

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 MOUËLMI

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

État

Abstraction de valeurs d'un objet

© Achref EL MOUELHI ©

État

Abstraction de valeurs d'un objet

Comment représenter un état en UML 2 ?

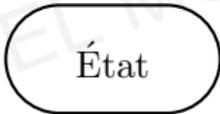


État

État

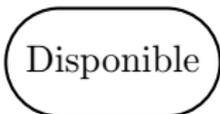
Abstraction de valeurs d'un objet

Comment représenter un état en UML 2 ?



État

Exemple : un livre peut avoir un état disponible



Disponible

Un objet de type `Livre` peut avoir plusieurs états

Disponible

Emprunté

© Achref EL MOUELHI ©

Un objet de type `Livre` peut avoir plusieurs états

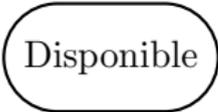
Disponible

Emprunté

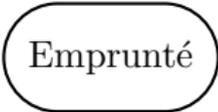
Tout objet a un état initial (schématisé différemment)



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

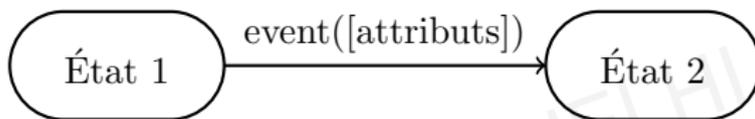
Exemple d'évènements pour un objet de type `Livre`

- Demande d'emprunt
- Enregistrement de retour

Transition

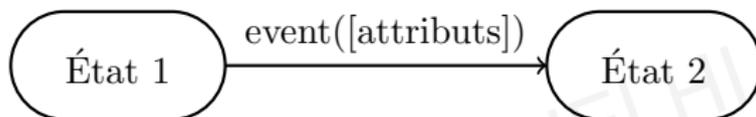
- Le passage d'un état vers un autre suite à un évènement
- Pouvant être automatique si on ne précise pas l'évènement déclencheur
- Représenté en **UML** par une flèche (arc)

États + transition + évènement



© Achref EL MOUJELHI ©

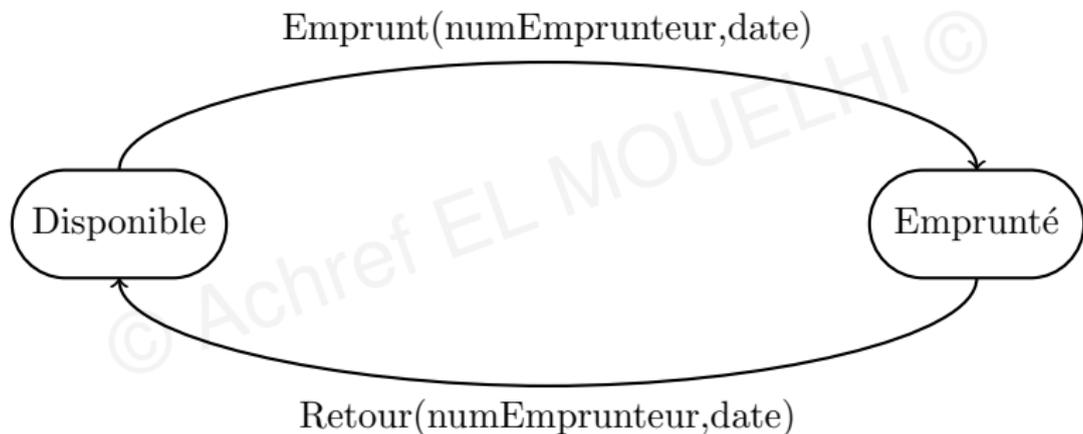
États + transition + évènement



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

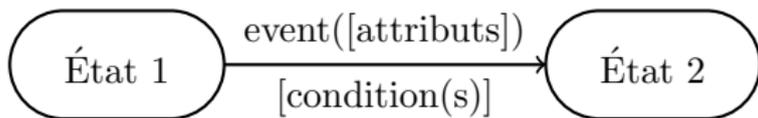
- D'après notre système, toute demande d'emprunt de livre est acceptée
- Dans le monde réel, il faut vérifier certaines conditions avant d'emprunter un livre
- Par exemple, il ne faut pas dépasser un nombre maximum d'emprunts par emprunteur

Solution

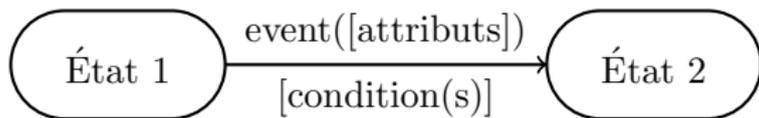
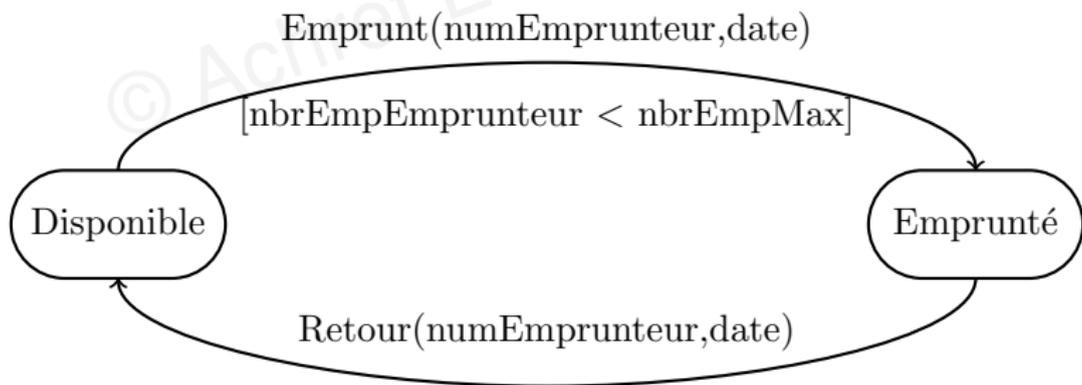
Définir des conditions de franchissement

Condition de franchissement (ou de garde)

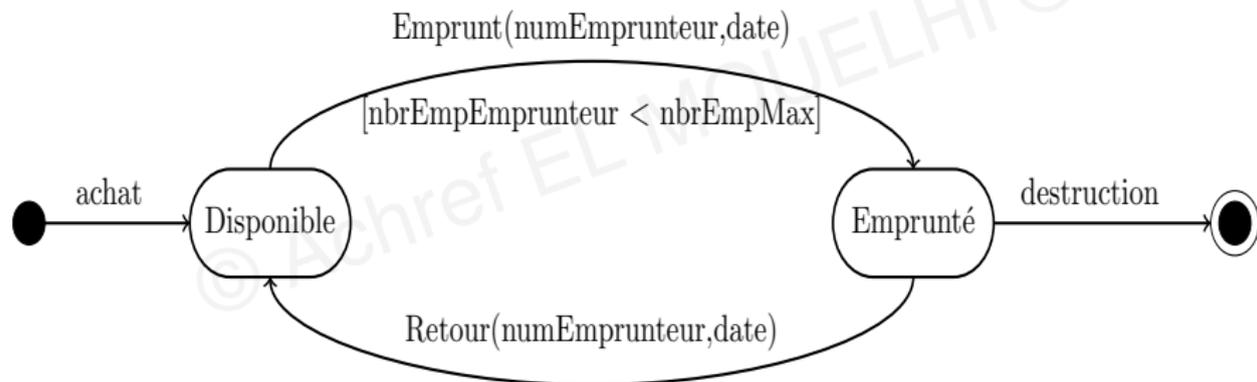
- une transition peut avoir une condition
- expression booléenne exprimée en langage naturelle (mathématique, logique...)
- évaluation uniquement lorsque l'événement se produit
 - si l'expression est fausse \Rightarrow la transition ne s'effectue pas
 - sinon, la transition s'effectue
- placée entre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

États + transition + évènement + condition de franchissement

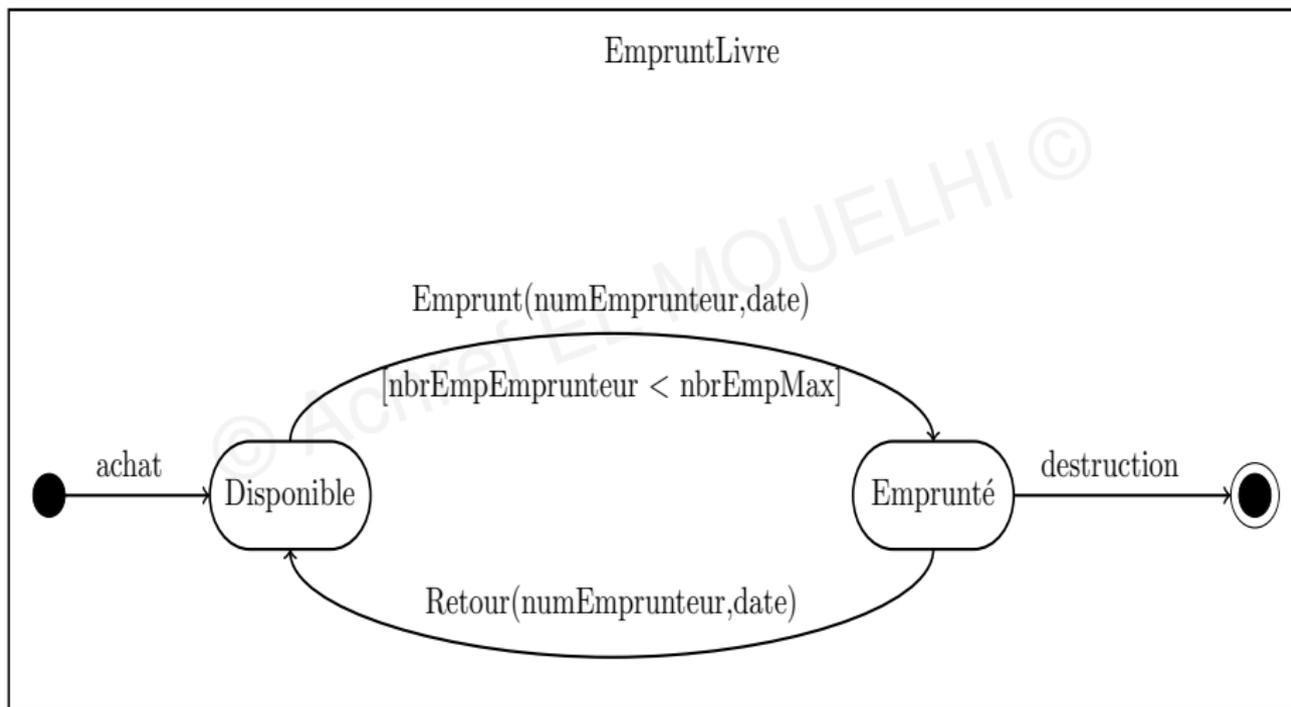
© Achref EL MOUELHI ©

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

© Achille

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

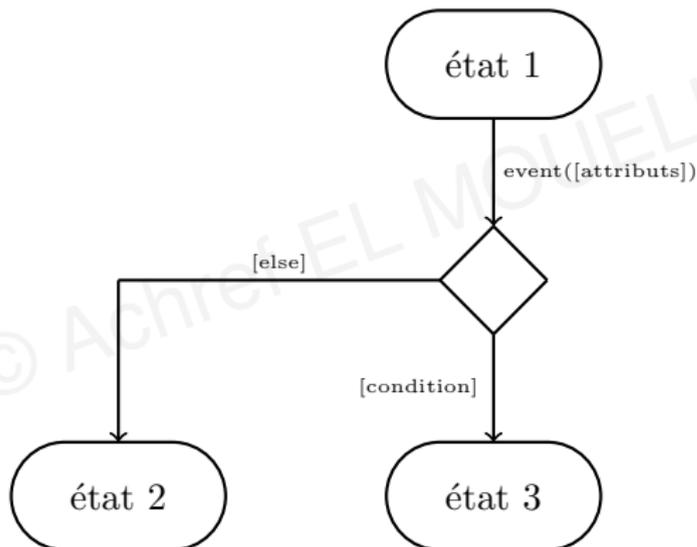
Dans ce cas, on a un évènement et deux conditions complémentaires

Solution : utiliser les points de décision

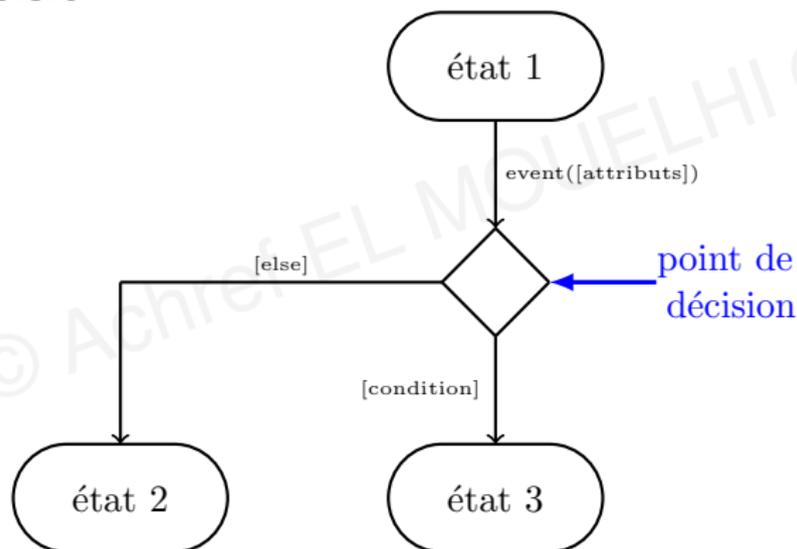
Point de décision

- 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

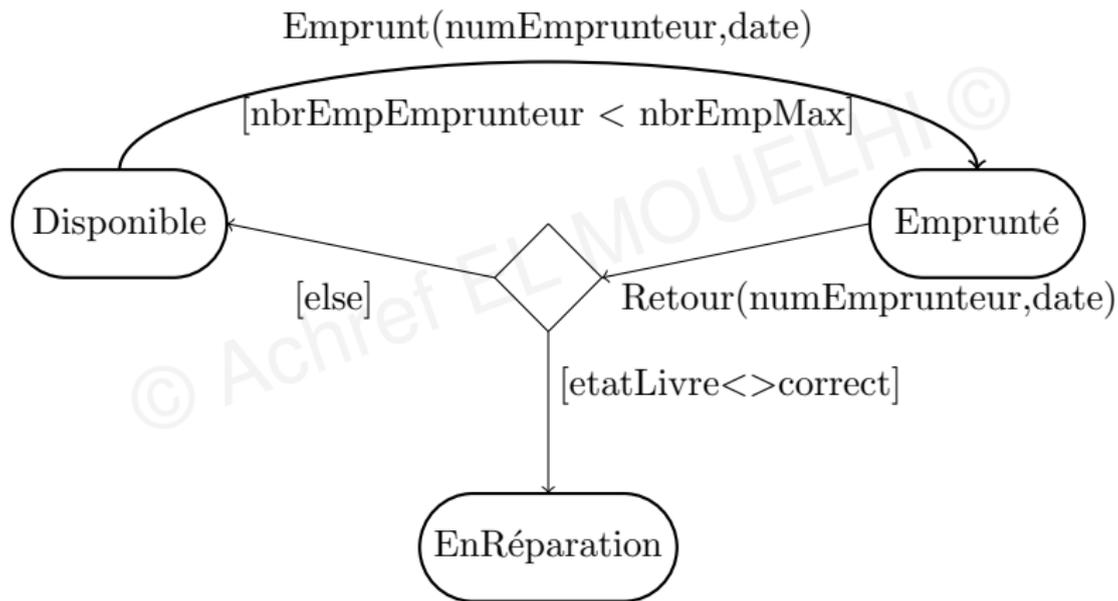
États + évènement + transitions + condition de franchissement + point de décision



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



Exemple



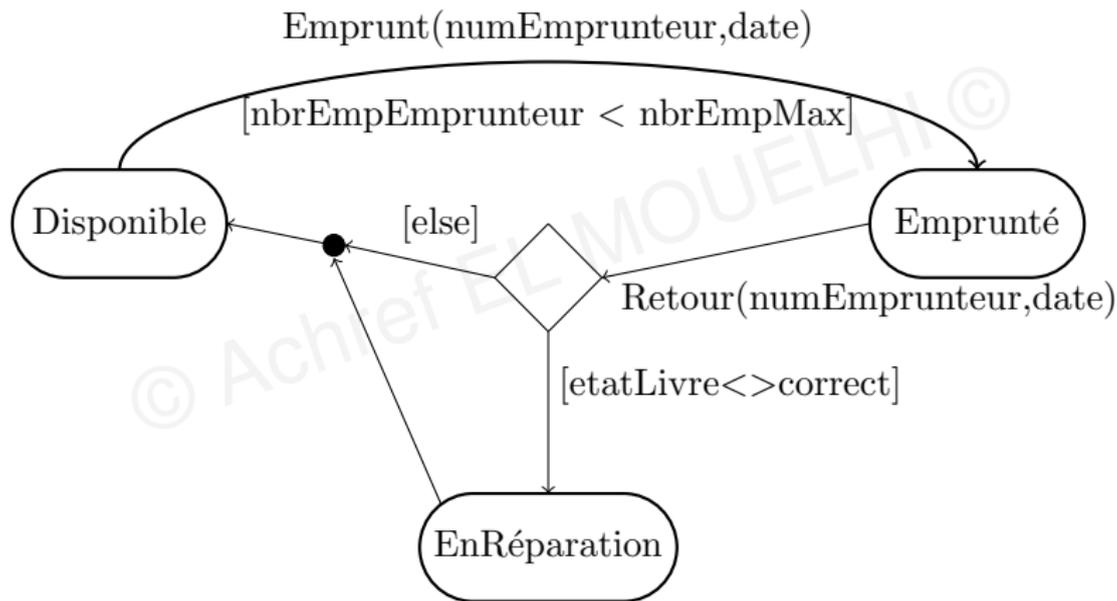
Remarques

- 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

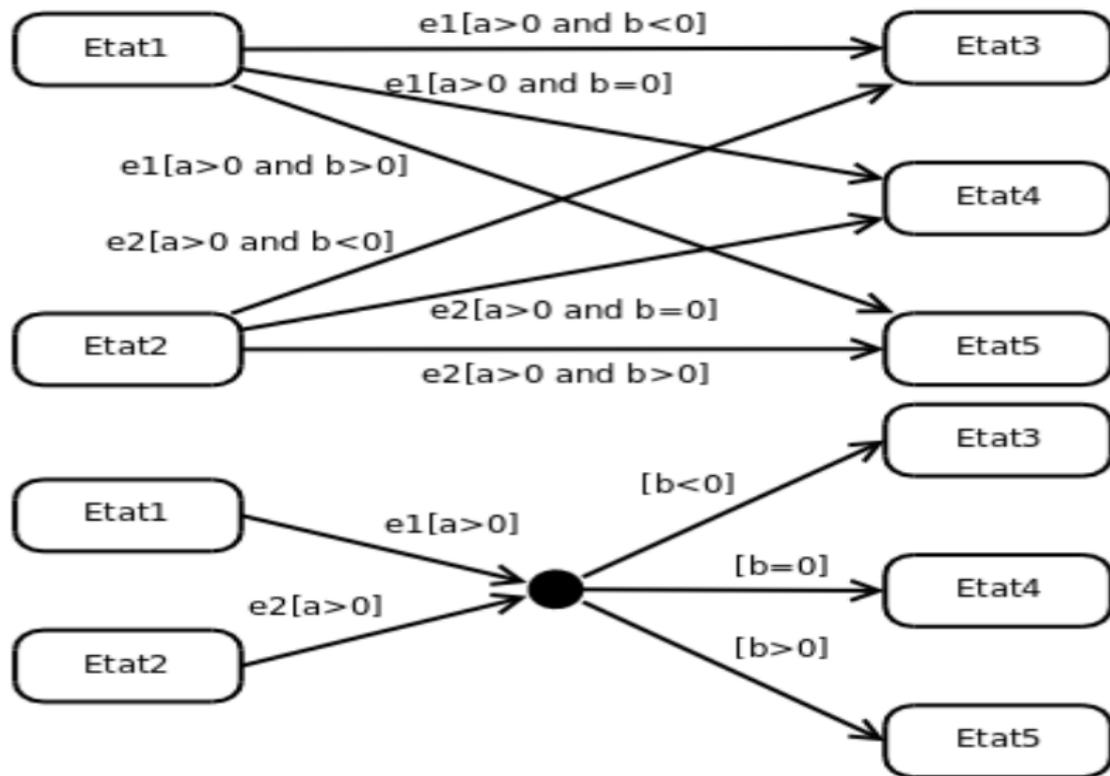
Point de jonction

- 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

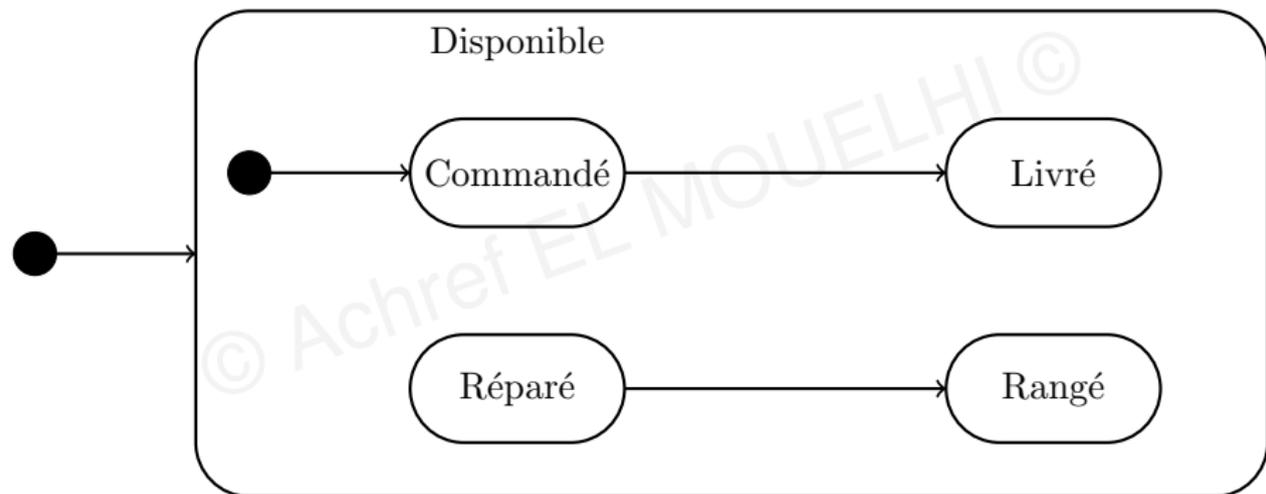


Intérêt des points de jonction



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 - Système de Gestion d'Emprunt de Livres

- Considérez le cas d'un système de gestion d'emprunt de livres pour une bibliothèque interne d'une entreprise. L'objectif de cet exercice est de créer un diagramme d'états-transitions pour un objet de type `Livre` avec les états suivants : `disponible`, `réservé`, `emprunté`, `en réparation`.
- Vérifiez les conditions suivantes :
 - Tout employé dont le nombre d'emprunts et de réservations n'a pas dépassé la limite maximale peut réserver un livre si ce dernier est dans l'état `disponible`.
 - Un livre peut rester sous réservation pendant 24 heures. S'il n'a pas été emprunté pendant ce délai, il redevient `disponible`. Sinon, il passe à l'état `emprunté`.
 - Lors du retour d'un livre, si la date de retour n'est pas respectée ou si l'état du livre est jugé inacceptable, l'employé sera sanctionné et perdra temporairement son droit d'emprunt.
 - Si le livre est retourné dans un état inacceptable, deux cas sont possibles :
 - Si le livre est réparable, une période de réparation de 15 jours sera nécessaire, après quoi le livre redevient `disponible`.
 - Si le livre est jugé irréparable, il atteint son état final.

Exercice 2 - Diagramme d'États-Transitions pour un Employé

- En vous basant sur l'exercice 1, créez un diagramme d'états-transitions pour un objet de type `Employé` avec les états suivants : `abonné`, `empruntant`, `sanctionné`, `bloqué`.
- Les règles suivantes doivent être respectées :
 - Un employé sans emprunts ni réservations en cours est dans l'état `abonné`.
 - Si un employé accumule un certain nombre de sanctions et n'a plus d'emprunts en cours, il perd le droit d'emprunter ou de réserver des livres, atteignant ainsi son état final.
 - Un employé ayant atteint le nombre maximal de sanctions mais possédant encore des emprunts en cours est placé dans l'état `bloqué`. Cet état persiste jusqu'à ce que tous les livres empruntés soient retournés. Une fois cela fait, l'employé atteint son état final.
 - Un employé ayant l'état `emprunteur` ou `abonné` ne peut pas être simultanément sous le coup de sanctions en cours.