

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

Axe de Modélisation

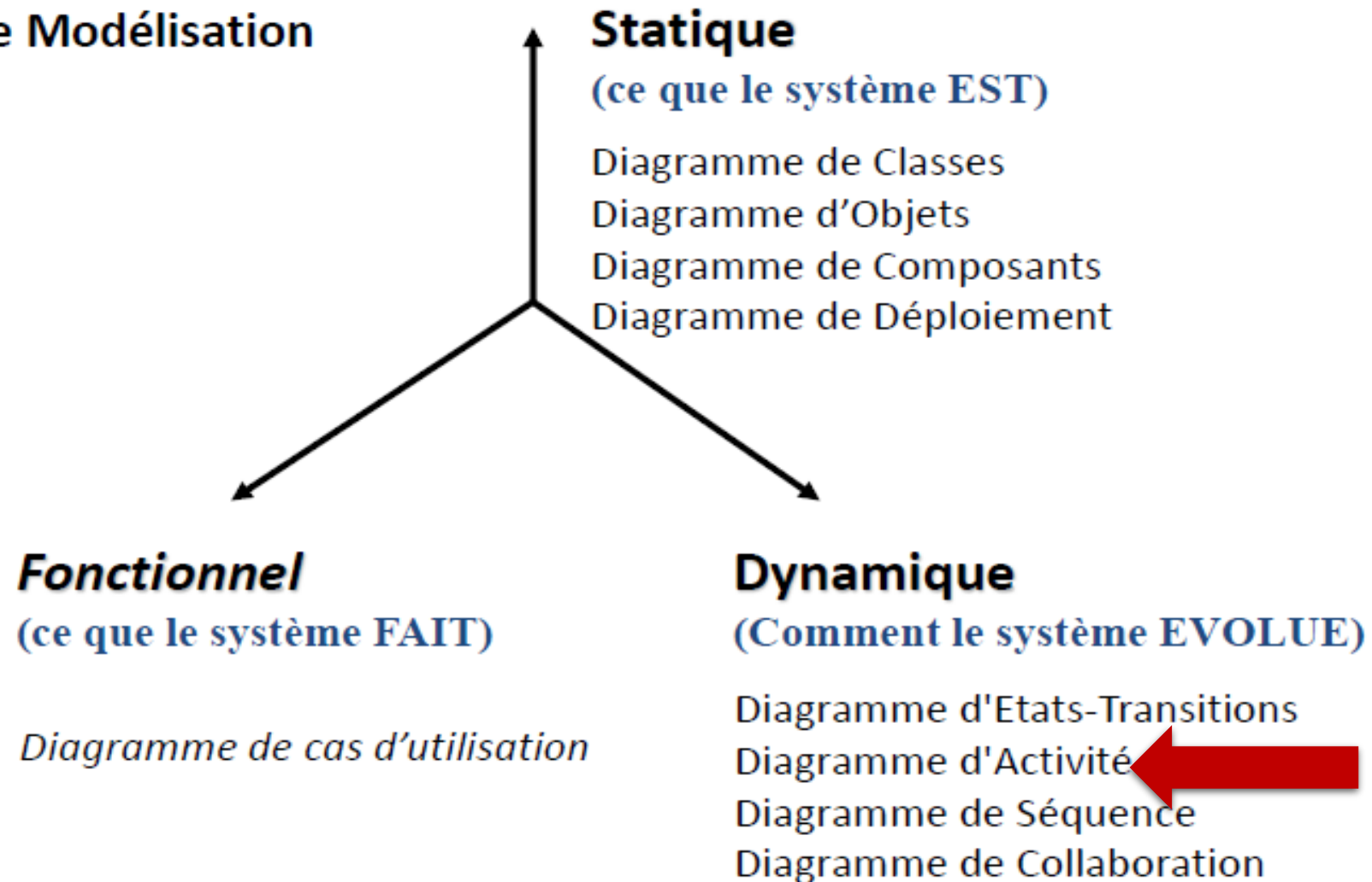


Diagramme d'activités

- Les **diagrammes d'activité** offrent une manière graphique et non ambiguë pour modéliser les traitements.
 - Comportement d'une méthode
 - Déroulement d'un cas d'utilisation
- Une **activité** représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles. Le passage d'une activité à l'autre est matérialisé par une **transition**.
- Ces diagrammes sont assez semblables aux états-transitions mais avec une interprétation différente.

Diagramme d'activités

- Les diagrammes d'**états-transitions** sont définis pour chaque classeur et n'en font pas intervenir plusieurs.
- A l'inverse, les **diagrammes d'activité** permettent une description s'affranchissant (partiellement) de la structuration de l'application en classeurs.
- La vision des diagrammes d'activités se rapproche des langages de programmation impérative (C, C++, Java)
 - Les états représentent des calculs
 - Il n'y a pas d'événements externes mais des attentes de fins de calculs
 - Il peut y avoir de la concurrence entre activités

Diagramme d'activités

Les **activités** décrivent un traitement.

- Le flot de contrôle reste dans l'activité jusqu'à ce que les traitements soient terminés.
- On peut définir des variables locales à une activité et manipuler les variables accessibles depuis le contexte de l'activité (classe contenant en particulier).
- Les activités peuvent être imbriquées hiérarchiquement : on parle alors d'activités composites.

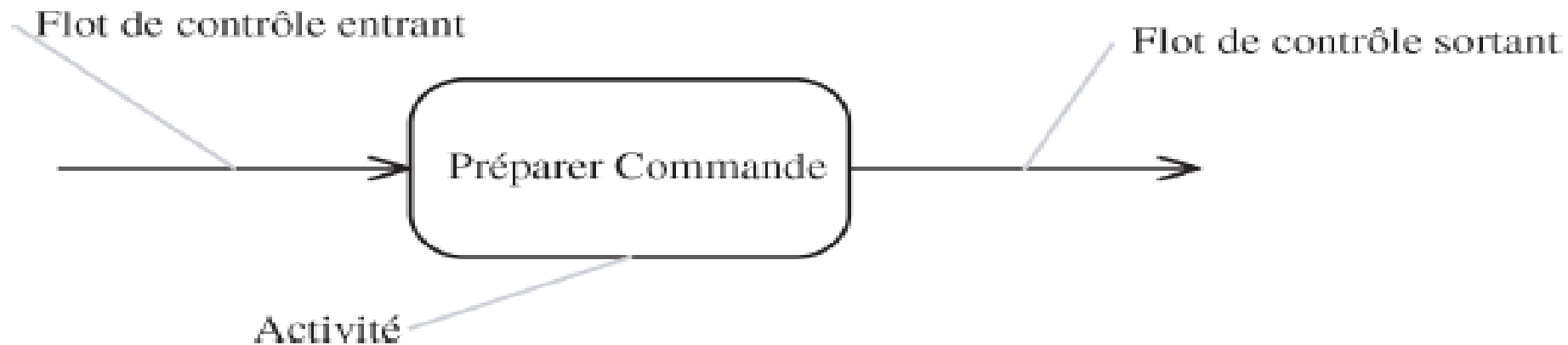


Diagramme d'activités (Transition)

- Les transitions spécifient l'enchaînement des traitements et définissent le flot de contrôle.
- Elles sont représentées par des èches pleines qui connectent les activités entre elles.
- Elles sont déclenchées dès que l'activité source est terminée.
- Elles provoquent automatiquement le début immédiat de la prochaine activité à déclencher (l'activité cible).
- Contrairement aux activités, les transitions sont franchies de manière atomique, en principe sans durée perceptible.

Diagramme d'activités (Structure de contrôle conditionnelle)

- Expression de conditions au moyen de transitions munies de gardes conditionnelles
- Ces **transitions ne peuvent être empruntées que si la garde est vraie.**
- On dispose d'une **clause [else] qui est validée si et seulement si toutes les autres gardes des transitions** ayant la même source sont fausses.
- Les **conditions sont notées entre crochets.**
- Pour mieux mettre en évidence un branchement conditionnel, on peut utiliser les **points de choix (losanges).**
- Les **points de choix** montrent un aiguillage du flot de contrôle.

Diagramme d'activités (Structure de contrôle conditionnelle -Exemple)

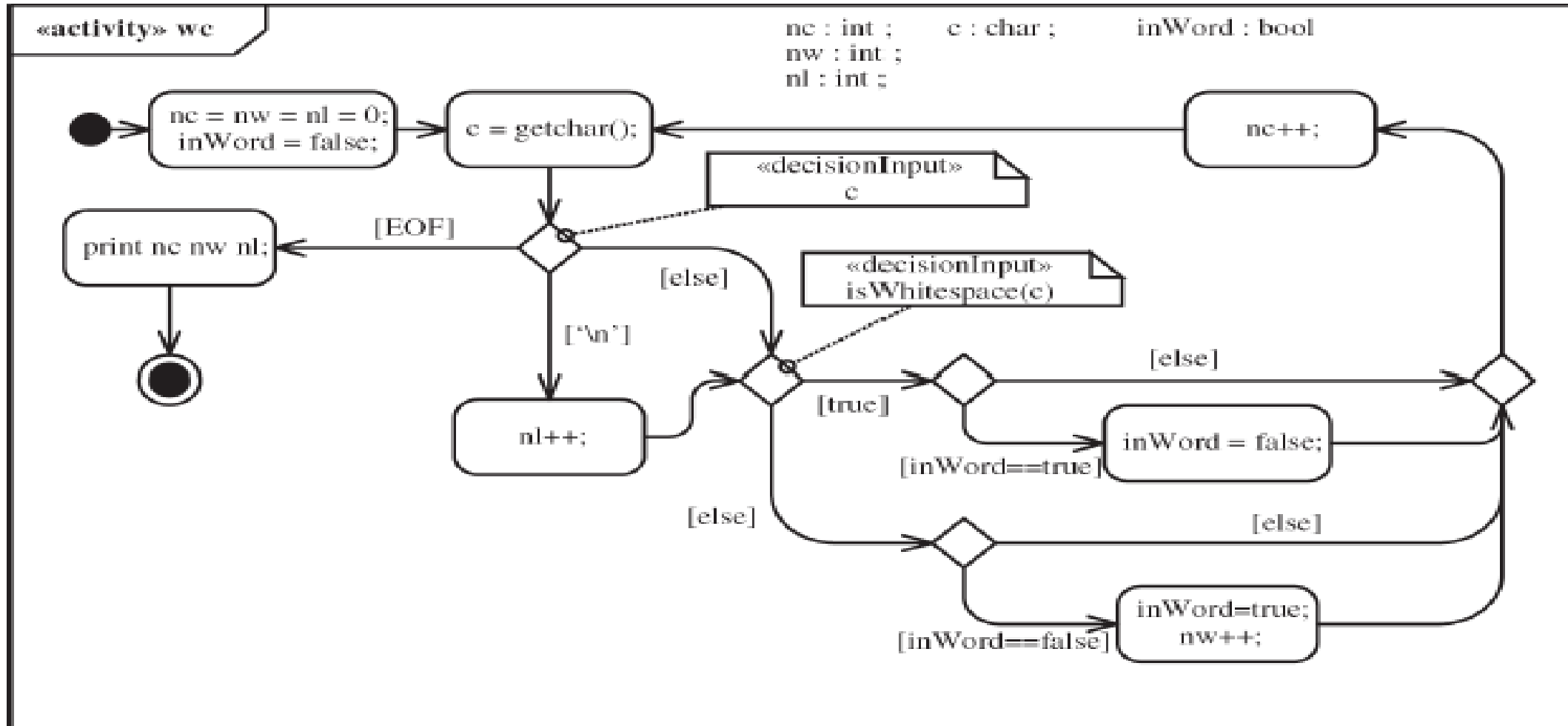


Diagramme d'activités (activités structurées)

- Les activités structurées utilisent les structures de contrôle usuelles (conditionnelles et boucle) à travers une syntaxe qui dépend de l'outil.
- La syntaxe précise de ces annotations pas définie dans la norme UML.
- Dans une activité structurée, on définit les arguments d'entrée et les sorties par des èches encadrées.

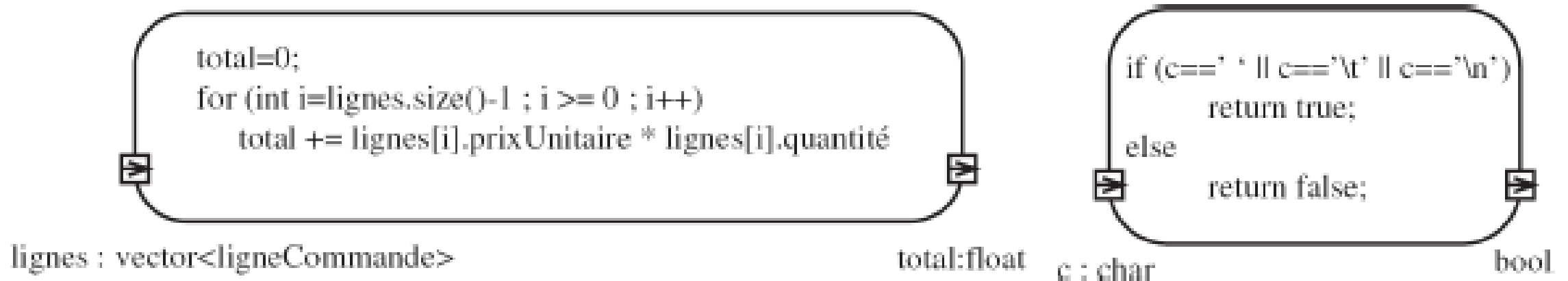


Diagramme d'activités (partitions)

- Pour modéliser un **traitement mettant en œuvre plusieurs classeurs**, on peut **spécifier le classeur responsable de chaque activité**.
- Les **partitions** permettent d'attribuer les activités à des éléments particuliers du modèle.
- Une **partition** peut elle-même **être décomposée en sous-partitions**.
- Pour spécifier qu'une activité est effectuée par un classeur particulier, on la positionne dans la partition correspondante.

«external»	«attribute» libelléService : Service	
Client	<u>Service comptable</u>	<u>Service livraison</u>

Diagramme d'activités (Partition – Exemple1)

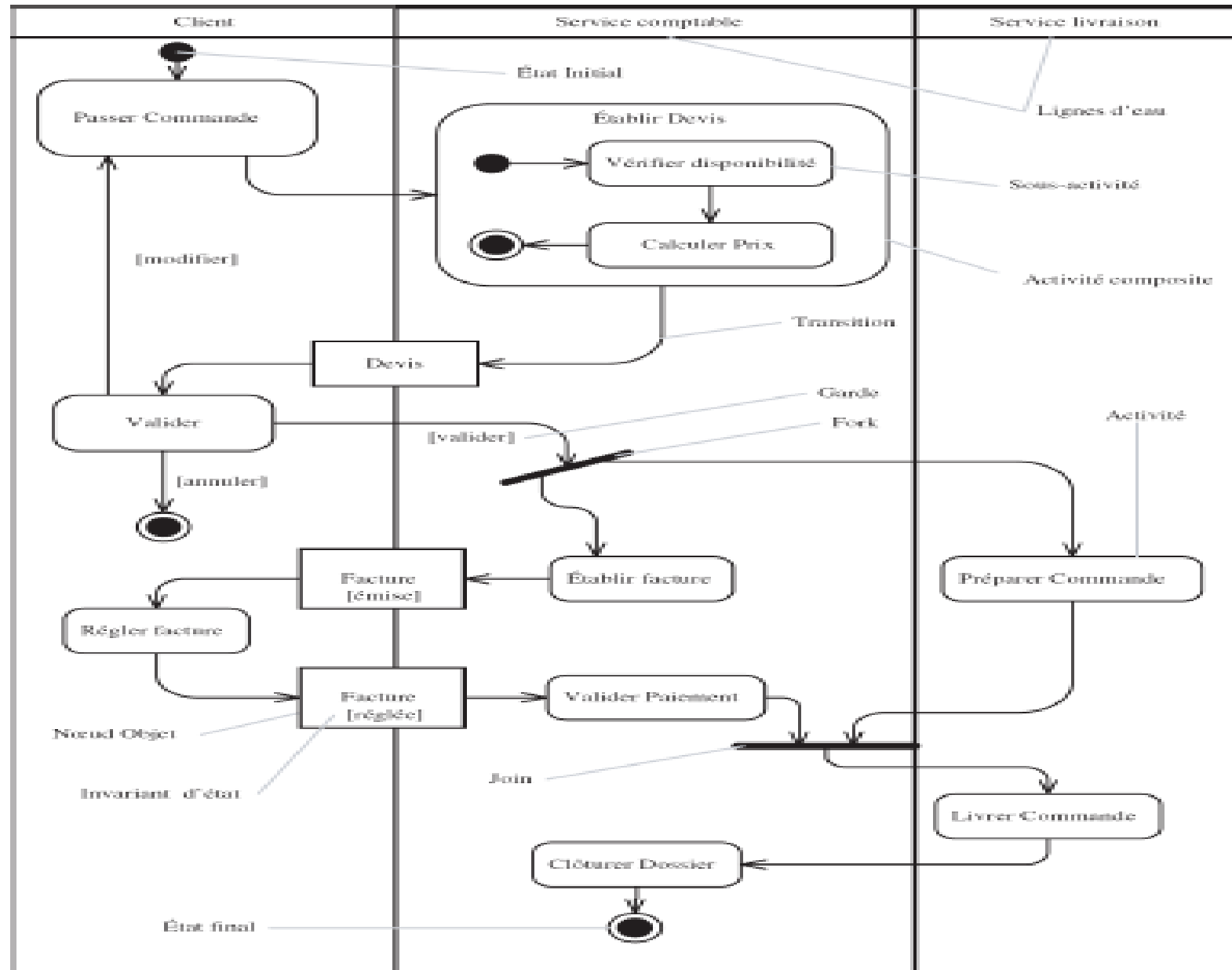


Diagramme d'activités (Partition –Exemple2)

- Partitions multidimensionnelle

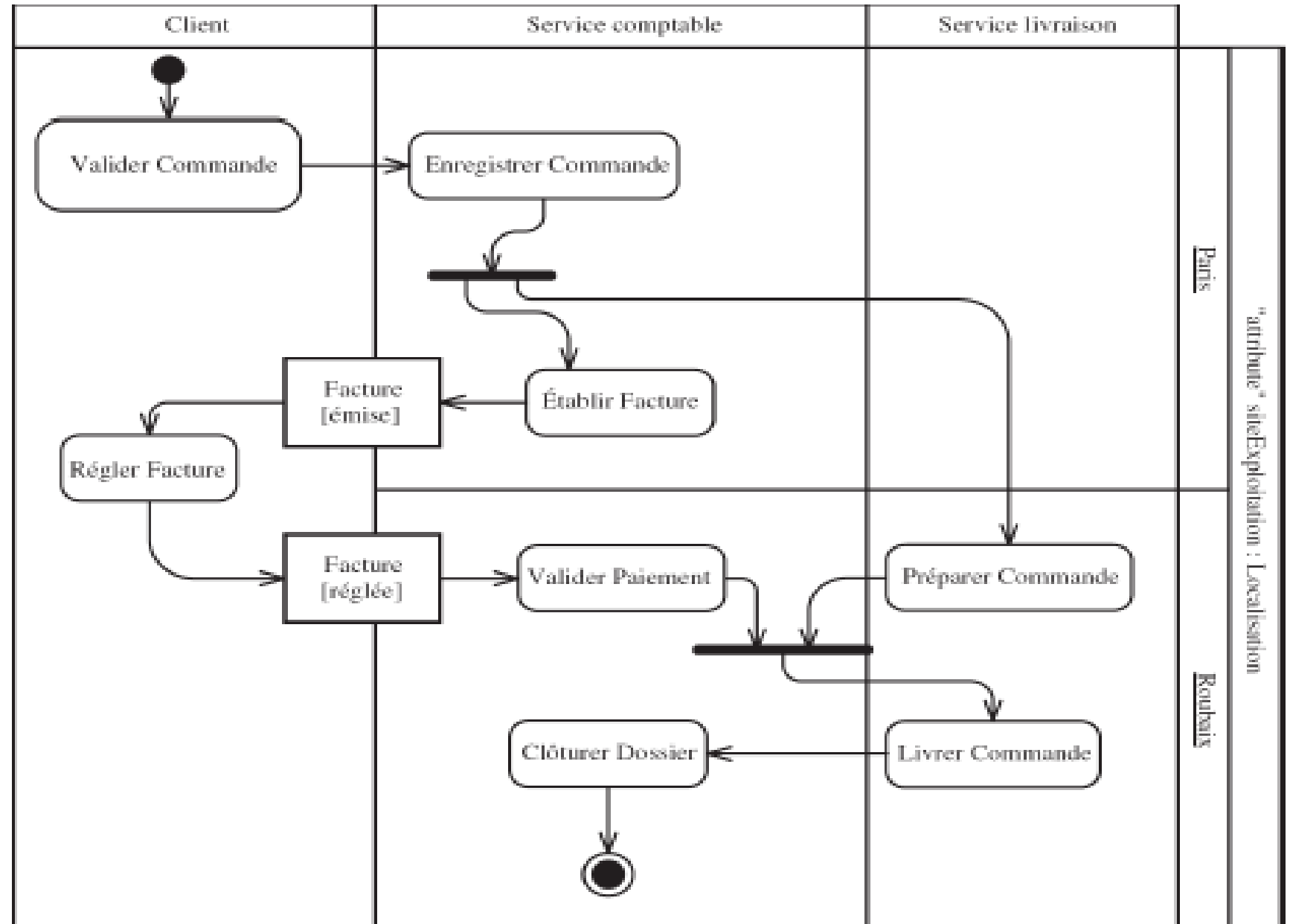


Diagramme d'activités Partition explicite)

Cette notation est moins encombrante graphiquement

Toutefois, elle met moins bien en valeur l'appartenance de groupes d'activités à un même conteneur.



Diagramme d'activités (les arguments et valeurs de retour de l'opération)

- Les diagrammes d'activités présentés jusqu'ici montrent bien le comportement du flot de contrôle.
- Pourtant, le flot de données n'apparaît pas.
- Si une activité est bien adaptée à la description d'une opération d'un classeur, il faut un moyen de spécifier les arguments et valeurs de retour de l'opération. C'est le rôle:
 - des pins,
 - des nœuds,
 - des Flots d'objets associés.

Diagramme d'activités (Pin)

- Un **pin** représente un point de connexion pour une action.
 - L'action ne peut débuter que si l'on affecte une valeur à chacun de ses pins d'entrée.
 - Les valeurs sont passées par copie.
- Quand l'activité se termine, une valeur doit être affectée à chacun des pins de sortie

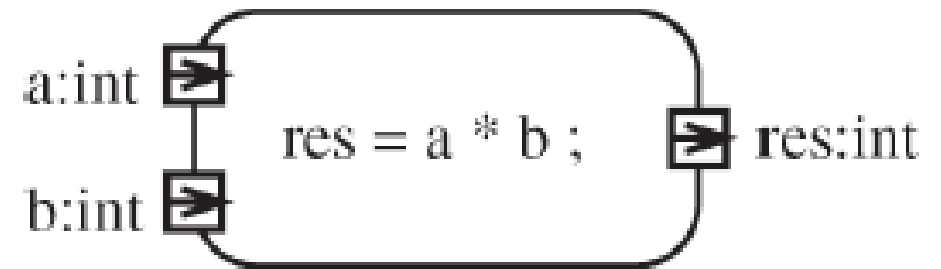
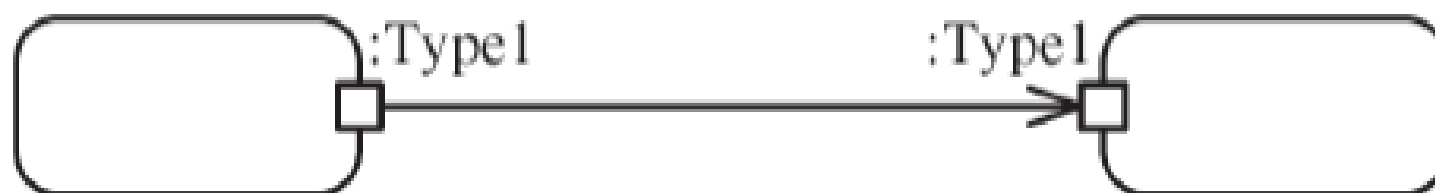


Diagramme d'activités (Flot de données)

- Un **flot d'objets** permet de passer des données d'une activité à une autre.
 - De fait, un arc qui a pour origine et destination un pin correspond à un flot de données.



- Un **noeud d'objets** permettent de mieux mettre en valeur les données.
 - C'est un conteneur typé qui permet le transit des données.



Diagramme d'activités (Activité Concurrente)

- Les diagrammes d'activités permettent en principe de représenter des **activités séquentielles**.
- Néanmoins, on **peut représenter des activités concurrentes** avec :
- Les barres de synchronisation,
- Les nœuds de contrôle de type flow final.

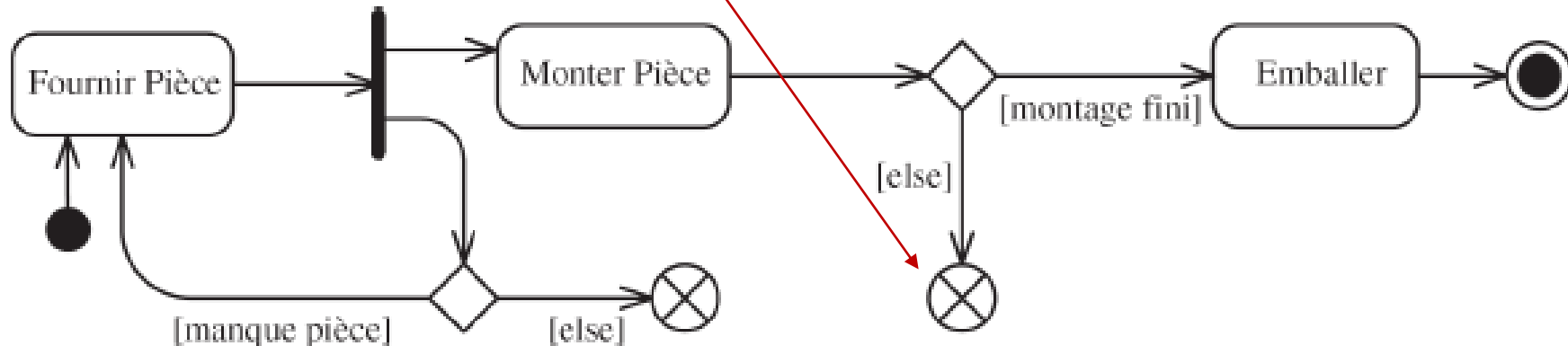


Diagramme d'activités (Barre de synchronisation)

- Plusieurs transitions peuvent avoir pour source ou pour cible une barre de synchronisation.
- Lorsque la barre de synchronisation a plusieurs transitions en sortie, on parle de **transition de type fork** qui correspond à une duplication du flot de contrôle en plusieurs flots indépendants.
- Lorsque la **barre de synchronisation** a plusieurs transitions en entrée, on parle de **transition de type join** qui correspond à un rendez-vous entre des flots de contrôle.
- Pour plus de commodité, il est possible de fusionner des barres de synchronisation de type join et fork.
- On a alors plusieurs transitions entrantes et sortantes sur une même barre.

Diagramme d'activités

- Un flot de contrôle qui atteint un noeud de contrôle de type « **flow final** » est détruit.
 - Les autres flots de contrôle ne sont pas affectés.



- Ce type de noeud est moins « fort » qu'un noeud de contrôle **final**.
 - Dans ce cas, tous les autres flots de contrôle de l'activité sont interrompus et détruits.



Diagramme d'activités (Actions de communication)

- Les actions de communication gèrent le passage et le retour de paramètres, les mécanismes d'appels d'opérations synchrones et asynchrones.
- Les actions de communication peuvent être employées comme des activités dans les diagrammes d'activité.

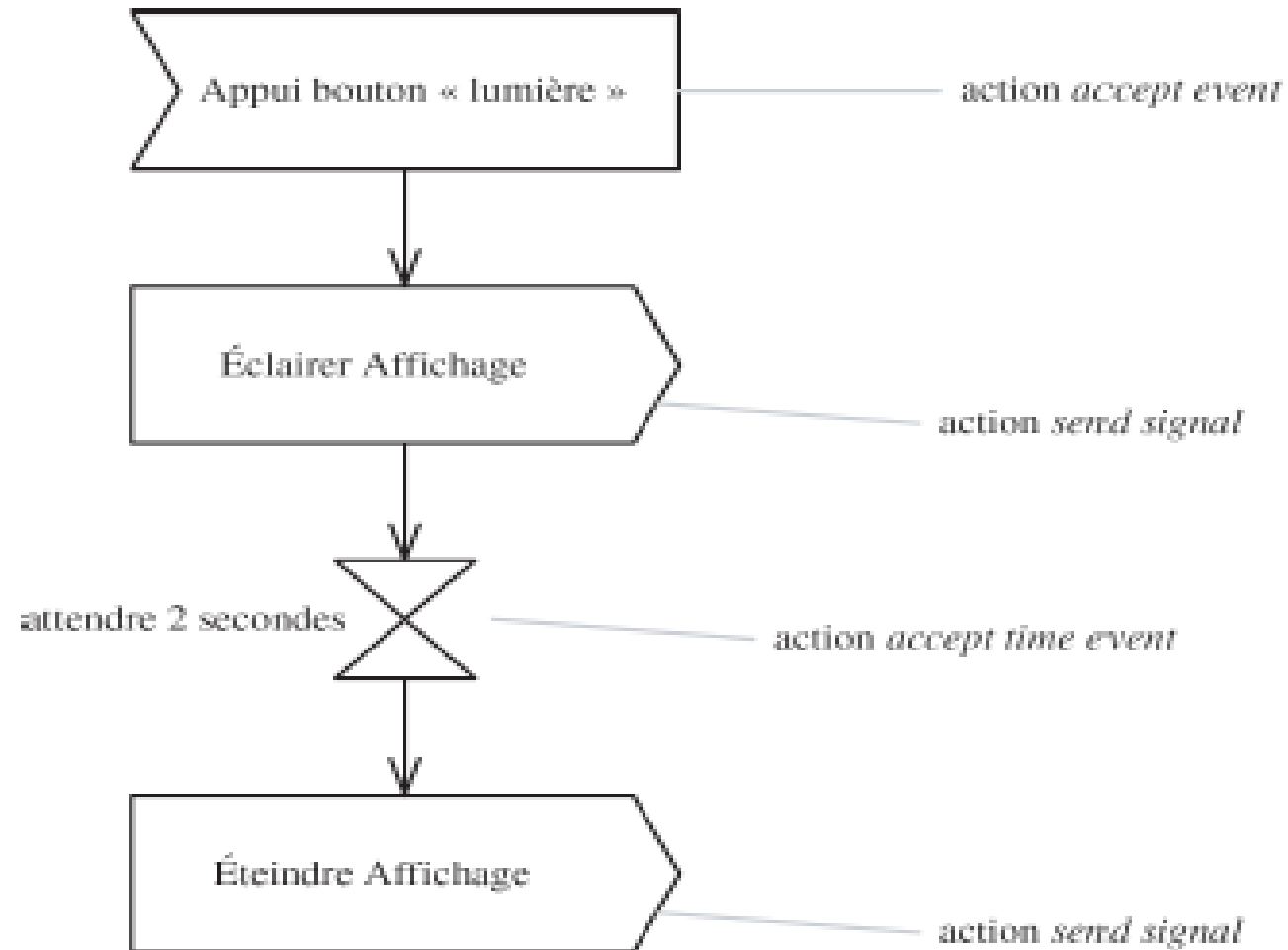


Diagramme d'activités (Exception)

Les **exceptions** permettent d'interrompre un traitement quand une situation qui dévie du traitement normal se produit. Elles assurent une gestion plus propre des erreurs qui peuvent se produire au cours d'un traitement.

On utilise des pins d'exception (avec un triangle) un pour gérer l'envoi d'exceptions et la capture d'exceptions.

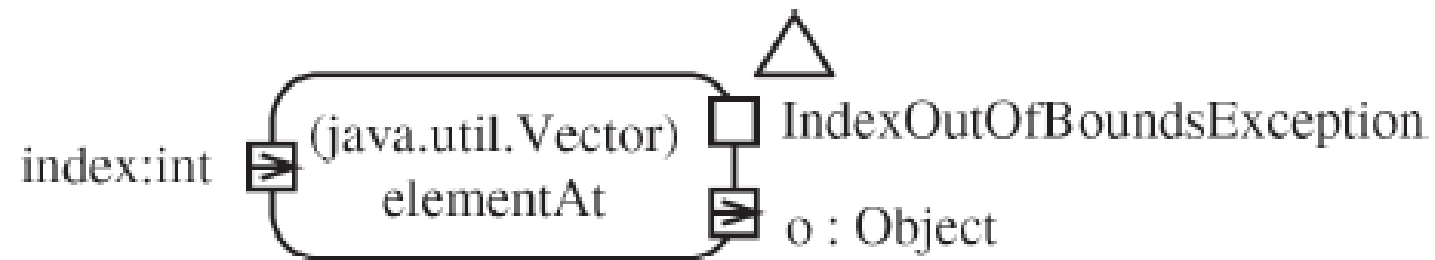


Diagramme d'activités (Traitement de l'exception)

Un flot de données correspondant à une exception est matérialisé par une flèche en « zigue zague ».

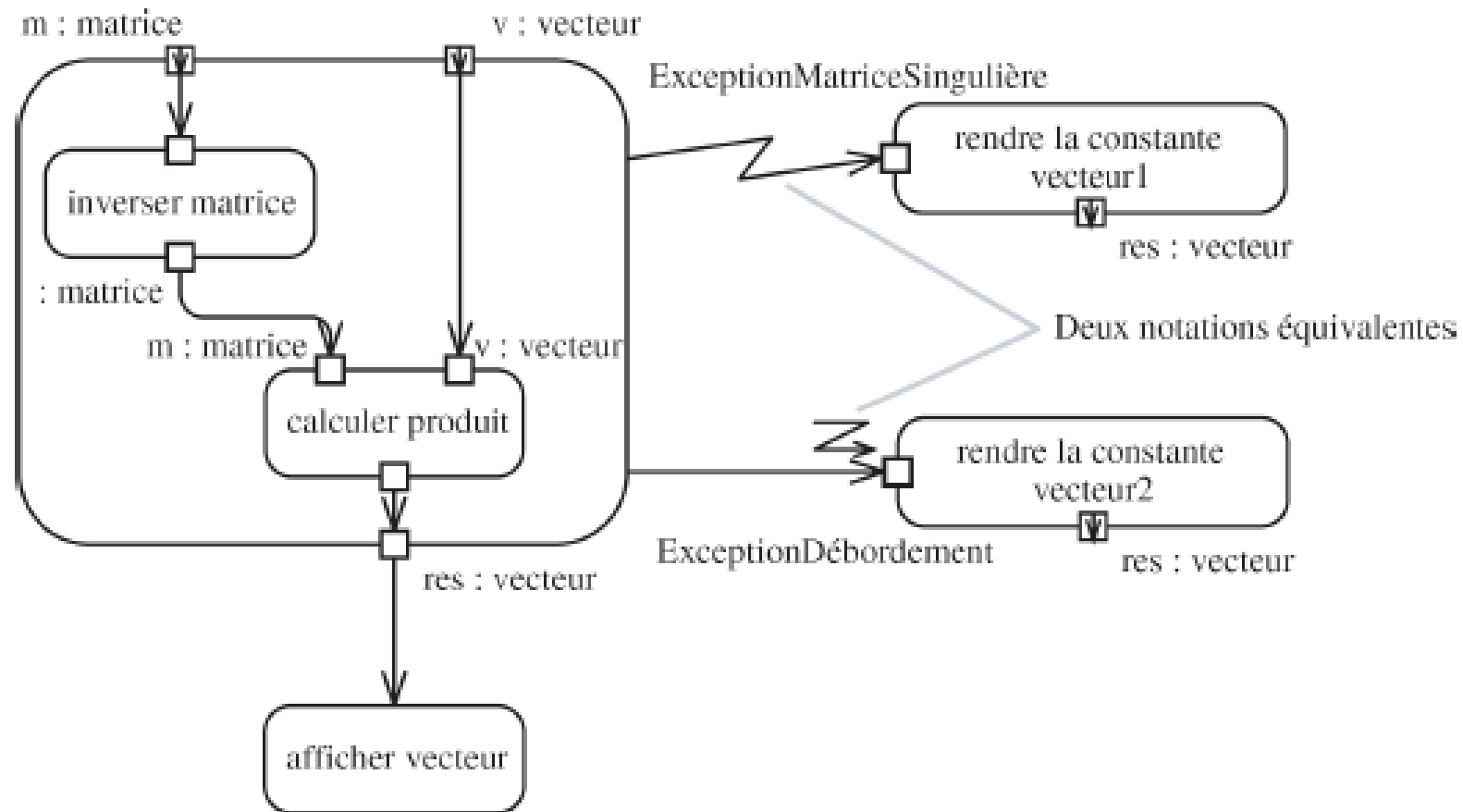


Diagramme d'activités (Région interruptible)

- Une région interruptible est représentée par un cadre arrondi en pointillés.
- Si l'événement d'interruption se produit, toutes les activités en cours dans la région interruptible sont stoppées et le flot de contrôle suit la flèche en zigzag qui quitte la région.

