**Chapter 4:
Dynamic Design**

Academic Year

2021-2022

STATE-TRANSITION DIAGRAM

- Behavioural diagram.

- Used for classes with complex behaviour.

- Is associated with an instance of a class (object) with several states.

- Describes how an object reacts to events depending on its state and how it moves to a new state.

- Oriented graph of states and transitions representing an automaton with finite states.
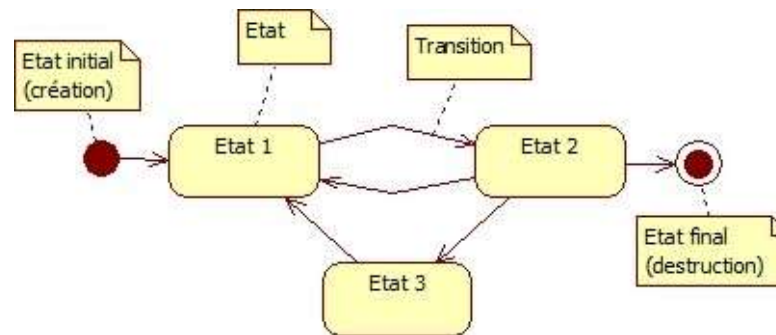
- Example: operation of a push-button switch
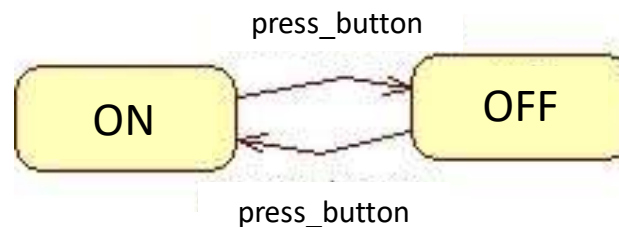
- When the push-button is pressed, the reaction of the machine depends on its current state: if it is switched on, it will stop and if it is switched off, it will start.
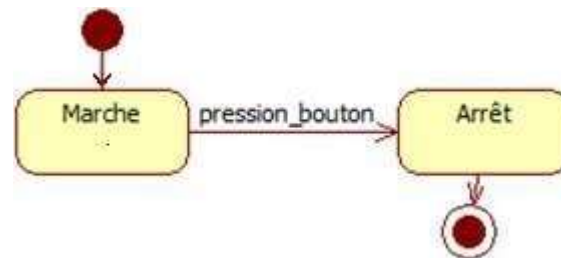
- State

  - Activity

- Transition

  - Event (Trigger)

  - Custody condition (guard condition)

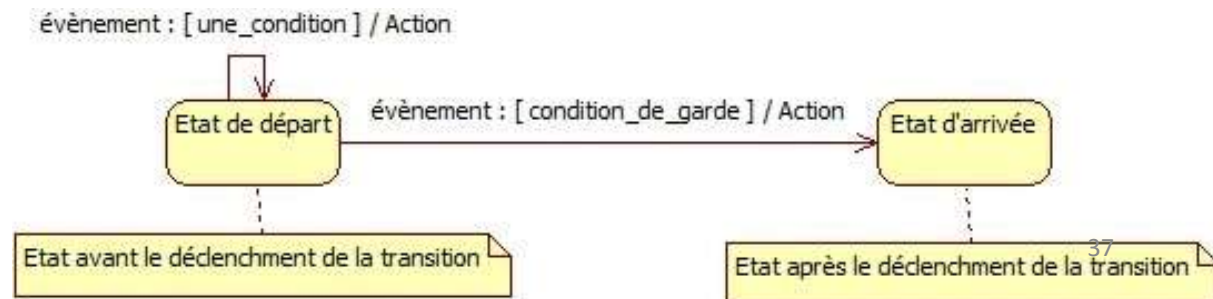 - Action

- Composite state

STATE

- An object goes through several states during its lifetime

- A state: a set of values for object properties

- An object in a given state:

  - Waits for an event

  - Performs an activity

- Initial state ●

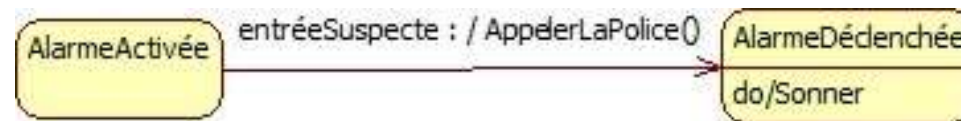- Final state ◉

- Example:

TRANSITION

- Describes the reaction of an object when an event occurs

- Enables you to move from one state to another

- A transition generally has :

  - A triggering event

  - A custody condition

  - An action

    - Operation associated with the transition

    - Is non-interruptible

  - A target state

évènement : [une_condition] / Action

Etat de départ   évènement : [condition_de_garde] / Action   Etat d'arrivée

Etat avant le déclenchment de la transition

Etat après le déclenchment de la transition
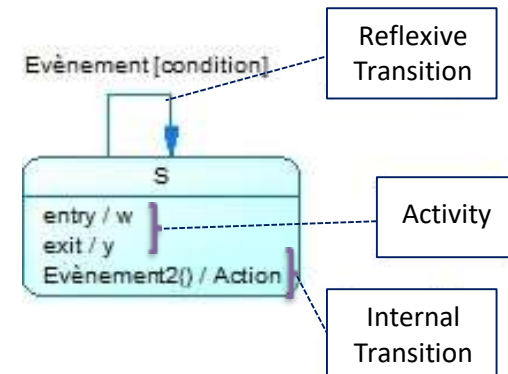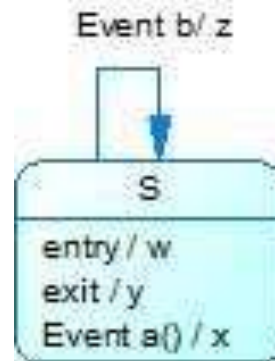
37

- A simple example:

- Reflexive transition

  – The initial state and the target state are identical.

  – The object leaves a state and returns to it.

- Internal transition

  – Transition triggered in a current state

  – The transition has a source state but no target state

    → The object does not leave the current state
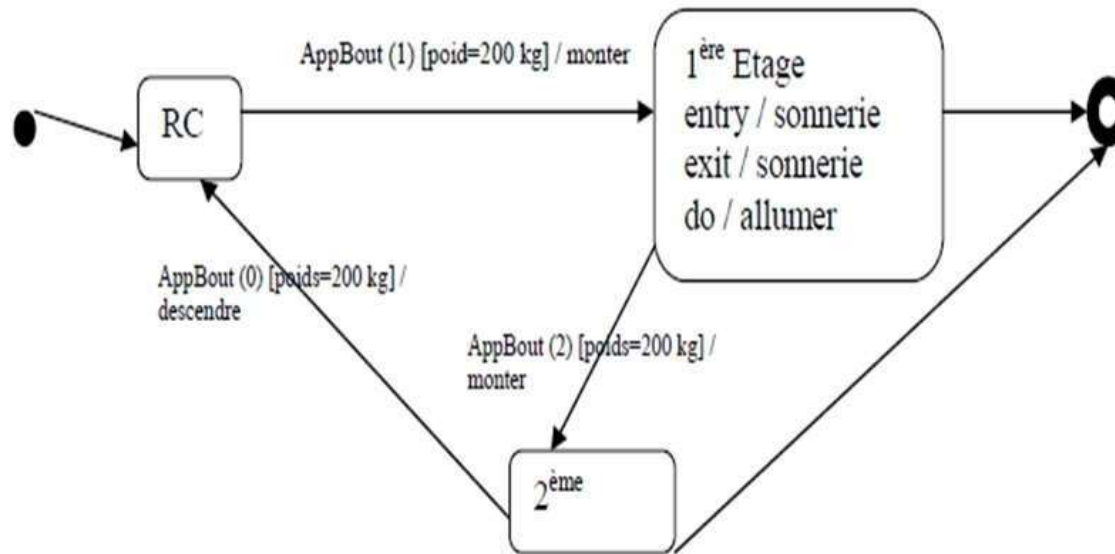
  – Is registered in the state

- Illustration:

- If we are in state **S** and there is an occurrence of **Event a**, then **x** is activated.

- If we are in state **S** and **Event b** occurs, then **y** (we first exit **S**), next **z** (triggered because is suffixing of **Event b**) and finally **w** (we re-enter **S**) are activated.

- Example 1: State-transition diagram of an elevator cabin

- Example 2: State-transition diagram of a coin mechanism of a drinks dispenser
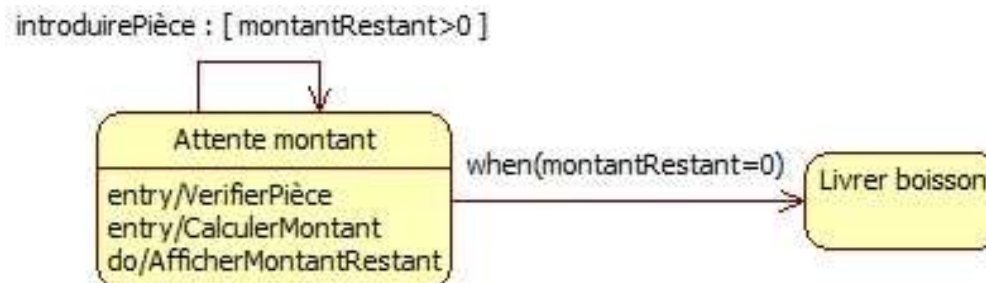
# EVENT

- Occurs at a given moment and has no duration

- Triggers a transition

- Types of events :

    – Method call type (call)

    – Signal type

        - Example: mouse click, I/O interrupt

    – Change of value type (true/false): evaluation of a Boolean

      expression: **when**(boolean_condition)

– Temporal type: event linked to the time flow

- After a specific duration: **after** (duration)

- At a specific time: **when** (date)

- Example:

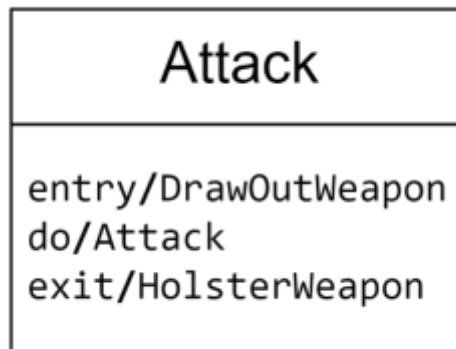– After two minutes of inactivity, the screensaver will be activated
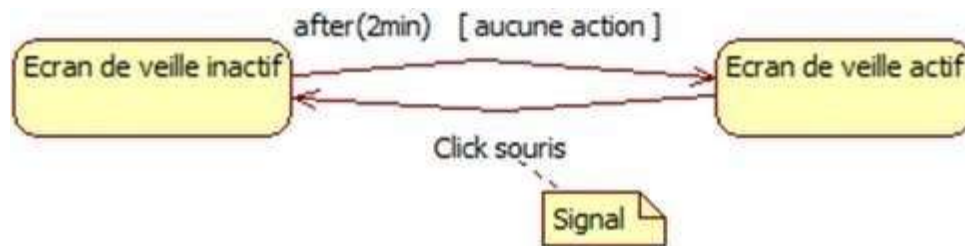
ACTIVITY

- Specifies an optional behaviour for the object when it reaches a new state (after a transition has been triggered).
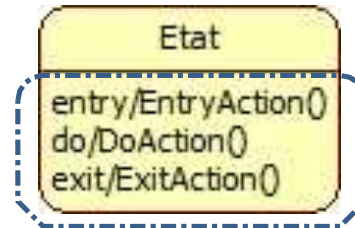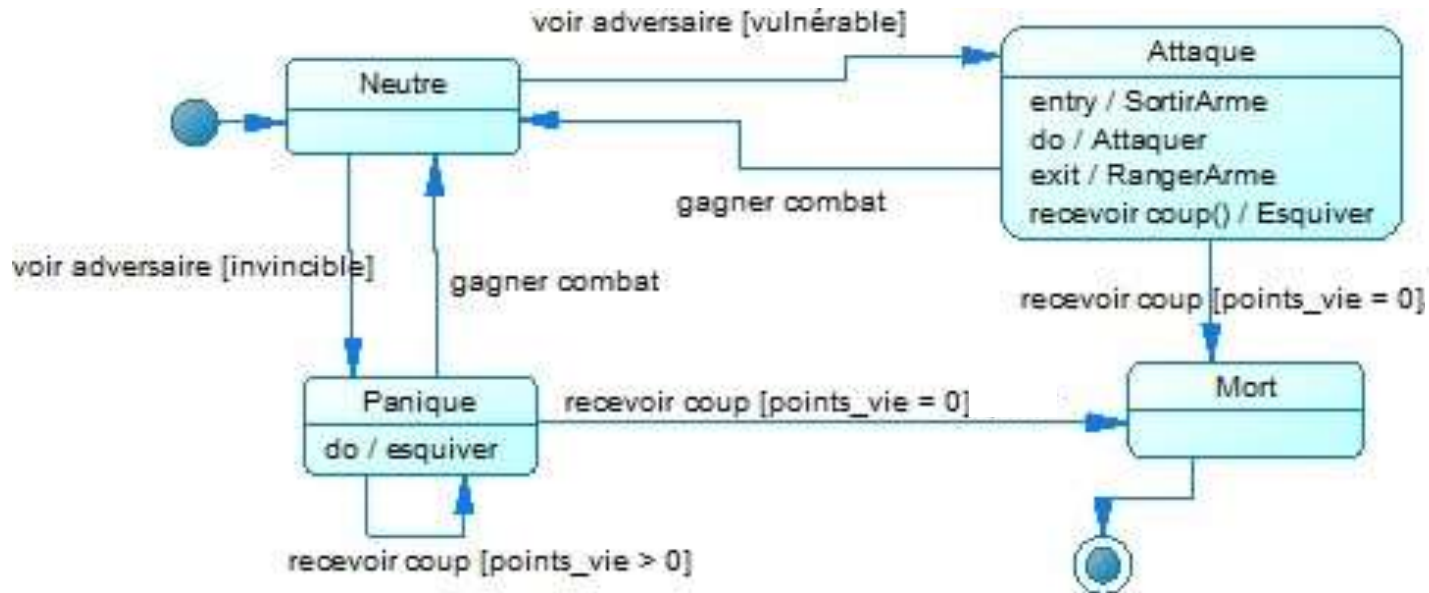

- Is interruptible.

- Types of activities :

  – **Entry** activity: actions performed when entering a state.

  – Sustainable activity (**do**): indicates work performed as long as the object is in a state.

  – **Exit** activity: actions performed when exiting a state.

- For example:

  – In a video game, the character in a state of attack will first

    draw out his weapon, attack his opponent and then holster

    his weapon when the attack is over.

```
┌─────────────────────────────┐
│          Attack             │
├─────────────────────────────┤
│ entry/DrawOutWeapon         │
│ do/Attack                   │
│ exit/HolsterWeapon          │
└─────────────────────────────┘
```
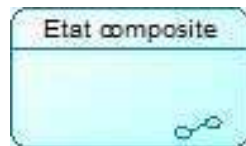
- Simulation of a video game character :

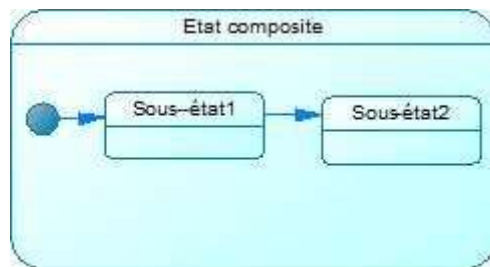# COMPOSITE STATE

- State containing many states.

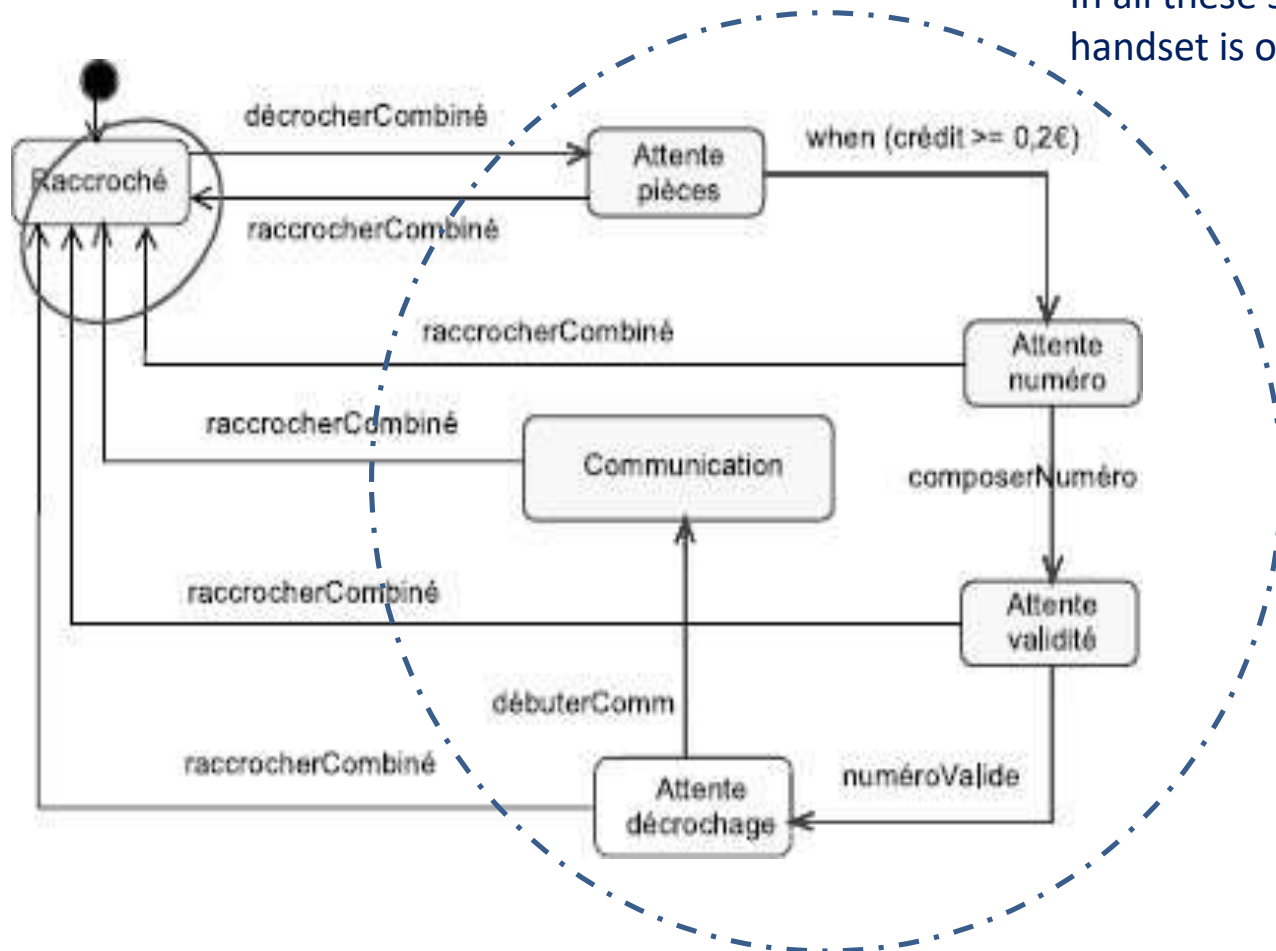→ Form 1: hidden decomposition
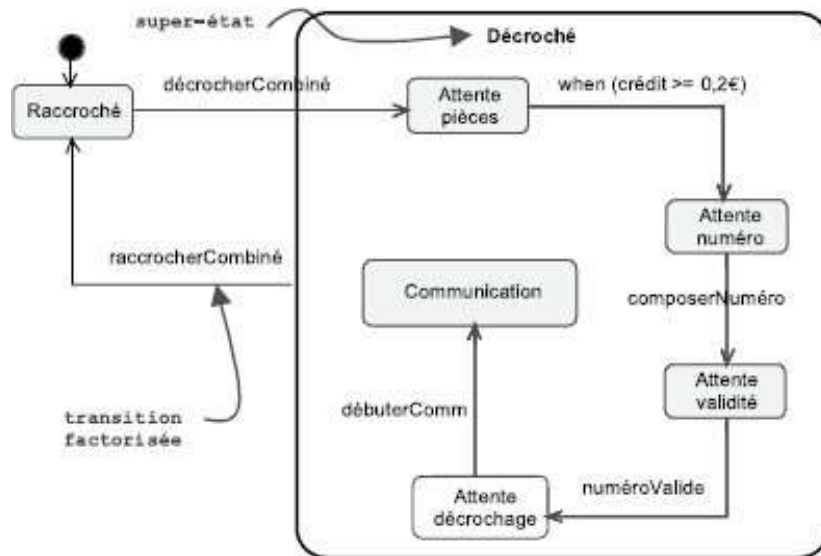

→ Form 2: explicit decomposition

Example:

In all these states, the handset is off-hook

Source : UML2 in practice

- Two possible representations for factoring the off-hook handset state:

Or