

# UML : diagramme états-transitions

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



**UNIFIED  
MODELING  
LANGUAGE**

# Plan

- 1 Introduction
- 2 État
- 3 Évènement
- 4 Transition
- 5 Condition de franchissement
- 6 Point de décision
- 7 Point de jonction
- 8 États Imbriqués
- 9 Comment construire les diagrammes états-transitions ?
- 10 Exercices d'application

## Diagramme états-transitions ?

- Un diagramme dynamique d'**UML**
- Permettant de représenter les différents états qu'un objet peut avoir et les transitions d'un état vers un autre
- Utilisant des automates déterministes à états-finis
  - un graphe orienté
  - déterministe : il y a toujours un chemin d'un état initial vers un état final
  - états-finis : le nombre d'états est fini

Un diagramme états-transitions pour chaque objet ?

Non, que pour les classes ayant un comportement temporel significatif pour le système

© Achref EL MOUELHI

# UML

## Un diagramme états-transitions pour chaque objet ?

Non, que pour les classes ayant un comportement temporel significatif pour le système

## Exemples

- Livre (emprunté, disponible, réservé...) et Emprunteur (autorisé, bloqué, sanctionné...) dans un système de gestion de bibliothèque
- Article (disponible, vendu, expédié, livré...) dans un système de vente et achat en ligne

## Mots-clés associés

- état
- évènement
- transition
- condition de franchissement
- point de décision
- point de jonction

# UML

## État

Abstraction de valeurs d'un objet

© Achref EL MOUELHI ©

## État

Abstraction de valeurs d'un objet

**Comment représenter un état en UML 2 ?**



État

A diagram of a UML state node, which is a rounded rectangle with a black border. The word "État" is centered inside the rectangle. A large, faint watermark "© Achref EL MOUJELHI ©" is visible in the background of the slide.

## État

Abstraction de valeurs d'un objet

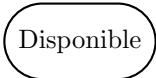
**Comment représenter un état en UML 2 ?**



État

A UML state node, represented as a rounded rectangle with a black border, containing the text "État".

**Exemple : un livre peut avoir un état disponible**



Disponible

A UML state node, represented as a rounded rectangle with a black border, containing the text "Disponible".

**Un objet de type Livre peut avoir plusieurs états**

Disponible

Emprunté

© Achref EL MOUELHI ©

**Un objet de type `Livre` peut avoir plusieurs états**



Disponible

Emprunté

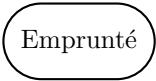
**Tout objet a un état initial (schématisé différemment)**



**Un objet de type `Livre` peut avoir plusieurs états**



Disponible



Emprunté

**Tout objet a un état initial (schématisé différemment)**



**Et un état final (schématisé aussi différemment)**



## Un objet peut avoir

- plusieurs états
- état initial : lorsqu'un objet est créé, il a cet état initial.
- état final : lorsqu'un objet a cet état, il ne peut plus changer d'état.

© Achref

## Un objet peut avoir

- plusieurs états
- état initial : lorsqu'un objet est créé, il a cet état initial.
- état final : lorsqu'un objet a cet état, il ne peut plus changer d'état.

## Question

Comment passer d'un état à un autre ?

## Évènement

- Occurrence ou fait ayant lieu à un moment donné
- Générant un changement d'état chez l'objet
- Pouvant être
  - appel d'une méthode
  - signal
  - changement temporel

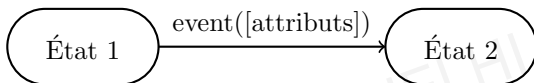
### Exemple d'évènements pour un objet de type `Livre`

- Demande d'emprunt
- Enregistrement de retour

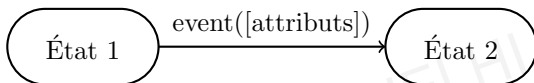
## Transition

- Le passage d'un état vers un autre suite à un évènement
- Pouvant être automatique si on ne précise pas l'évènement déclencheur
- Représenté en **UML** par une flèche (arc)

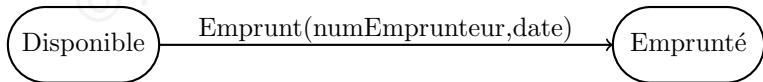
## États + transition + évènement



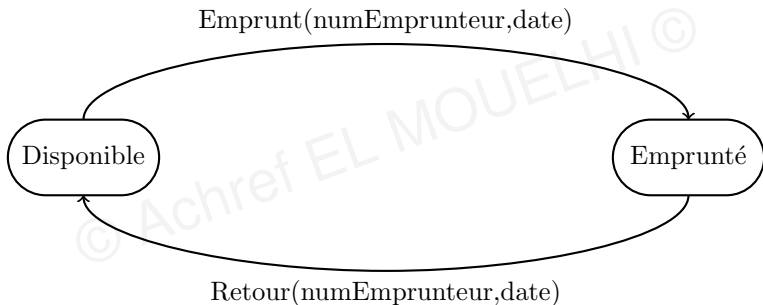
## États + transition + évènement



## Exemple



## Plusieurs transitions entre deux états



## Remarques

- D'après notre système, toute demande d'emprunt de livre est acceptée
- Dans le monde réel, il faut vérifier certaines conditions avant d'emprunter un livre
- Par exemple, il ne faut pas dépasser un nombre maximum d'emprunts par emprunteur

## Remarques

- D'après notre système, toute demande d'emprunt de livre est acceptée
- Dans le monde réel, il faut vérifier certaines conditions avant d'emprunter un livre
- Par exemple, il ne faut pas dépasser un nombre maximum d'emprunts par emprunteur

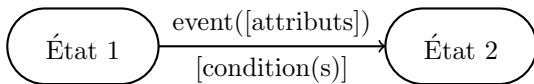
## Solution

Définir des conditions de franchissement

## Condition de franchissement (ou de garde)

- une transition peut avoir une condition
- expression booléenne exprimée en langage naturelle (mathématique, logique...)
- évaluation uniquement lorsque l'événement se produit
  - si l'expression est fausse  $\Rightarrow$  la transition ne s'effectue pas
  - sinon, la transition s'effectue
- placée entre crochet `[condition(s)]`

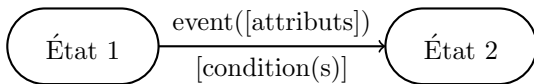
## UML

**États + transition + évènement + condition de franchissement**

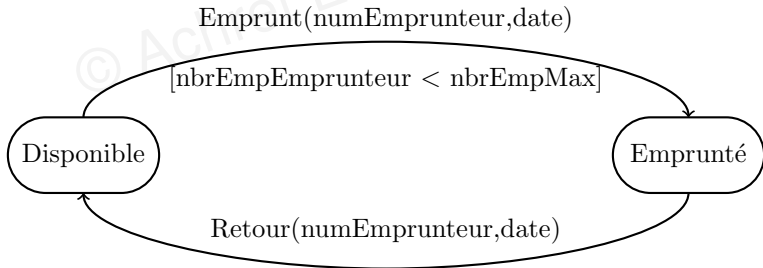
© Achref EL MOUELHI ©

## UML

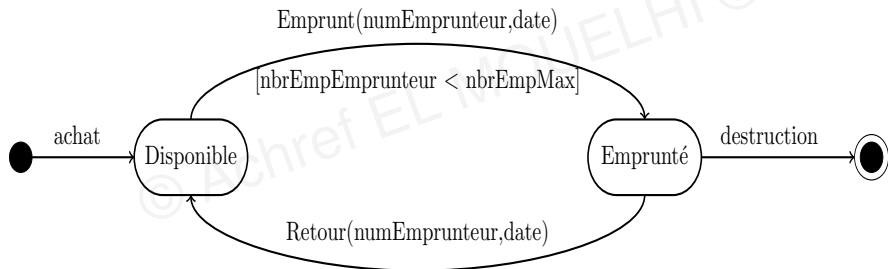
## États + transition + évènement + condition de franchissement



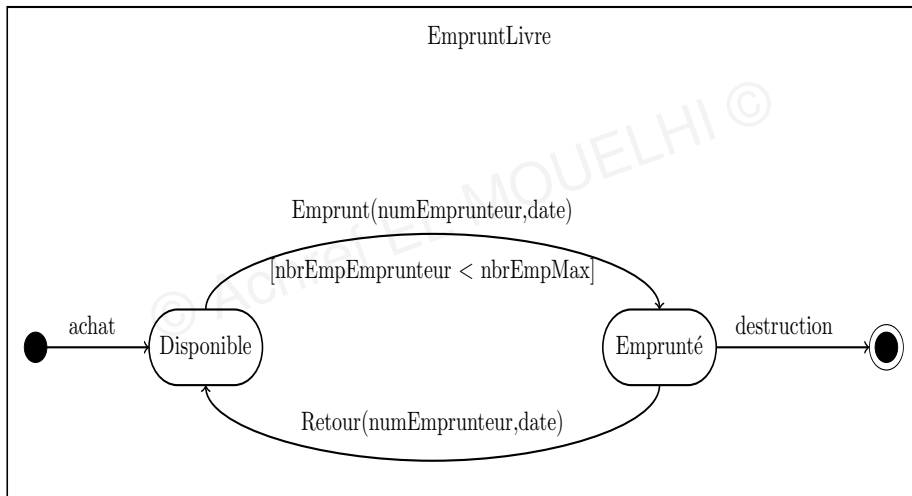
## Exemple



**En réunissant tous les éléments précédents, on obtient le diagramme états-transitions suivant**



## On peut même définir un contexte pour notre diagramme



## Hypothèse

Supposant qu'au retour d'un livre, le livre peut avoir deux états différents

- correct : on le remet à disponible
- autre (mouillé, déchiré...) : le livre doit être réparé

## Hypothèse

Supposant qu'au retour d'un livre, le livre peut avoir deux états différents

- correct : on le remet à disponible
- autre (mouillé, déchiré...) : le livre doit être réparé

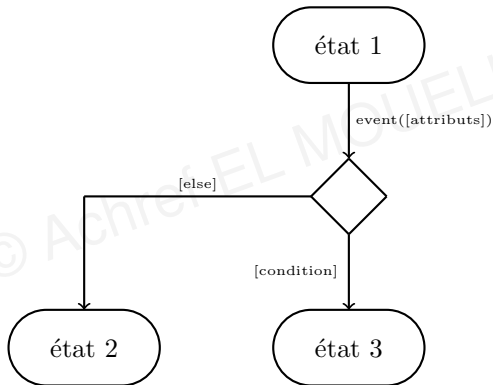
Dans ce cas, on a un événement et deux conditions complémentaires

Solution : utiliser les points de décision

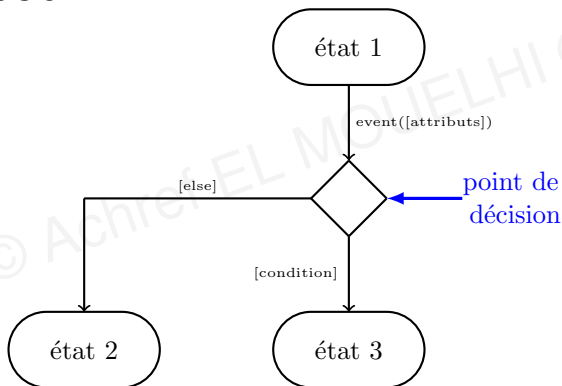
## Point de décision

- Permettant de préciser sur quel état il faut aller quand l'évènement est déclenché et que la condition de franchissement soit vraie ou fausse
- Modélisé en **UML 2** par un losange ayant une entrée et au moins deux sorties

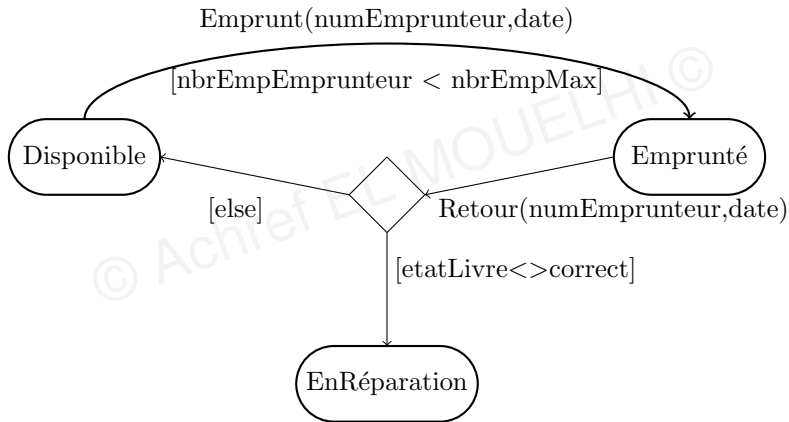
## États + évènement + transitions + condition de franchissement + point de décision



## États + évènement + transitions + condition de franchissement + point de décision



## Exemple



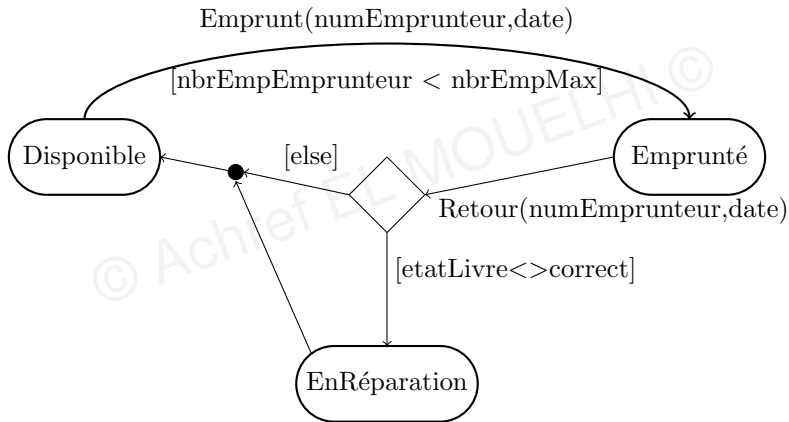
## Remarques

- Lorsque l'évènement est déclenché, il faut que les conditions de franchissement couvrent tous les cas possibles
- Il est possible d'avoir plusieurs conditions différentes sur le même point de décision

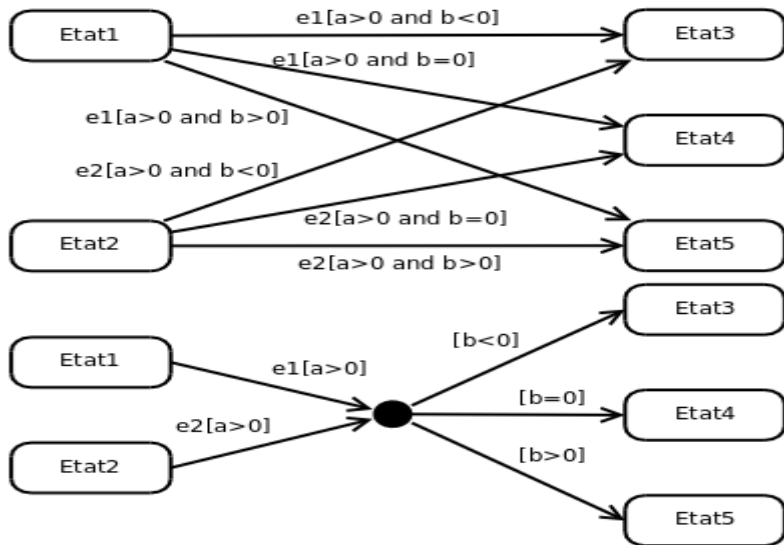
## Point de jonction

- nœud permettant de partager certaines transitions et de rendre le diagramme plus lisible
- Permettant de préciser sur quel état il faut aller quand l'évènement est déclenché et que la condition de franchissement soit vraie ou fausse
- Modélisé en **UML** 2 par un cercle plein
- Pouvant avoir une ou plusieurs entrée(s) et une ou plusieurs sortie(s)

## Exemple

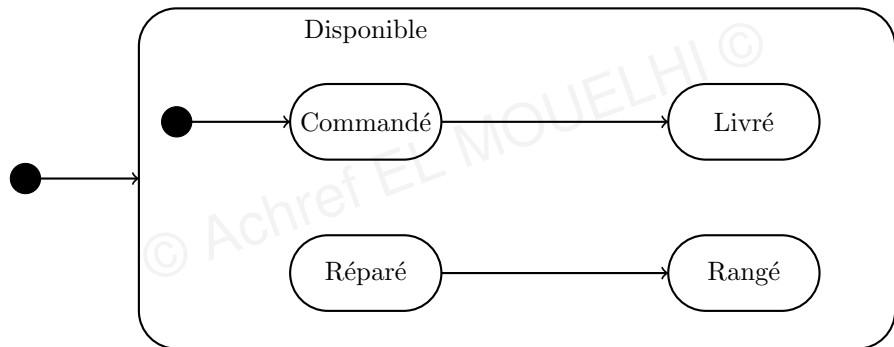


## Intérêt des points de jonction



Source : *UML 2 de l'apprentissage à la pratique* de Laurent Audibert

## Il est possible d'imbriquer les états



## Terminologie

- **L'état** Disponible : état composite
- **Les états** Commandé, Réparé, Livré et Rangé : états imbriqués

## Étapes

- Identifier les classes ayant un comportement temporel significatif
- Déterminer les différents états de chaque objet de la liste précédente
- Trouver les évènements et les conditions de franchissement
- Élaborer le diagramme états-transitions
- Essayer de le simplifier avec les points de décision et les imbrications

## Remarques

Chaque état doit avoir

- une transition entrante
- une transition sortante

## Exercice 1

- On considère le cas d'un système de gestion d'emprunt de livre pour une bibliothèque interne d'une entreprise. Le but de cet exercice est d'élaborer un diagramme états-transitions pour un objet de type `Livre` ayant les états suivants : disponible, réservé, emprunté, en réparation).
- Vérifier les conditions suivantes
  - tout employé, dont le nombre d'emprunt et réservation n'a pas dépassé le nombre max, peut réserver un livre si ce dernier a un état disponible
  - un livre peut rester 24 heures sous réservation, s'il n'a pas été emprunté il reprendra son état disponible. Sinon il passe à l'état emprunté
  - lors d'un retour, si la date de retour prévue n'a pas été respectée ou si le livre retourné est dans un état inacceptable, l'employé sera sanctionné. Il n'a donc plus le droit d'emprunter pour une durée bien déterminée
  - un livre retourné dans un état inacceptable est soit réparable (délai de réparation = 15 jours), après réparation, il sera disponible, soit irrécupérable et dans ce cas il passe à l'état final

## Exercice 2

- En se basant sur l'exercice 1, élaborer un diagramme états-transitions pour un objet de type `Employé` ayant les états suivants : abonné, empruntant, sanctionné, bloqué
- Vérifier les conditions suivantes
  - l'état de chaque employé n'ayant pas d'emprunt ou réservation encours est abonné
  - après un certain nombre de sanction, un employé, qui n'a pas d'emprunt encours, n'a plus le droit d'emprunter ni de réserver de livre, il arrive donc à son état final
  - un employé, ayant atteint le nombre max de sanction et ayant des emprunts encours, a l'état bloqué tant qu'il n'a pas retourné tous les livres empruntés. Lorsque cela est fait, il passe à l'état final