

# SÉANCE 3

## DIAGRAMMES D'ACTIVITÉS UML

# SUJETS

## Révision de modélisation de logiciel

### Diagrammes d'activités UML

- Activités
- Actions
- Flux de contrôle
- Flux d'objets
- Décision et nœuds de fusion
- Fork et join
- Fil d'exécution conditionnel
- Partition
- Signal
- Région Interruptible
- Région d'Expansion

### Modélisation de domaine et processus de logiciel en utilisant les diagrammes d'activités

# MODÉLISATION DE LOGICIEL

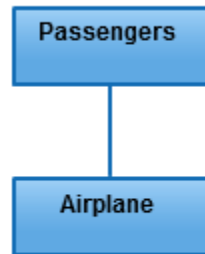
**UML définit treize types de diagrammes de base, divisées en deux catégories générales:**

- Modélisation Structurelle
- Modélisation Comportementale

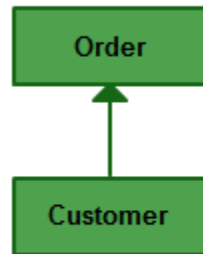
**Modèles structurels définissent l'architecture statique d'un logiciel**

- Ils sont utilisés pour modéliser les «choses» qui composent un modèle - les classes, les objets, les interfaces et les composants physiques
- En outre, ils sont utilisés pour modéliser les relations et dépendances entre les éléments

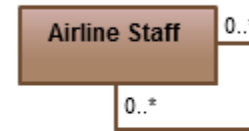
# RÉVISION DES ASSOCIATIONS UML



Association



Directed Association



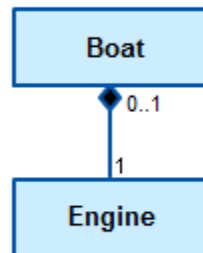
Reflexive Association



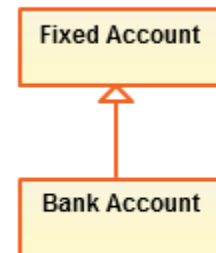
Multiplicity



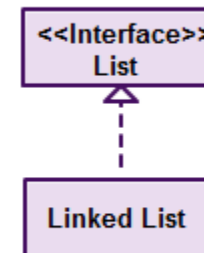
Aggregation



Composition



Inheritance



Realization

# ÉVÉNEMENTS DANS UML

**Dans UML, on peut modéliser quatre types d'événements:**

- Signaux: objet envoyé **d'une façon asynchrone** par un objet et reçu par un autre
- Appels: appels de méthode entre objets (souvent d'une façon synchronisée)
- Passage du temps
- Changement d'état

**Les événements peuvent être externes ou internes**

- Les événements externes sont ceux qui passent entre le système et ses acteurs (ex. peser un bouton de GUI)
- Les événements internes sont ceux qui sont passés à travers les objets qui résident dans le système (ex. exception IO)

# DIAGRAMMES D'ACTIVITÉS

## UML

**En UML un diagramme d'activité est utilisé pour afficher la séquence des actions produits par un système**

**Ils montrent le flux de travail dès le début jusqu'à la fin**

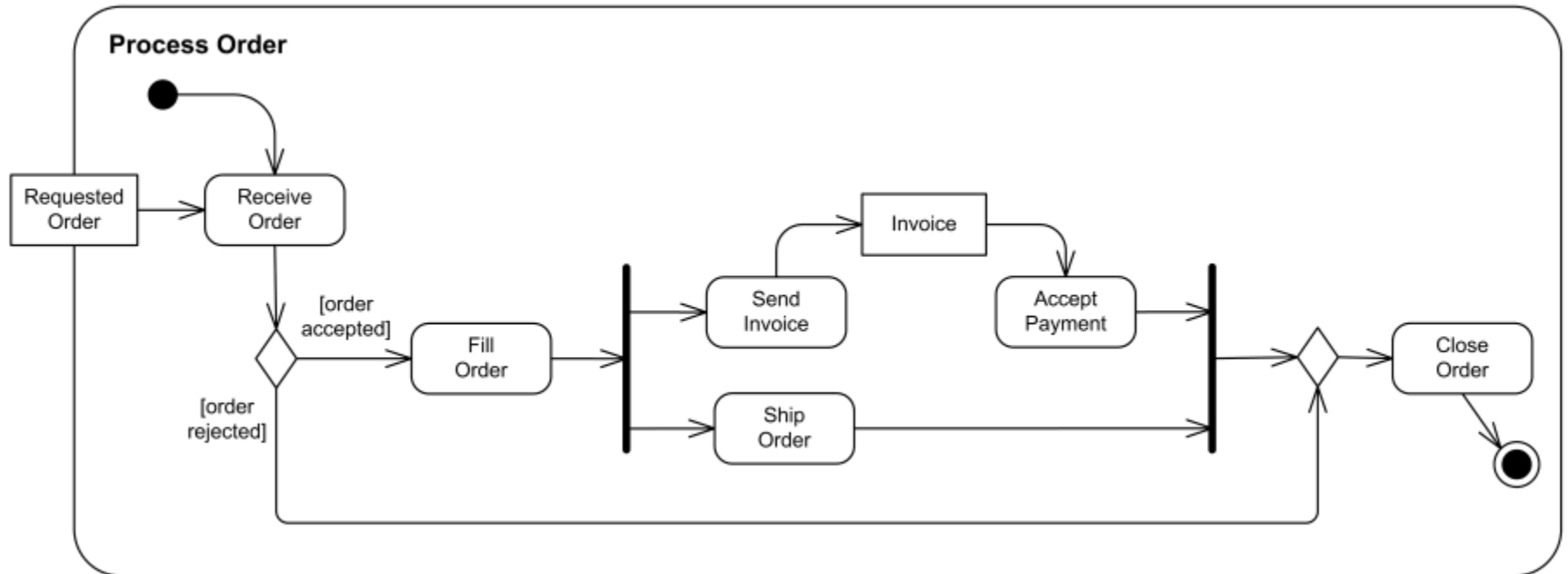
- **Détail les nombreux trajets de décision qui existent dans la progression des événements contenus dans l'activité**

**Très utile pour modéliser le comportement de concurrence**

# DIAGRAMMES D'ACTIVITÉS UML

Un exemple d'un diagramme d'activité est illustrée ci-dessous

*(Nous reviendrons sur ce diagramme)*

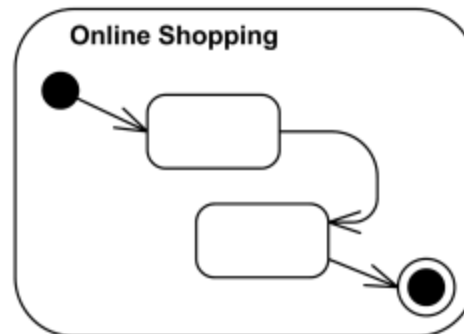


# ACTIVITÉ

**Une activité est la spécification d'une séquence paramétrée de comportement**

- Ça prend du temps
- Semblable à un état, où le critère pour quitter l'état est l'achèvement de l'activité

**Représenté par un rectangle aux coins arrondis renfermant toutes les actions et les flux de contrôle**

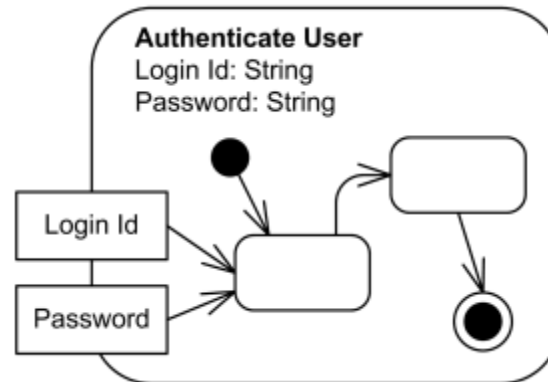




# PARAMÈTRES D'ACTIVITÉS

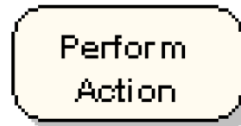
Les paramètres d'activité sont affichés sur le bord et listés en-dessous du nom d'activités tel que:

*nom du paramètre: type du paramètre*

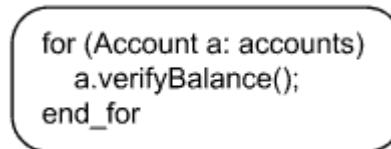


# ACTIONS

**Une action représente une étape unique dans une activité**



**L'action pourrait être exprimée dans un langage d'action dépendant de l'application**



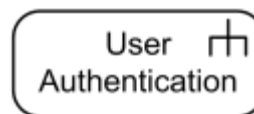
**Il existe quatre façons desquelles une action peut être déclenchée**

1. Aussitôt qu'une activité commence
2. Durant la durée de vie de l'activité
3. En réponse à un événement
4. Juste avant la complétion de l'activité

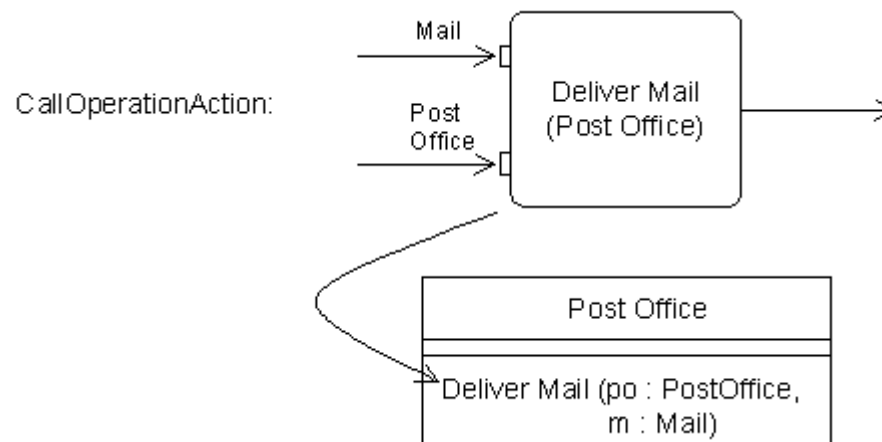
# ACTIONS D'APPEL

**Action d'appel d'activité** : permet d'appeler une autre activité

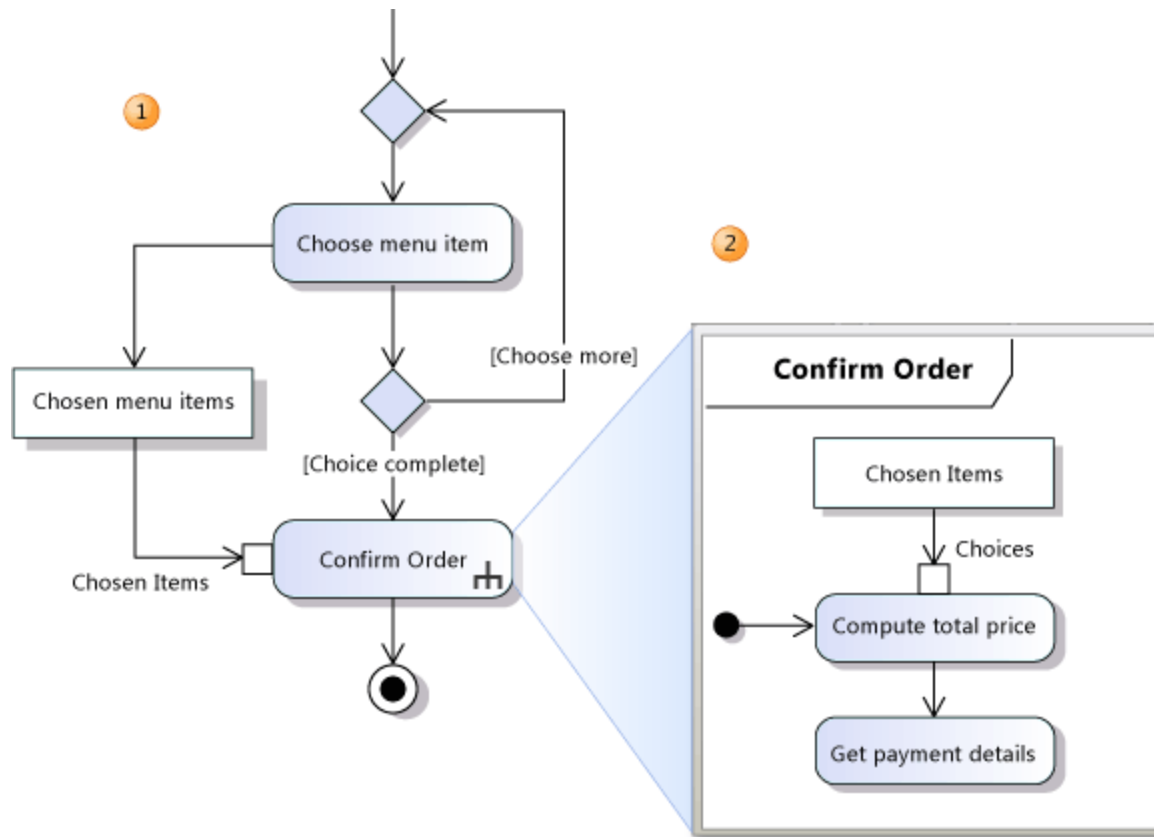
- Ceci évite les définitions redondantes d'activités



**Action d'appel d'opération** : appelle le comportement d'un élément de structure (opération d'une classe)

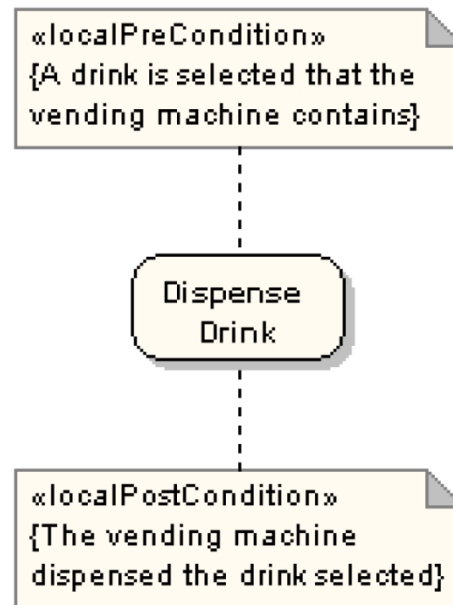


# ACTIONS D'APPEL D'ACTIVITÉ



# ACTIONS ET CONTRAINTES

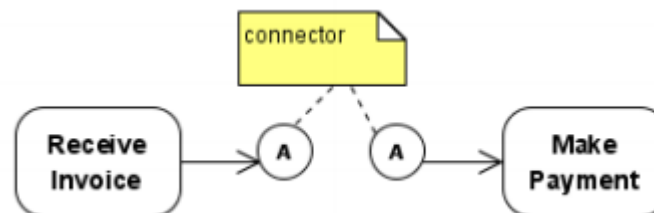
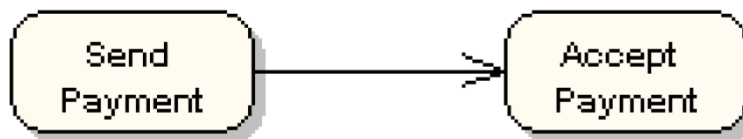
Les contraintes peuvent être attachés aux actions



# FLUX DE CONTRÔLE

Indique le flux de contrôle d'une action à la prochaine

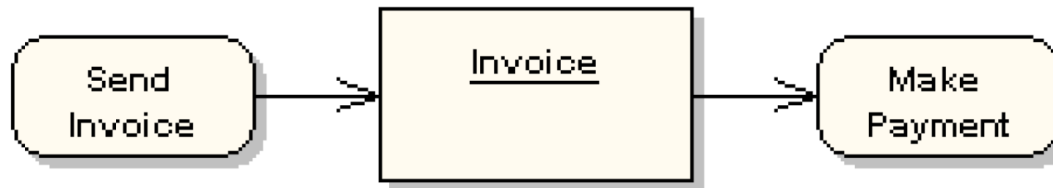
- Sa notation est une flèche



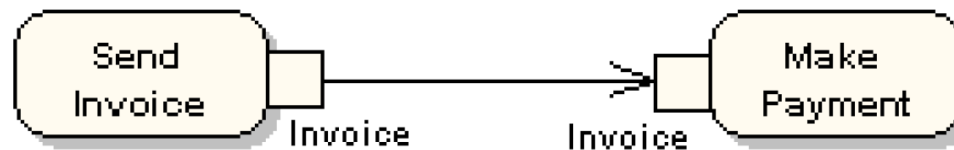
# FLUX D'OBJETS

Un flux d'objet est un trajet le long duquel des objets peuvent passer

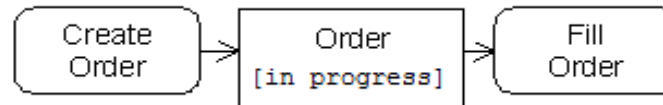
- Un objet est représentée par un rectangle



Un autre façon pour montrer la même chose:



On peut aussi montrer l'état de l'objet qui passe (montré en-dessous du nom de l'objet, entre parenthèses)



# NŒUDS INITIAUX ET FINAUX

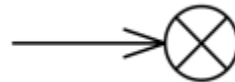
**Un nœud initial** est un nœud de contrôle auquel le flux commence lorsque l'activité est invoquée



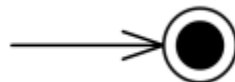
**Les activités peuvent avoir plus qu'un seul nœud initial**

- Dans ce cas, lorsqu'on invoque une activité, plusieurs flux commencent, un flux à chaque nœud initial

**Un nœud final de flux** est un nœud final de contrôle qui termine un flux



**Un nœud final d'activité** est un nœud final de contrôle qui arrête tous les flux de l'activité

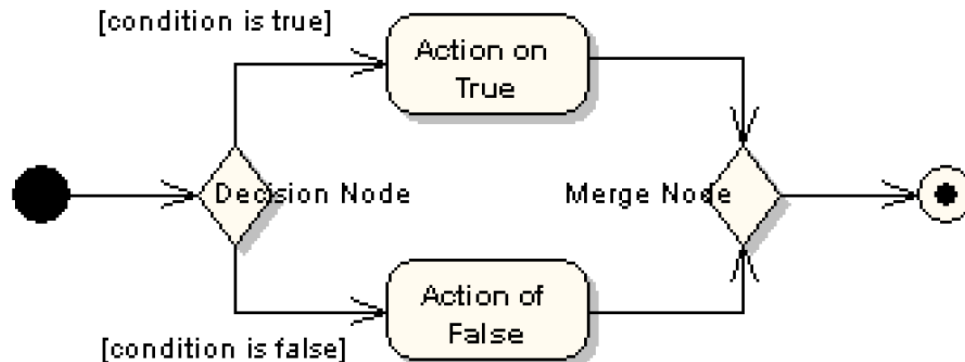




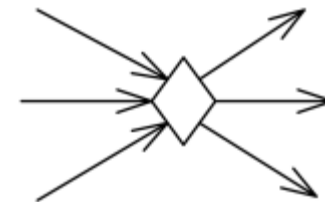
# DÉCISION ET NŒUDS DE FUSION

Les nœuds de décision et fusion ont la même notation: une forme de diamant

Le flux de contrôle qui sort d'un nœud de décision sont associés avec des conditions



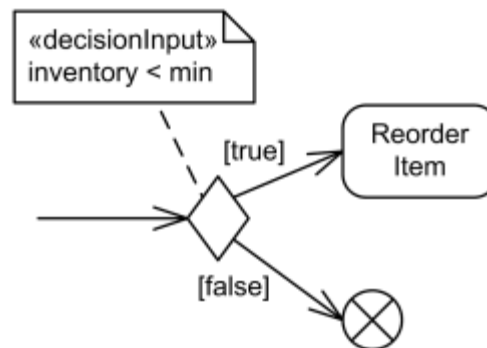
Nœuds de décision et fusion combinés



# NŒUDS DE DÉCISION

**Une décision peut avoir un comportement de l'entrée de décision spécifié**

- Les comportements de l'entrée de décision fut introduits dans UML afin d'éviter les re-calculs redondants dans les gardes
- Ils sont spécifiés par le mot-clé «decisionInput» et une condition est placée dans un symbole de note

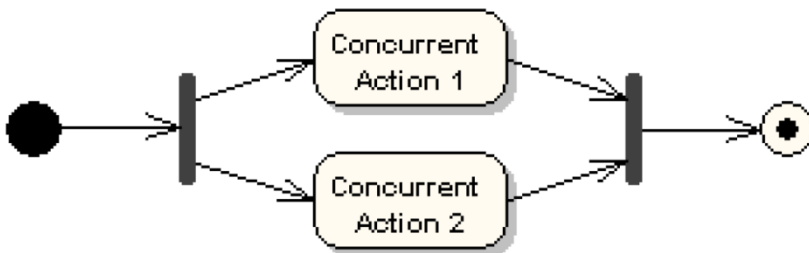


# NŒUDS DE FORK ET JONCTION (JOIN)

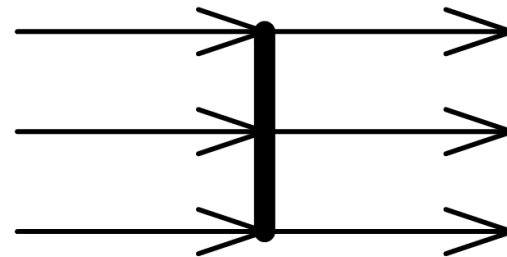
Les nœuds de fork et join ont la même notation: soit une barre horizontale ou verticale

Ils indiquent le début et la fin des threads (fils) de contrôle simultanés

- « Join » synchronise deux entrées et produit une seule sortie
- Le flux de sortie de « join » ne peut pas exécuter jusqu'à ce que toutes les flux d'entrées ont été reçues

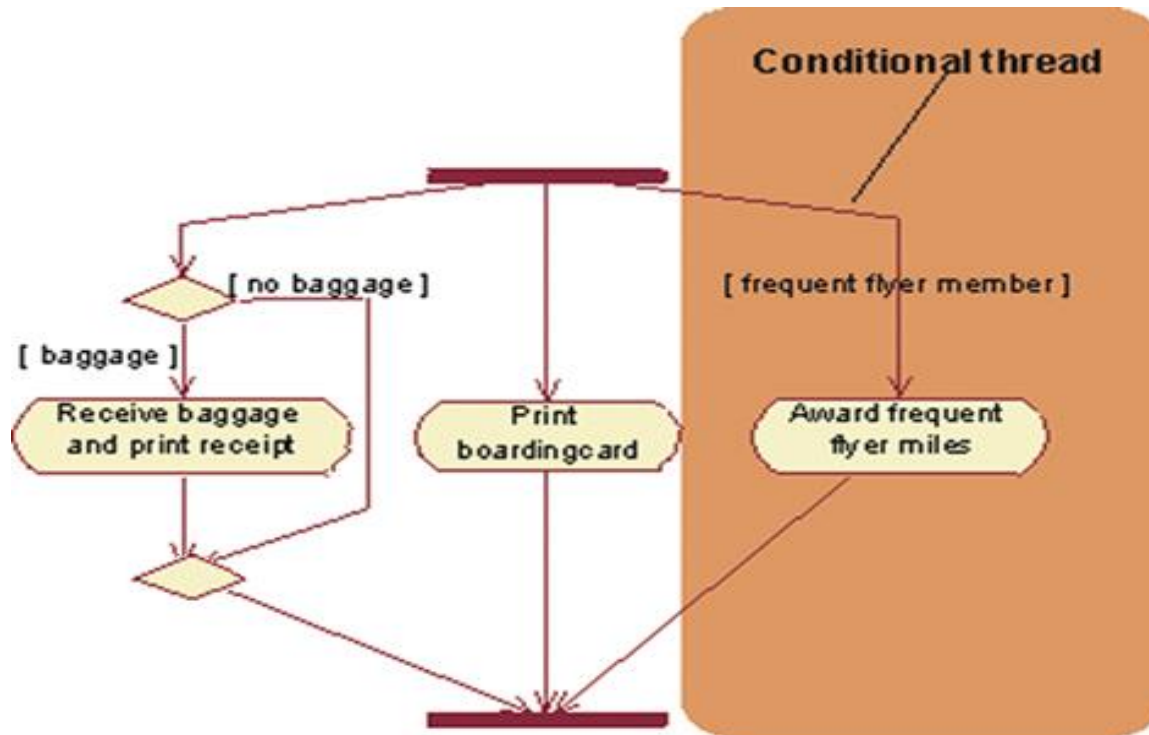


Nœuds fork et join combinés

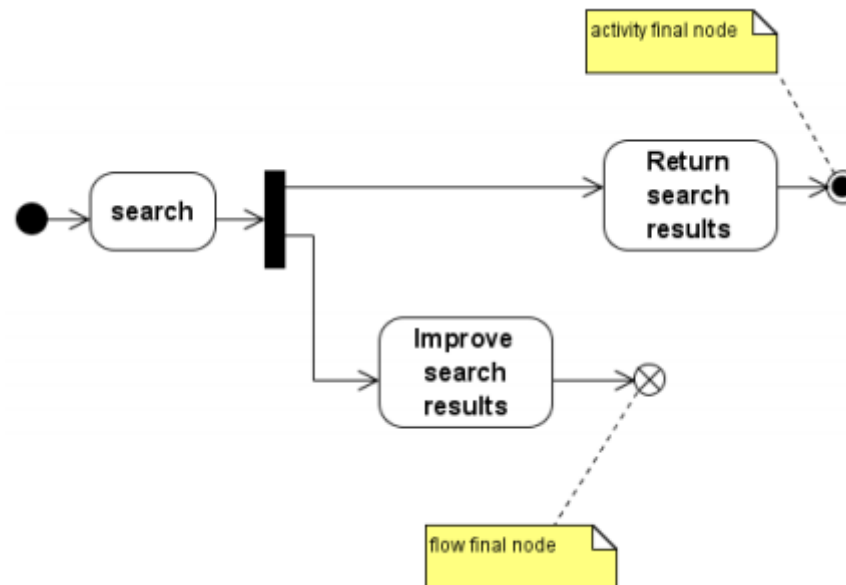


# FIL D'EXÉCUTION CONDITIONNEL

Les conditions de garde peuvent être utilisées pour montrer qu'un d'un ensemble de threads simultanés est conditionnel



# NŒUDS INITIAUX ET FINAUX

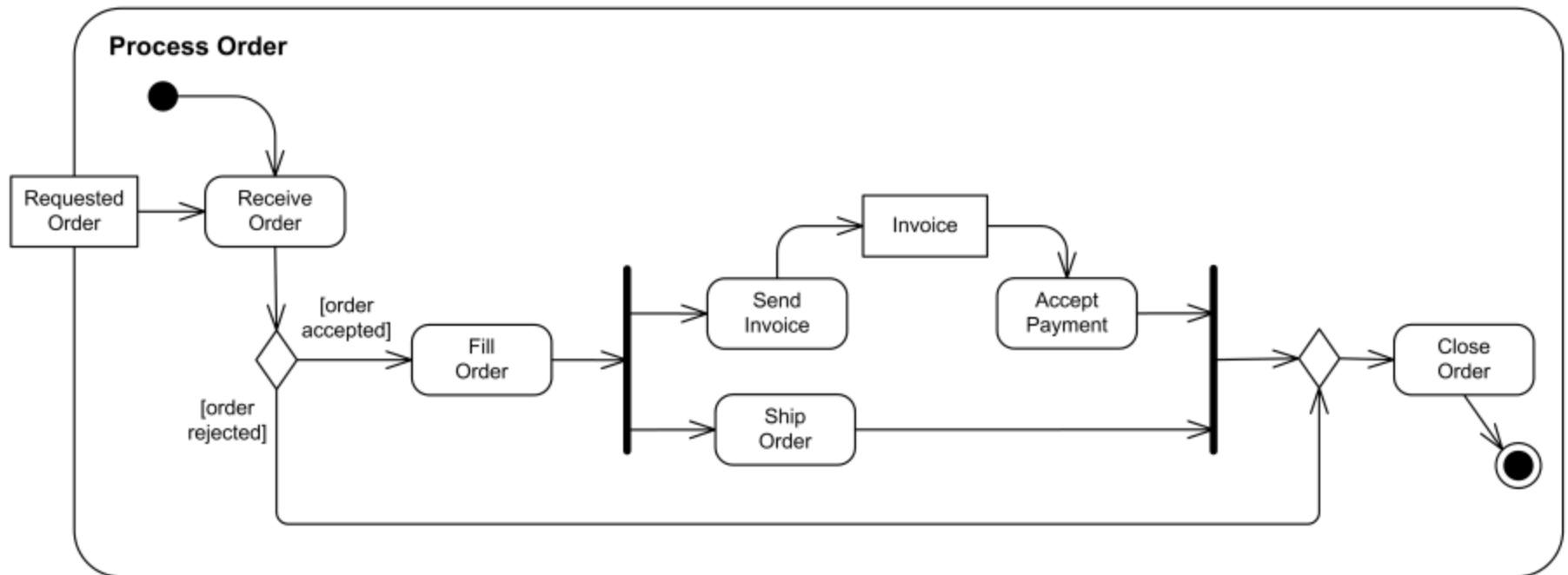


**Attention:** soyez prudents lorsque vous utilisez un nœud final de flux après un fork

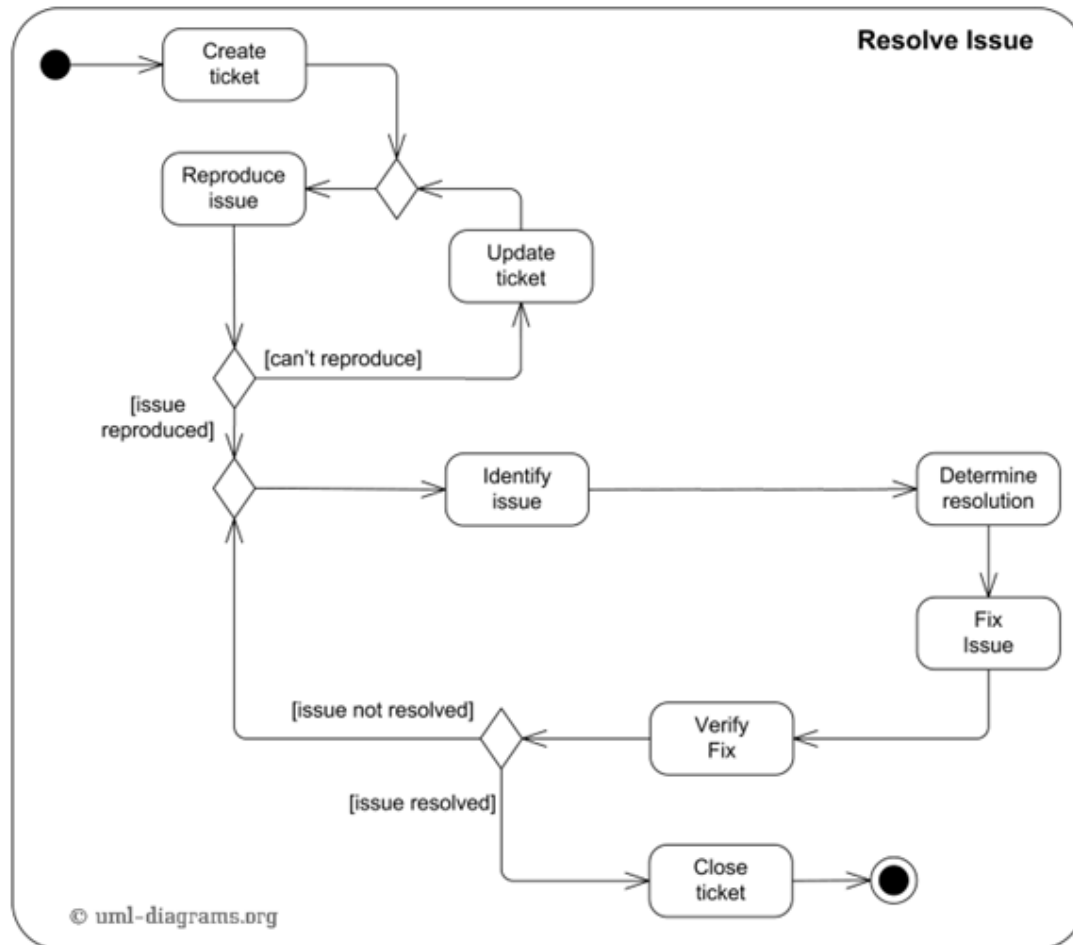
- Dès que le nœud final de l'activité est atteint, toutes les autres actions dans l'activité (incluant ceux avant le nœud final de flux) se terminent

# DIAGRAMMES D'ACTIVITÉS UML

Revenant à notre exemple initial



# LA GESTION DES PROBLÈMES DANS LES PROJETS DE LOGICIEL

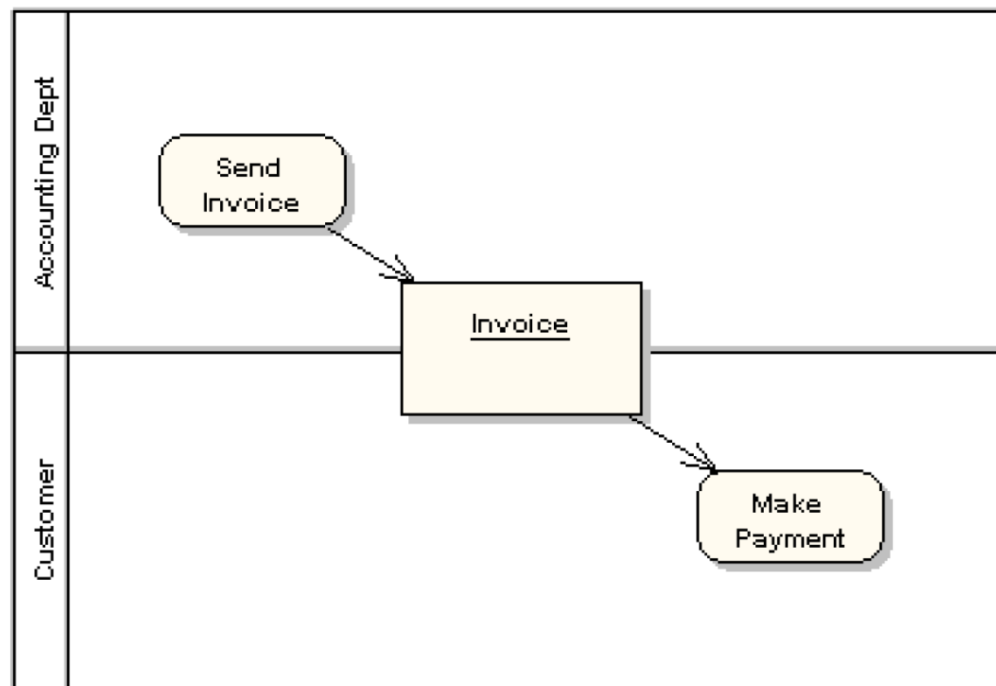


Courtesy of [uml-diagrams.org](http://uml-diagrams.org)

# PARTITION

## Représenté comme couloir de natation horizontal ou vertical

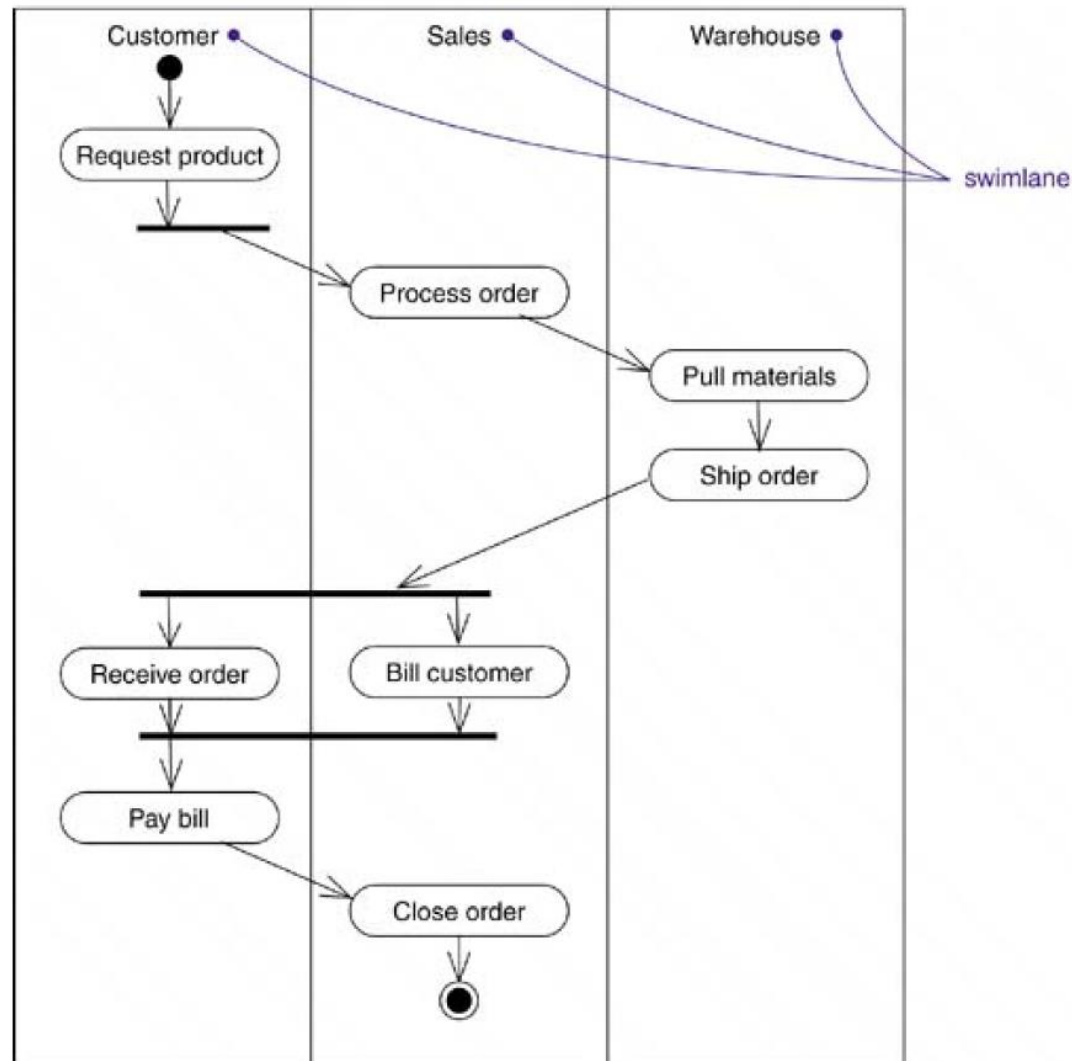
- Représente un groupe d'actions qui ont des caractéristiques communes







# PARTITION



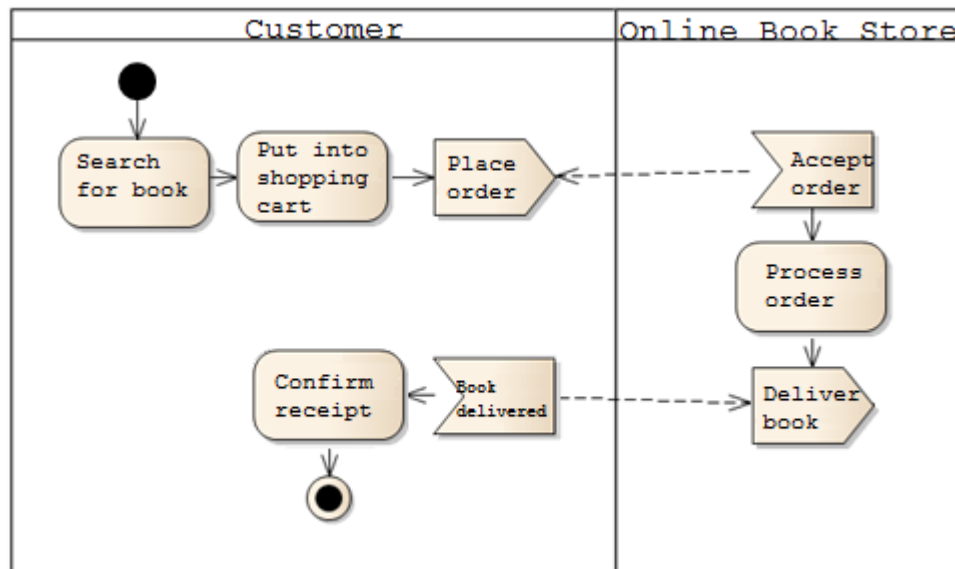
# ENVOYER ET RECEVOIR DES SIGNAUX ET DES ÉVÉNEMENTS DE TEMPS

**Les flux de contrôle ou les flux d'objets connectent des actions**

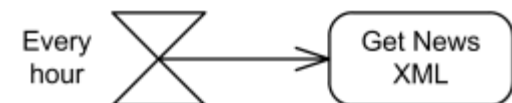
- Ils définissent des processus "synchrones" où le flux est déterminé par une séquence ordonnée d'étapes

**Grâce à l'utilisation de signaux, les processus peuvent être désaccouplés**

- Nous pouvons atteindre une communication asynchrone



Action d'événement de temps produit une sortie chaque heure

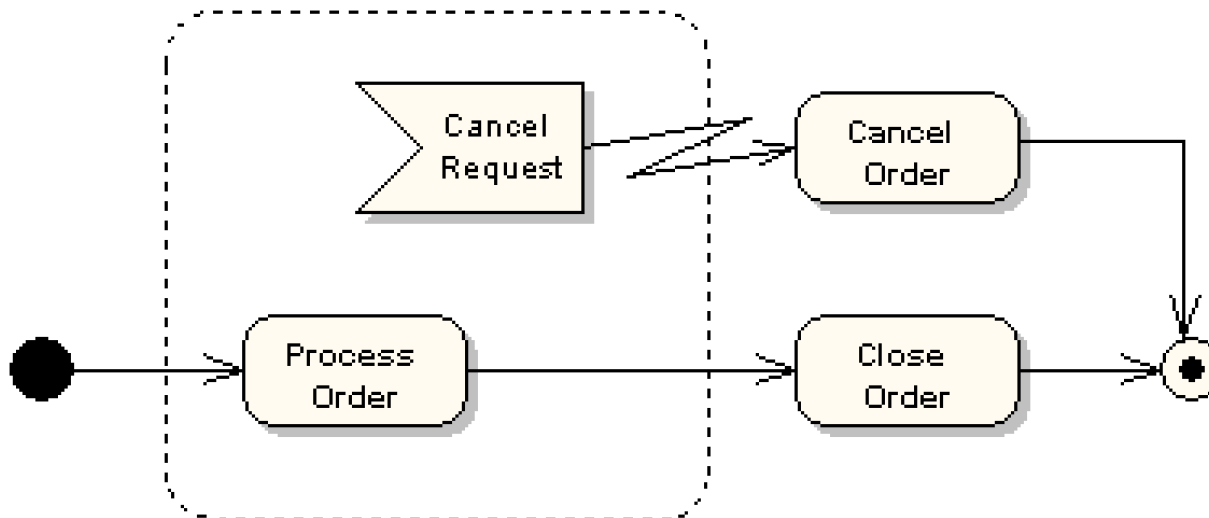


# RÉGIONS D'ACTIVITÉ INTERRUPTIBLE

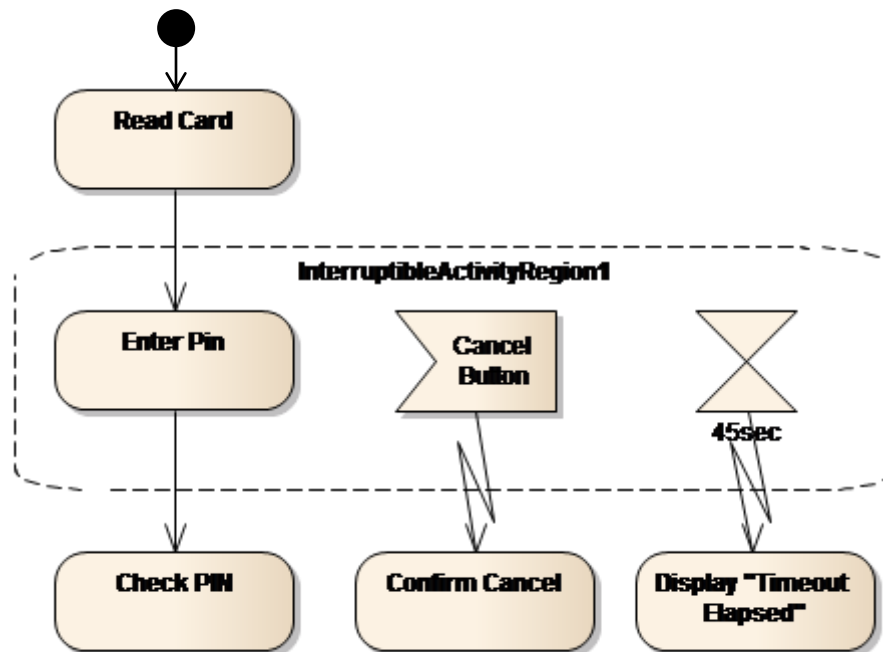
Entoure un ensemble d'actions qui peut être interrompu

Exemple ci-dessous:

- L'action « Process Order » va exécuter jusqu'à la fin où le contrôle va être passé vers l'action « Close Order », à moins que l'Interruption « Cancel Request » est reçu, ce qui va causer le contrôle d'être passé vers l'action « Cancel Order ».



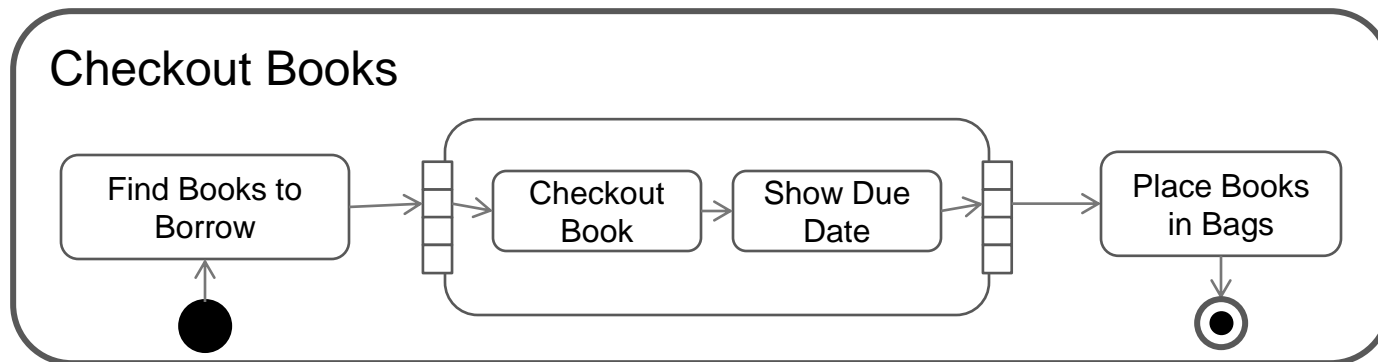
# RÉGIONS D'ACTIVITÉ INTERRUPTIBLE



# RÉGIONS D'EXTENSION

Une région d'extension est une zone d'activité qui exécute à plusieurs reprises pour consommer tous les éléments d'une collection d'entrée

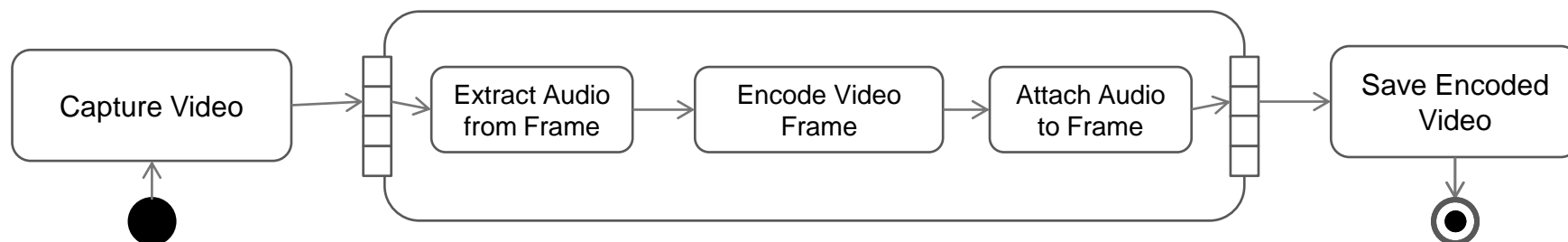
Exemple d'un « checkout » des livres dans une bibliothèque modélisée à l'aide d'une zone d'expansion



# RÉGIONS D'EXTENSION

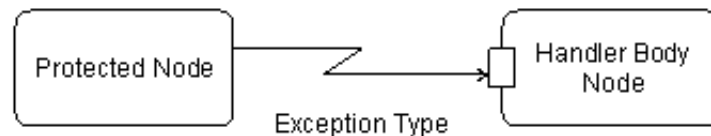
## Un autre exemple: Encodage de vidéo

### Encode Video



# GESTIONNAIRES D'EXCEPTIONS

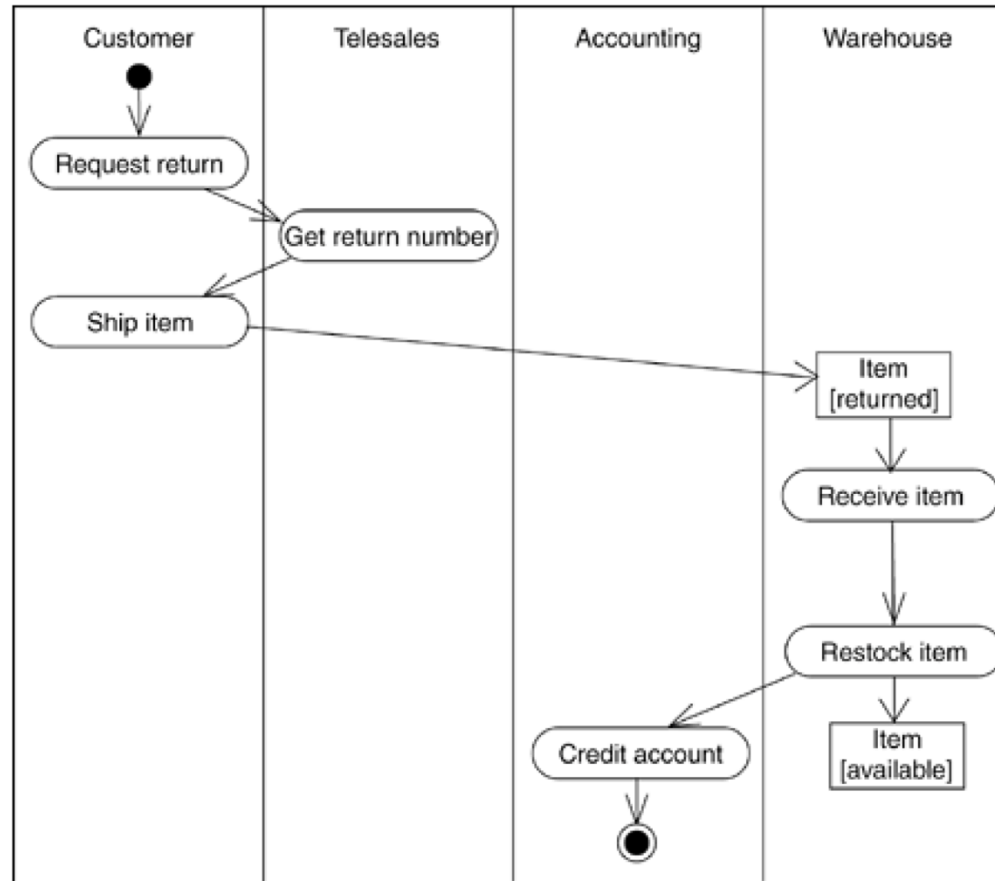
**Un gestionnaire d'exception est un élément qui spécifie quoi exécuter dans le cas où l'exception spécifiée se produit pendant l'exécution du nœud protégé**



## En Java

- “Try block” correspond à un “Protected Node”
- “Catch block” correspond à un “Handler Body Node”

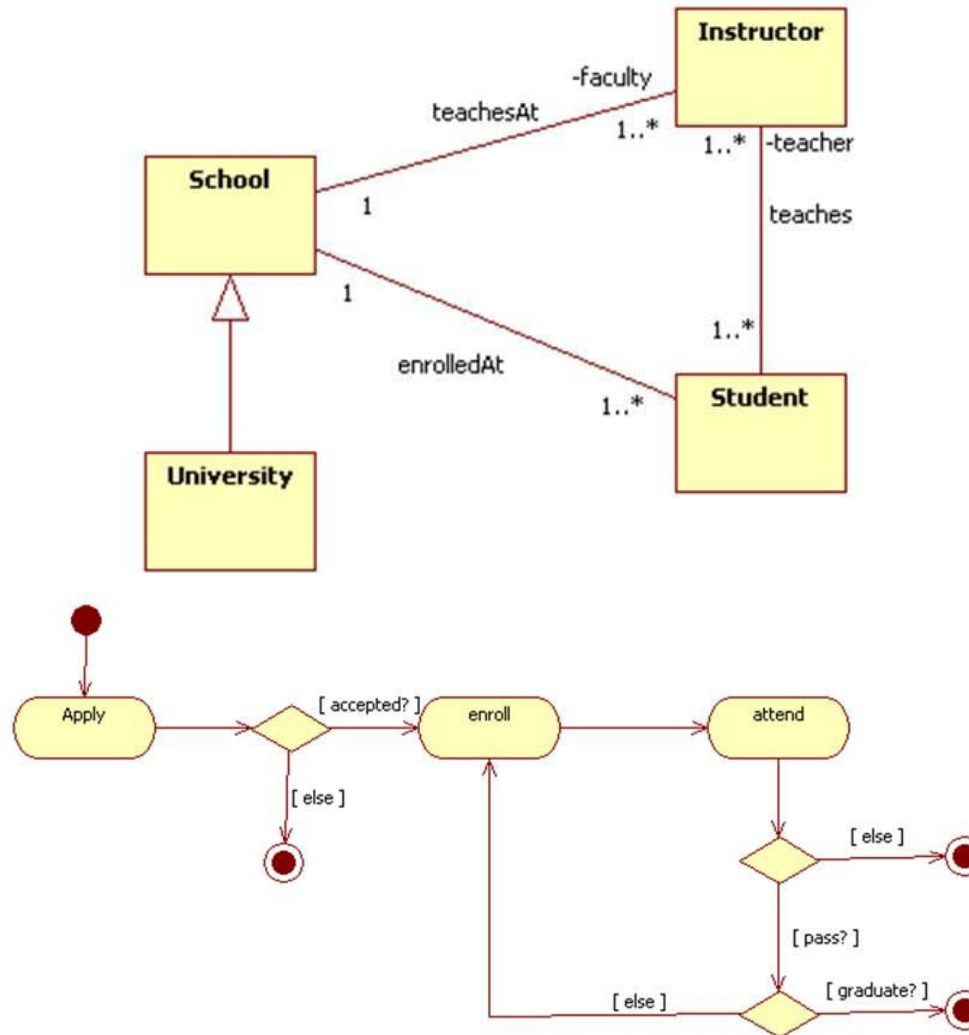
# DIAGRAMMES D'ACTIVITÉ POUR MODÉLISER UN FLUX DE TRAVAIL



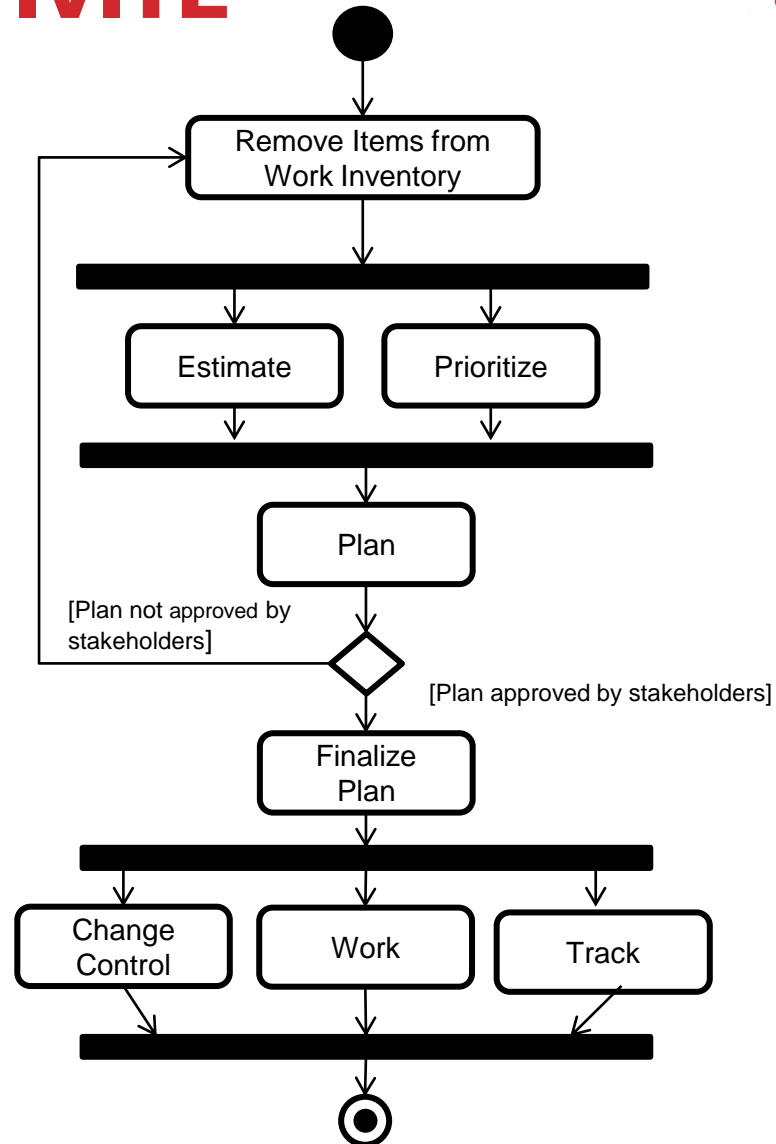
Retour d'un  
article acheté



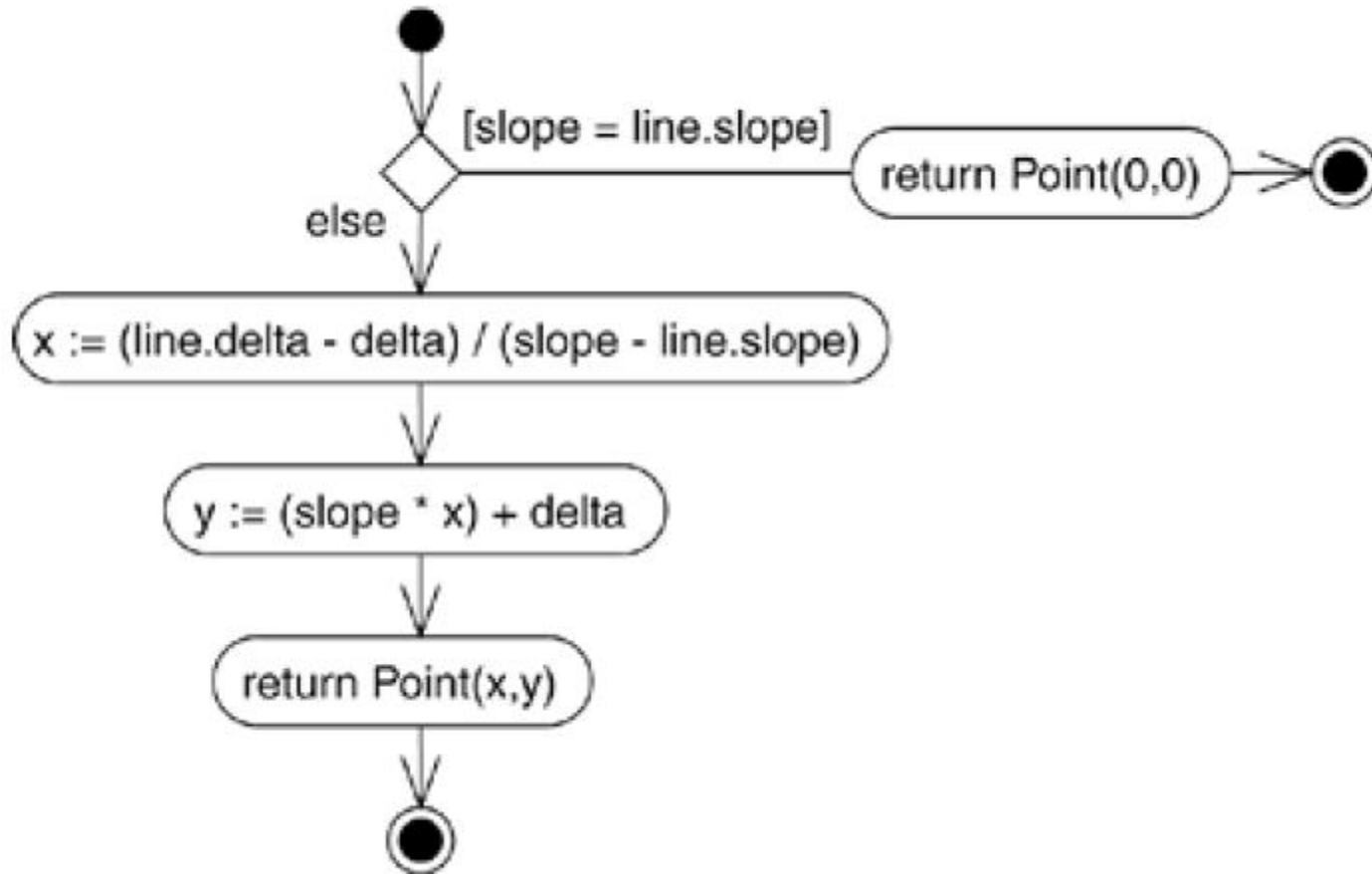
# DIAGRAMMES D'ACTIVITÉ UTILISÉS DANS LA MODÉLISATION DE DOMAINES



# MODÉLISATION D'UN PROCESSUS LOGICIEL À L'AIDE D'UN DIAGRAMME D'ACTIVITÉ



# DIAGRAMMES D'ACTIVITÉ POUR MODÉLISER UNE OPÉRATION



# COMMENT CONSTRUIRE DES DIAGRAMMES D'ACTIVITÉS

1. Identifier les scénarios clés des cas d'utilisation du système
2. Combinez les scénarios pour produire des processus complets décrits en utilisant des diagrammes d'activité
3. Lorsqu'un comportement significatif d'objet est déclenché par un processus, ajoutez des flux d'objets aux diagrammes
4. Lorsque les processus passent à travers les frontières de la technologie, utilisez des couloirs de natations
5. Affiner les activités compliquées de haut niveau

# QUAND UTILISER LES DIAGRAMMES D'ACTIVITÉ

## Utilisez-les pour

- Analyser les cas d'utilisation
- Comprendre le flux de travail dans de nombreux cas d'utilisation
- Modéliser les applications avec plusieurs flux de travail concurrents

## Ne les utilisez pas

- Pour voir comment les objets collaborent
- Pour voir comment un objet se comporte au cours de son cycle de vie

# MERCI

## QUESTIONS?