

# DIAGRAMME D'ACTIVITÉS



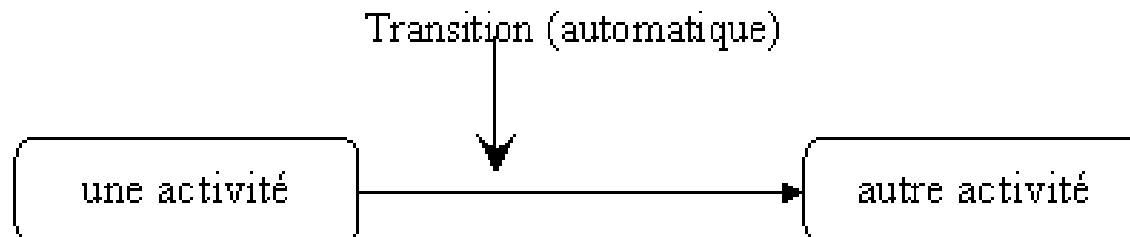
# Introduction



2

- Un diagramme d'activités (activités et transitions) **est une variante du diagramme d'états-transitions** (états et transitions). Les deux types de diagrammes permettent d'avoir deux vues différentes sur des automates donnés.
- Un diagramme d'activités visualise **un graphe d'activités qui modélise le comportement interne d'une méthode** (une réalisation d'une opération), d'un cas d'utilisation ou plus généralement d'un processus impliquant un ou plusieurs classificateurs (classes / cas d'utilisation / paquetages /...).
- Un diagramme d'activités représente l'état d'exécution d'un mécanisme, sous la forme d'un déroulement d'étapes regroupées séquentiellement dans des branches parallèles de flots de contrôle. Il ne représente ni la collaboration ni le comportement des objets. Il est utile pour la représentation des processus métiers et les cas d'utilisation.

- Le diagramme d'activités comprend :
  - **des activités** (une activité = une étape d'exécution, état-activité). Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles. Le passage d'une activité vers une autre est matérialisé par une transition.
  - **des transitions** qui sont automatiques entre activités, il est inutile également de préciser les événements. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre.
- En théorie, tous les mécanismes dynamiques pourraient être décrits par un diagramme d'activités, mais seuls les mécanismes complexes ou intéressants méritent d'être représentés.
- Activités et transition, notation :



# Action

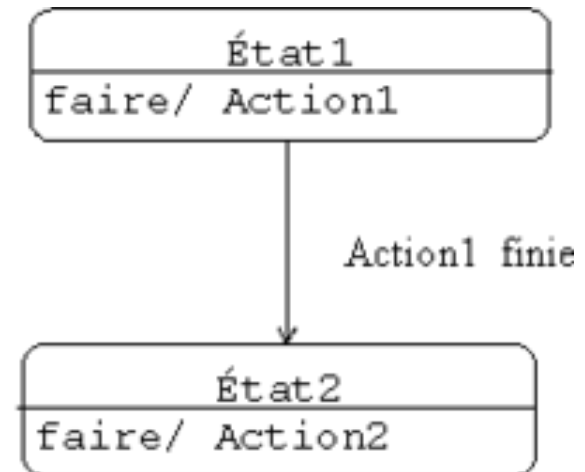
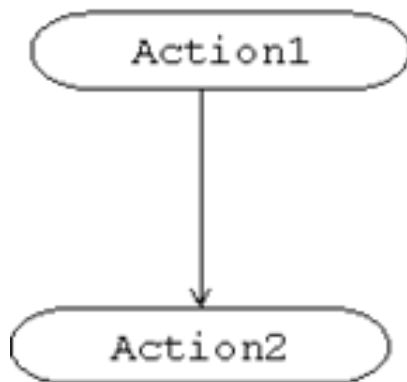
- Une action est le plus petit traitement qui puisse être exprimé en UML.
- Une action a une incidence sur l'état du système ou en extrait une information.
- Les actions sont des étapes discrètes à partir desquelles se construisent les comportements.
- La notion d'action est à rapprocher de la notion d'instruction élémentaire d'un langage de programmation (comme C++ ou Java).
- Une action peut être, par exemple :
  - ▣ une affectation de valeur à des attributs ;
  - ▣ un accès à la valeur d'une propriété structurelle (attribut ou terminaison d'association) ;
  - ▣ la création d'un nouvel objet ou lien ;
  - ▣ un calcul arithmétique simple ;
  - ▣ l'émission d'un signal ;
  - ▣ la réception d'un signal ;

# Activité & Groupe d'activités

- Une activité définit un comportement décrit par un séquençement organisé d'unités dont les éléments simples sont les actions.
- Le flot d'exécution est modélisé par des nœuds reliés par des arcs (transitions). Le flot de contrôle reste dans l'activité jusqu'à ce que les traitements soient terminés.
- Une activité est un comportement et à ce titre peut être associée à des paramètres.
- Un groupe d'activités est une activité regroupant des nœuds et des arcs. Les nœuds et les arcs peuvent appartenir à plus d'un groupe.
- Un diagramme d'activités est lui-même un groupe d'activités.

- Un **état-action** modélise une étape dans l'exécution d'un algorithme.
- Il consiste en un état simplifié dans lequel figure une action d'entrée et avec au moins une transition automatique vers un autre état, déclenchée par la fin de l'action d'entrée.
- Il correspond à l'exécution d'instructions atomiques.

- Un **état-activité** correspond à une exécution non atomique. Un état d'activité peut être décomposé en d'autres états-activité et/ou états-action.



# Définition & Exemple

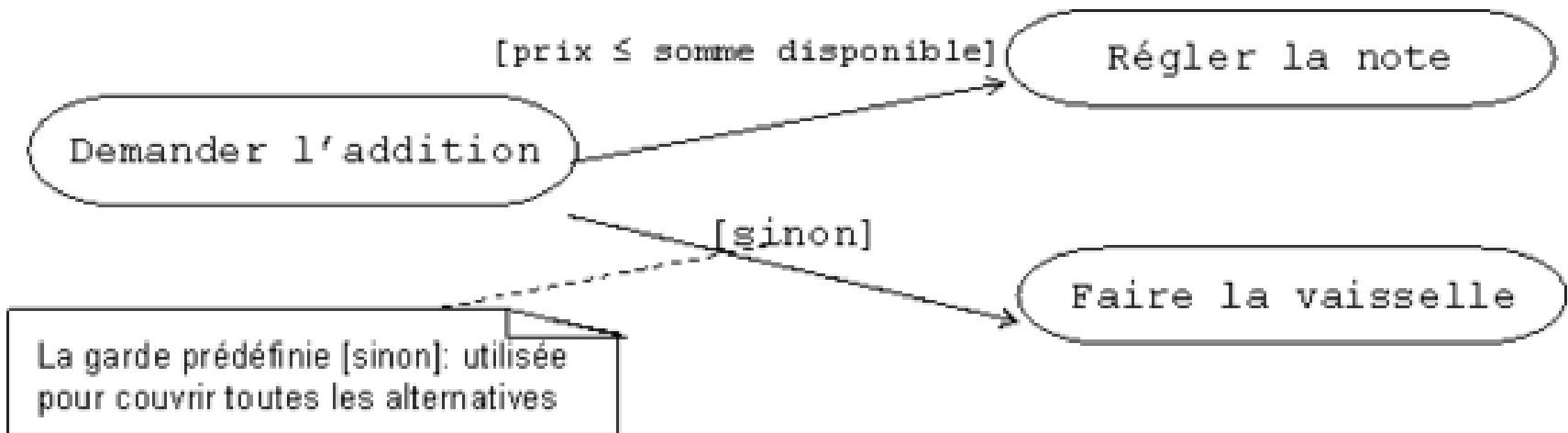
- Un état-action ou un état-activité est une étape particulière du traitement englobant.
- Ces étapes sont reliées par des transitions de complétude : **transition automatique** sans déclencheur ou aussi **transition de terminaison** (fin de l'activité actuelle déclenche automatiquement l'activité suivante).





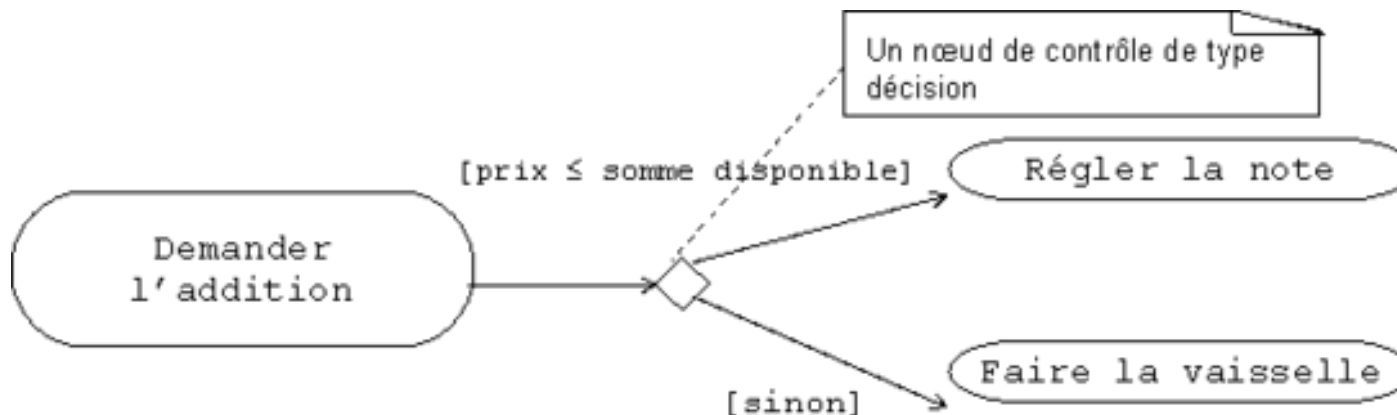
# Transition conditionnelle

- Il est possible de représenter des **transitions conditionnelles** en utilisant des gardes (appelées aussi décisions) qui doivent être mutuellement exclusives.



# Transition composite

- Une décision peut aussi utiliser des **transitions composites** et créer un point de jonction. Le point de jonction (décision) :
  - ▣ matérialisé par un losange
  - ▣ en entrée une seule transition
  - ▣ en sortie plusieurs transitions

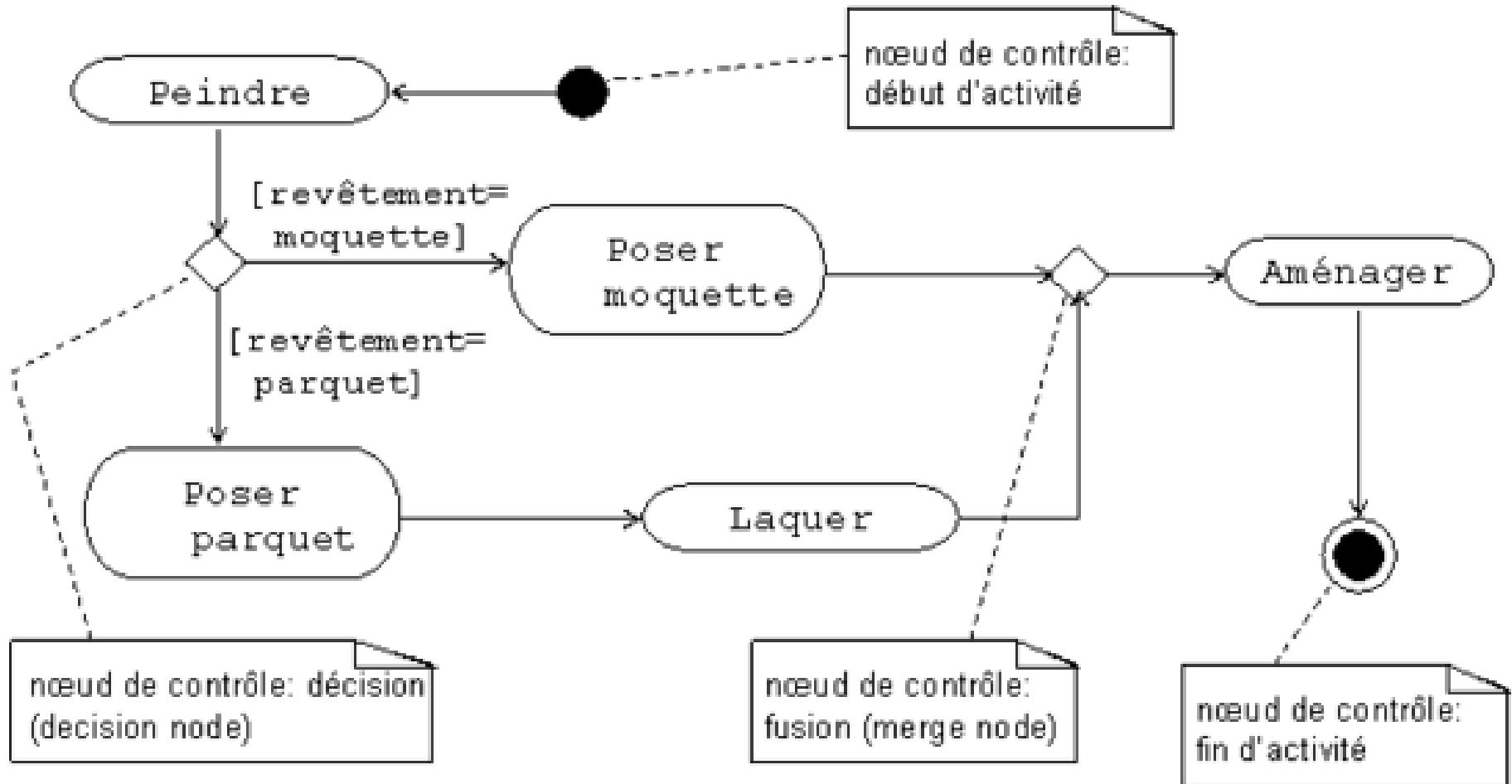


# Transition composite (Suite)

- À un point de jonction peuvent se rejoindre des chemins séparés par de précédentes décisions.
- Ce point reçoit plusieurs **transitions gardées** ou non en entrée (flots entrant) mais il permet une seule transition en sortie (flot sortant).
- Il n'est pas utilisé pour synchroniser des flots concurrents mais pour accepter un parmi plusieurs

# Transition composite (Suite)

12

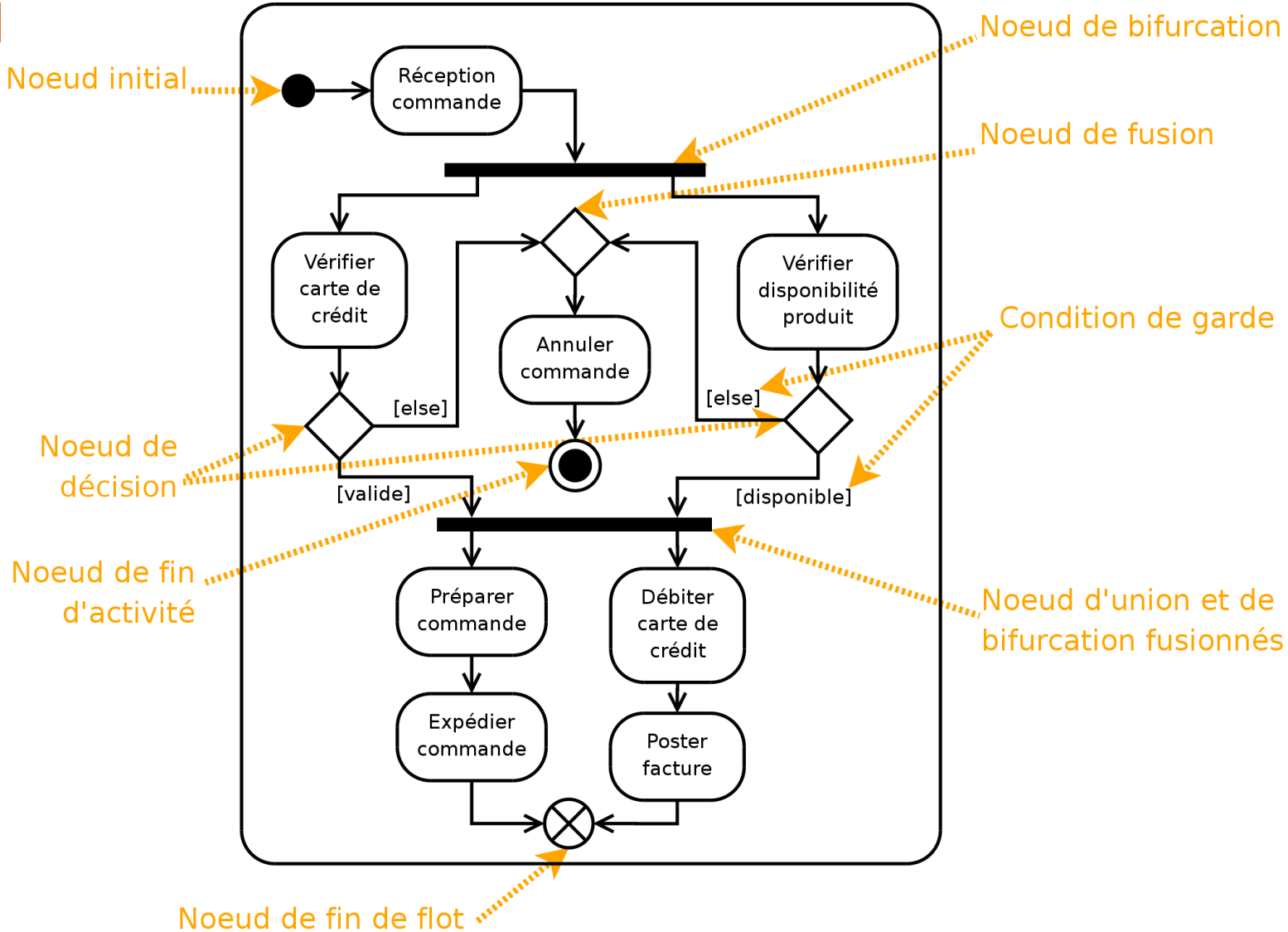


# Nœuds de contrôle

- Un nœud de contrôle est un nœud d'activité abstrait utilisé pour coordonner les flots entre les nœuds d'une activité.
- Il existe plusieurs types de nœuds de contrôle :
  - ▣ nœud initial (initial node) ;
  - ▣ nœud de fin d'activité (final node) ;
  - ▣ nœud de fin de flot (flow final node) ;
  - ▣ nœud de décision (decision node) ;
  - ▣ nœud de fusion (merge node) ;
  - ▣ nœud de bifurcation (fork node) ;
  - ▣ nœud d'union (join node).

# Types de nœuds contrôle

14



# Types de nœuds de contrôle (Suite)



15

## □ Nœud initial

- Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Une activité peut avoir plusieurs nœuds initiaux. Un nœud initial possède un arc sortant et pas d'arc entrant.
- Graphiquement, un nœud initial est représenté par un petit cercle plein .

## □ Nœud final

- Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant.

## □ Nœud de fin d'activité

- Lorsque l'un des arcs d'un nœud de fin d'activité est activé, l'exécution de l'activité enveloppante s'achève et tout nœud ou flot actif au sein de l'activité enveloppante est abandonné.
- Graphiquement, un nœud de fin d'activité est représenté par un cercle vide contenant un petit cercle plein.

# Types de nœuds de contrôle (Suite)

## □ Nœud de fin de flot

- Lorsqu'un flot d'exécution atteint un nœud de fin de flot, le flot en question est terminé, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante.
- Graphiquement, un nœud de fin de flot est représenté par un cercle vide barré d'un X.

## □ Un nœud de décision :

- est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. Ces derniers sont généralement accompagnés de conditions de garde pour conditionner le choix.
- Graphiquement, on représente un nœud de décision par un losange



# Types de nœuds de contrôle (Suite)

- **Nœud de fusion (merge node)**
  - Un nœud de fusion est un nœud de contrôle qui rassemble plusieurs flots alternatifs entrants en un seul flot sortant. Il n'est pas utilisé pour synchroniser des flots concurrents (c'est le rôle du nœud d'union) mais pour accepter un flot parmi plusieurs.
  - Graphiquement, on représente un nœud de fusion, comme un nœud de décision, par un losange .

# Types de nœuds de contrôle (Suite)

- **Nœud de bifurcation ou de débranchement (fork node)**
  - ▣ Un nœud de bifurcation, également appelé nœud de débranchement est un nœud de contrôle qui sépare un flot en plusieurs flots concurrents. Un tel nœud possède donc un arc entrant et plusieurs arcs sortants.
  - ▣ Graphiquement, on représente un nœud de bifurcation par un trait plein
- **Nœud d'union ou de jointure (join node)**
  - ▣ Un nœud d'union, également appelé nœud de jointure est un nœud de contrôle qui synchronise des flots multiples. Un tel nœud possède donc plusieurs arcs entrants et un seul arc sortant.
  - ▣ Graphiquement, on représente un nœud de union, comme un nœud de bifurcation, par un trait plein.

# Synchronisation :

## Disjonction & Conjonction d'activités



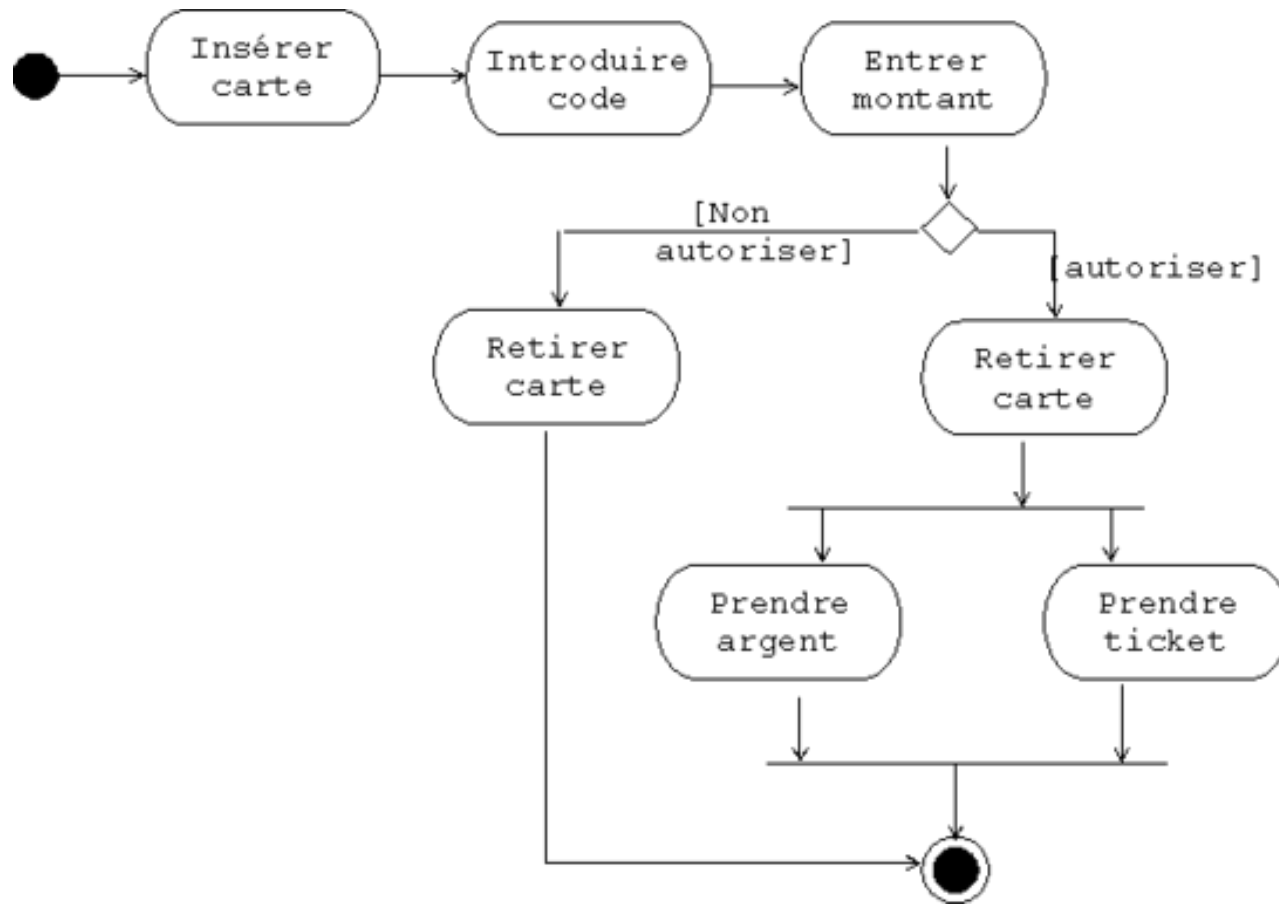
19

- Il est possible de synchroniser les transitions à l'aide des "**barres de synchronisation**" (comme dans les diagrammes d'états-transitions).
- Une barre de synchronisation permet d'ouvrir et de fermer des branches (chemins) parallèles au sein d'un flot d'exécution d'une méthode ou d'un cas d'utilisation.
- Les transitions qui partent d'une barre de synchronisation ont lieu en même temps.
  - ▣ **Union** : Une barre de synchronisation ne peut être franchie que lorsque toutes les transitions en entrée sur la barre sont déclenchées.
  - ▣ **Bifurcation** : Les transitions de débranchement au départ d'une barre de synchronisation sont déclenchées simultanément.

# Exemple : Synchronisation

20

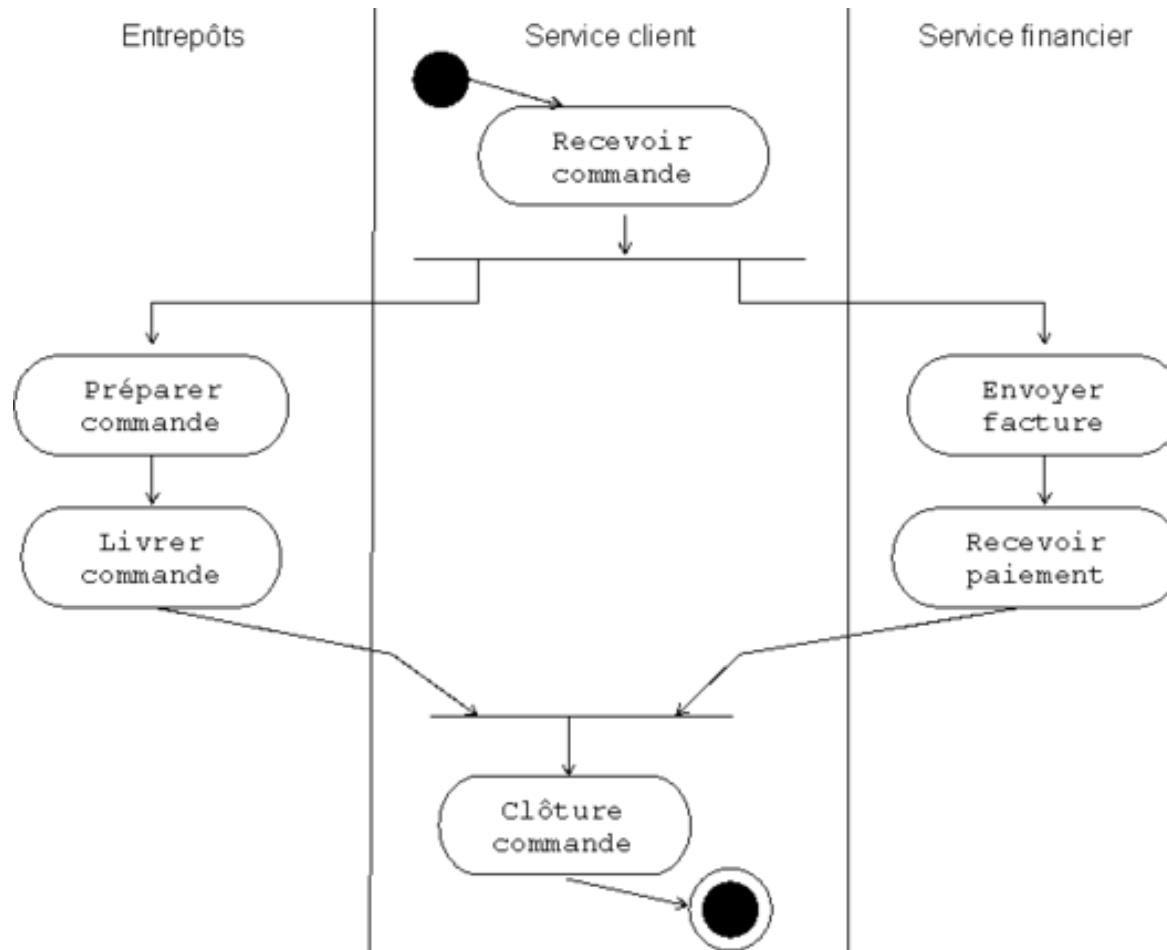
- Dans un diagramme d'activités, la même action peut être exécutée plusieurs fois.



- Les diagrammes d'activités indiquent ce qui se passe sans préciser qui fait quoi (en terme de programmation, ils ne précisent pas quelle classe est responsable et en terme de processus métier, ils ne précisent pas quelle partie de l'organisation exécute chaque action).
- Il est possible de diviser un diagramme d'activités en partitions ou couloirs d'activités (travées, swimlanes). Chaque partition montre quelles actions sont exécutées par une classe ou une unité organisationnelle.

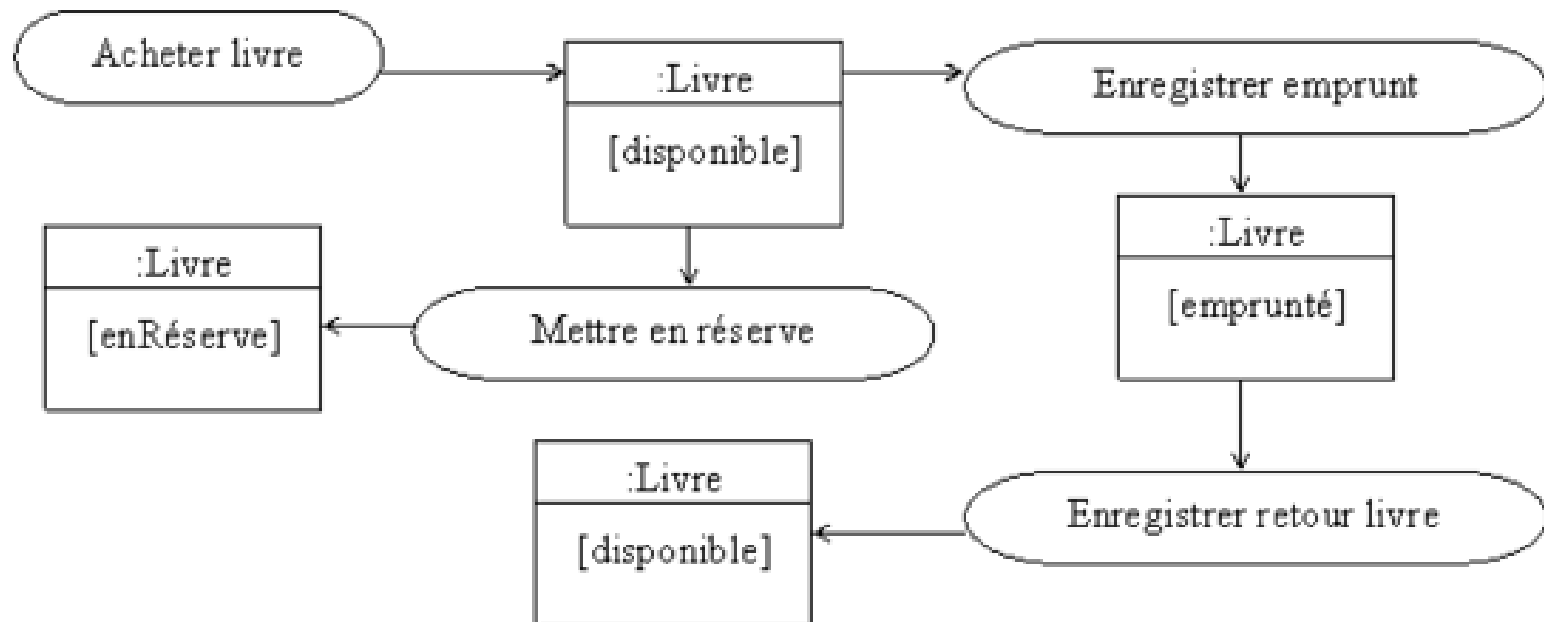
# Exemple : Couloirs d'activités

## □ Diagramme d'activité avec trois couloirs d'activité



# Exemple 1: Flots d'objets

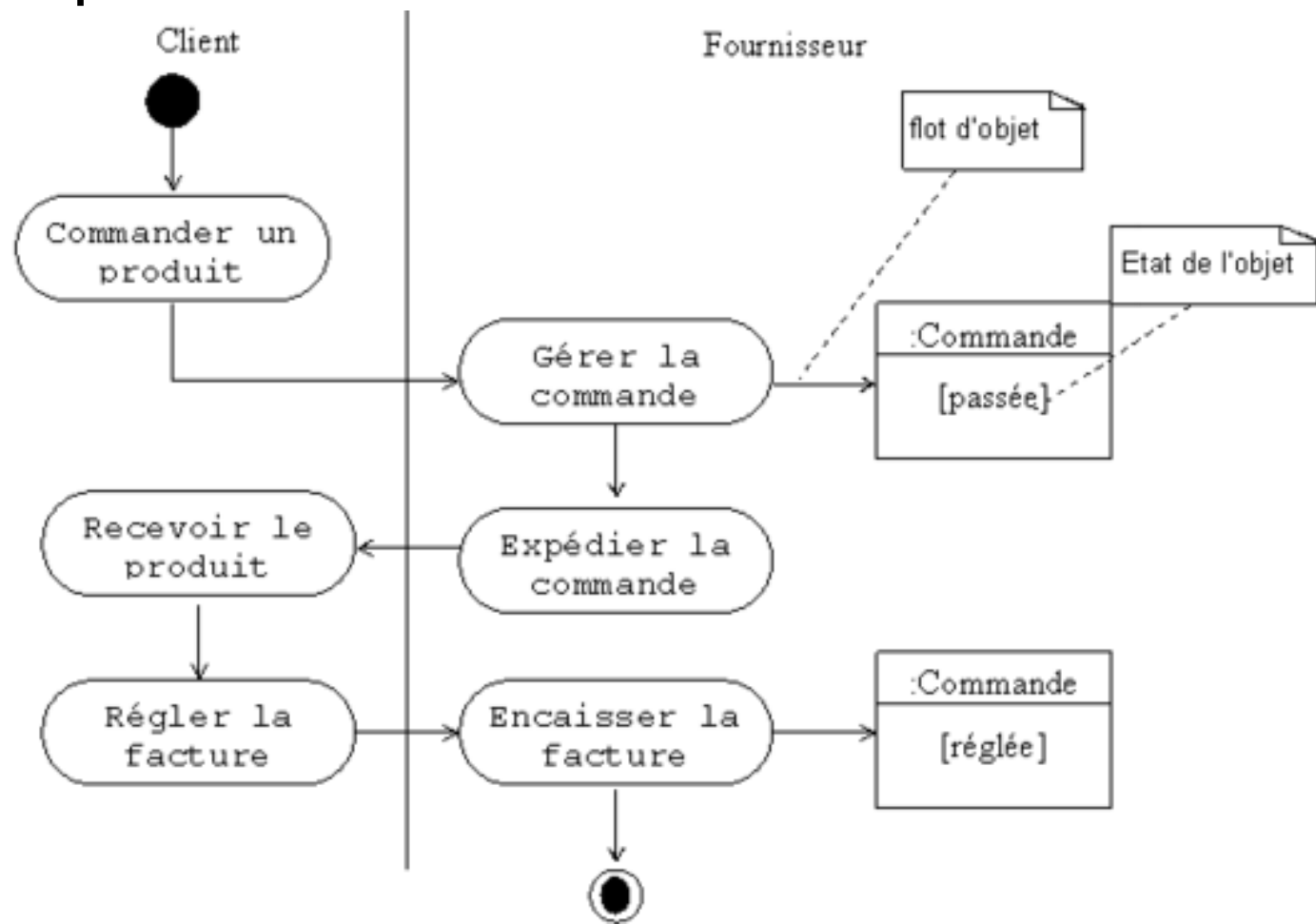
- Dans le diagramme d'activités, les objets peuvent être représentés. Ils sont utilisés ou modifiés par des actions. Ils peuvent aussi initier des actions.



# Exemple 2 : Flots d'objets

24

- Diagramme d'activité avec deux couloirs d'activité et un objet





# Exemple récapitulatif

25

