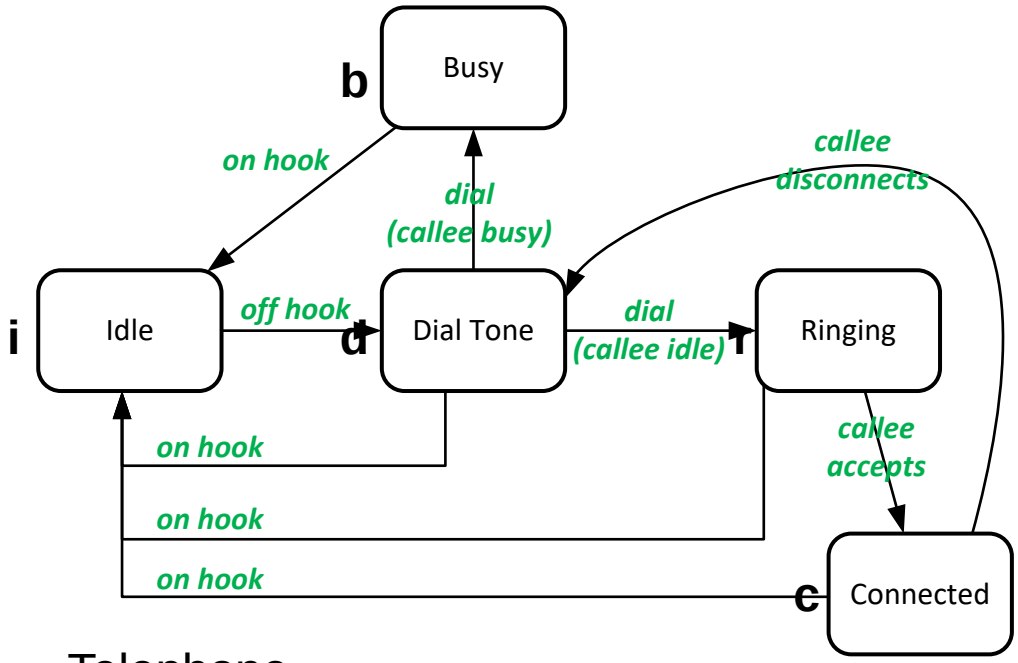
1. o g y r
2. o y g r
3. g y r
4. o g y r g y r
5. o g y r g y r g y r

o (g y r)+

Telephone example

- http://youtu.be/PuYPOC-gCGA?t=4m12s

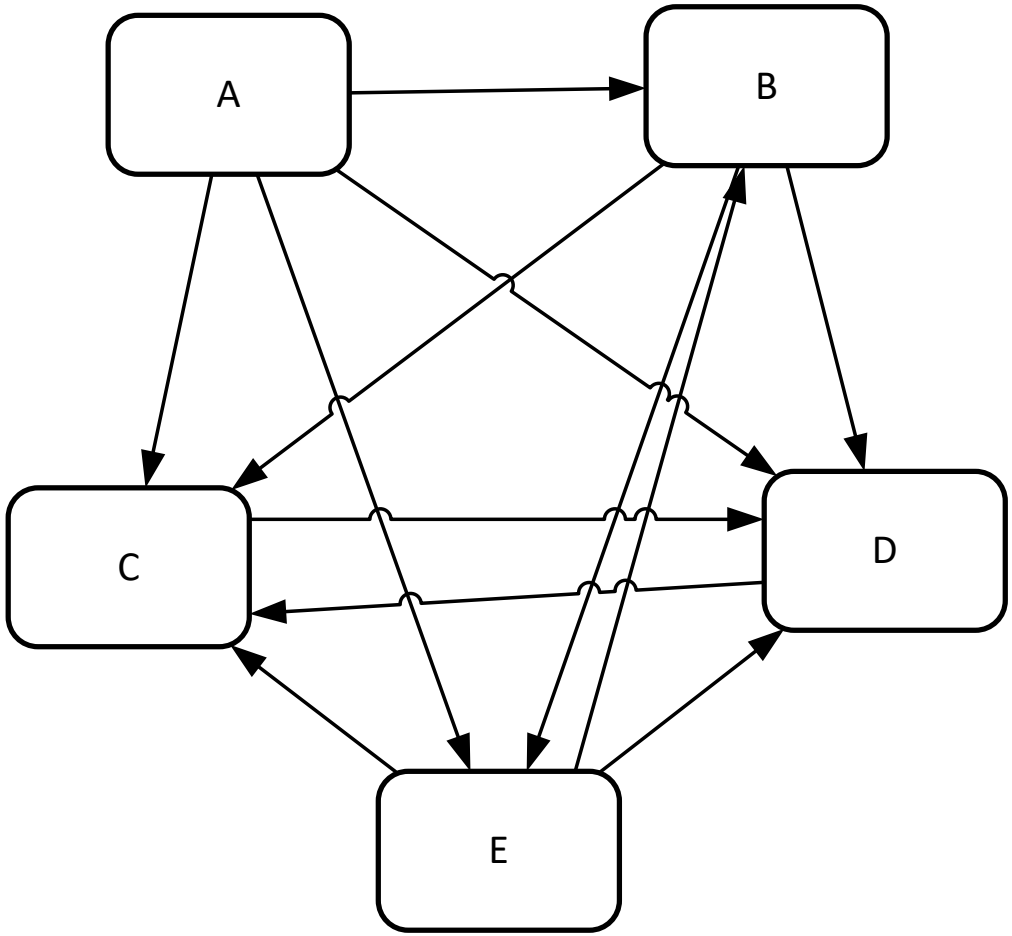
Tracing



Telephone

1. i d r c i
2. i d i
3. i d r d
4. i d b i
5. i d b i d b i
6. i d b i d b i d b i
7. i (d b i)*

Neat, but...



Statecharts to the rescue

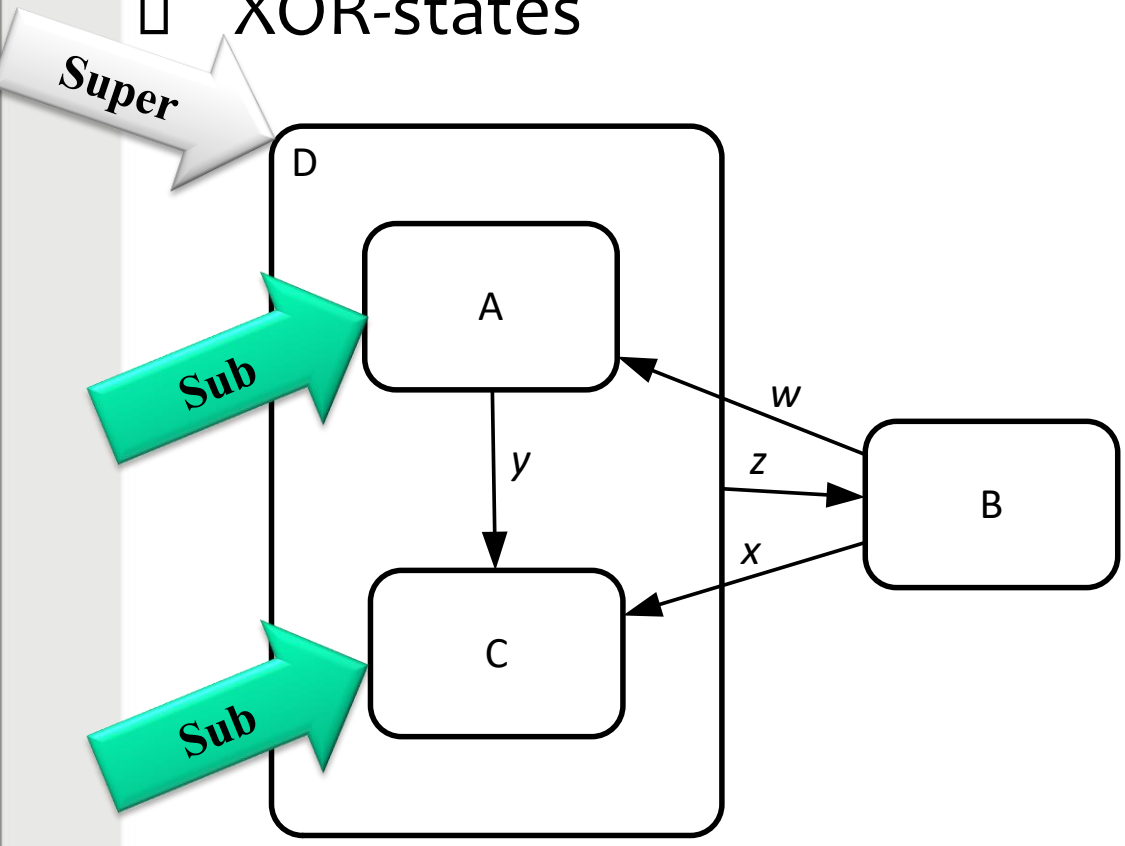
- Proposed by David Harel, in 1987 (*Statecharts: A visual formalism for complex systems*)
- State diagrams with additional mechanisms to:
  - handle complexity
  - improve expresiveness

Currently...

- Part of UML
- Tool support
  - Modeling, simulation and code generation
- Formalization
  - Reasoning

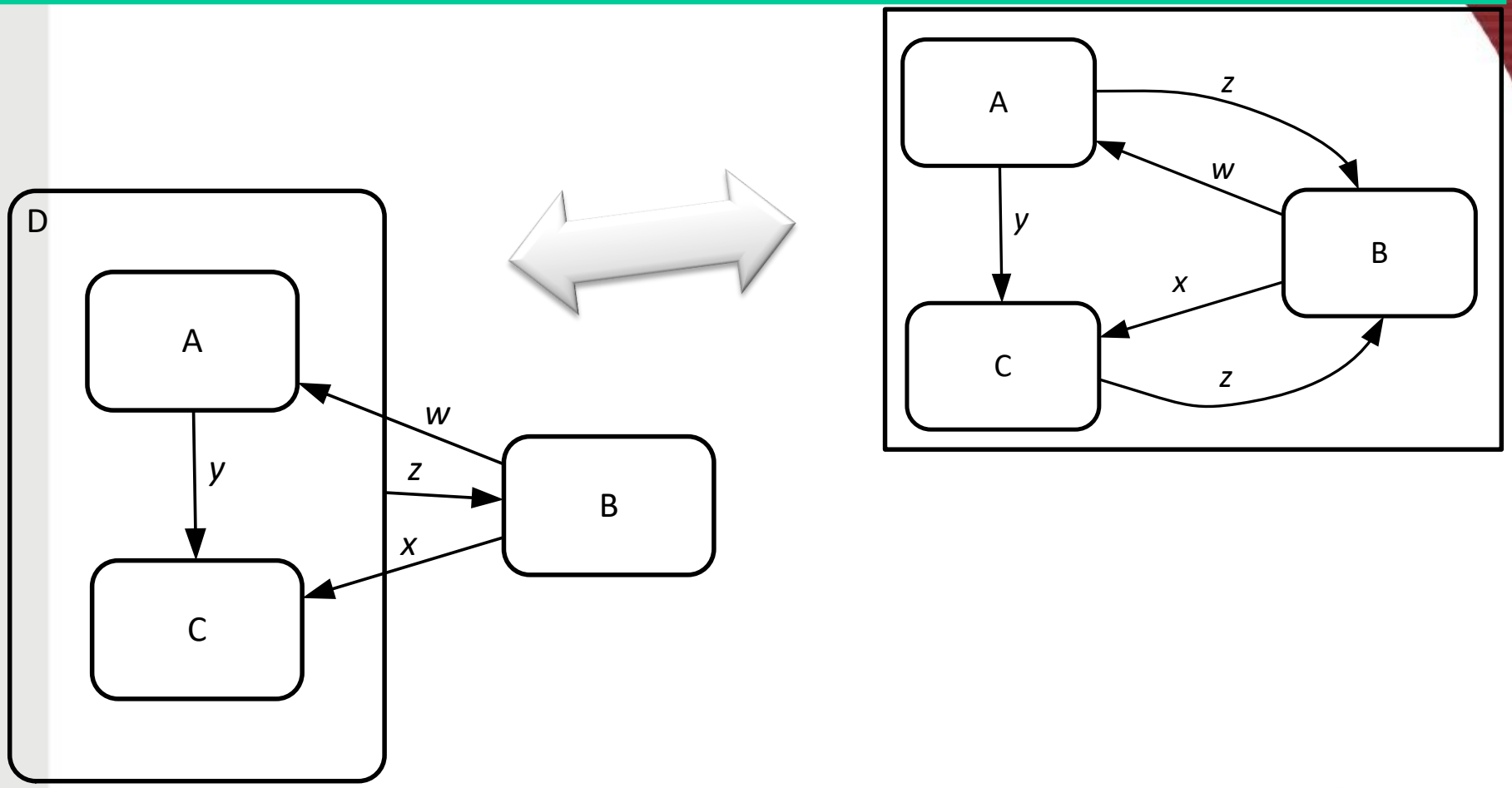
Hierarchy

- Super and sub states
- XOR-states

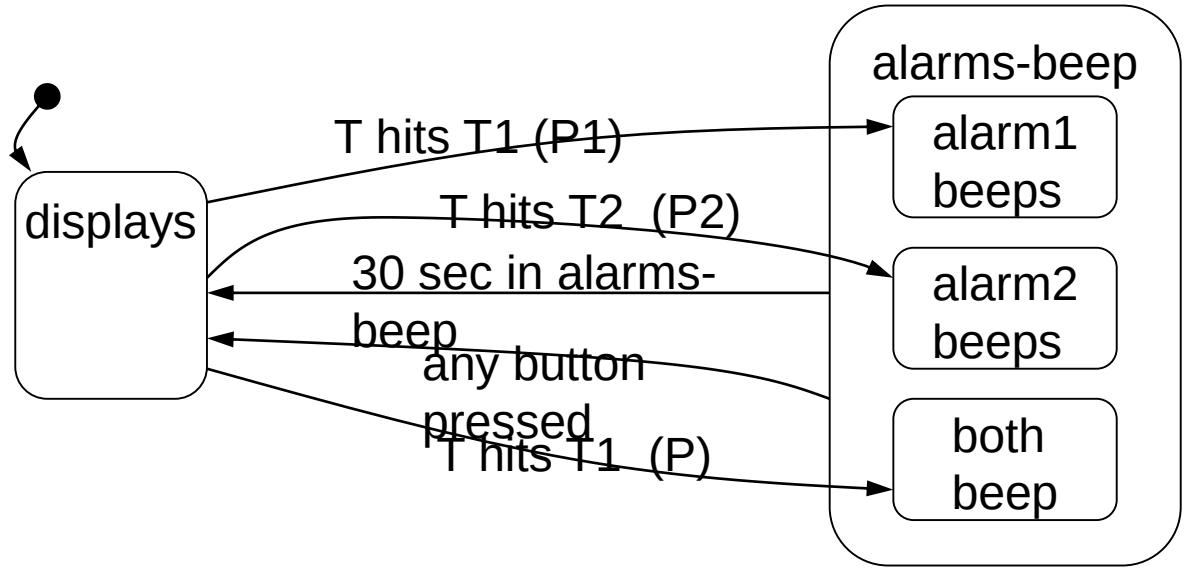


1. B A B
2. B C B
3. B A C B
4. B C A B
5. B A B A B A B
6. B A B C B A B
7. B C C B
8. B A A B

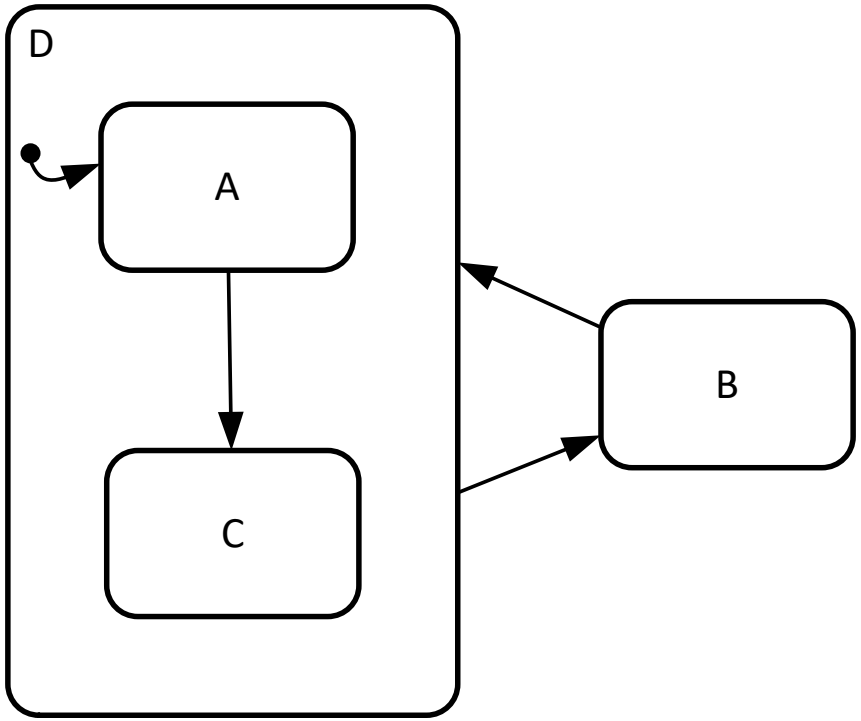
Hierarchy



Hierarchy

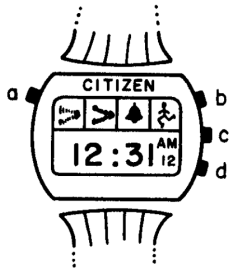

alarms-beep
displays
T hits T1 (P1)
T hits T2 (P2)
30 sec in alarms-beep
any button pressed
T hits T1 (P)
alarm1 beeps
alarm2 beeps
both beep

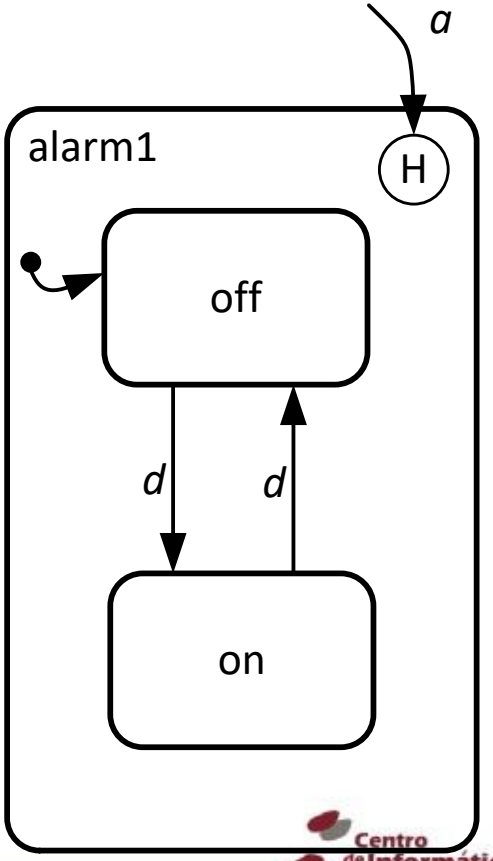
Default state



History

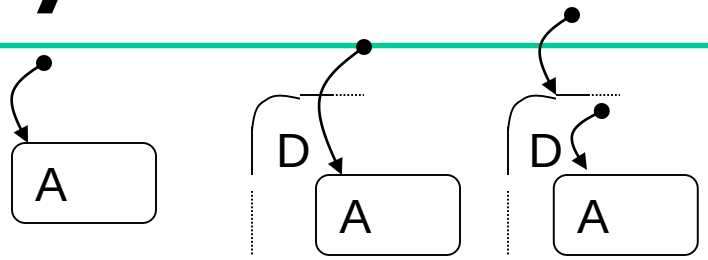
- Enter the last active sub-state on that state
- Example:
  - When displaying an alarm, it will be either on or off, according to whether it was on or off before

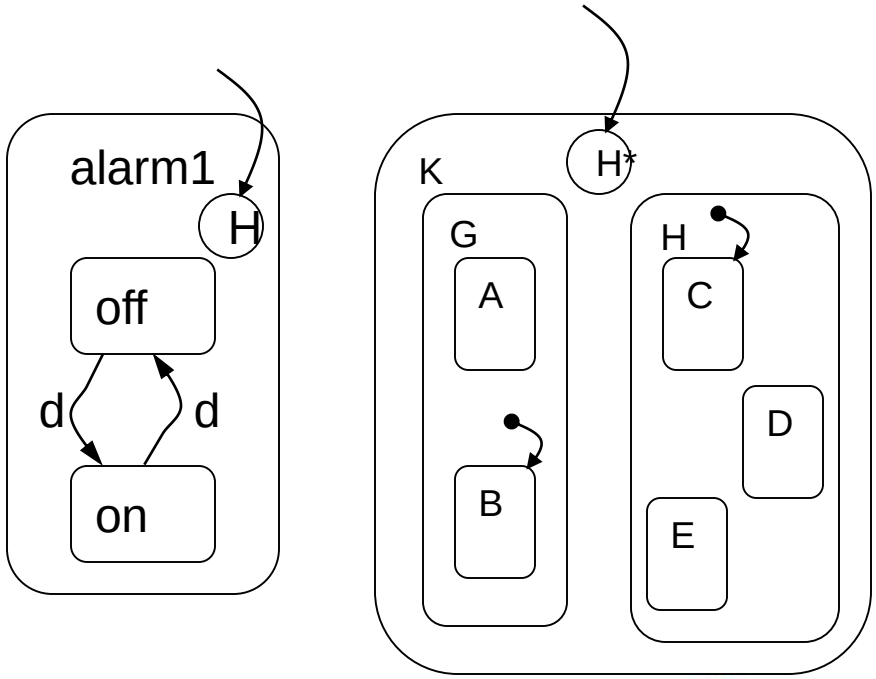




Default and History States

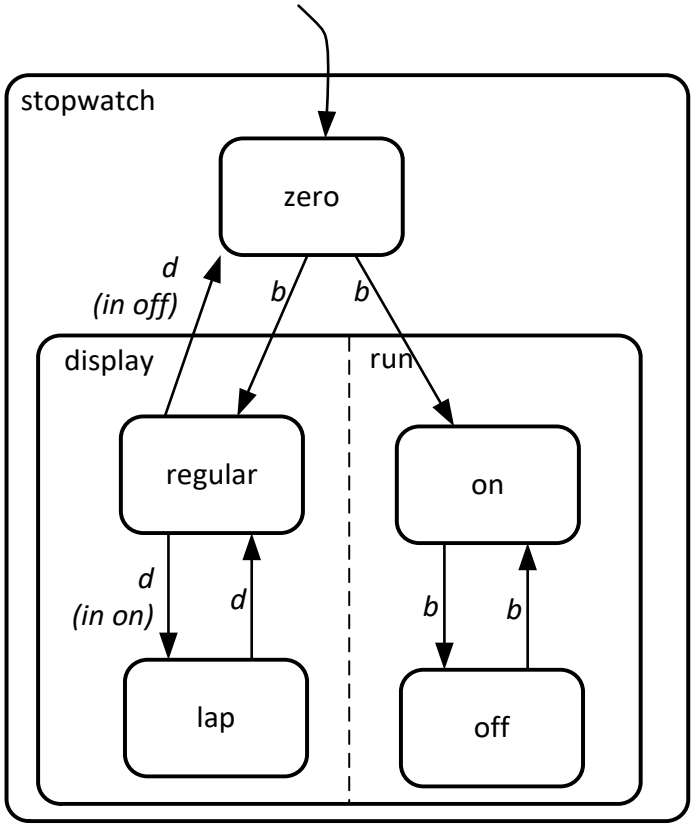
- Default states
  - Equivalent to start states for FSM
- History
  - (H) Record of the last state at the current level
  - (H*) Deep history stores last state at all levels current and below





Orthogonality

- Independence and concurrency
- AND-states

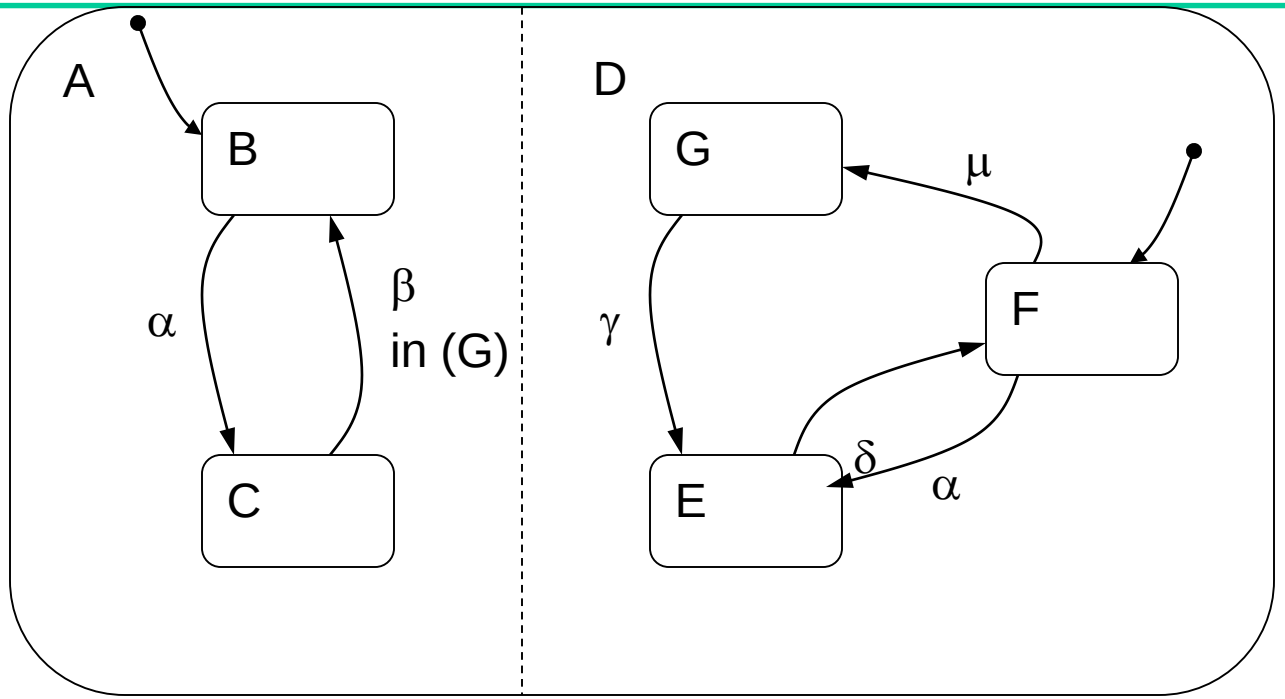


Orthogonality

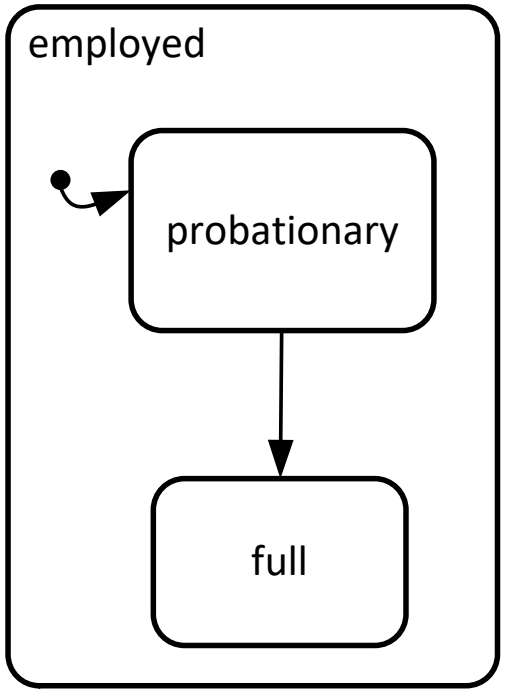
- AND combination of states
- Concurrency and synchronization
  - Simultaneous transitions in component states
- Independence
  - Independent transition in one of the component state
- Orthogonality = concurrency + independence
- Communication among states by common events

Orthogonality

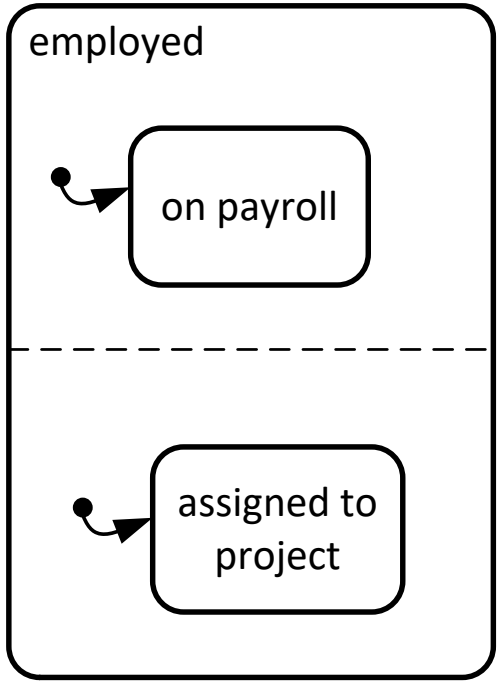
Y



Comparing hierarchy and orthogonality



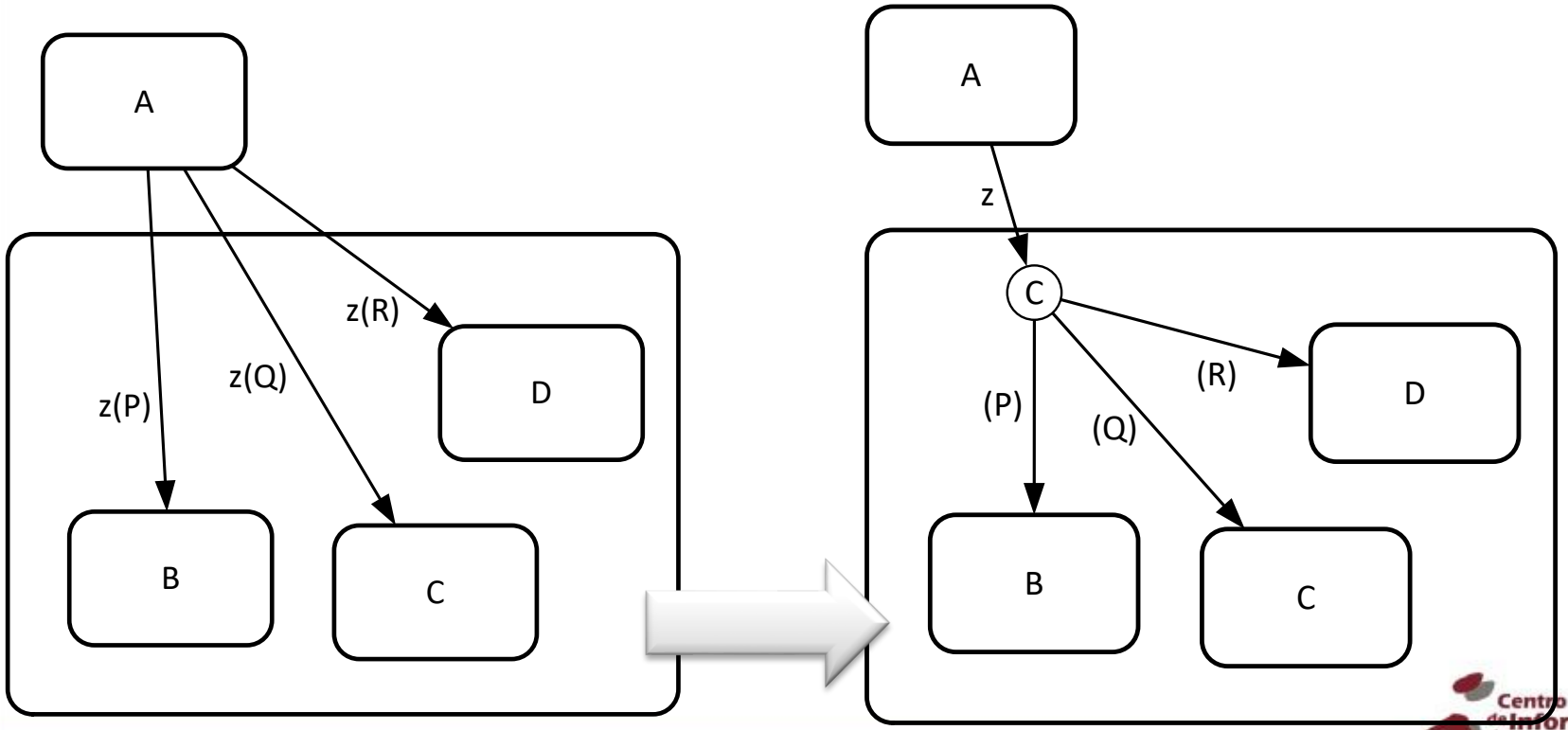
probationary XOR full



on payroll AND assigned to project

Conditional entrance

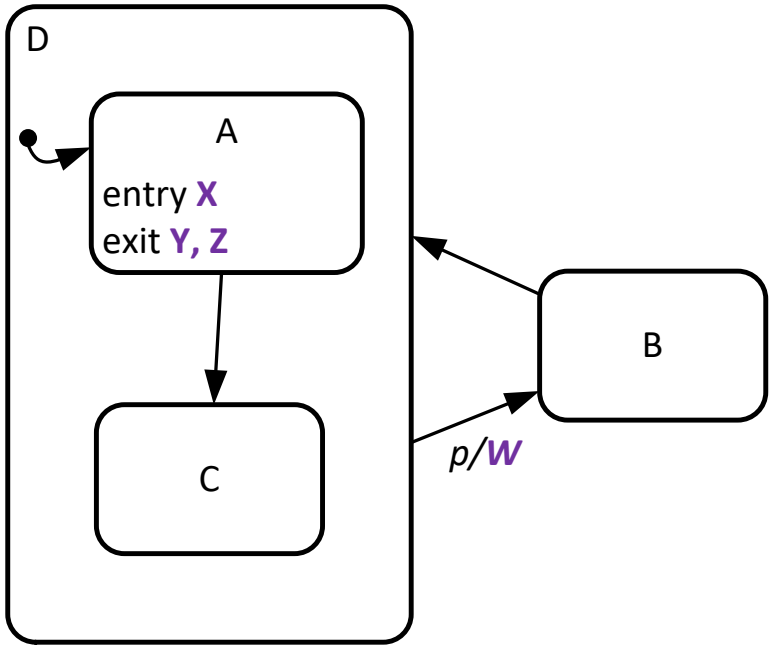
- Groups a set of transitions with a single event and multiple conditions



Actions

- Can be defined as state transition label
  - –Event / action(s)
  - –Event [condition] / action(s)
- Actions
  - –Executed as a result of state transition
  - –Executes instantaneously at state transition
  - –Terminates itself
- Entry Actions
  - Defined for a given state and executes on entry to this state from any state
- Exit Actions
  - Defined for a given state and executes on exit from this state to any state

Actions



Activities

- Activity
  - –Executes for duration of state
    - •Enable Activity on entry to state
    - •Disable Activity on exit from state
- Examples of activities
  - –Increase Speed
    - •Executes for duration of Accelerating state
  - –Maintain Speed
    - •Executes for duration of Cruising state

Example: StateChart with Activities, Auto Cruise Controller

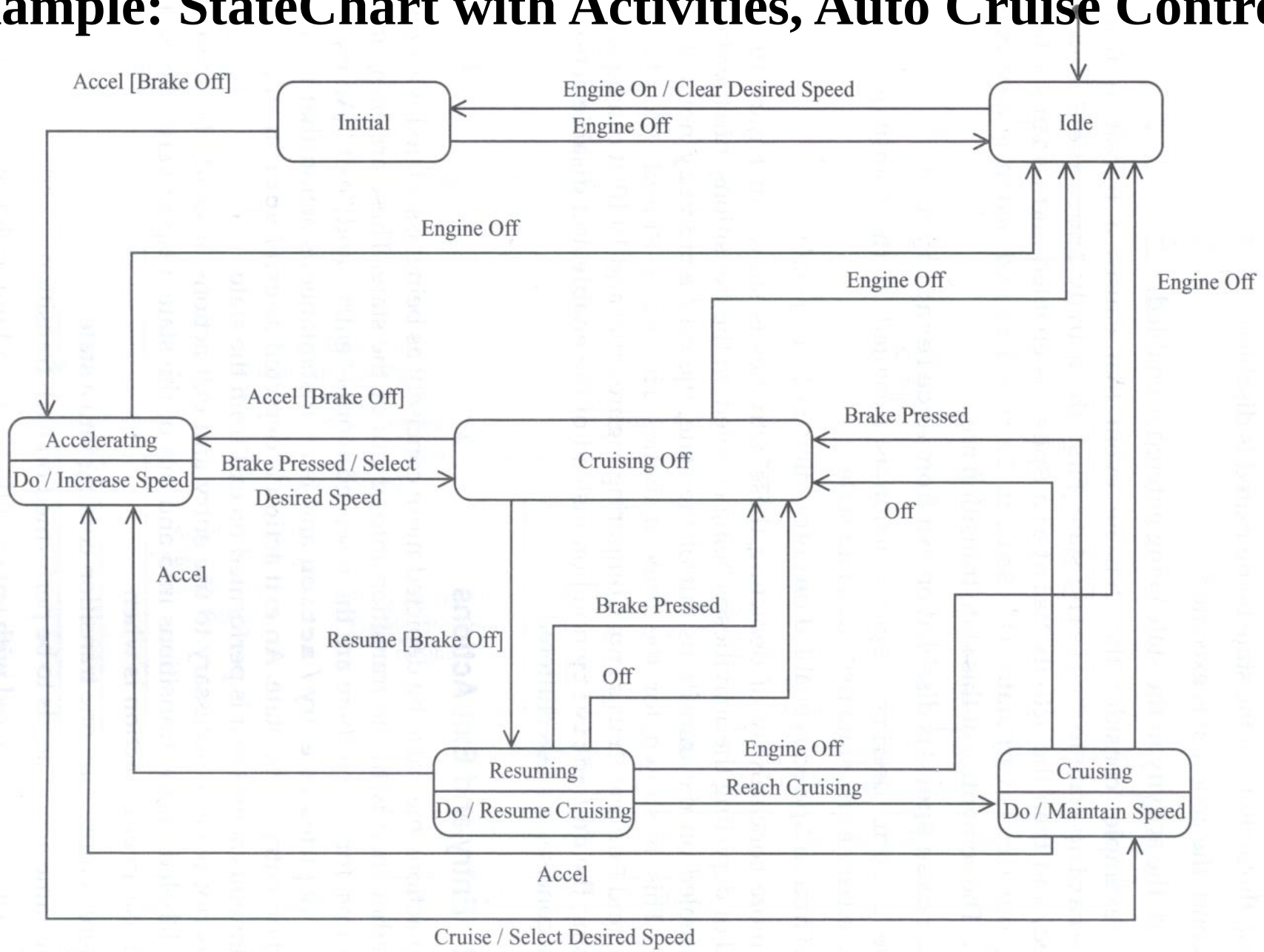
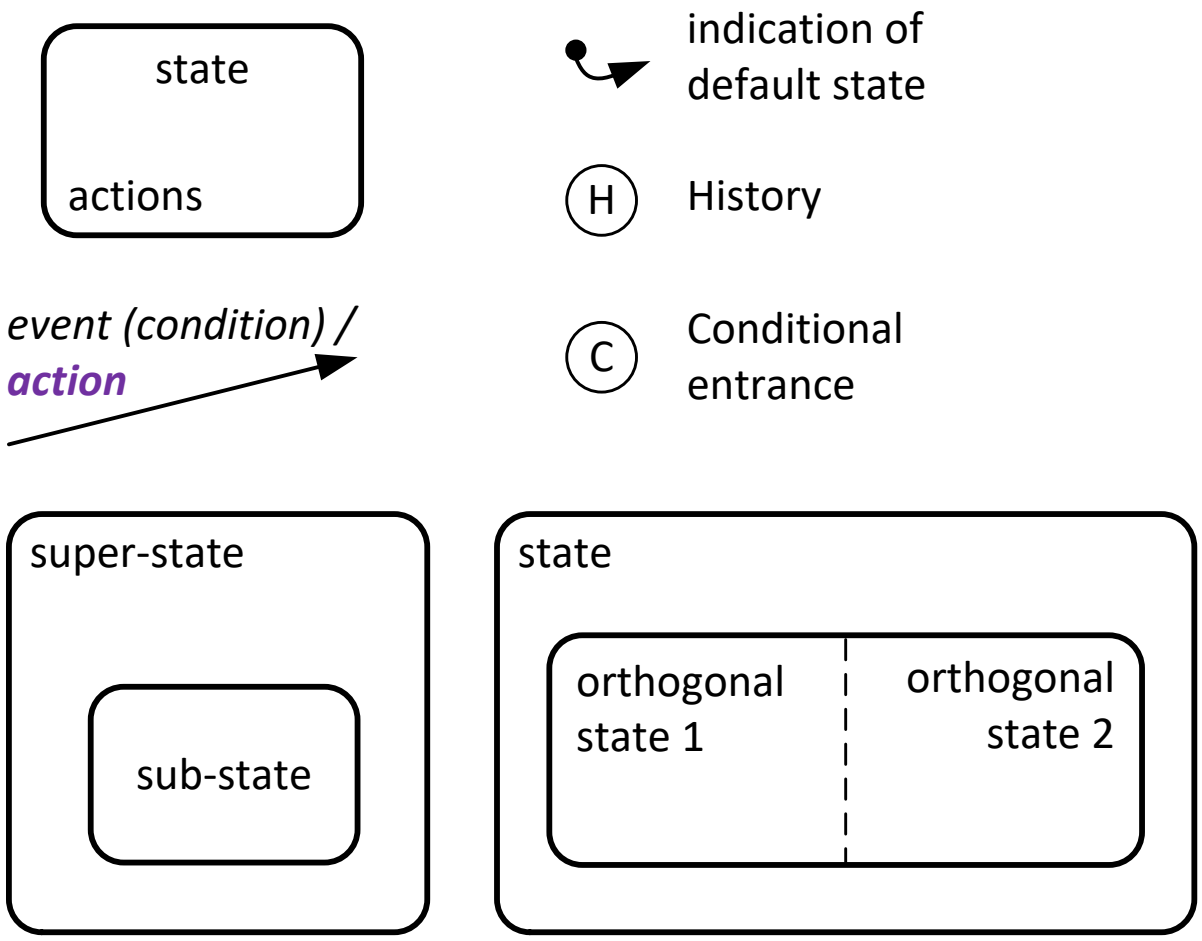


Figure 10.10 *Cruise Control statechart with activities*

Notation



Examples

Finite State Machine: ATM_Controller

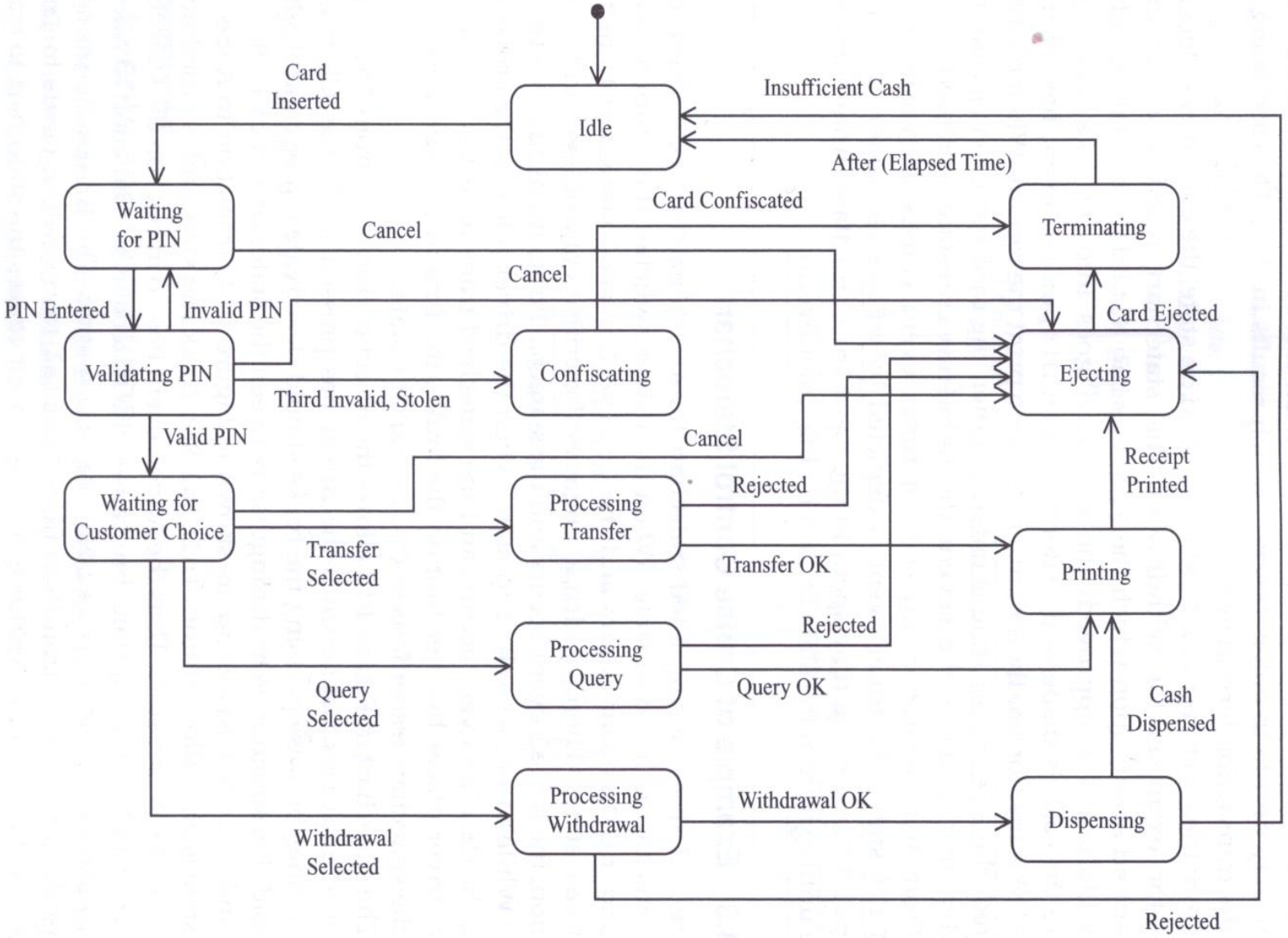


Figure 10.2 *Example of flat ATM statechart*

Example: Macro States, Hierarchical StateCharts ATM system

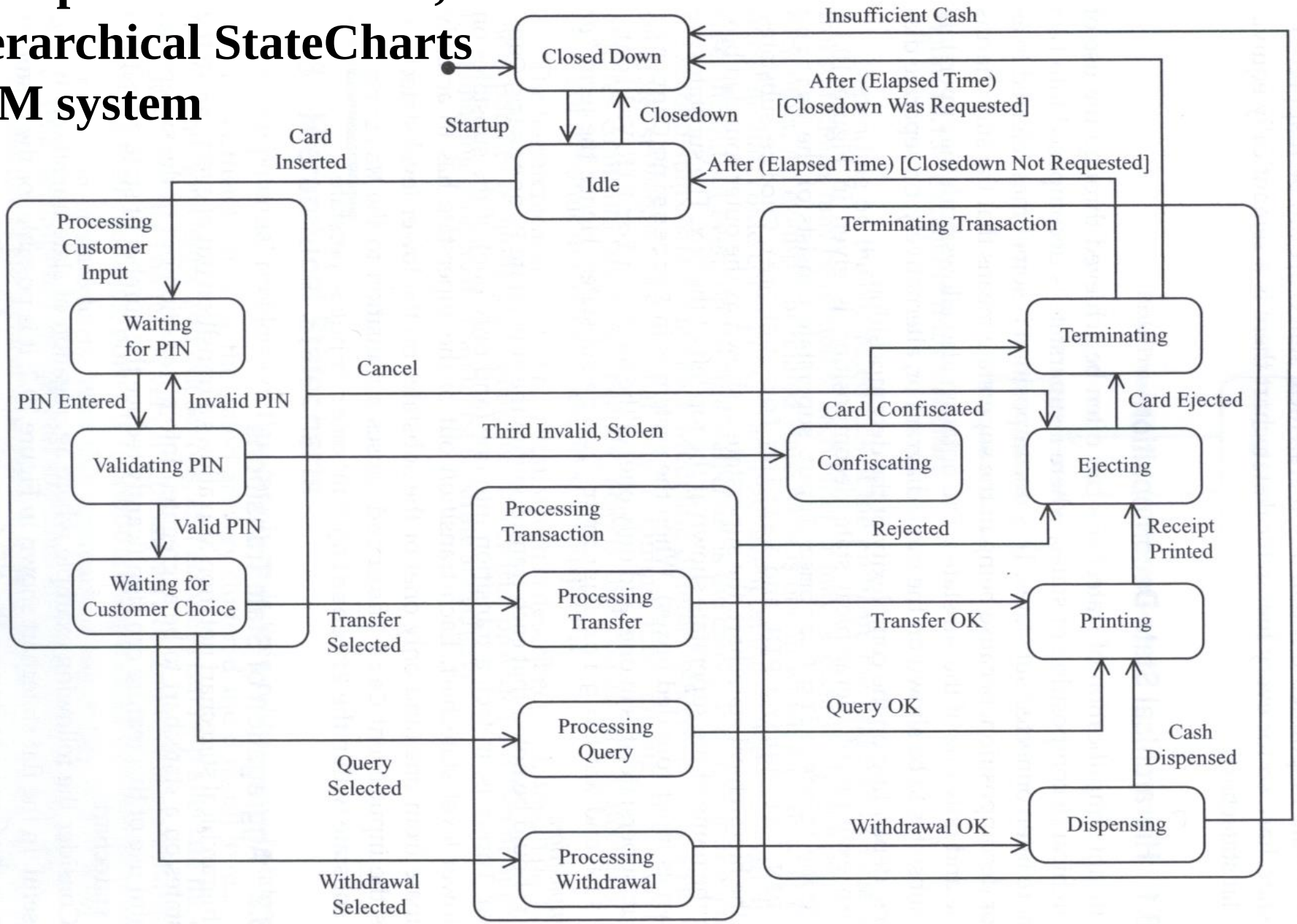
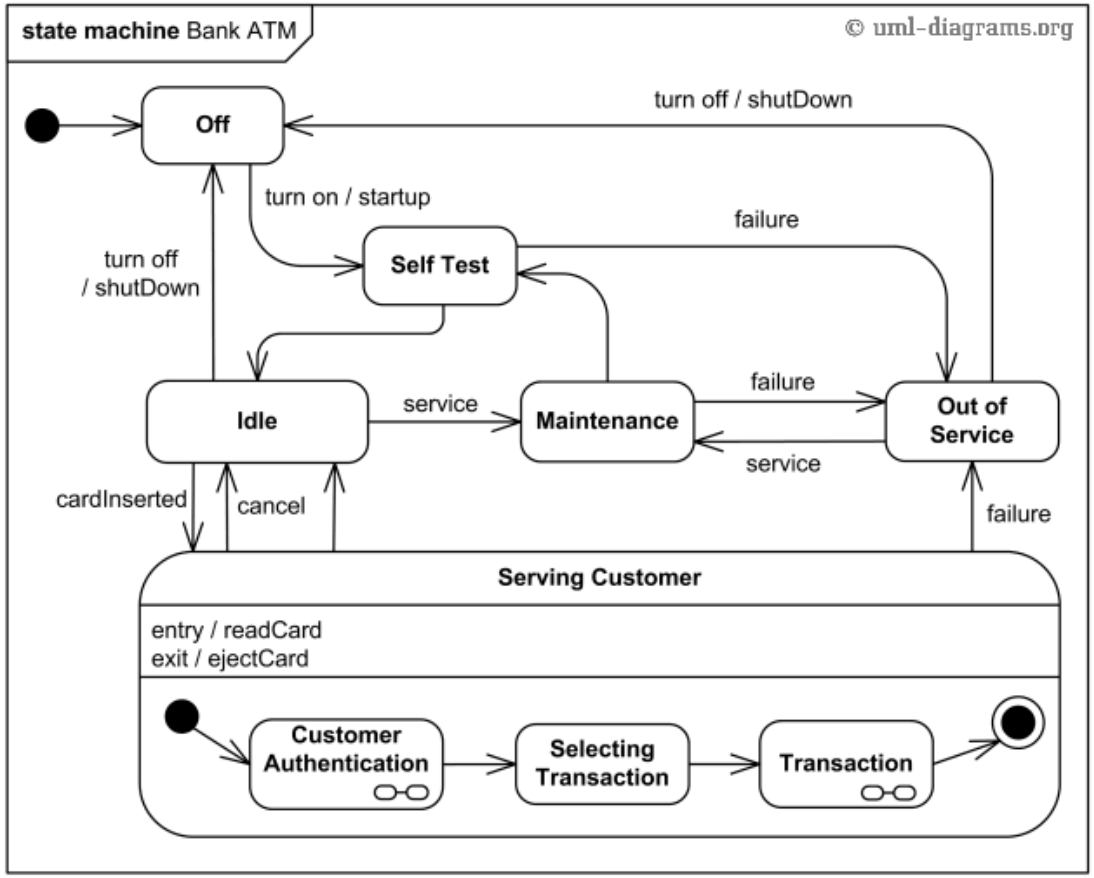
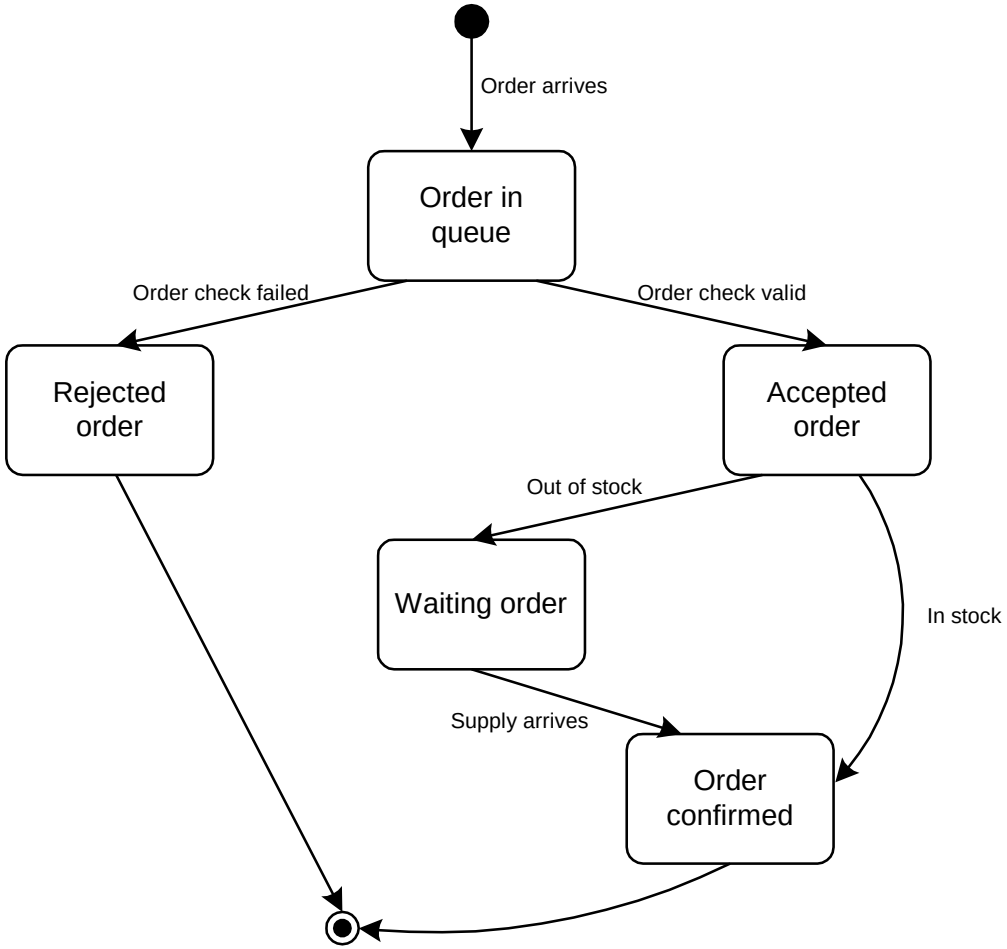


Figure 10.14 *Example of hierarchical statechart*

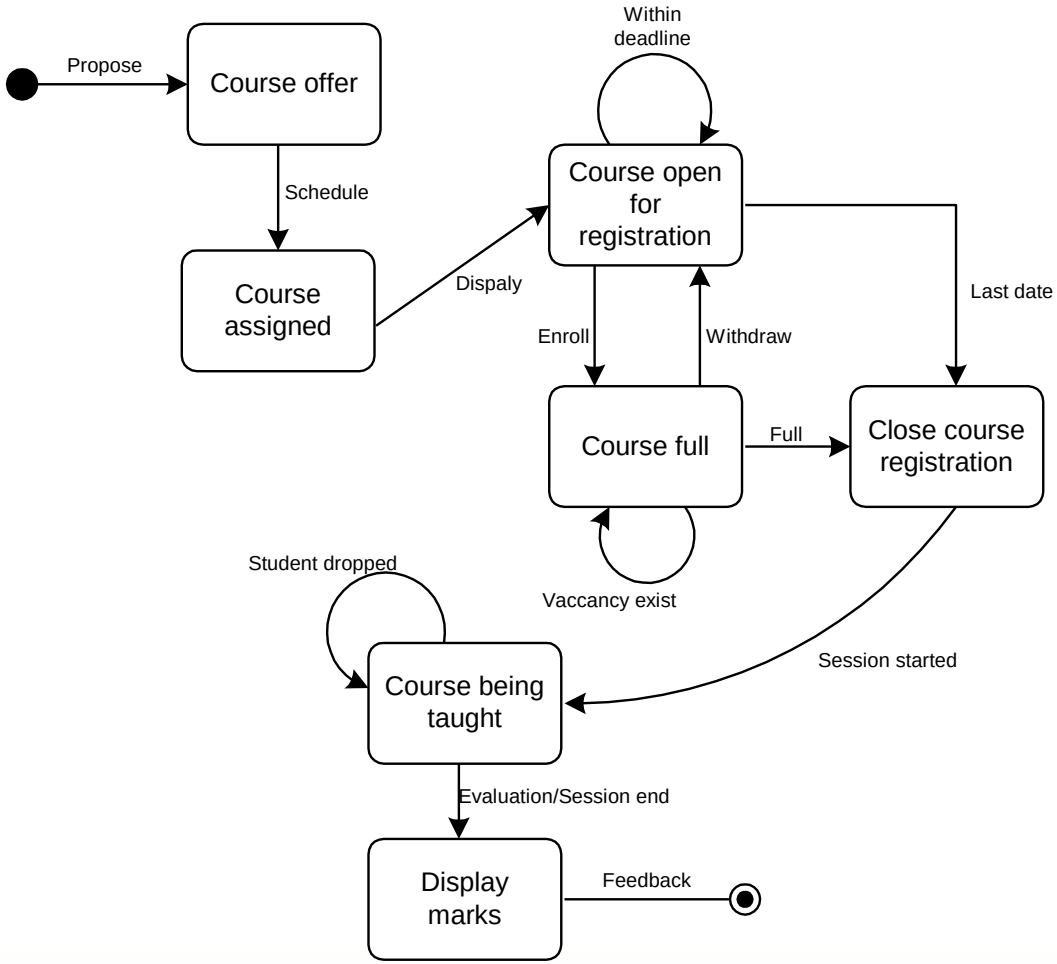
Finite State Machine – FSM : BANK ATM



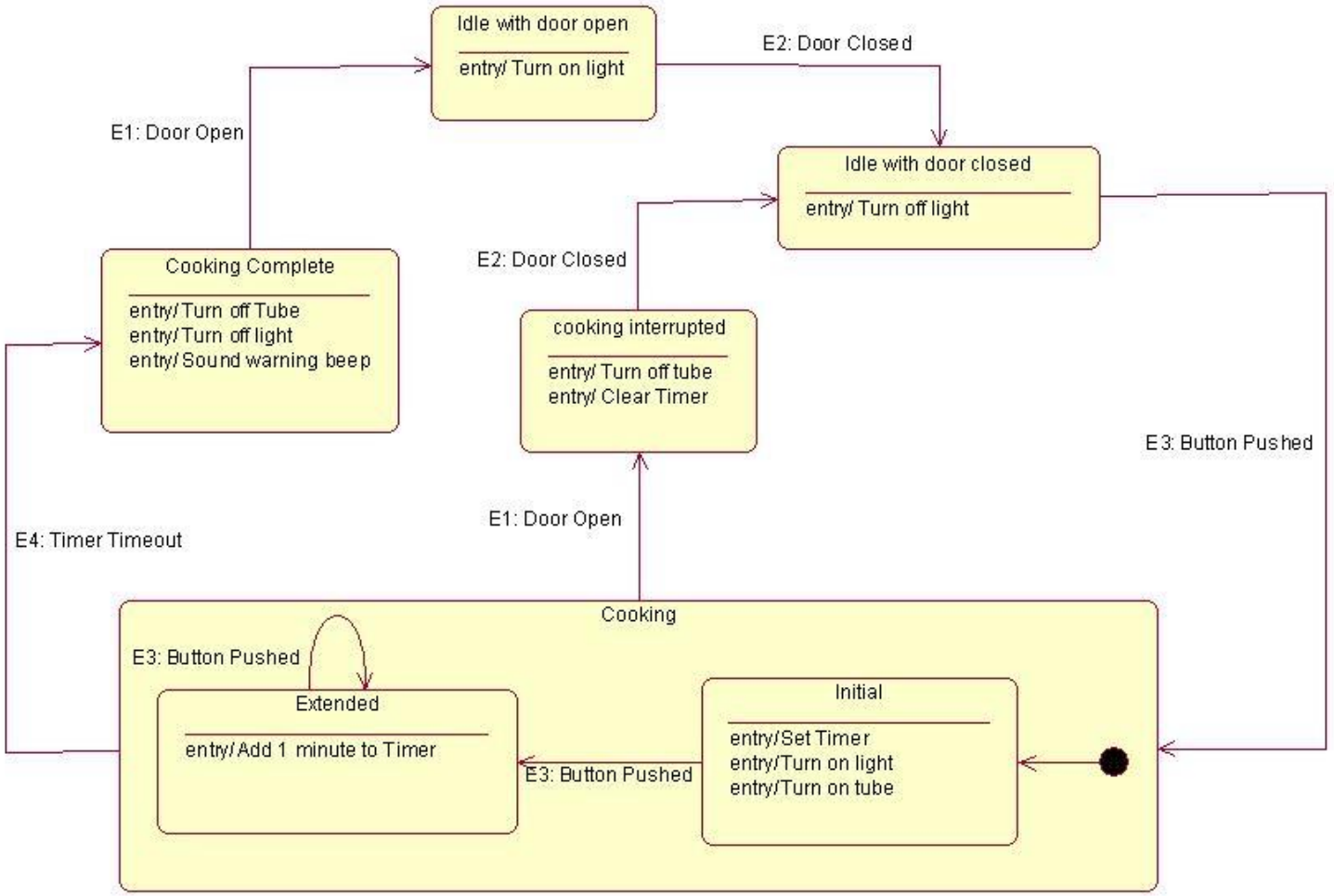
Order processing

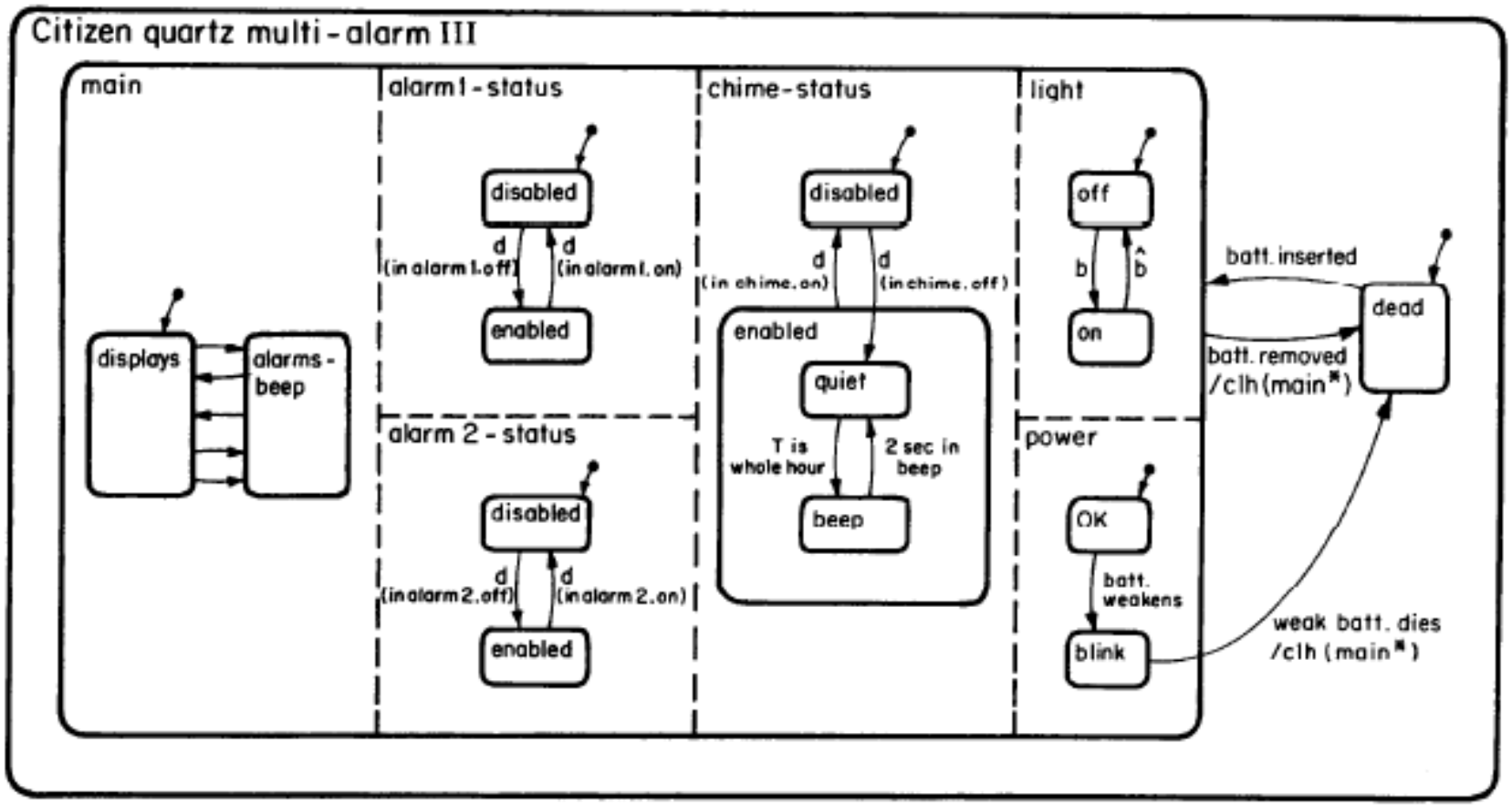


Course registration



Microwave


Idle with door open
entry/ Turn on light
E2: Door Closed
E1: Door Open
Idle with door closed
entry/ Turn off light
Cooking Complete
entry/ Turn off Tube
entry/ Turn off light
entry/ Sound warning beep
E2: Door Closed
cooking interrupted
entry/ Turn off tube
entry/ Clear Timer
E3: Button Pushed
E1: Door Open
E4: Timer Timeout
Cooking
E3: Button Pushed
Extended
entry/ Add 1 minute to Timer
E3: Button Pushed
Initial
entry/Set Timer
entry/Turn on light
entry/Turn on tube


Citizen quartz multi-alarm III
main
displays
alarms-beep
alarm1-status
disabled
d (in alarm1.off)
d (in alarm1.on)
enabled
alarm 2-status
disabled
d (in alarm2.off)
d (in alarm2.on)
enabled
chime-status
disabled
d (in chime.on)
d (in chime.off)
enabled
quiet
T is whole hour
2 sec in beep
beep
light
off
b
b̂
on
power
OK
batt. weakens
blink
batt. inserted
dead
batt. removed
/clh(main*)
weak batt. dies
/clh(main*)

Practice

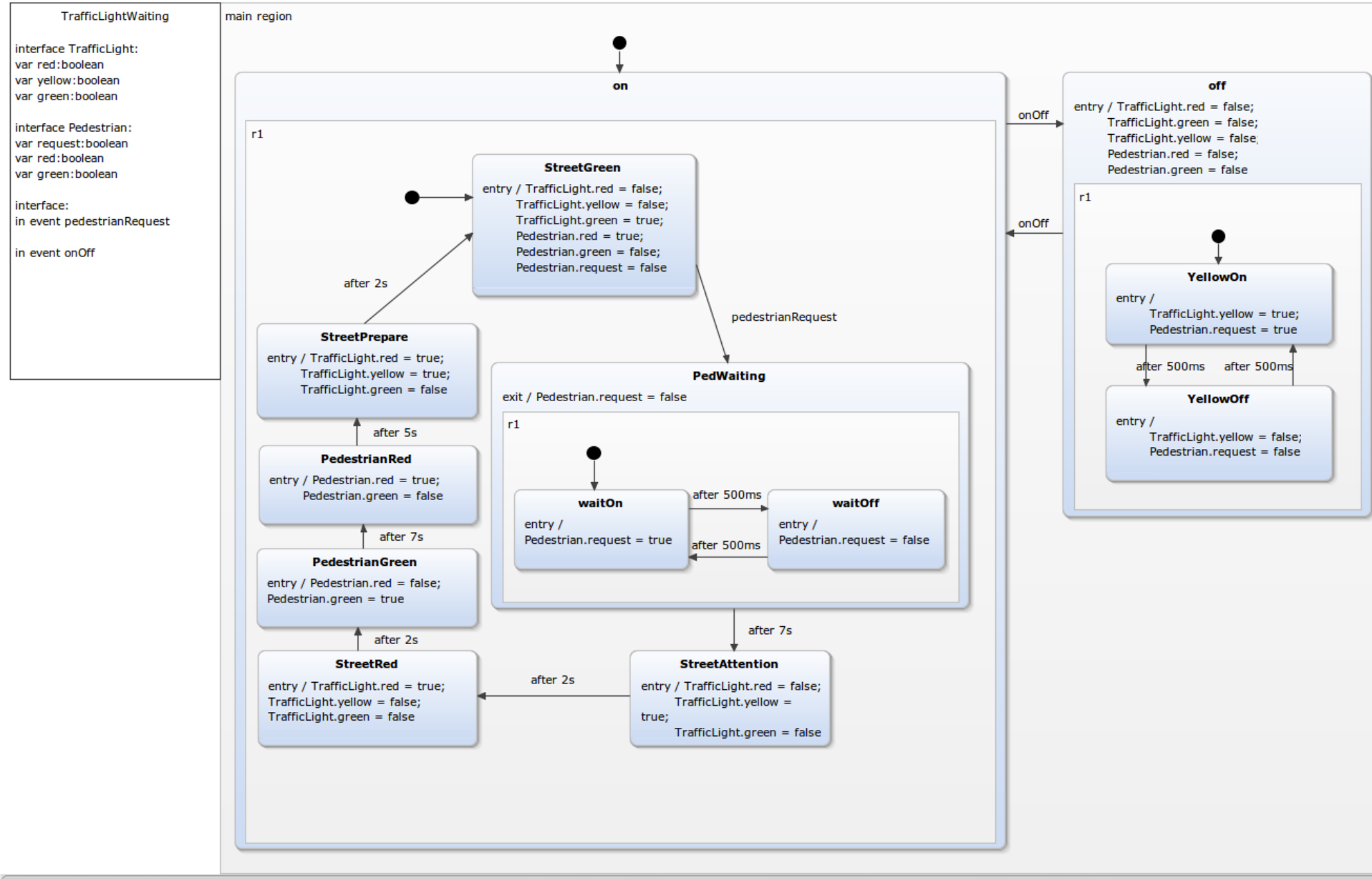
Practice

- Design the statechart for a traffic light with a button for pedestrians

A pedestrian can press a button to indicate, she/he wants to cross the street. Then a blinking light indicates, that the traffic light has recognized the request. After a few seconds, the traffic light for the street turns to red and the pedestrian traffic light turns to green. Then the pedestrian traffic light turns to red and the street traffic light changes to green again

References

- David Harel. *Statecharts: A visual formalism for complex systems (1987).*
- Slides from Betty H. C. Cheng: Dynamic Modeling: Modeling “State”.
- Slides from Debasis Samanta: Statechart Diagrams


TrafficLightWaiting
interface TrafficLight:
var red:boolean
var yellow:boolean
var green:boolean
interface Pedestrian:
var request:boolean
var red:boolean
var green:boolean
interface:
in event pedestrianRequest
in event onOff
main region
on
r1
StreetGreen
entry / TrafficLight.red = false;
TrafficLight.yellow = false;
TrafficLight.green = true;
Pedestrian.red = true;
Pedestrian.green = false;
Pedestrian.request = false
after 2s
pedestrianRequest
StreetPrepare
entry / TrafficLight.red = true;
TrafficLight.yellow = true;
TrafficLight.green = false
after 5s
PedestrianRed
entry / Pedestrian.red = true;
Pedestrian.green = false
after 7s
PedestrianGreen
entry / Pedestrian.red = false;
Pedestrian.green = true
after 2s
StreetRed
entry / TrafficLight.red = true;
TrafficLight.yellow = false;
TrafficLight.green = false
PedWaiting
exit / Pedestrian.request = false
r1
waitOn
entry /
Pedestrian.request = true
after 500ms
waitOff
entry /
Pedestrian.request = false
after 500ms
after 7s
StreetAttention
entry / TrafficLight.red = false;
TrafficLight.yellow =
true;
TrafficLight.green = false
after 2s
onOff
onOff
off
entry / TrafficLight.red = false;
TrafficLight.green = false;
TrafficLight.yellow = false;
Pedestrian.red = false;
Pedestrian.green = false
r1
YellowOn
entry /
TrafficLight.yellow = true;
Pedestrian.request = true
after 500ms
after 500ms
YellowOff
entry /
TrafficLight.yellow = false;
Pedestrian.request = false