Chapitre 5: Diagrammes UML : vue dynamique

2022-2023

Dr. Kouah Sofia

Plan

1. Diagrammes d’interaction (séquence et collaboration)
2. Diagramme d’états / transitions
3. Diagramme d’activités

Les diagrammes UML

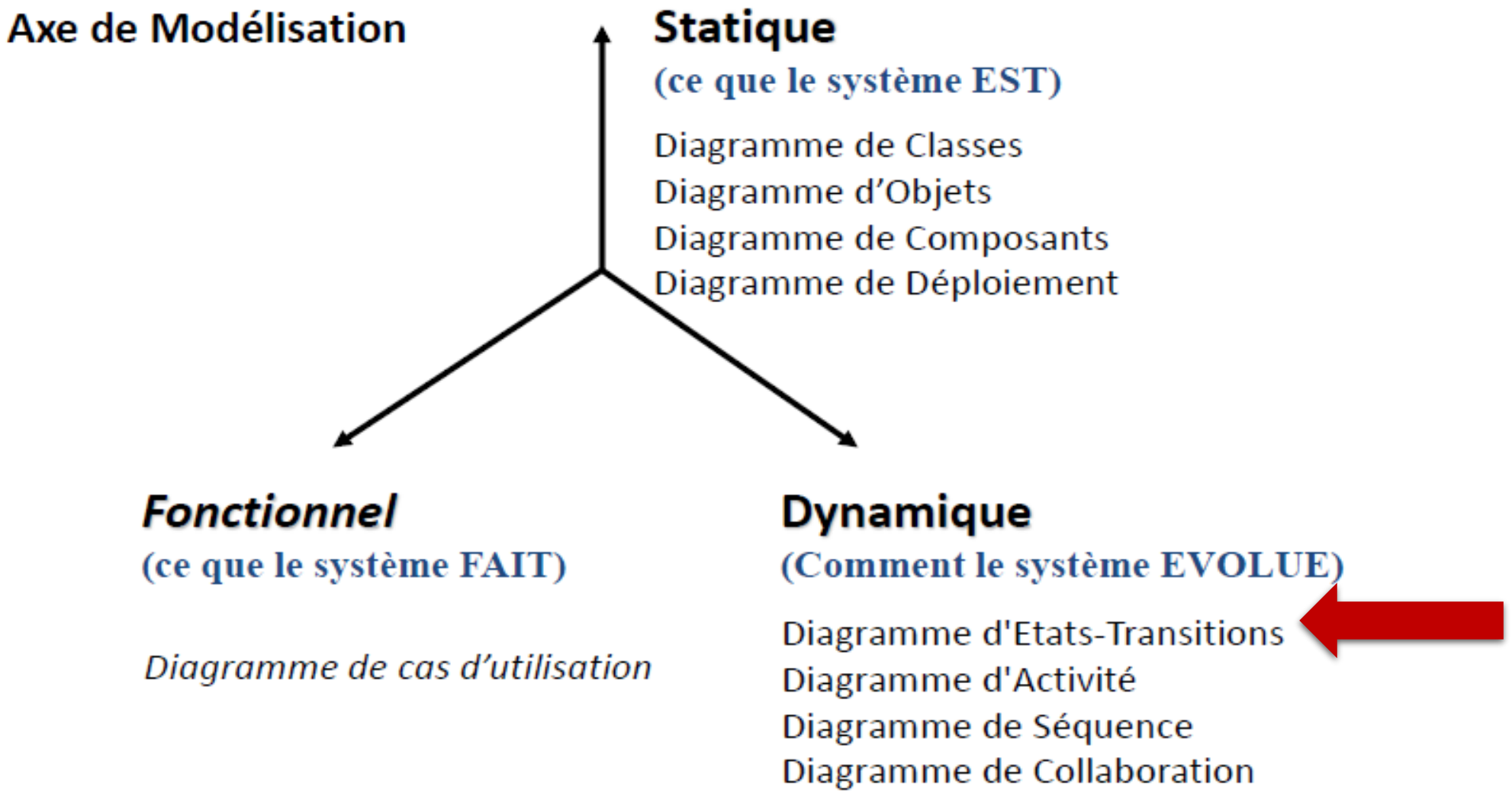


Diagramme d’états / Transitions

Un automate à états finis est la **spécification de la séquence d'états** que subira un **objet** au cours de son cycle de vie.

Un tel automate représente le **comportement** d'un **classeur** dont les sorties:

- ne dépendent pas seulement de ses entrées,

- mais aussi d'un **historique** des sollicitations passées.

Cet **historique** est caractérisé par un état.

Les objets **changent d'état en réponse à des événements extérieurs** donnant lieu à des transitions entre états.

Diagramme d’états / Transitions

- Les **états** sont représentés par des rectangles aux coins arrondis
- Les **transitions** sont représentées par des arcs orientés liant les états entre eux.
- Certains états, dits « **composites** » , peuvent contenir des sous-diagrammes.

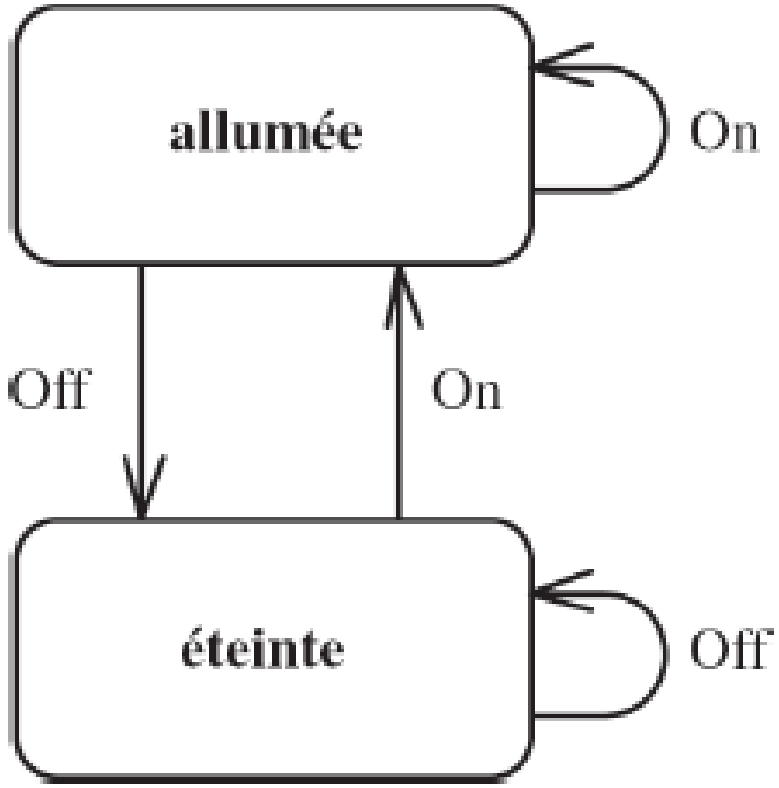


Diagramme d'états / Transitions

- L'organisation des états et des transitions pour un classeur donné est représentée dans un **diagramme d'états-transitions**.
- Le modèle dynamique comprend **plusieurs diagrammes d'états**.

Attention

Chaque diagramme d'états ne concerne qu'une seule classe.

Chaque automate à états finis s'exécute concurremment et peut changer d'état de façon indépendante des autres.

Diagramme d’états / Transitions (Exemple)

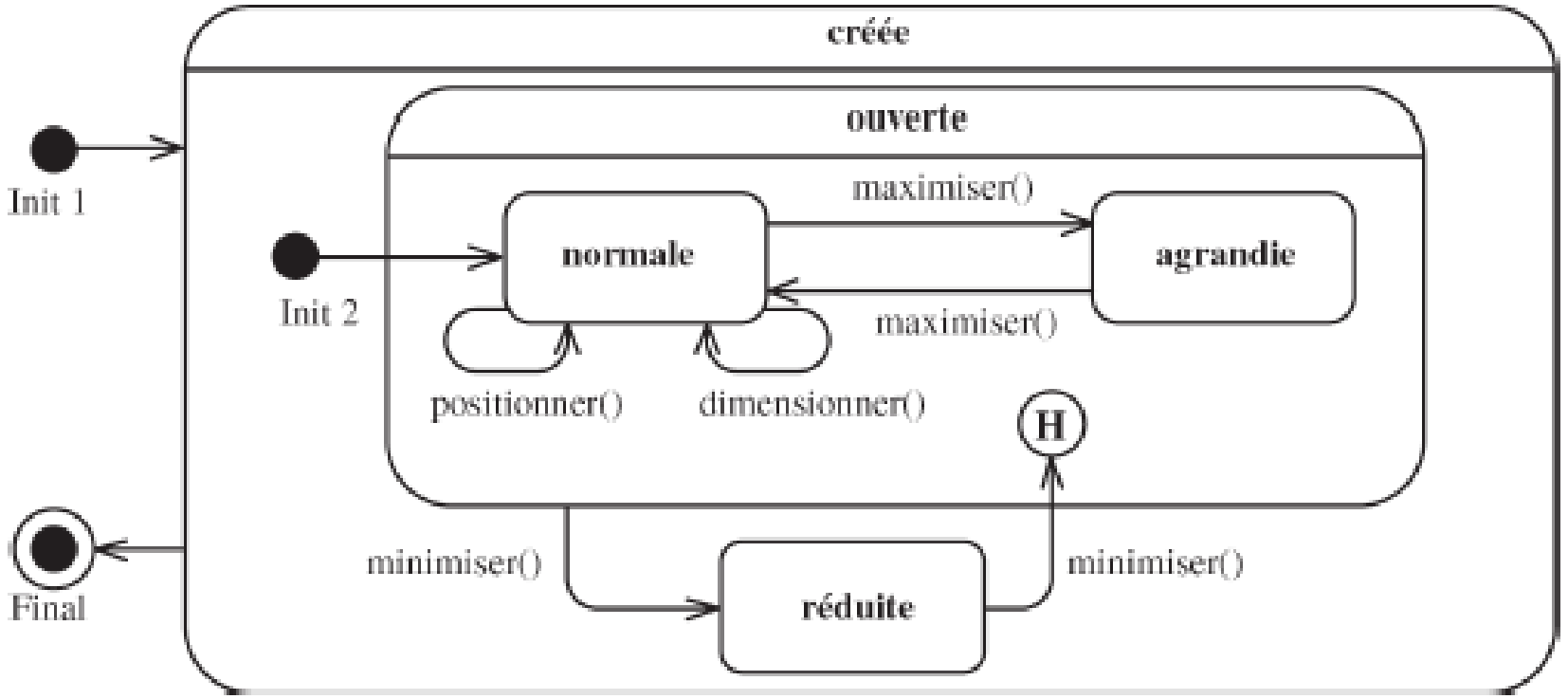


Diagramme d’états / Transitions

- L'état initial définit le point de départ par défaut pour l'automate ou le sous-état.
- Lorsqu'un objet est créé, il entre dans l'état initial:

- L'état final indique que l'exécution de l'automate ou du sous-état est terminée:

Diagramme d'états / Transitions

- Les transitions d'un diagramme d'états-transitions sont déclenchées par des **événements déclencheurs**.
  - Un appel de méthode sur l'objet courant génère un événement de type **call**.
  - Le passage de faux à vrai de la valeur de vérité d'une condition booléenne génère implicitement un événement de type **change**.
  - La réception d'un signal asynchrone, explicitement émis par un autre objet, génère un événement de type **signal**.
  - L'écoulement d'une durée déterminée après un événement donné génère un événement de type **after**. Par défaut, le temps commence à s'écouler dès l'entrée dans l'état courant.

L'événement déclencheur est indiqué à côte de la èche représentant la transition

Diagramme d’états / Transitions

Transition simple:

- Une transition entre deux états est représentée par un arc qui les lie l'un à l'autre.
- Elle indique qu'une instance peut changer d'état et exécuter certaines activités, si un événement déclencheur se produit et que les conditions de garde sont vérifiées.

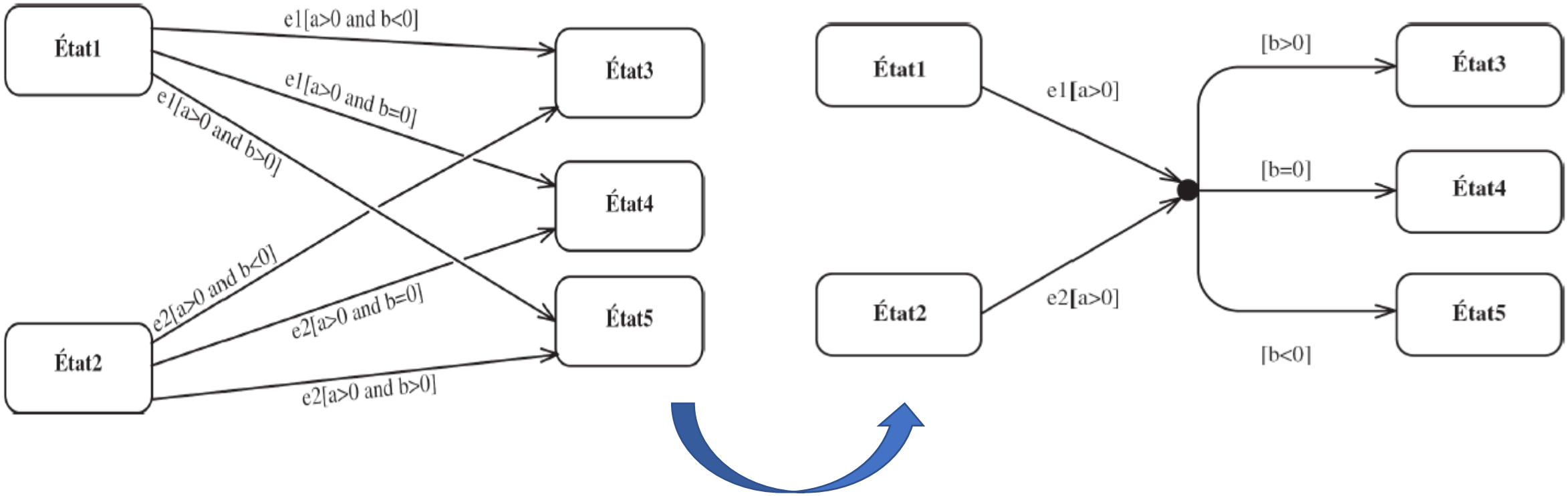
Diagramme d’états / Transitions

Point de Décision:

- On peut représenter des alternatives pour le franchissement d’une transition.
- On utilise pour cela des pseudo-états particuliers :
- **Les points de jonction (petit cercle plein)** permettent de partager des segments de transition.
- Ils ne sont qu'un raccourci d'écriture.
- Ils permettent des représentations plus compactes.
- **Les points de choix (losange)** sont plus que des raccourcis d'écriture.

Diagramme d’états / Transitions

Pont de Jonction



Pour emprunter un chemin, toutes **les gardes** le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.

Diagramme d’états / Transitions

Exemple d’un Point de jonction

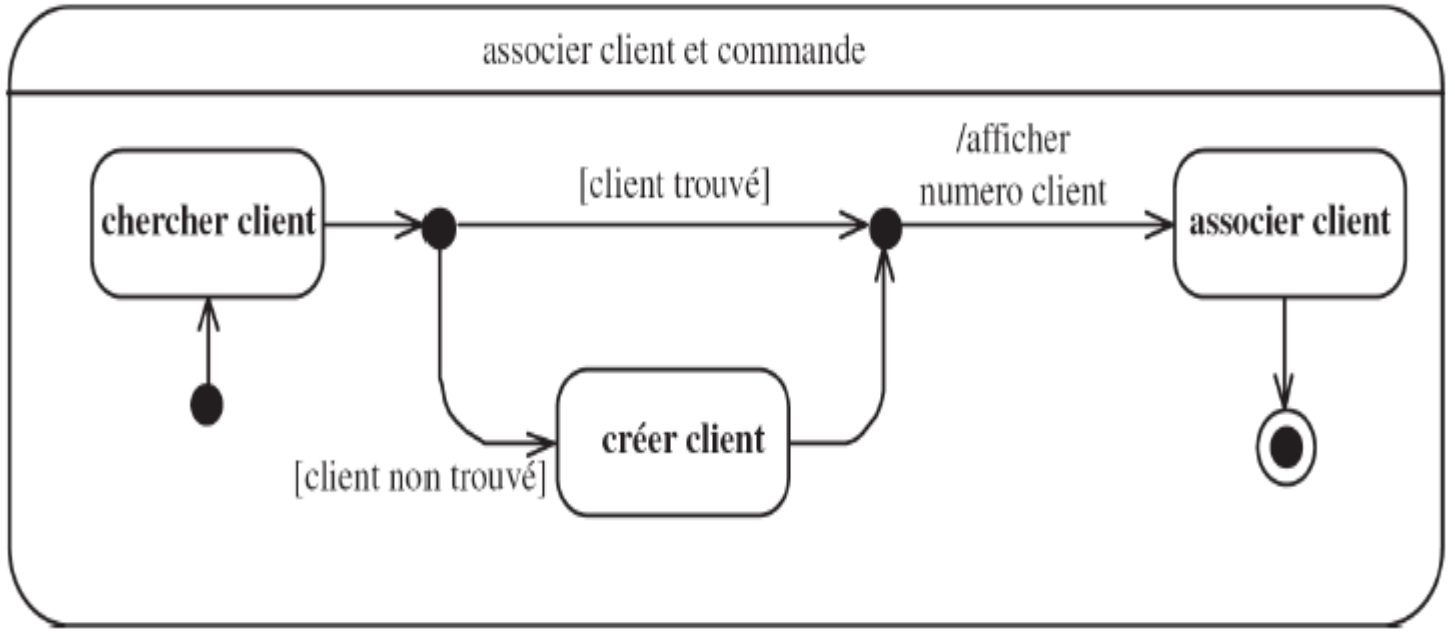


Diagramme d’états / Transitions

Point de choix

Les **gardes après le point de choix sont évaluées au moment où il est atteint**.

Cela permet de baser le **choix sur des résultats obtenus** en franchissant le segment avant le point de choix.

Si, quand le point de choix est atteint, aucun segment en aval n’est franchissable, le **modèle est mal formé.**

Contrairement aux points de jonction, les points de choix **ne sont pas** de simples raccourcis d'écriture.

Diagramme d’états / Transitions

Exemple d’un Point de choix

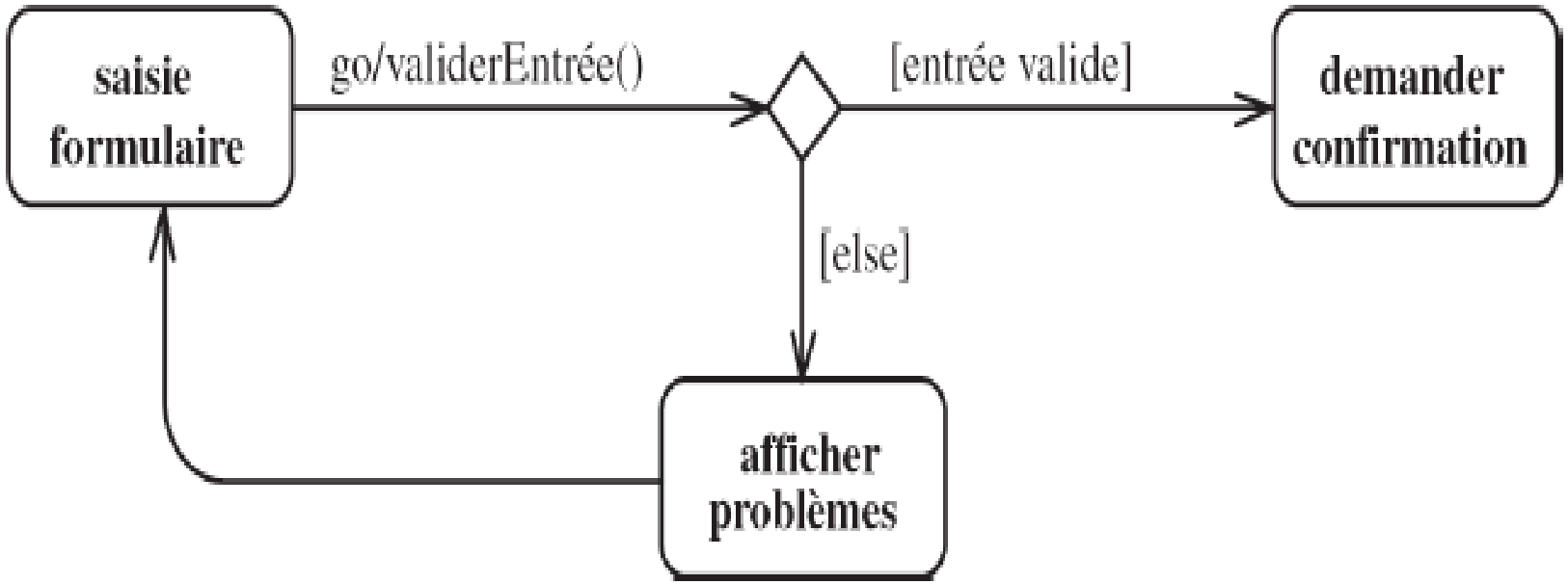


Diagramme d’états / Transitions

Transition Interne

- Un objet reste dans un état durant une certaine durée et **des transitions internes** peuvent intervenir.
- Une **transition interne** ne modifie pas l'état courant, mais suit globalement les règles d'une transition simple entre deux états.
- Trois déclencheurs particuliers sont introduits permettant le tir de transitions internes : **entry/, do/, et exit/.**

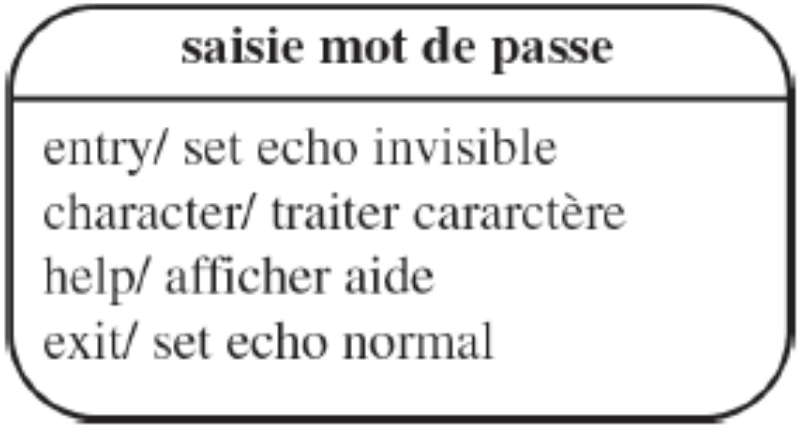


Diagramme d’états / Transitions

- **entry** : une activité à effectuer à chaque fois que **l'on rentre dans l'état considéré**.
- **exit:** activité à exécuter **quand on quitte l'état**.
- **do** : activité continue **qui est réalisée tant que l'on se trouve dans l'état**, ou jusqu'à ce que le calcul associé soit terminé.

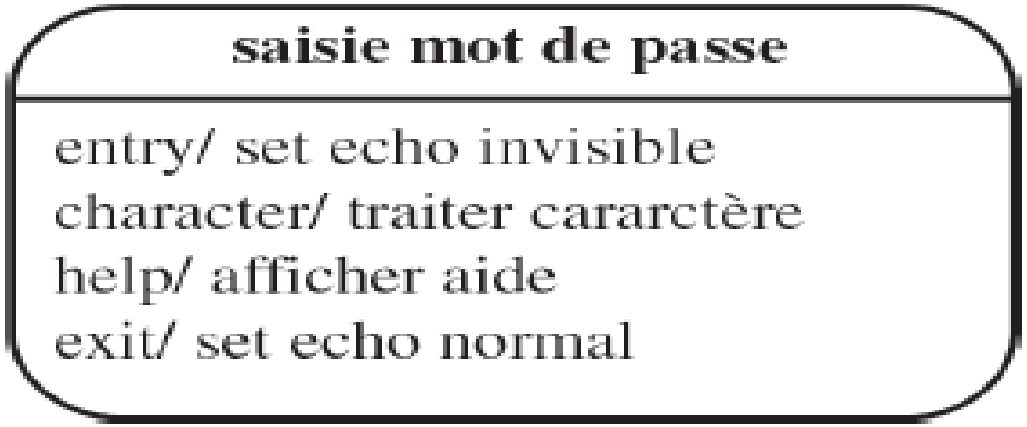
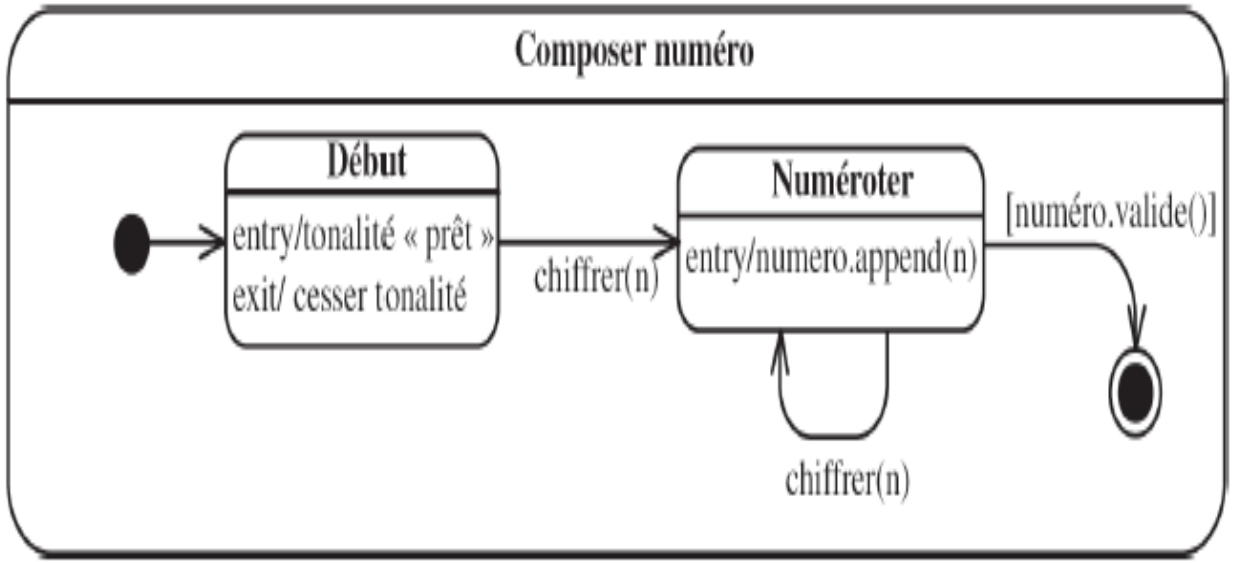


Diagramme d'états / Transitions

Etat composite

- **Un état composite**, par opposition à un état dit simple , est **<u>décomposé en deux ou plusieurs sous-états.</u>**
- Un **état composite** est représenté comme suit:



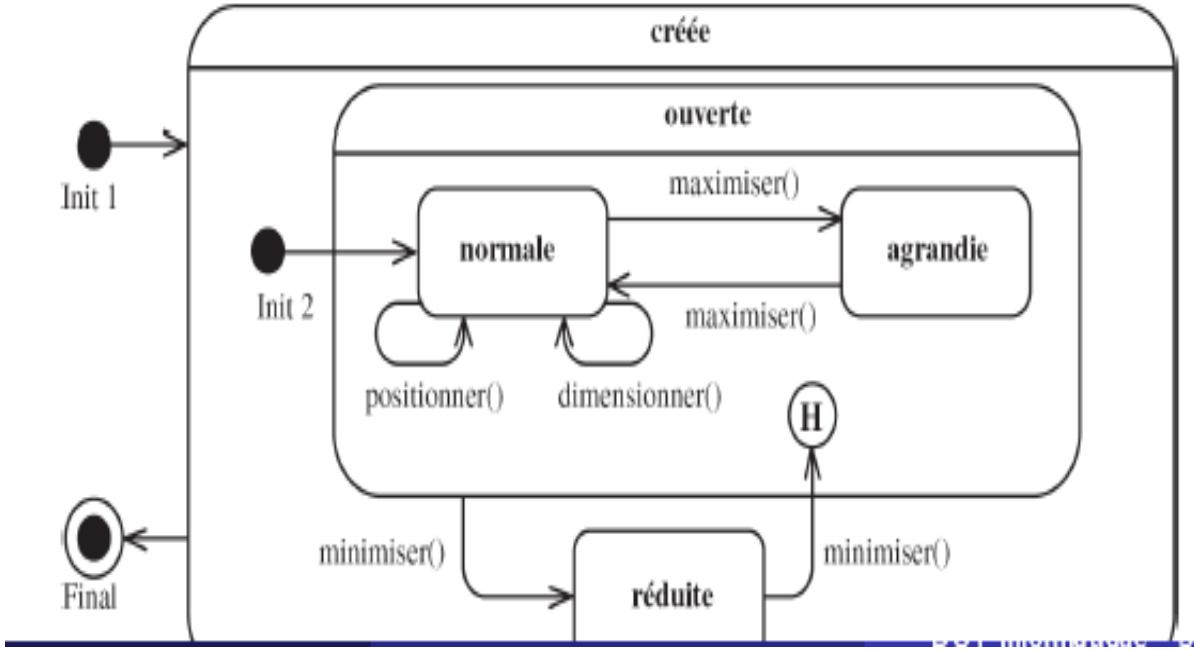


Diagramme d’états / Transitions

Pour pouvoir représenter un sous état indépendamment d’un macro-état, on a recours à des **points de connexion**.

Avec un X pour les **points de sortie**

Vides pour les **points d'entrée**

Ces interfaces permettent d'abstraire les sous-états des macro-états (réutilisabilité).

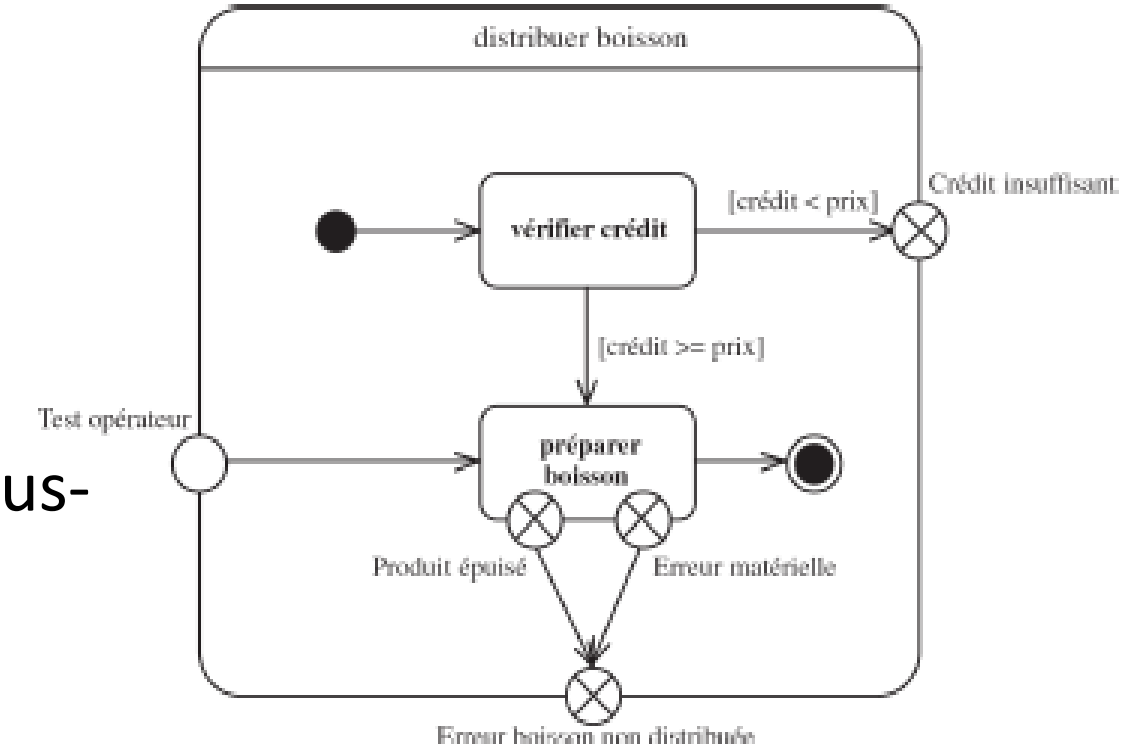


Diagramme d’états / Transitions

Etat Concurrent:

- Avec un **séparateur en pointillés**, on peu représenter **plusieurs automates s'exécutant indépendamment.**
- Un objet peut alors **être simultanément dans plusieurs états concurrents**.

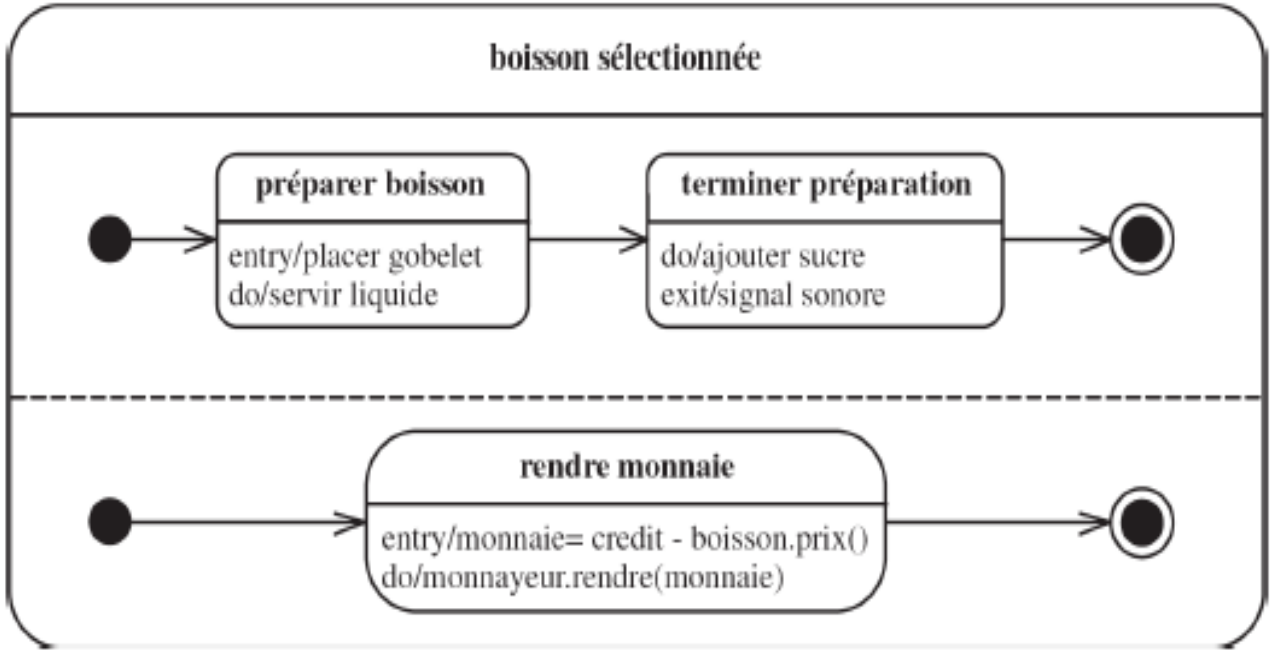


Diagramme d’états / Transitions (composants)

Transition Concurrente:

Une **transition fork** correspond à la création de deux états concurrentes.

Une **transition join** correspond à une barrière de synchronisation qui supprime la concurrence.

Pour pouvoir continuer leur exécution, toutes les **tâches concurrentes** doivent préalablement être prêtes à franchir la transition.