



# Le langage UML : Les diagrammes d'états-transitions

---

**Lydie du Bousquet**  
[Lydie.du-bousquet@imag.fr](mailto:Lydie.du-bousquet@imag.fr)



# Plan

---

1. Rappels
2. Introductions
3. Concepts de base
4. Concepts avancés



# 1. Rappels

---

- Aspects statiques
  - Diagramme de classe
  - Diagramme d'objets
- Les interactions
  - Diagramme de cas d'utilisation
  - Diagramme de séquence



## 2. Diagramme d'états-transitions

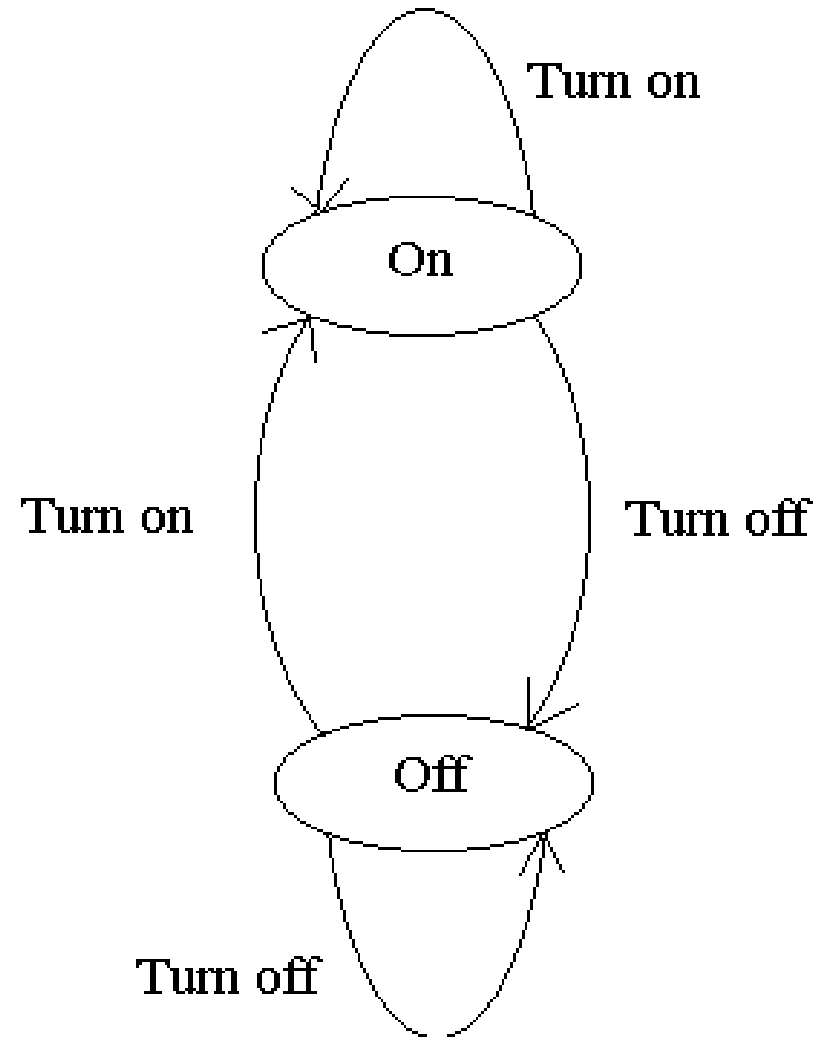
---

- Cycle de vie de l'objet
  - États = différentes étapes qui vont être suivies (activité / inactivité)
  - Événement ou signal qui fait changer d'état
  - Transition : passage d'un état à un autre
- Diagramme d'états-transitions
  - Représente les états et les transitions
  - Pour un objet
- **Automate**



## 2. Exemple

---





## 2. Objectifs

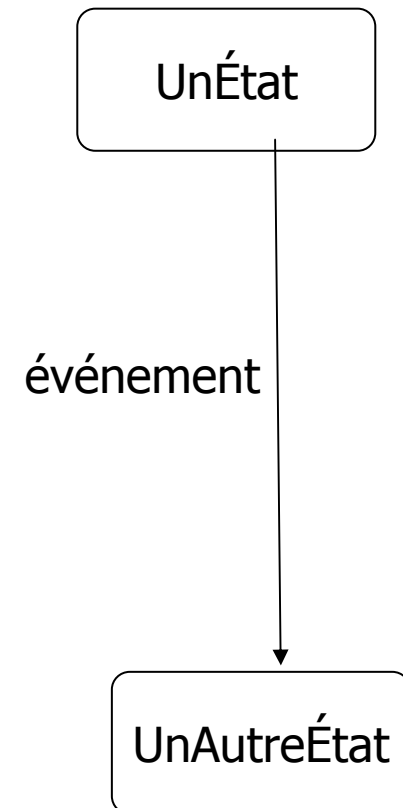
---

Pour un objet donné, le diagramme d'états-transitions décrit :

- Tous les états
- Tous les changements d'états possibles en réponse à des sollicitations

## 3. Les concepts de base

1. Les états
2. Les changements d'états
  - Transition
  - Événement
  - Condition de garde
3. Les activités

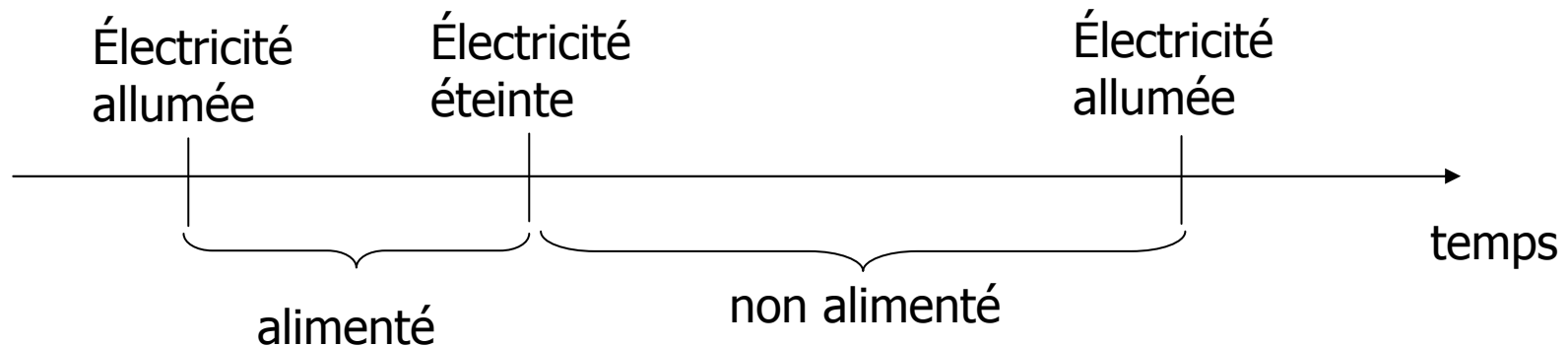




## 3.1 Les états

---

- C'est un moment dans le cycle de vie
- Traduit l'activité ou l'inactivité







## 3.1 Les états

---

- **Abstraction** des valeurs ou des liens d'un objet
  - Ex : banque solvable ou non
- Correspond souvent à
  - des participes présents (attendant, numérotant)
  - Stabilité d'une condition (allumé, inférieur\_à\_zéro)

## 3.1 Les états : notation UML

- État quelconque
  - Boîte aux coins arrondis
  - Nom en gras avec une capitale
- État initial
  - Un point noir
- État final
  - Point noir entouré par un cercle





## 3.2 Le changement d'état : notion d'événement et de signal

---

- Événement
  - Fait qui déclenche un changement d'état
  - Lié à la réception d'un **signal** = message
  - Le signal peut être émis par tout objet, y compris celui qui attend le signal
- Signal
  - Peut être décrit par une classe

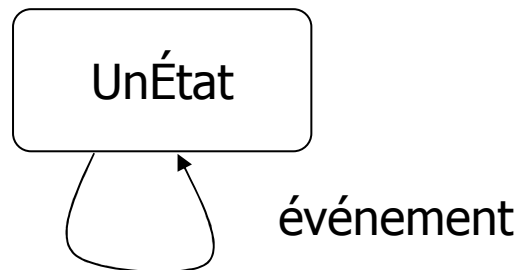
## 3.2 Le changement d'état : transition

- Lien orienté entre 2 états
- Exprime la **possibilité** de passer de l'état d'origine à l'état de destination
- Est **franchie** lorsque l'objet réalise le passage
- Généralement associé à un événement
  - Transition est franchie **instantanément**
    - si l'objet est dans l'état d'origine et
    - s'il reçoit l'événement



## 3.2 Le changement d'état : transition réflexive

- Quand la transition ne provoque pas de changement d'état



## 3.2 Conditions de garde

- Il est possible d'associer une condition à une transition
- Pour que la transition soit franchie, il faut alors
  - Que l'événement soit reçu **et**
  - Que la condition soit satisfaite
- La condition est exprimée
  - entre crochets,
  - à droite du nom de l'événement





## 3.3 Activité

---

- Activité = série d'actions
- Action =
  - Affecter une valeur à un attribut
  - Créer ou détruire un objet
  - Effectuer une opération, envoyer un signal
  - Etc...



## 3.3 Activité

---

- Il est possible de spécifier des activités
  - Lors d'un franchissement d'une transition
  - Pendant un état
  - À l'entrée ou la sortie d'un état







## 3.3 Activité

---

- Il est possible de spécifier des activités
  - Lors d'un franchissement d'une transition
  - Pendant un état
  - À l'entrée ou la sortie d'un état

Etat1

**entry/** action  
**do/** activité  
**on événement/** action  
**exit/** action

Saisie mot de passe

**entry/** ne plus afficher entrées clavier  
**exit/** réactiver l'affichades des entrées clavier  
**on aide/** afficher l'aide  
**do/** gérer entrées de l'utilisateur

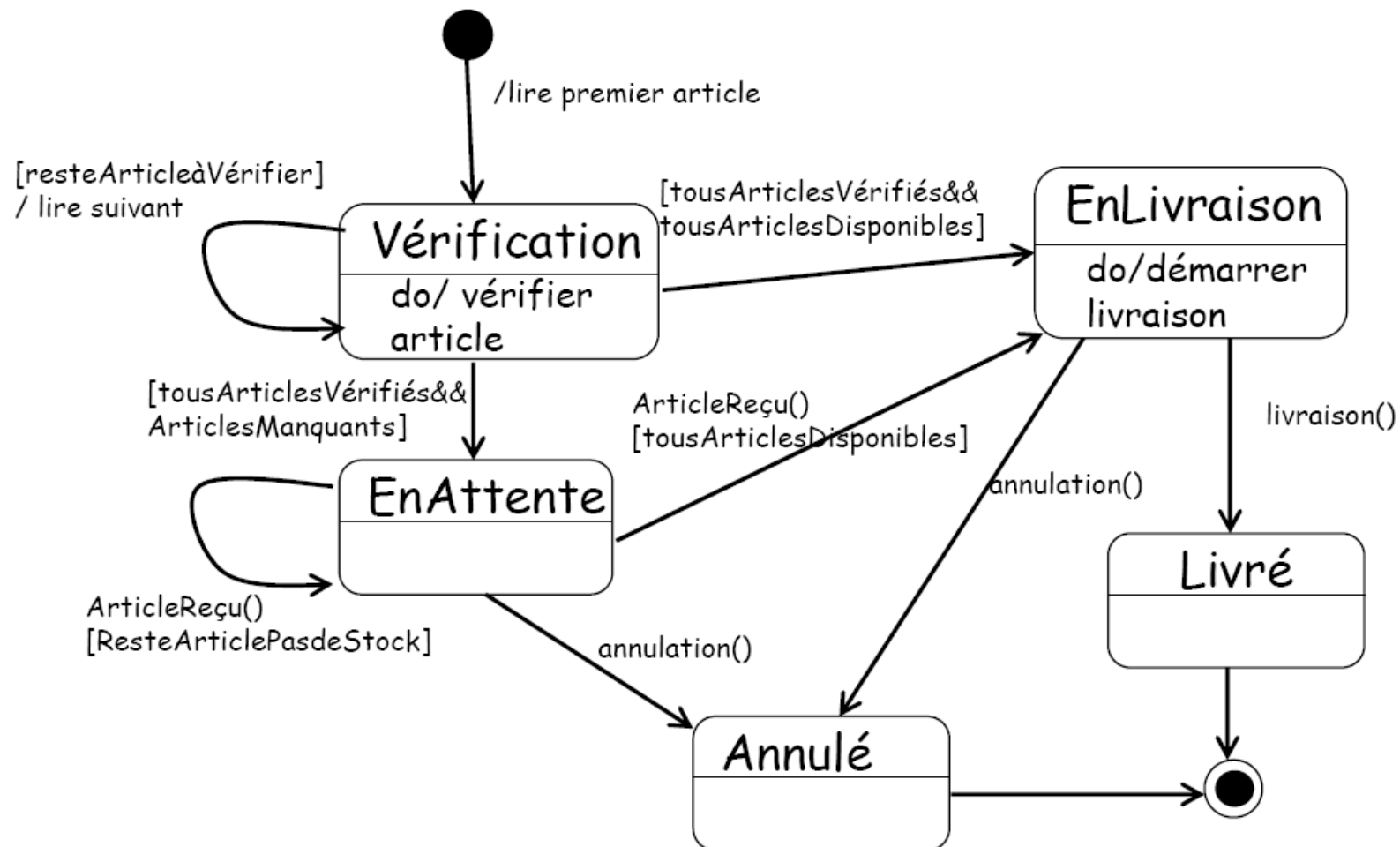


# Bilan - syntaxe

---

- Événement
  - nomÉvénement (param1:type, param2:type,...)
  - *Ex: allumedLed()*
- [Garde]
  - Condition booléenne
  - la transition n'a lieu que si la condition est vraie
  - *Ex: [ageCapitaine>50]*
- Action
  - Pseudo code
  - *Ex. solde = solde – débit*
- Action interne à un état
  - ActionLabel/Action
  - Action Label : entry, do, exit
  - *Ex. : do/calculateAverage(x:array)*

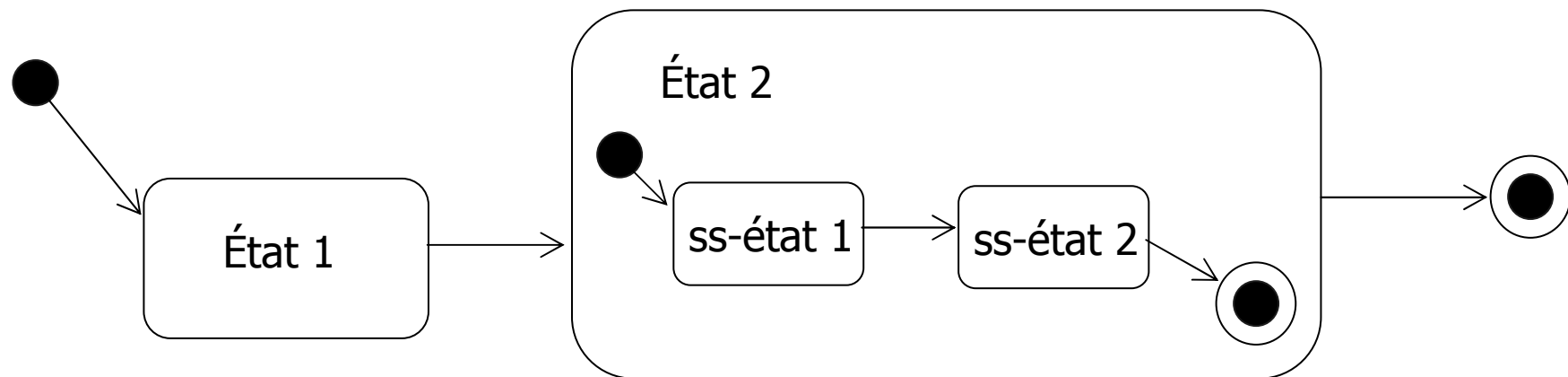
# Exemple : réalisation d'une commande



# 4. Concepts avancés

## Les états composés

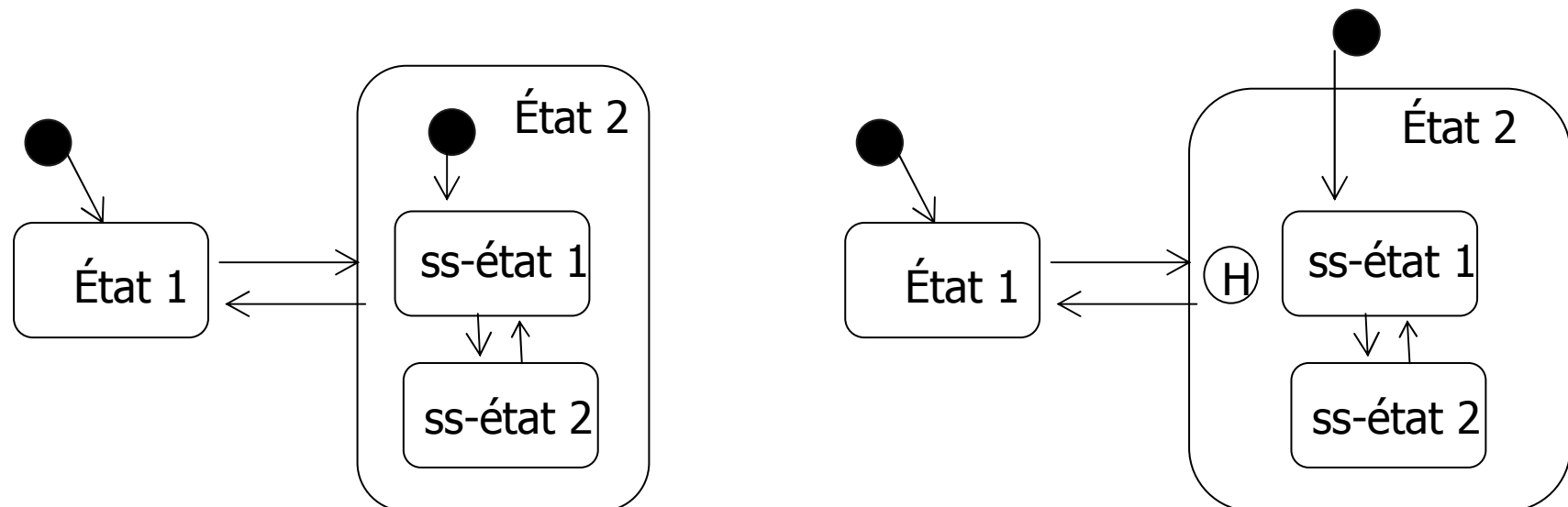
- Un état peut être décrit lui-même par un diagramme états-transitions
- Quand l'objet passe dans un état composé, il passe dans le sous-état initial



# 4. Concepts avancés

## Les états composés

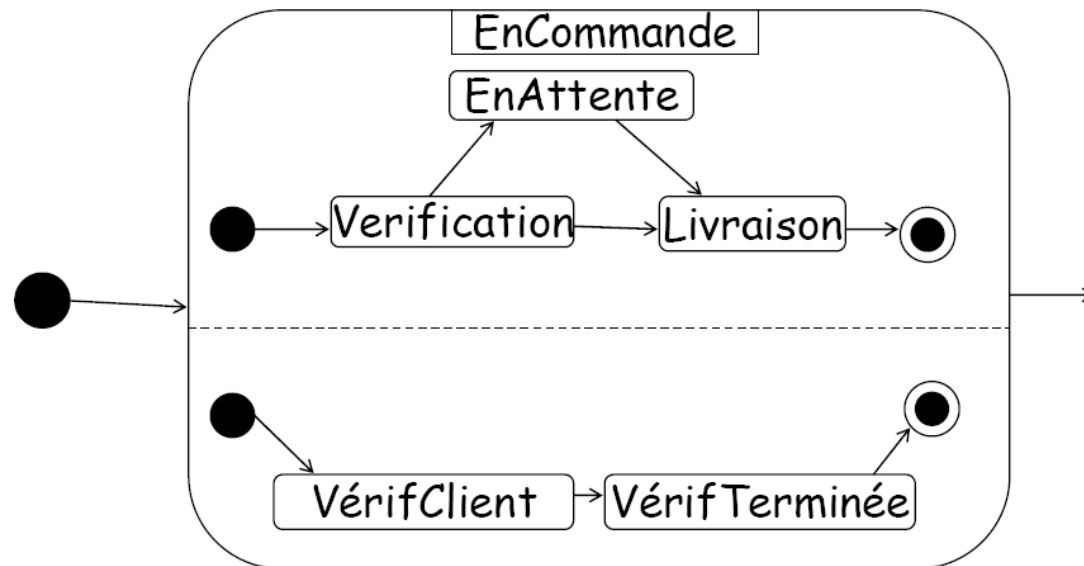
- L'état composé
  - Peut ne pas avoir d'état final
  - Peut posséder plusieurs états finaux
- Lorsque l'on quitte l'état composé
  - On peut mémoriser ou non l'état dans lequel on est



# 4. Concepts avancés

## Les états composés

- Au sein d'un état composé, il est possible d'avoir des états qui évoluent en parallèle



## 4. Concepts avancés

### Synchronisation d'états concurrents

- La barre de synchronisation n'est franchie que lorsque toutes les transitions entrantes ont été franchies

