

**GÉNIE LOGICIEL ORIENTÉ OBJET (GLO-2004)
ANALYSE ET CONCEPTION DES SYSTÈMES ORIENTÉS OBJETS (IFT-2007)**

Automne 2016

**Module 14 - Diagrammes d'activités :
flots de contrôle et flots de données**

Martin.Savoie@ift.ulaval.ca

Bachelier Génie logiciel, Chargé de cours,
département d'informatique et de génie logiciel

Questions sur le projet

UML : 14 types de diagrammes (ouf!)

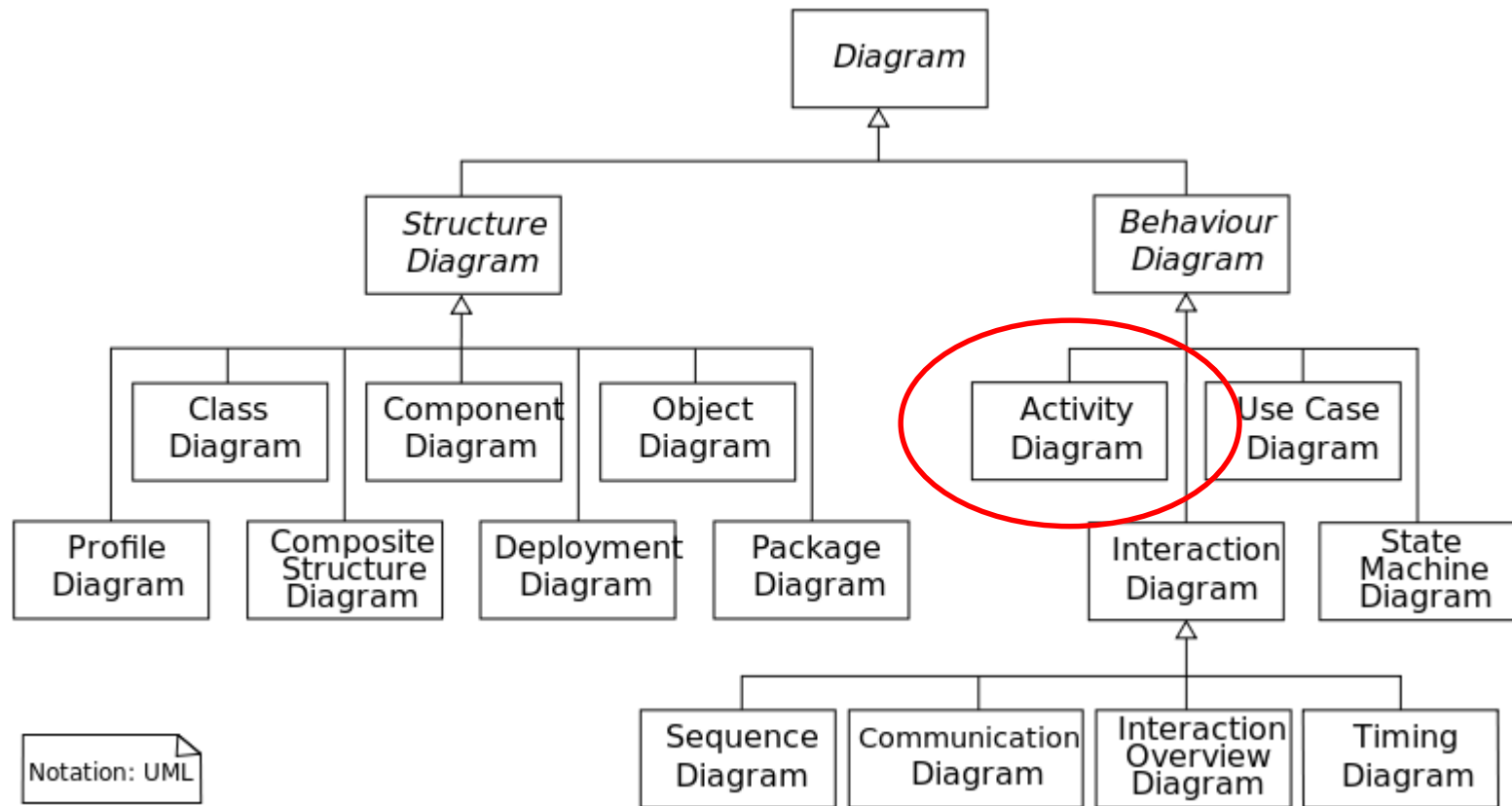


Diagramme d'activité

- Un autre diagramme servant à représenter le comportement dynamique du système (et son effet sur l'environnement)
- Montre en même temps les **flots de contrôle** et les **flots de données**
- Extrêmement utile... quand on en a besoin (pour plusieurs projets ce n'est pas nécessaire; pour d'autres c'est indispensable)
- On s'en sert à la fois pour modéliser des éléments de très haut niveau (modélisation des processus d'affaire) et de très bas niveau (algorithmes)

Exemple introductif – Construction immeuble locatif

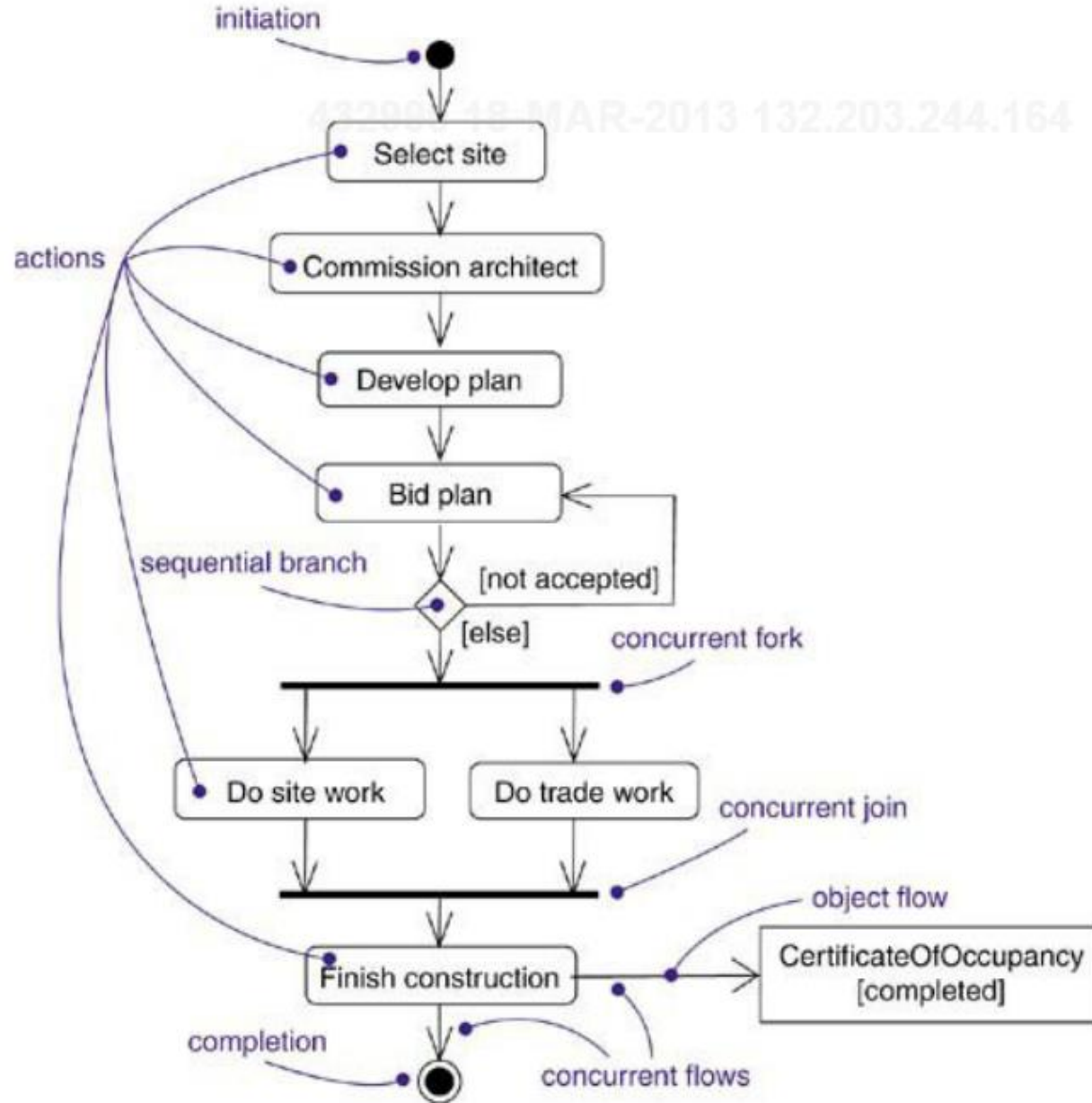
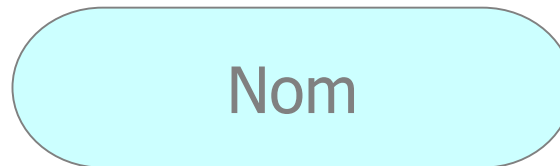


Diagramme d'activité

- Dans un diagramme d'états, ce sont les états d'un système/objet qui sont mis en évidence alors (de même que les transitions entre ceux-ci)
- Dans un **diagramme d'activité**, ce sont ses actions d'un ou plusieurs systèmes/objets qui sont mis en évidence
- On peut aussi montrer l'impact qu'ont ces actions sur l'état des objets
- En bref: un diagramme d'activité représente une exécution d'un processus quelconque ou encore un déroulement d'étapes séquentielles

Action

- Une action modélise une étape dans l'exécution d'un algorithme ou d'un flot de travail (workflow)
- Représentation
 - Symbole



- Le nom n'est pas nécessairement unique dans un diagramme d'activité donné

Action

- L'action peut être étiquetée en langage naturel, en pseudocode ou avec un langage de programmation
 - Exemples
 - appel de procédure
 - création ou destruction d'un objet
 - envoi d'un signal

Ouvrir porte

Dossier.Ouvrir(fichier)

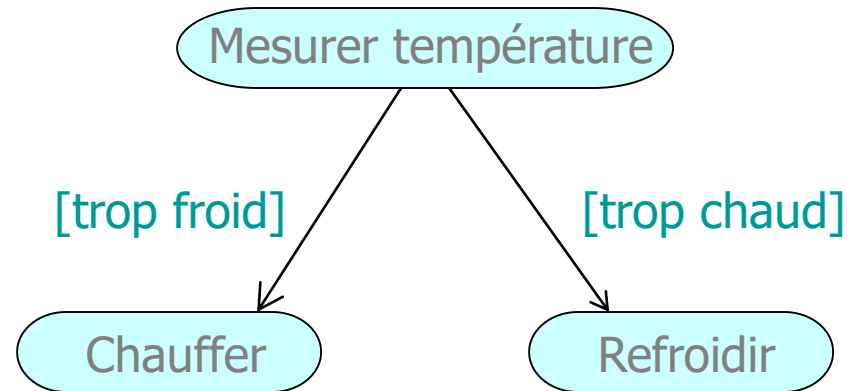
$i := i + 1$

Transitions

- Les **actions** sont reliées par des **transitions** représentées par des flèches
- Lorsque l'action se termine, la transition est automatiquement déclenchée et l'action du prochain état-action démarre
- Il est généralement inutile de mettre un nom d'événement sur la transition dans un diagramme d'activité

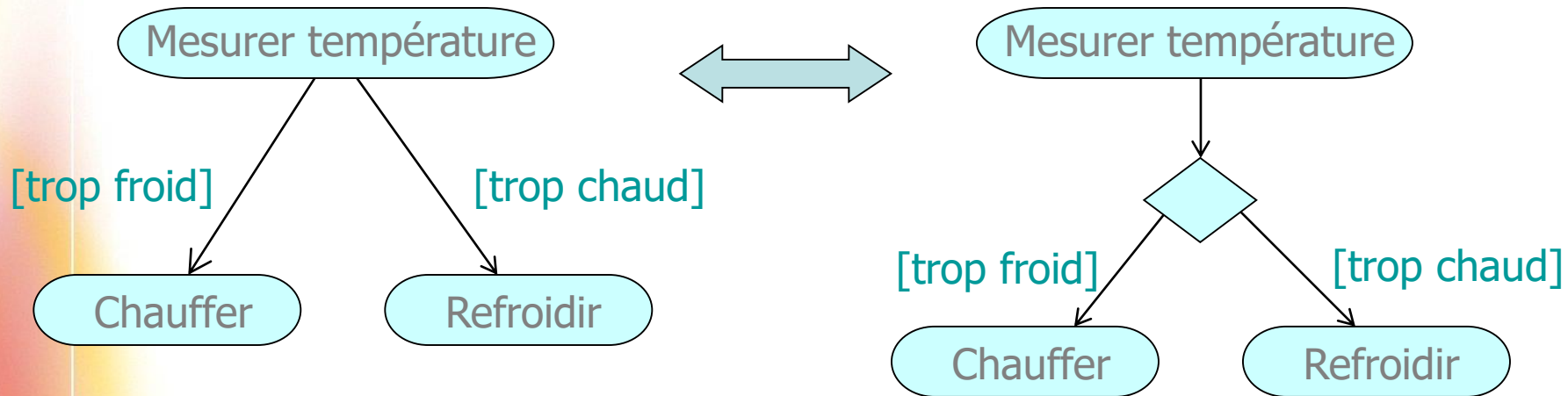
Conditions de garde

- Les transitions peuvent prendre des conditions de garde booléennes, mutuellement exclusives appelées décisions



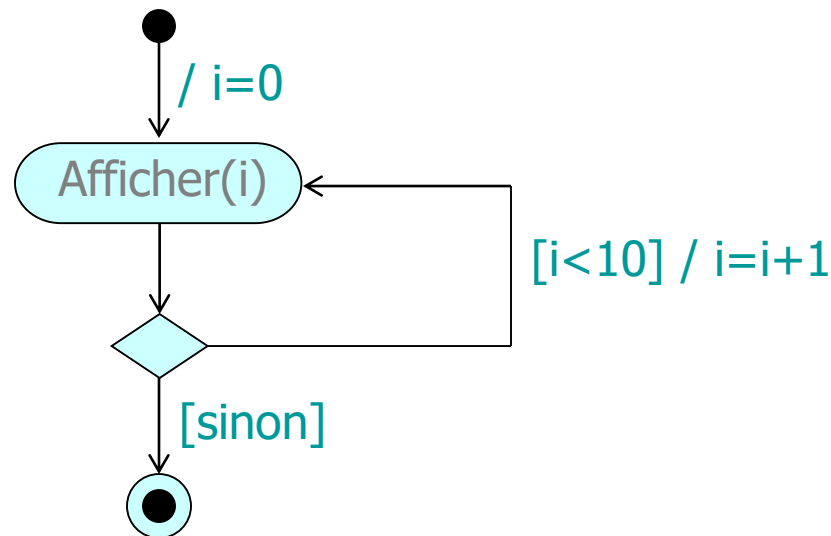
Branchement

- Une alternative peut être exprimée en utilisant un branchement:



Branchement

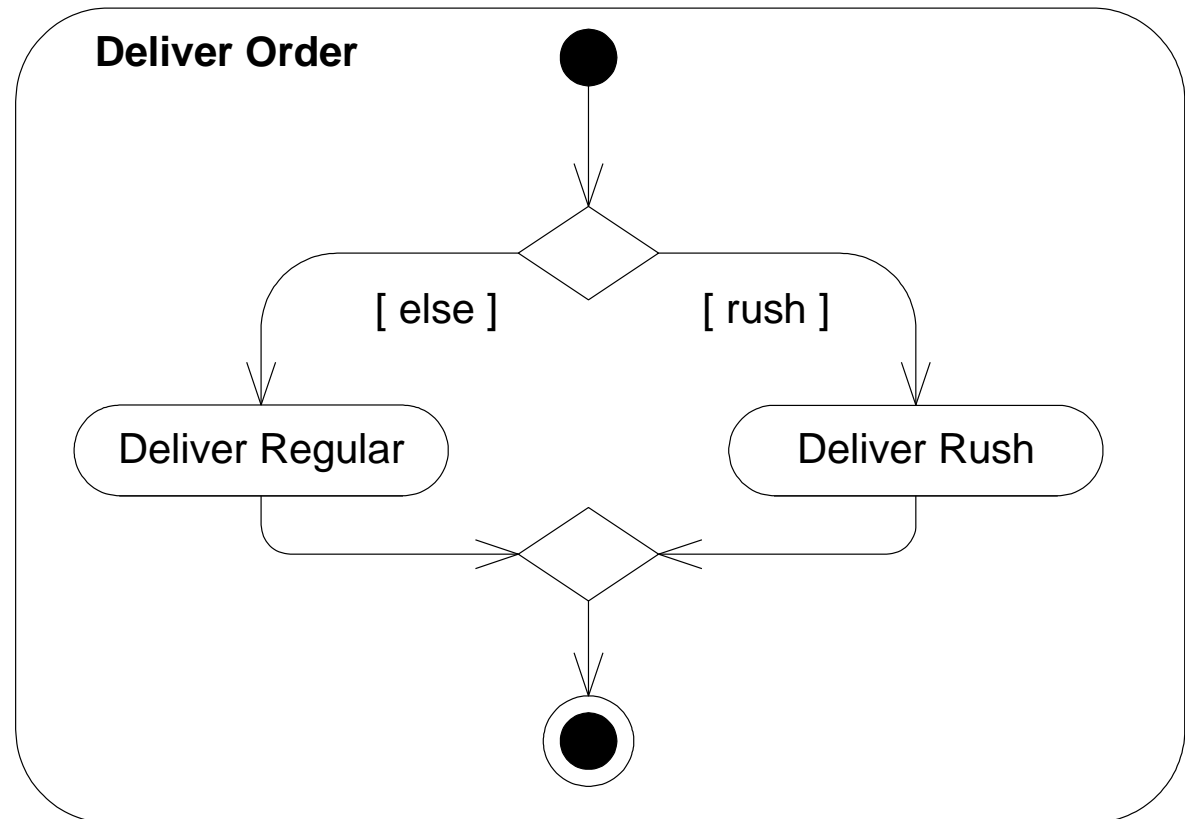
- Il existe une garde prédéfinie [Sinon] (ou [else])
- Une transition avec une garde «Sinon» est effectuée si aucune autre garde des transitions en aval du point de jonction n'est évaluée à vrai
- Exemple : Utilisation d'une garde [Sinon] pour introduire une procédure d'affichage itérative (affichage des entiers de 0 à 9)



Fusion

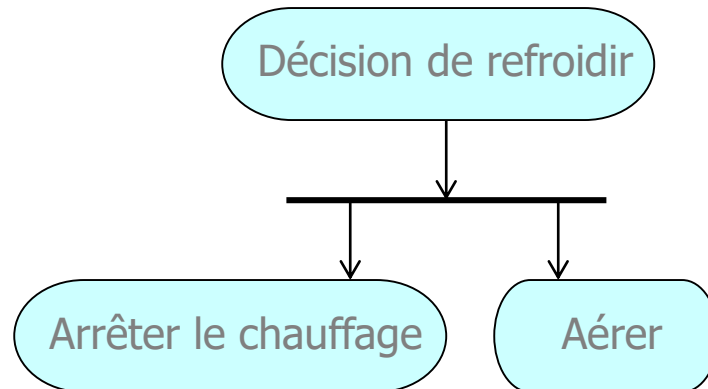
Decision: Any branch happens. Mutual exclusion

Merge: Any input leads to continuation. This is in contrast to a *join*, in which case *all* the inputs have to arrive before it continues.



Synchronisation

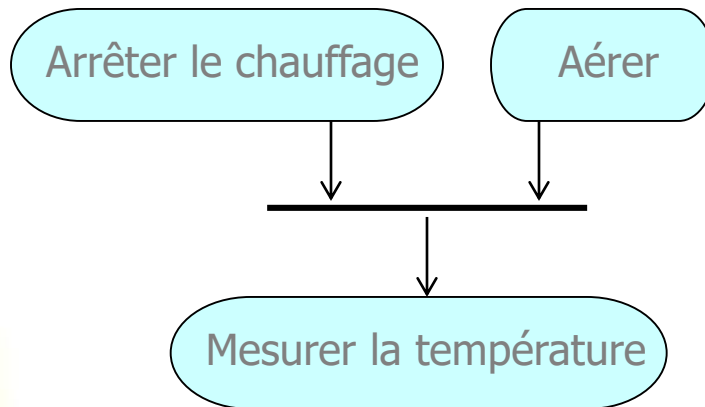
- Une **barre de synchronisation** permet d'ouvrir et de fermer des branches parallèles dans un flot d'exécution d'une méthode ou d'un use-case
- Les transitions qui quittent une barre de synchronisation sont déclenchées simultanément



Pour refroidir, il faut arrêter le chauffage **et** ouvrir les fenêtres

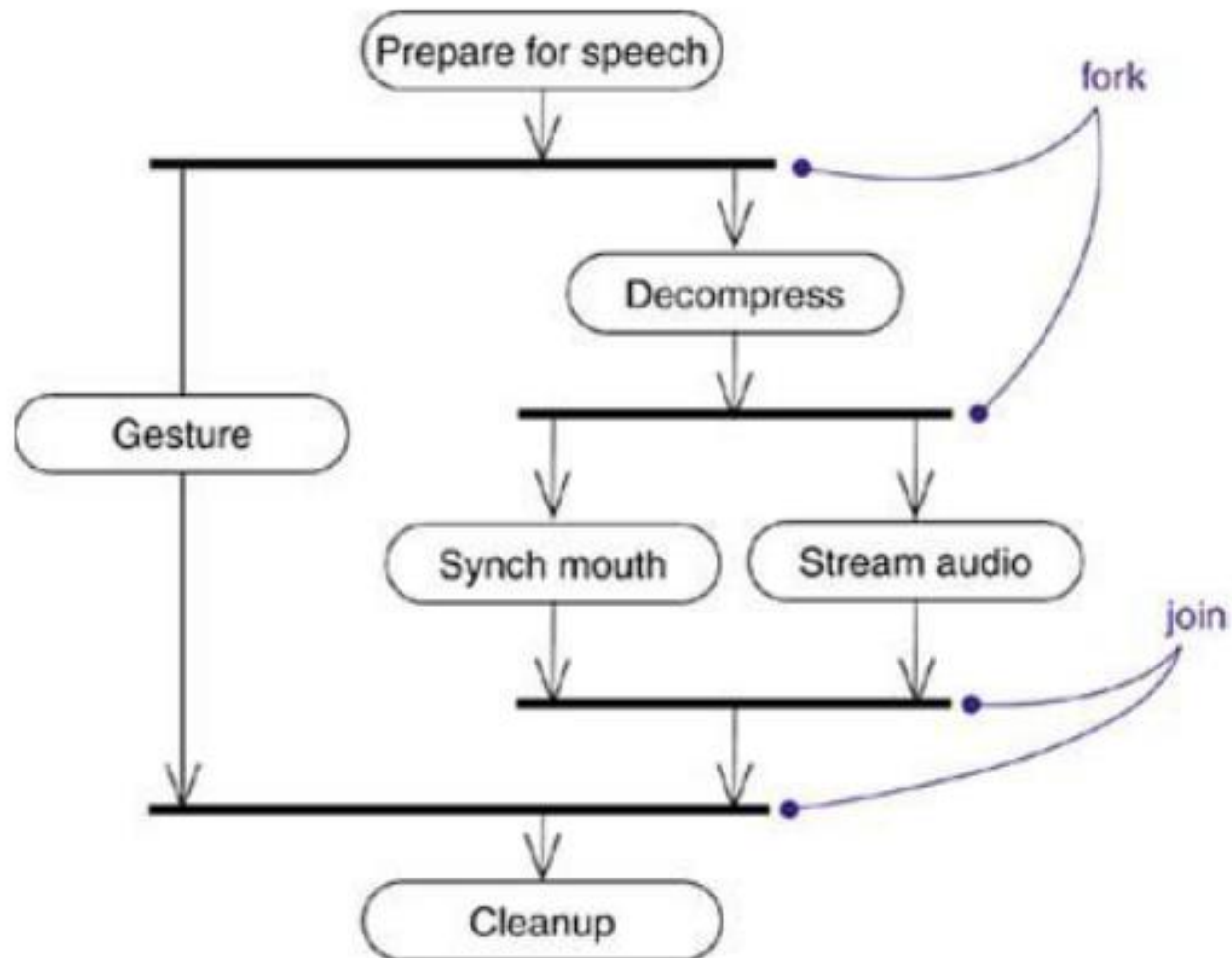
Synchronisation

- Inversement, une barre de synchronisation ne peut être franchie que lorsque toutes les transitions en entrée sur la barre sont déclenchées
 - Exemple : fusion de flots de contrôle parallèles



La mesure de la température est effectuée une fois que le chauffage est arrêté et la pièce est aérée

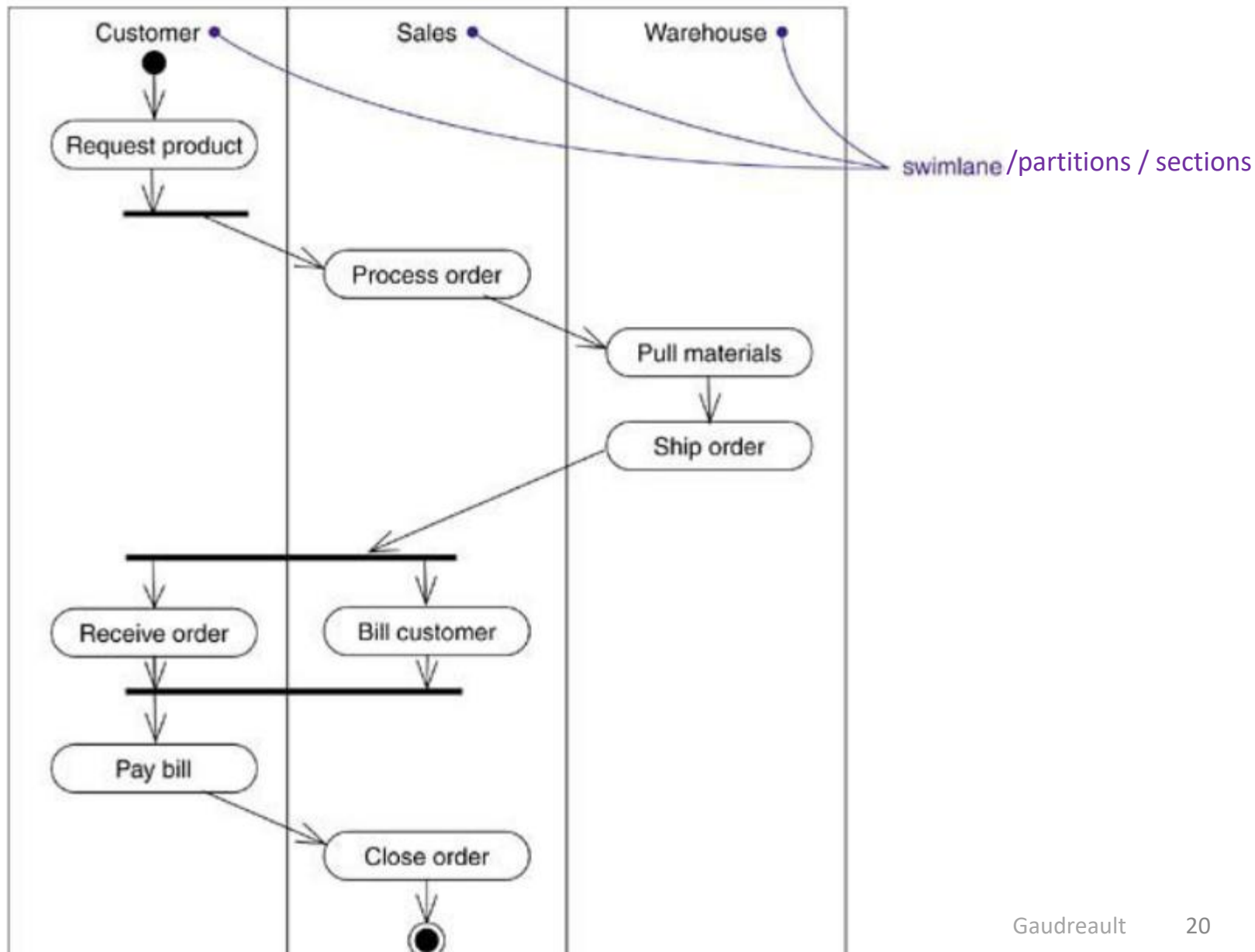
Actions réalisées en parallèle



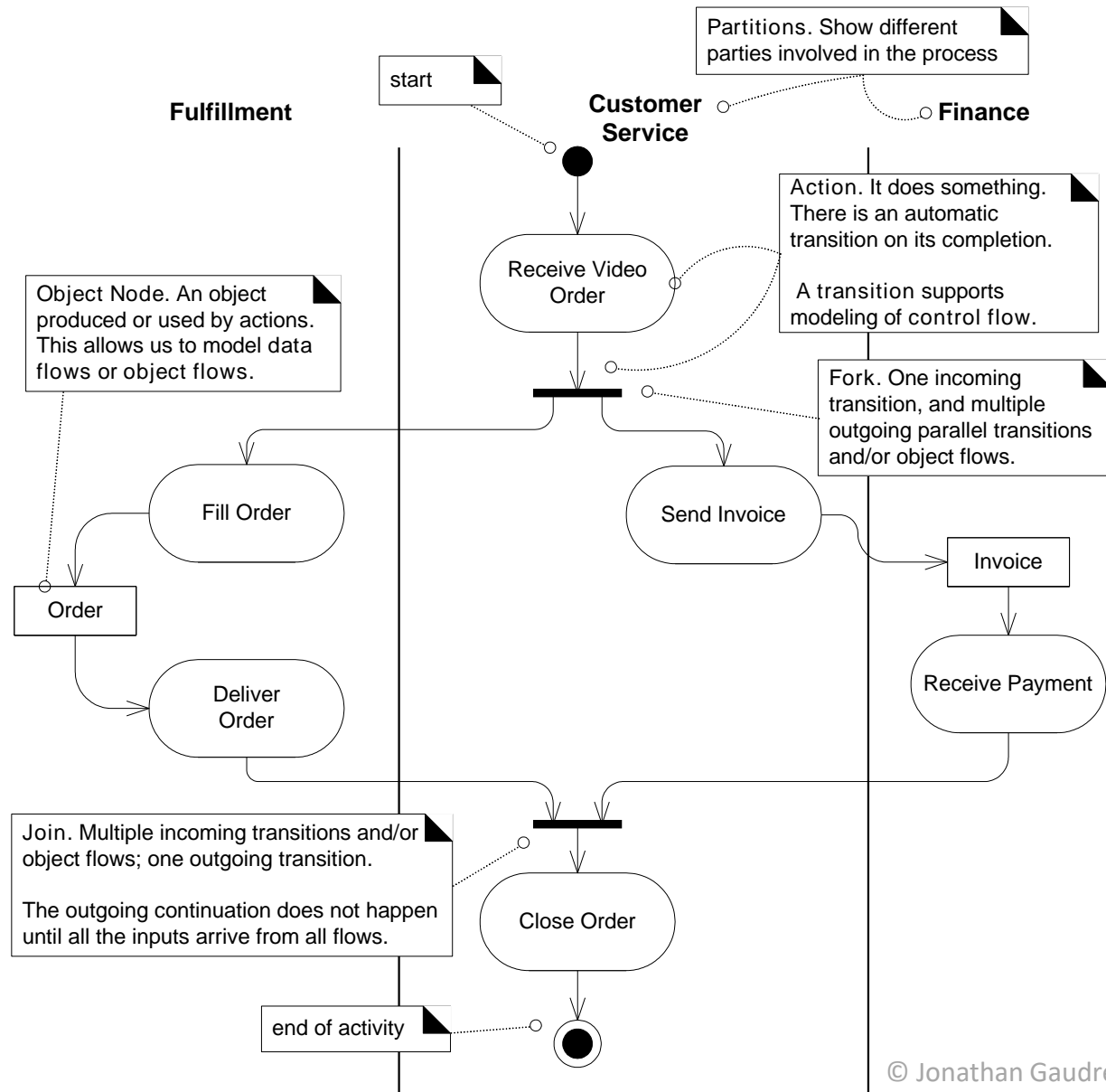
Travées / swimlanes / sections

- Les diagrammes d'activités peuvent être découpés en plusieurs travées (sections, ou swimlanes) pour montrer les responsabilités au sein d'un mécanisme ou d'une organisation
- Chaque responsabilité est assurée par un ou plusieurs **objets/système/sous-système/personnes**
- Les positions relatives des travées ne sont pas significatives. Les transitions peuvent traverser librement les travées

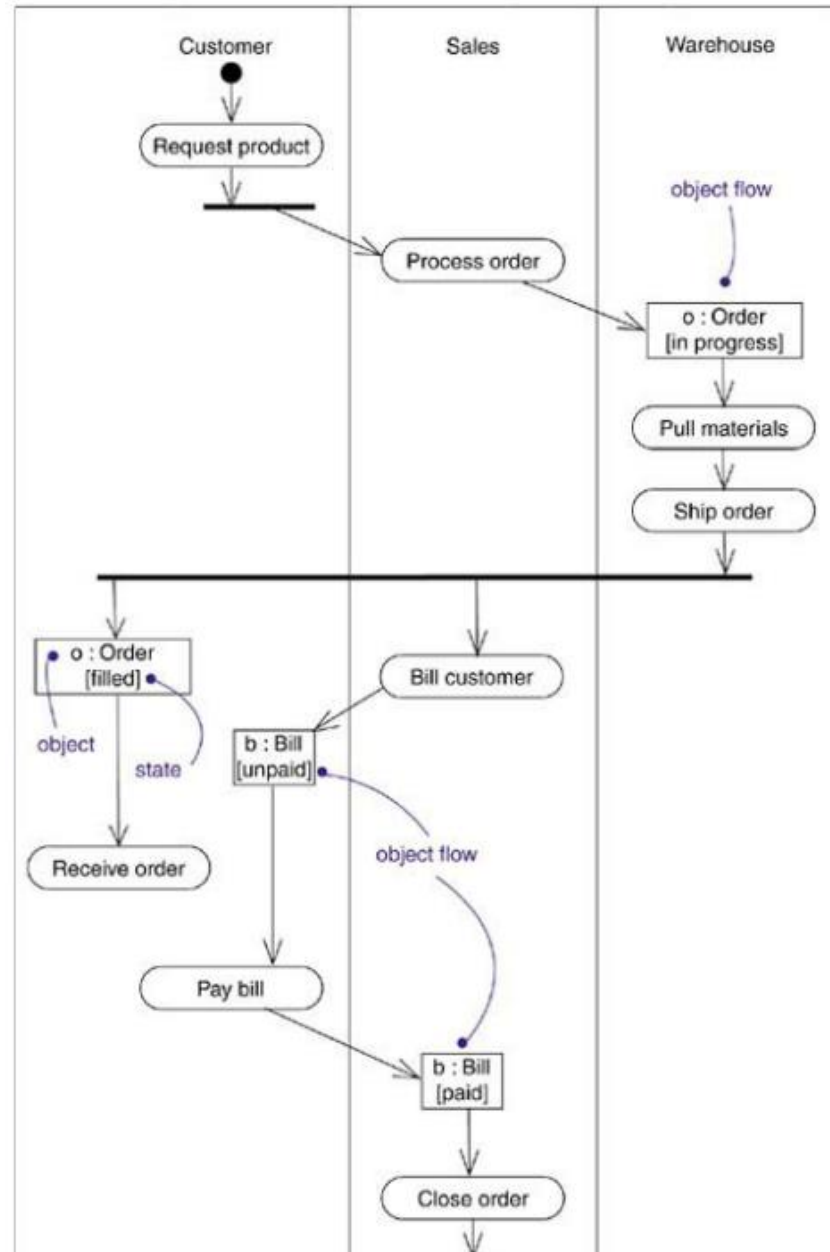
Travées pour illustrer les parties impliquées



Spécifier les objets (ex: achat d'un DVD sur internet)



Spécifier les objets, classes et états



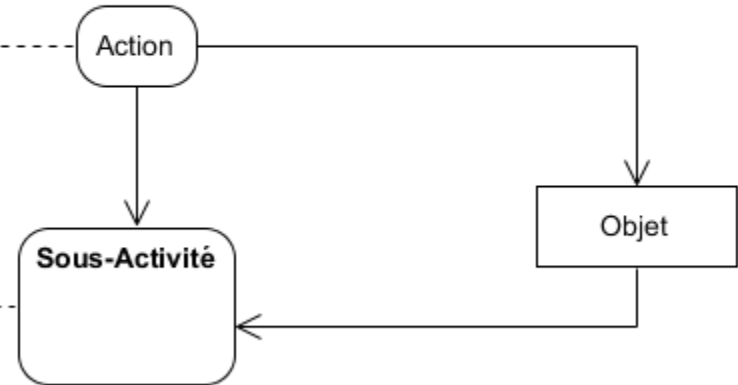
Sous-activité et sous-diagramme

Visual Paradigm for UML Standard Edition (Université Laval)

Le diagramme complet décrit une activité.

Il contient une action

Et une activité (ou sous-activité) elle pouvant elle-même être décrite à l'aide d'un diagramme d'activité.



Dans un "diagramme d'activité" le mot "activité" est toujours au singulier car **le diagramme au complet décrit UNE activité.**

Cette activité (le diagramme donc) est normalement composée d'actions (chaque petite boîte est une action). Chaque action est considérée comme indivisible.

Si une action se subdivise alors elle n'est alors pas réellement une action.

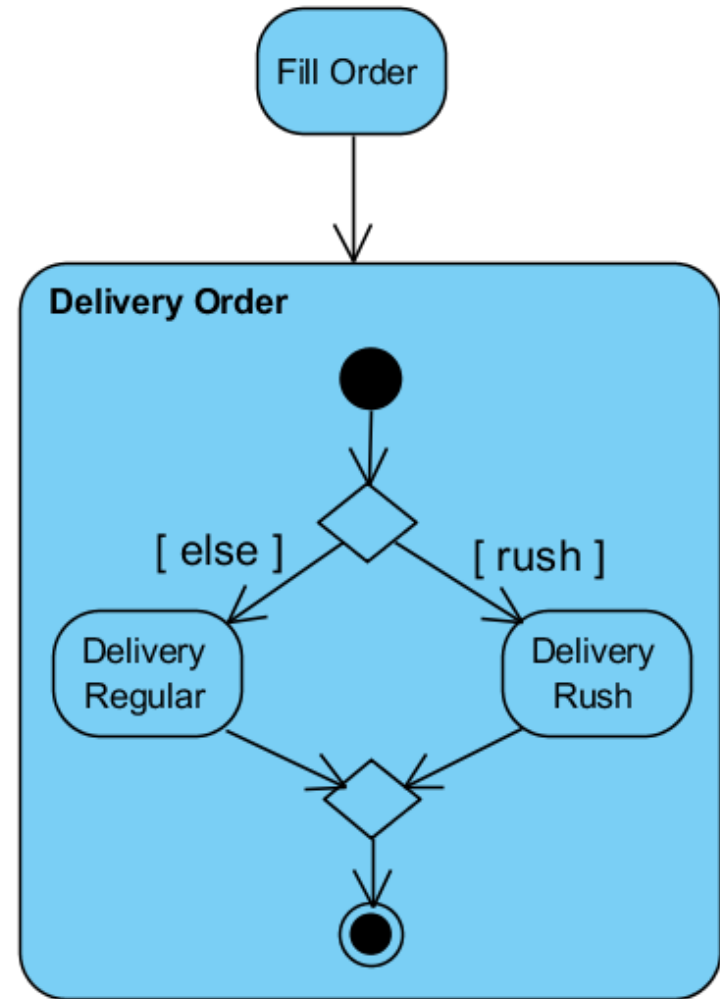
C'est une activité (sous-activité) qui pourrait elle-même être décrite par un diagramme d'activité.

Déclarer une sous-activité

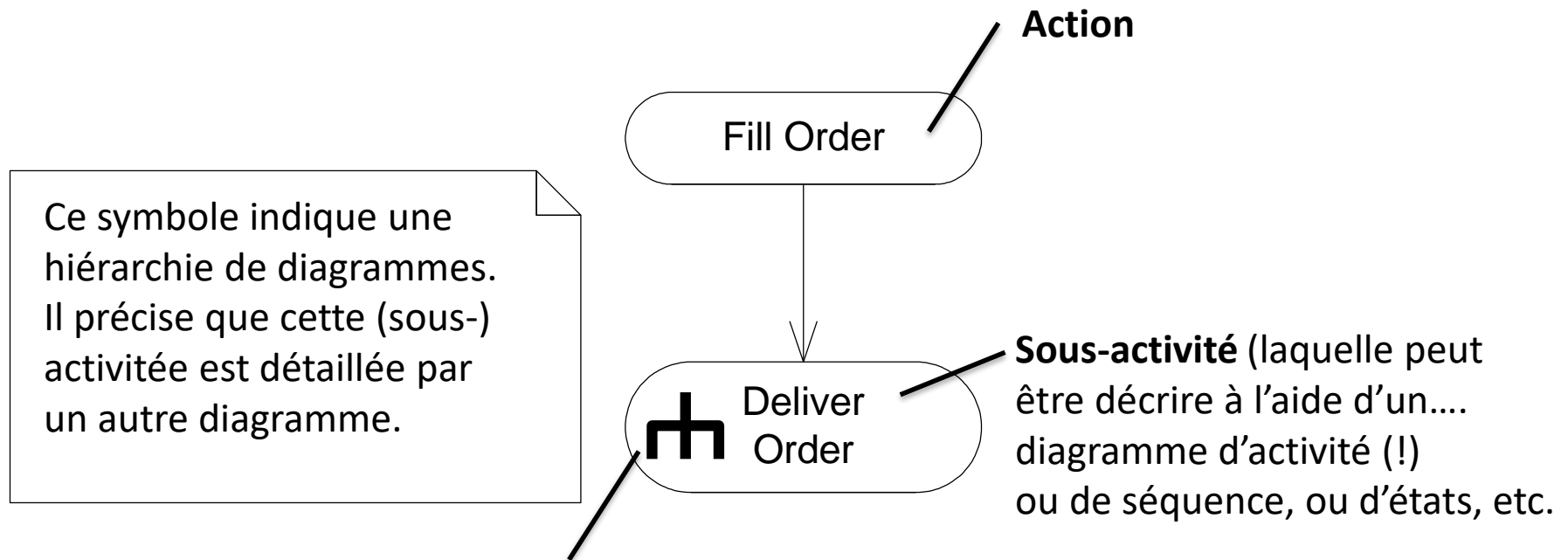
« Delivery Order » n'est pas une action, c'est une sous-activité.

Visual Paradigm nous permet de représenter les sous-activités directement dans le diagramme d'activité.

Toutefois, nous utiliserons normalement un sous-diagramme (voir page suivante).



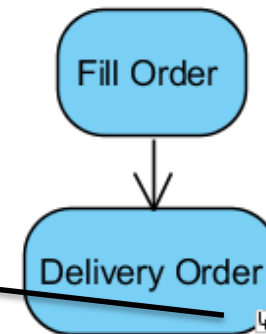
Déclarer un sous-diagramme



Ce symbole n'apparaît pas dans les spécifications de UML 2.0 (?) de même que dans Visual Paradigm!

Visual Paradigm le représente plutôt par un icône en bas à droite du rectangle de l'action.

(activer via l'option « Model Indicator » du menu « View »).

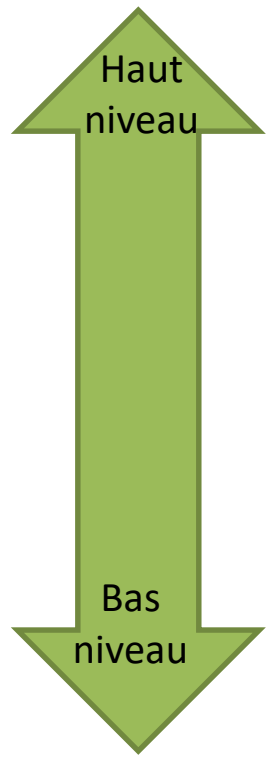


Usages

- Un diagramme d'activité est employé dans diverses phases de développement
- Il peut modéliser le comportement interne d'une méthode (la réalisation d'une opération), d'un use-case ou plus généralement d'un processus
- On modélise mécanismes complexes ou intéressants

Un formalisme, différents usages

- Du général (très haut niveau) au très spécifique (très bas niveau)



Élément modélisé	
1. Modélisation des processus d'affaires	(les “boîtes” correspondent à des “grosses” activités de la vraie vie; certaines sous la responsabilité du système à développer, d'autres non) (il est également possible que les boîtes correspondent à des cas d'utilisation)
2. Décrire de manière visuelle un scénario d'un cas d'utilisation	(chaque boîte correspond à une étape du scénario – possibilité d'éclater en sous-diagrammes)
3. Détailler une étape en particulier d'un cas d'utilisation	(permet de décrire de façon visuelle et détaillée le traitement qui est fait, les objets impliqués, etc.)
4. Un algorithme	(le diagramme d'activité spécifie en détail ce qui se passe lorsqu'une ou des méthodes sont invoquées)

Ou pour décrire tout autre élément pour lequel on sent que ça va nous aider ou bien favoriser la communication

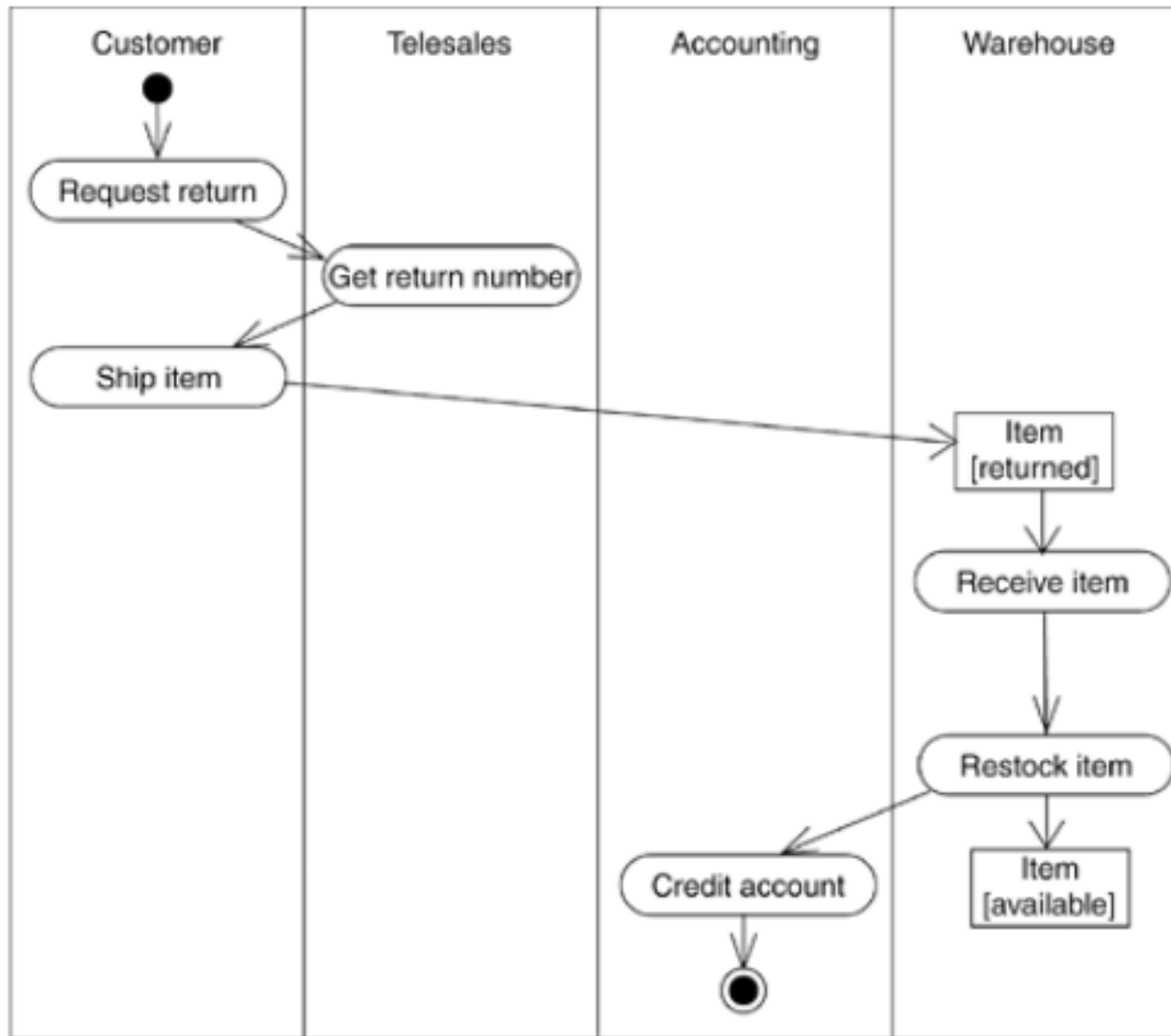
	Activités (appelées <i>disciplines</i> dans le Processus Unifié)	Modèles et artefacts générés
Analyse	Modélisation domaine d'affaires / Business modeling / Modélisation métier	Modèle du domaine: (1) diagramme de classe « conceptuel », (2) parfois un diagramme d'activité
	Analyse des besoins / Exigences / Requirements	(3) Énoncé de vision Modèle de cas d'utilisation / Use-case model : (4) diagramme des cas d'utilisation, (5) texte des cas d'utilisation, (6) diagramme de séquence système (7) Spécifications supplémentaires (8) Glossaire
	Design / Conception	Modèle de conception / Design model : (9) diagrammes de classes, (10) diagrammes d'interaction, (11) tout autre diagramme UML pertinent selon le contexte
	Implémentation	(12) Code
	...	

Très utile ici

Parfois pratique pour décrire un scénario

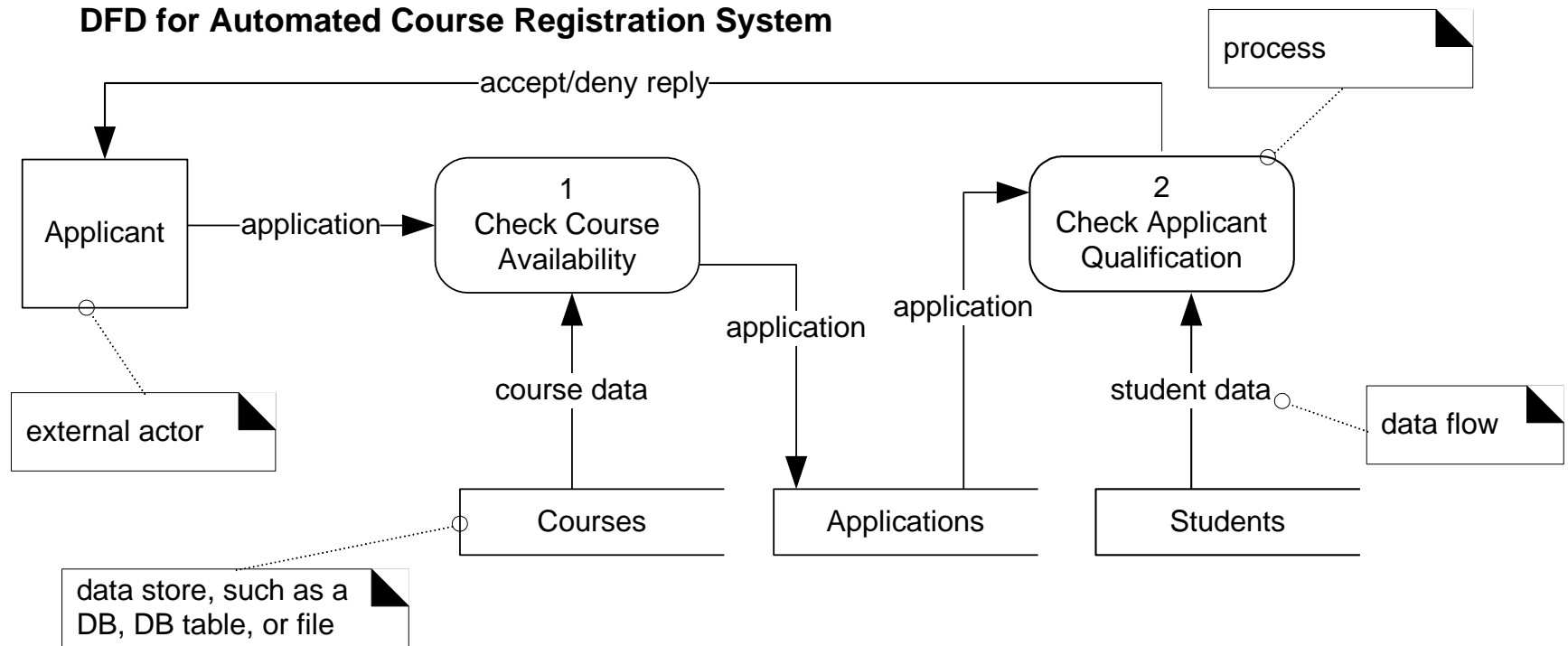
Pratique pour décrire un aspect particulier du système

1. Modélisation des processus d'affaires

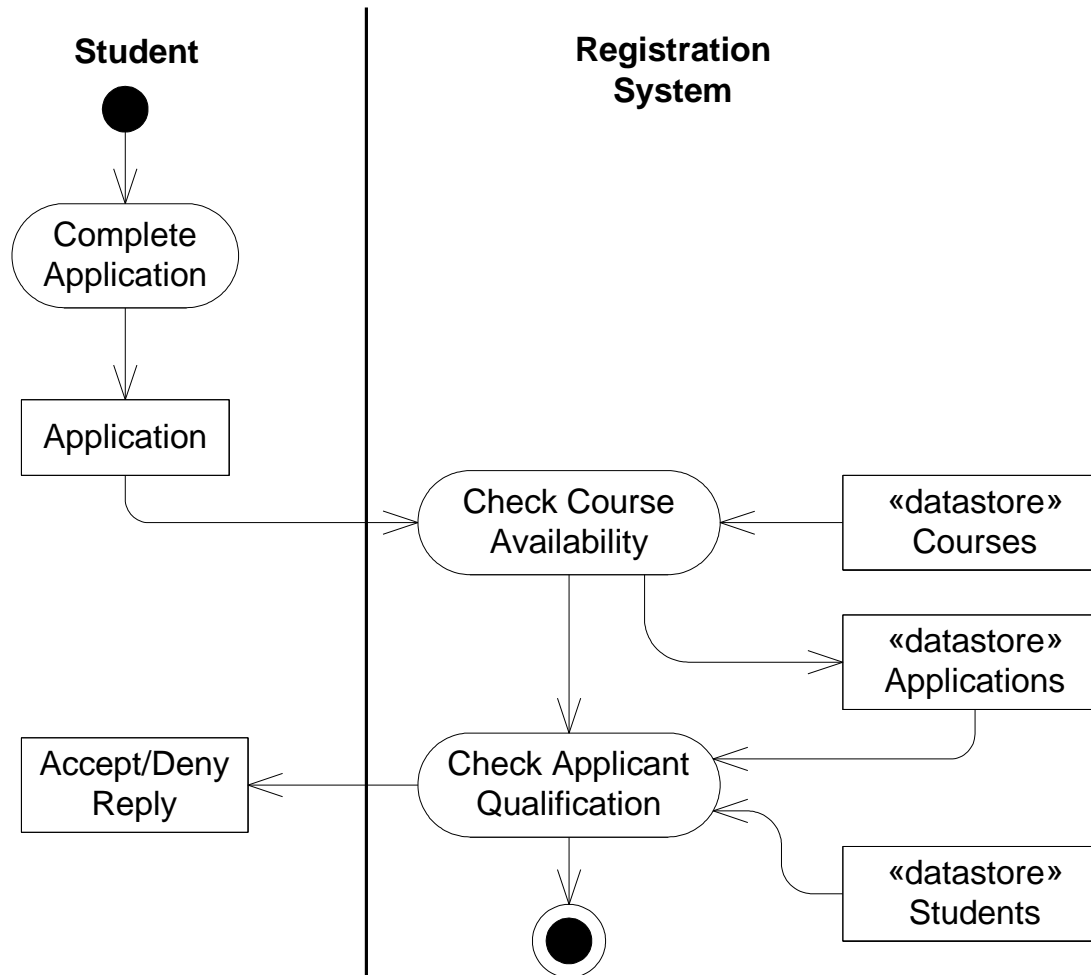


Remarquez
l'état des
objets

Rappel: Diag. de flux de données (DFD)

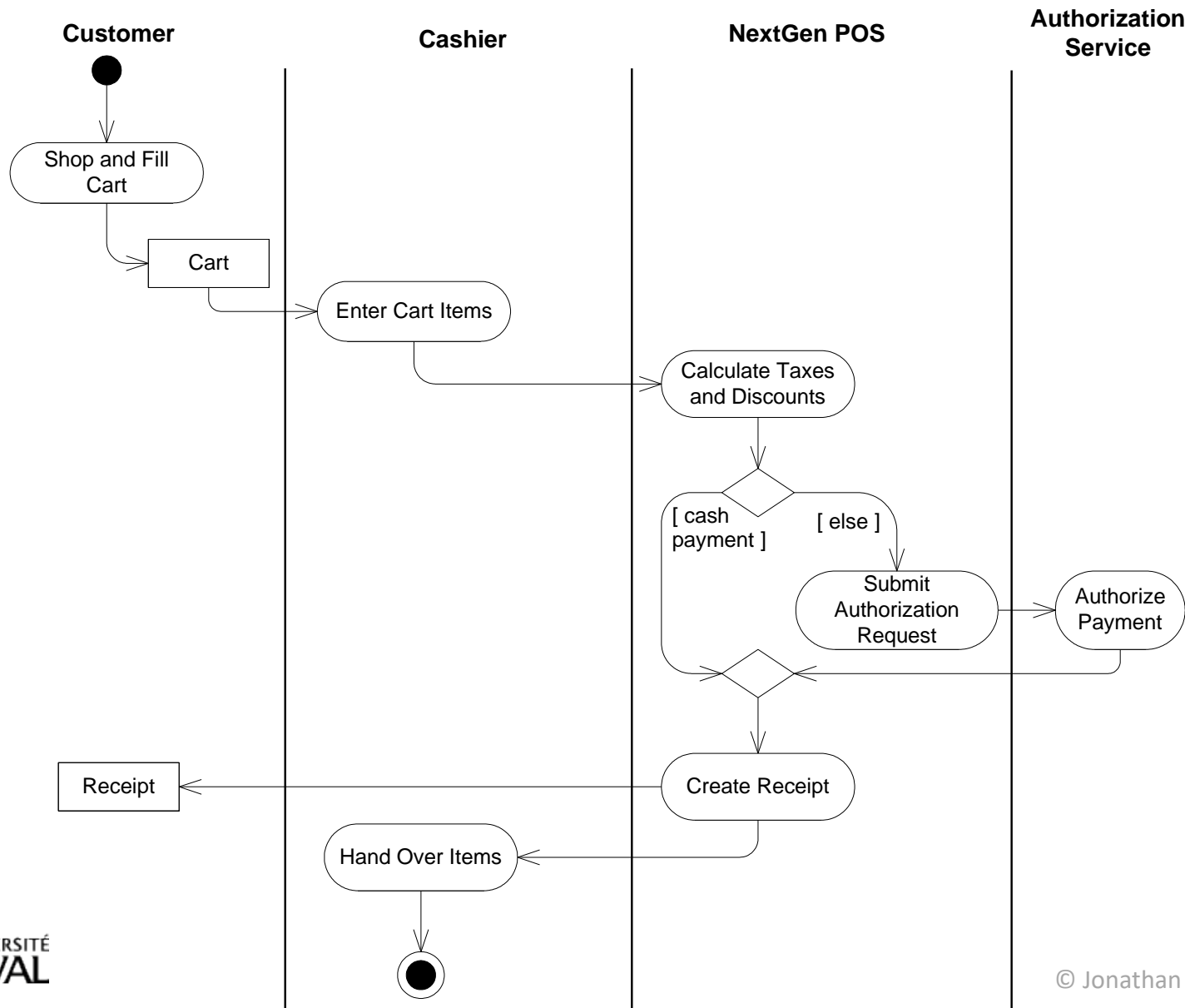


Même exemple avec formalisme UML



2. Modélisation d'un cas d'utilisation

Exemple NexGen pour cas d'utilisation “Process Sale”



3. Décrire une partie d'un cas d'utilisation

- Exemple: le cas d'utilisation « Exécuter une procédure » de l'application Robo sapiens

Robo sapiens



* A.ROB *

Sortie 1



Sortie 2



☐ Lier les sorties

☒ Mémoriser

Interrupteur 1:



Interrupteur 2:



Exécuter une
procédure



Ouvrir



Nouveau



Enregistrer



Quitter



Aide

```
; La procédure « Longe_un_mur » fait ceci
; ■ le robot avance jusqu'à rencontrer un
;   obstacle (interrupteur 1)
; ■ il se tasse d'une largeur et recommence
;   jusqu'à ce qu'il se retrouve a l'extrémité
;   de l'obsacle

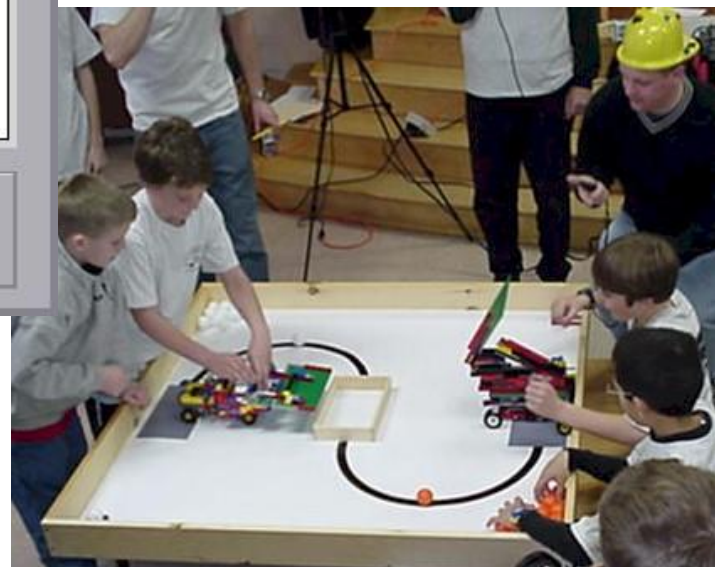
; Ce fichier contient d'autres procédure
; intéressantes (utilisées par Longe_un_mur)
; Avance_Un_Peu, Recule_Un_Peu
; FonceDansObstacle, Avance_D'une_Longueur
; Tasse_D'une_Largeur
```

```
[Longe_Un_mur]
FonceDansObstacle
Tant que il actif Fais Tasse_d'une_largeur
```

```
[TasseD'uneLargeur]
ReculeUnPeu
Droite90
attends 0,5
AvanceD'uneLongueur
Gauche90
AvanceUnPeu
```

```
[ReculeUnPeu]
recule
attends 0,3 sec
arreteetout
```

```
[AvanceUnPeu]
Avance
attends 0,4 sec
ArreteTout
```



Le cas d'utilisation le plus complexe: « Exécuter une procédure »

[Ma_procédure]

Si interrupteur 1 actif, alors moteur 1 positif,

Sinon, moteur 1 négatif

Attends 5 secondes

Moteur 1 arrêté



Il faut bâtir un
interpréteur !

a. Transformation du texte en éléments appelés« jetons »

[Ma_procédure]

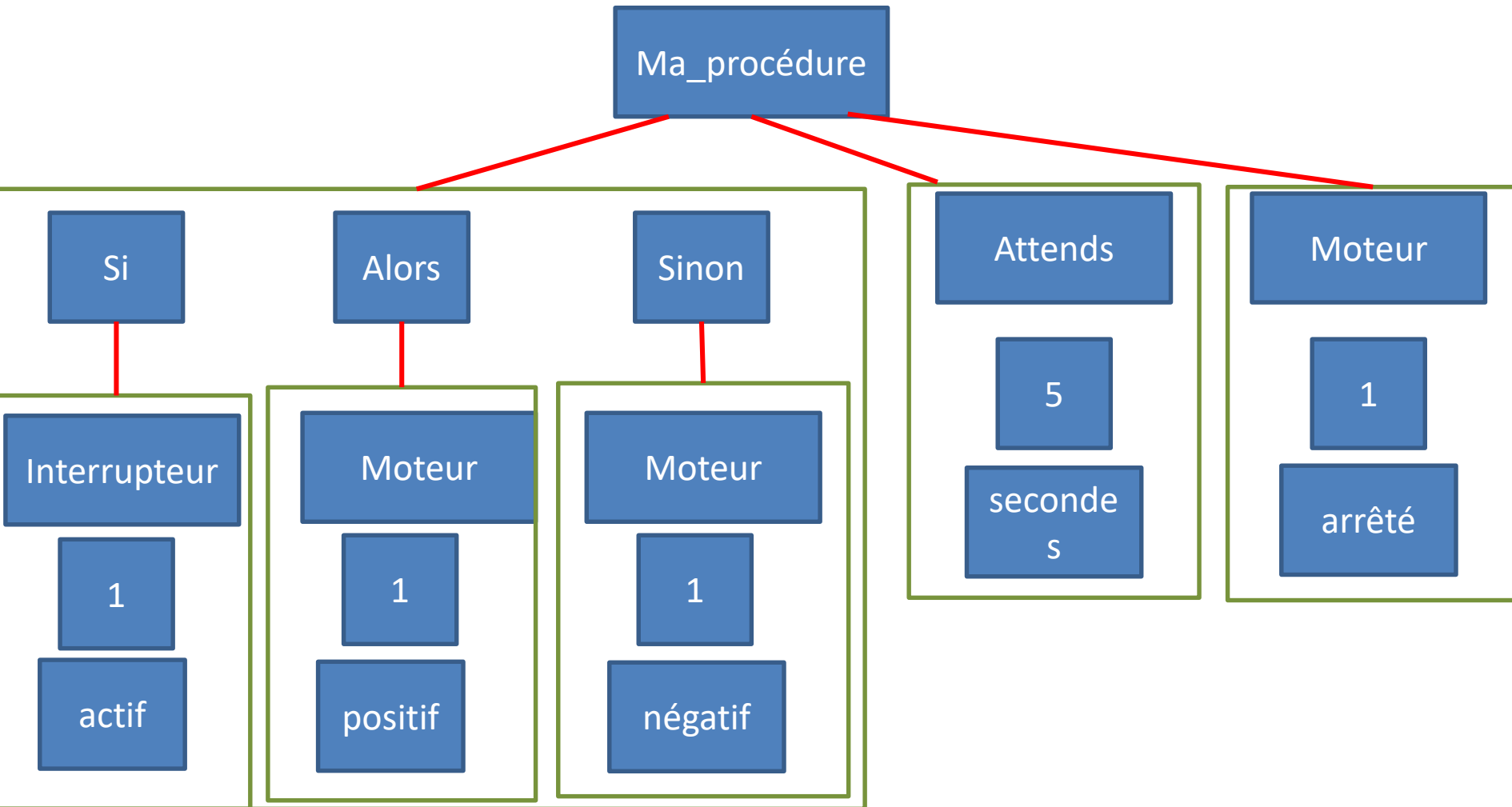
Si interrupteur 1 actif, alors moteur 1 positif,

Sinon, moteur 1 négatif

Attends 5 secondes

Moteur 1 arrêté

b. Mettre en évidence la structure du programme rédigé par l'utilisateur



Réflexion

- Il aurait été plus pratique de créer un diagramme d'activité dès le départ...
- Assurément avant de produire le diagramme de séquence...
- Probablement en parallèle de la réalisation de notre modèle de conception et notre architecture logique.

Diagramme d'activité correspondant

Visual Paradigm for UML Standard Edition (Université Laval)

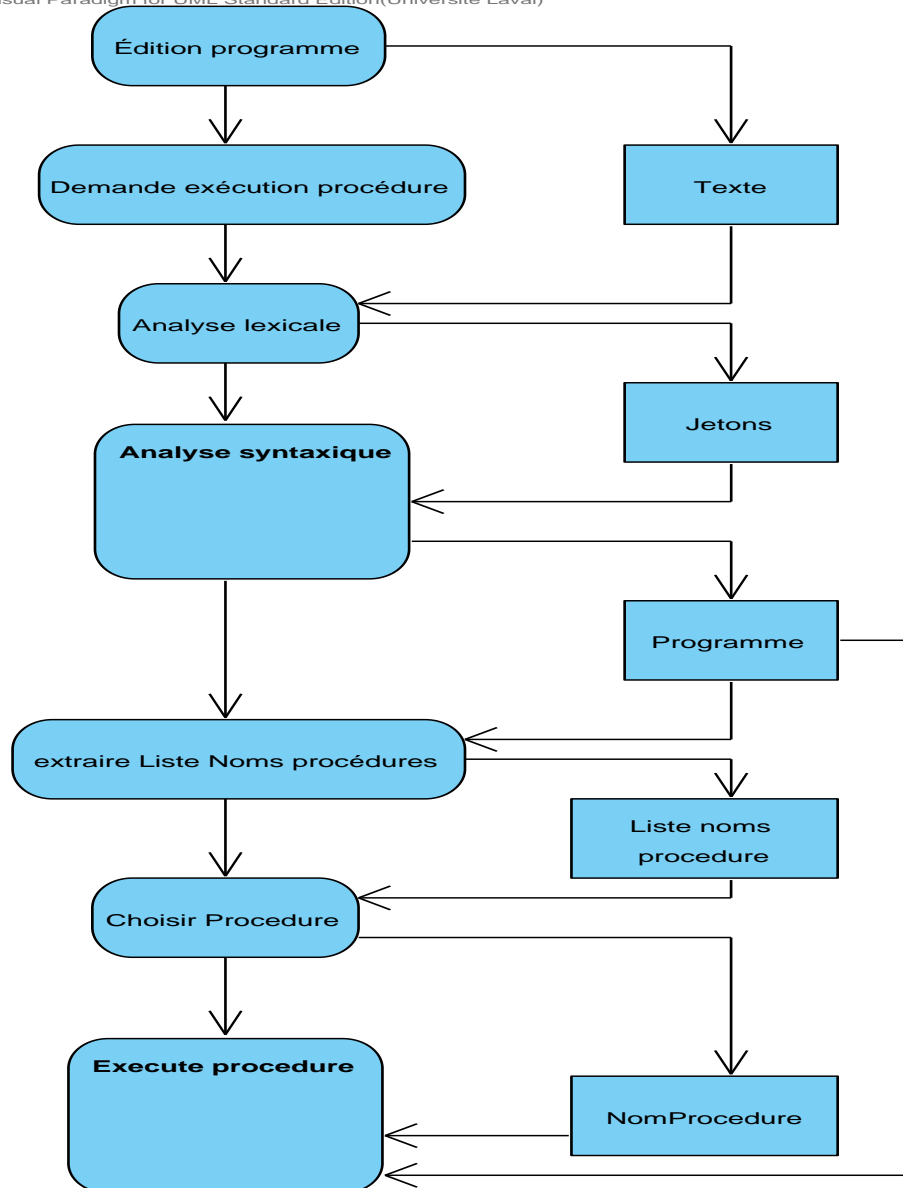
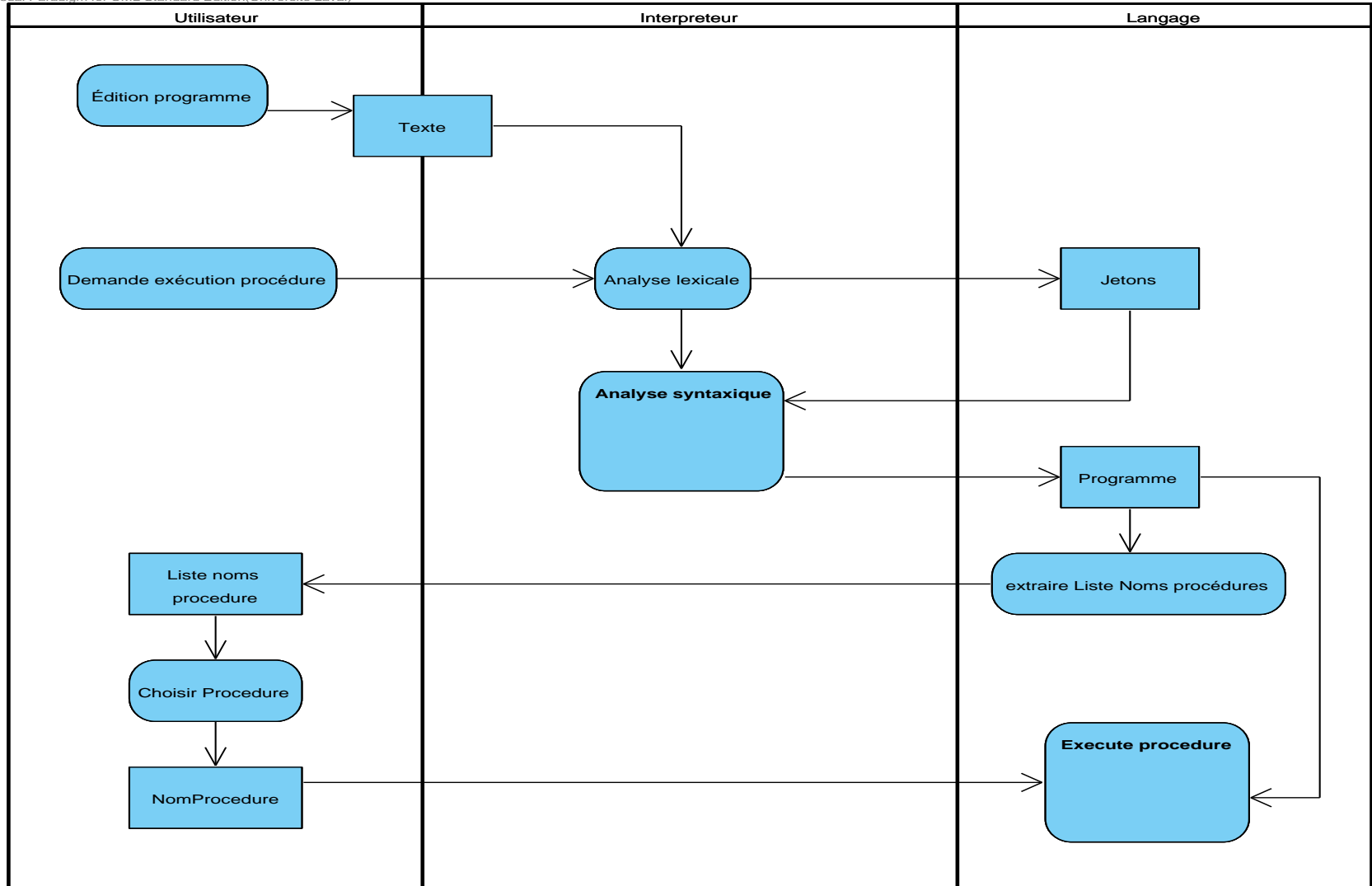
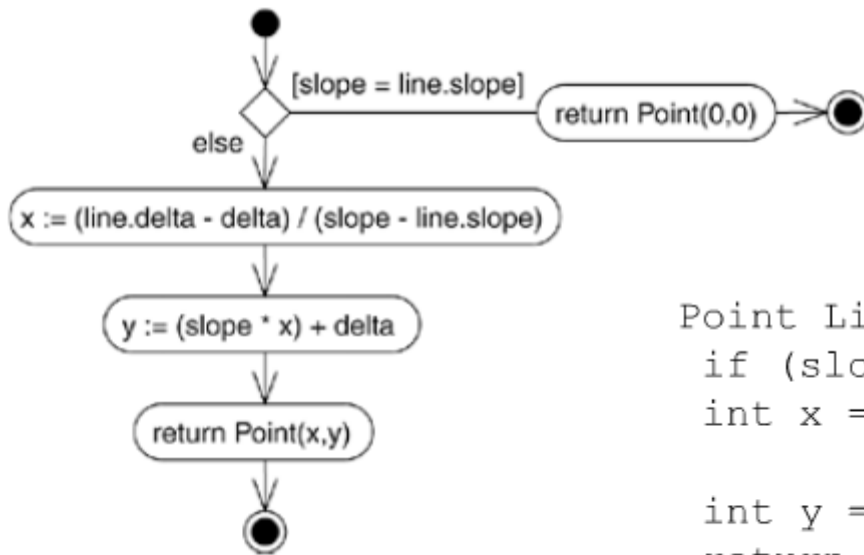


Diagramme d'activité correspondant

Visual Paradigm for UML Standard Edition (Université Laval)



4. Algorithme

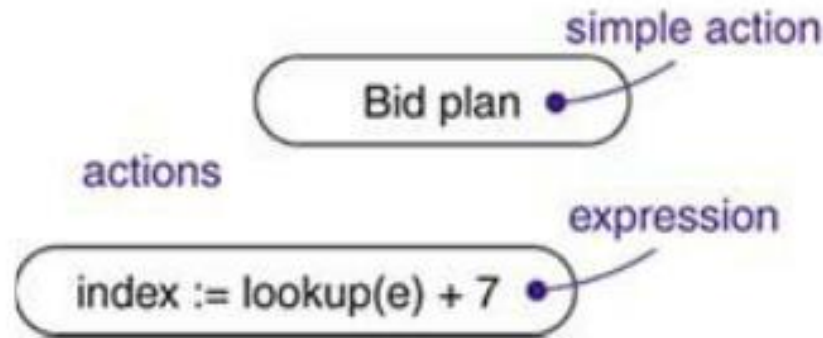


```
Point Line::intersection (line : Line) {  
    if (slope == line.slope) return Point(0,0);  
    int x = (line.delta - delta) /  
            (slope - line.slope);  
    int y = (slope * x) + delta;  
    return Point(x, y);  
}
```

- Exemple extrême... dans ce cas-ci cela revient à coder l'algorithme
- Dans quel cas serait-ce plus pertinent?

Niveau de détail des actions

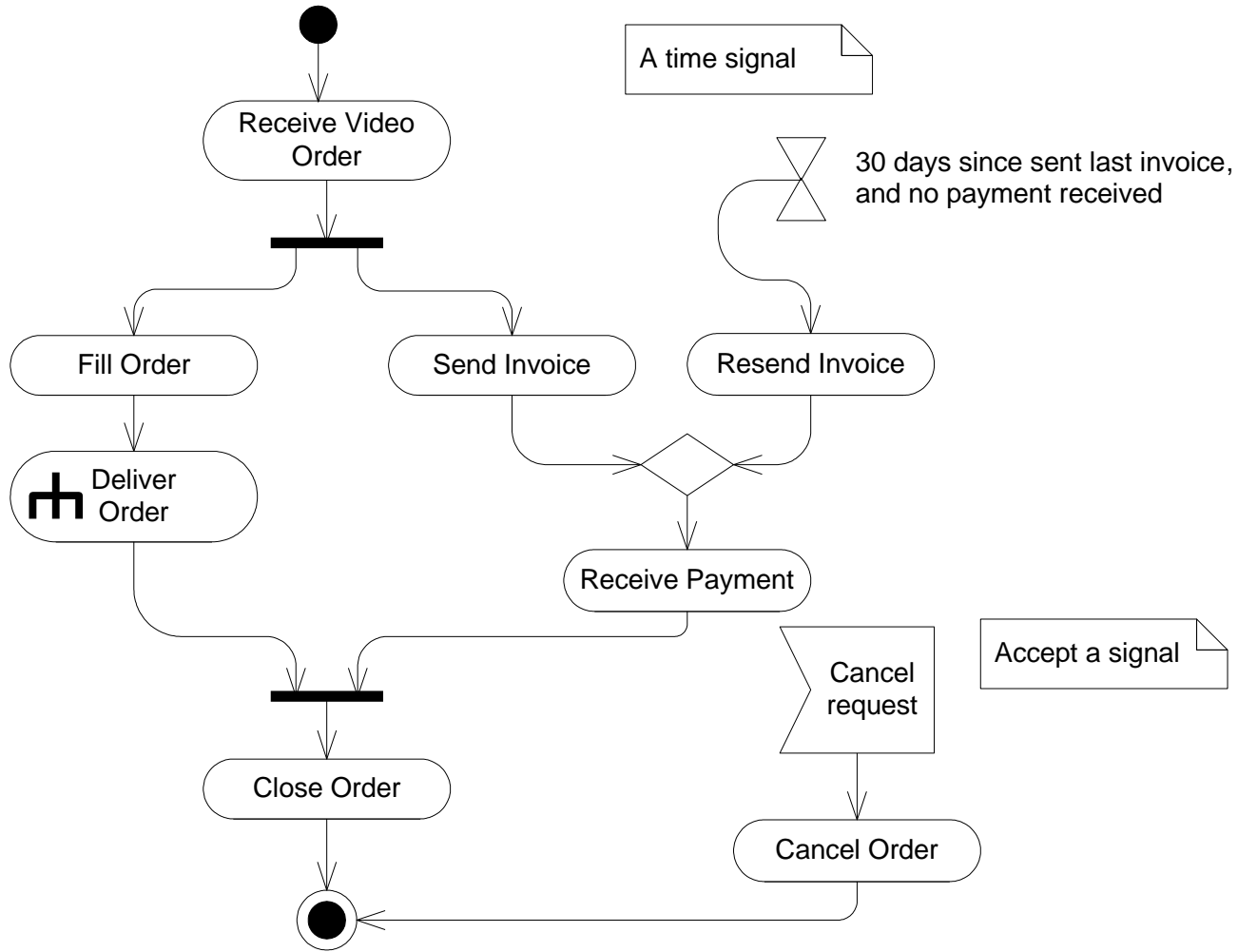
- Le diagramme permet différents niveaux d'abstraction (niveaux de détail).



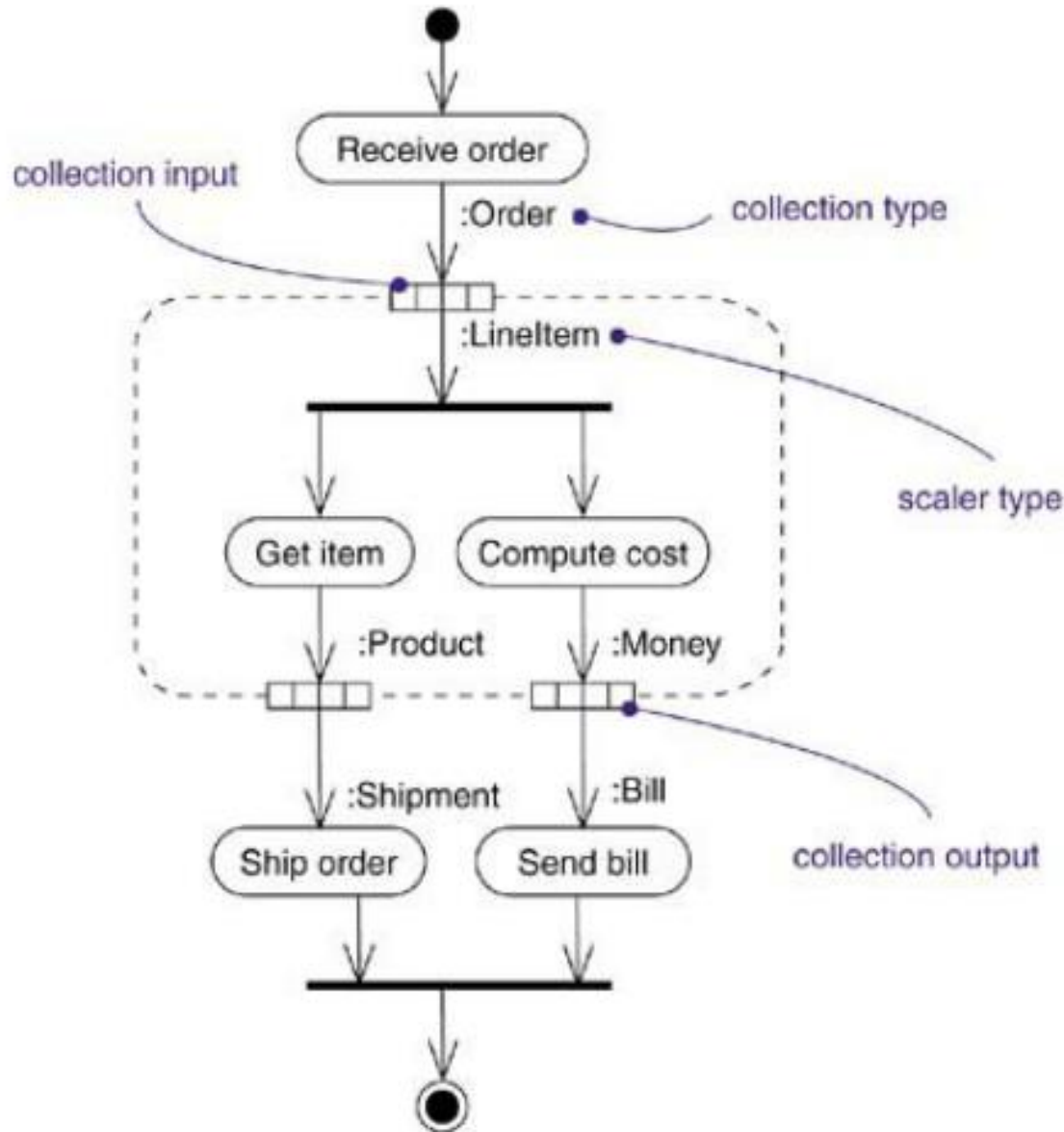
- Pour que le diagramme soit cohérent, il faut utiliser un niveau de détail uniforme pour l'ensemble d'un même diagramme (en considérant l'utilisation potentielle de sous-activités).

Concepts avancés

Signal



« Expansion » - Traitement itératif



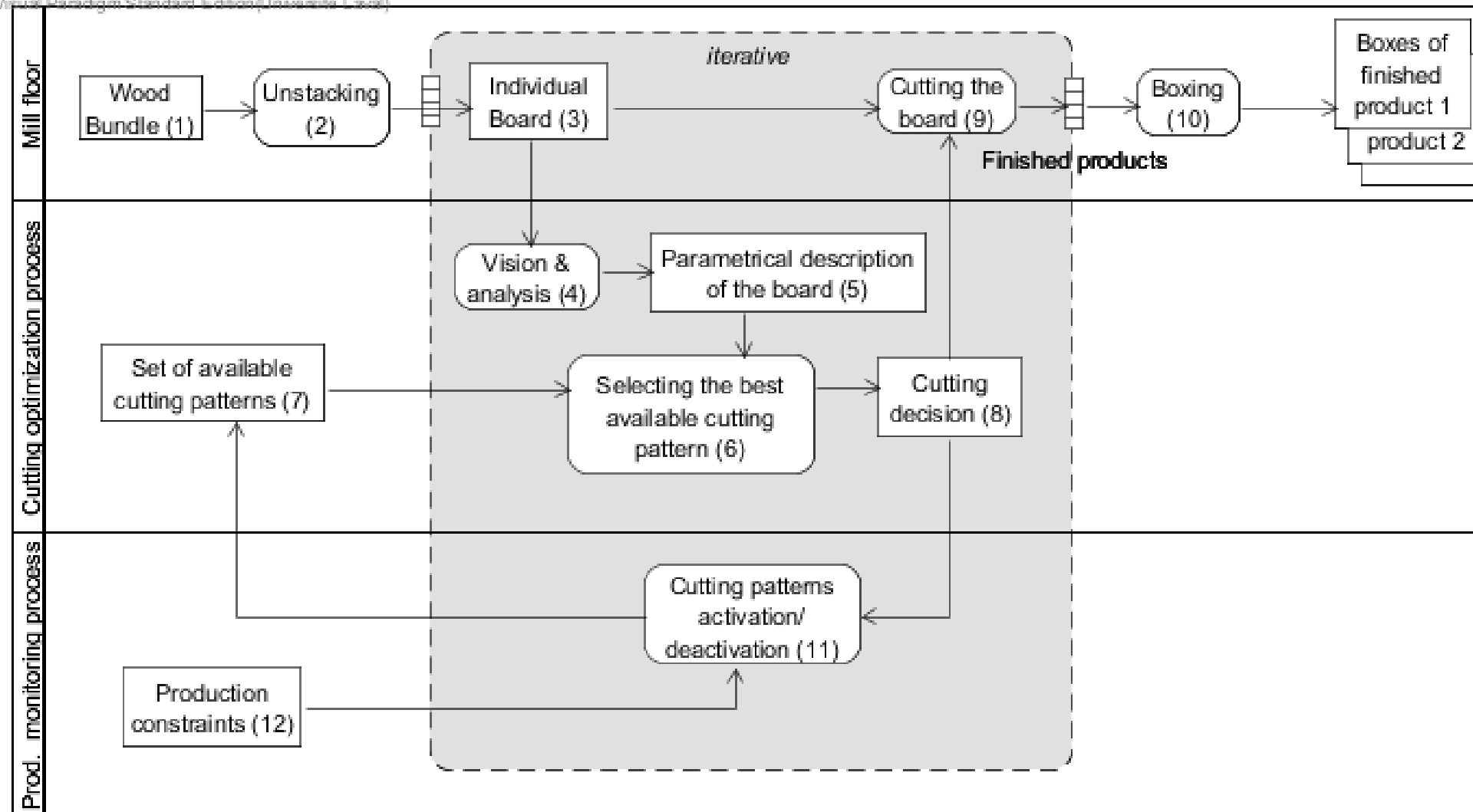
Facilite la modélisation des itérations sur une *collection d'objets*.

Dans l'exemple ci-contre, nous modélisons une boucle sur l'ensemble des lignes de vente en **entrées**.

Pour chacune de ces lignes, la **sortie** sera un produit et un montant.

Exemple – Traitement itératif

Visual Paradigm Standard Edition (University Level)

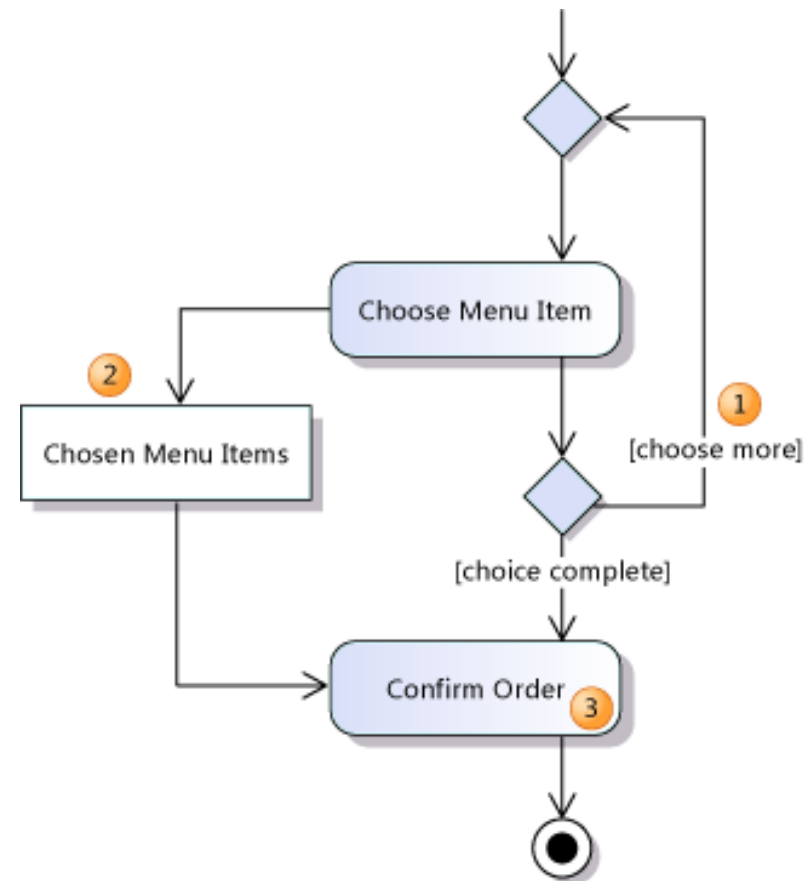


Buffers

Un objet peut agir comme un “buffer” de plusieurs objets.

Dans cet exemple, le client commande plusieurs items à l’aide d’une boucle (1) qui sont stockés dans l’objet (2).

Lorsqu’il a terminé, une autre action (3) permet de confirmer la commande et fera usage de tout ce qui a été conservé.



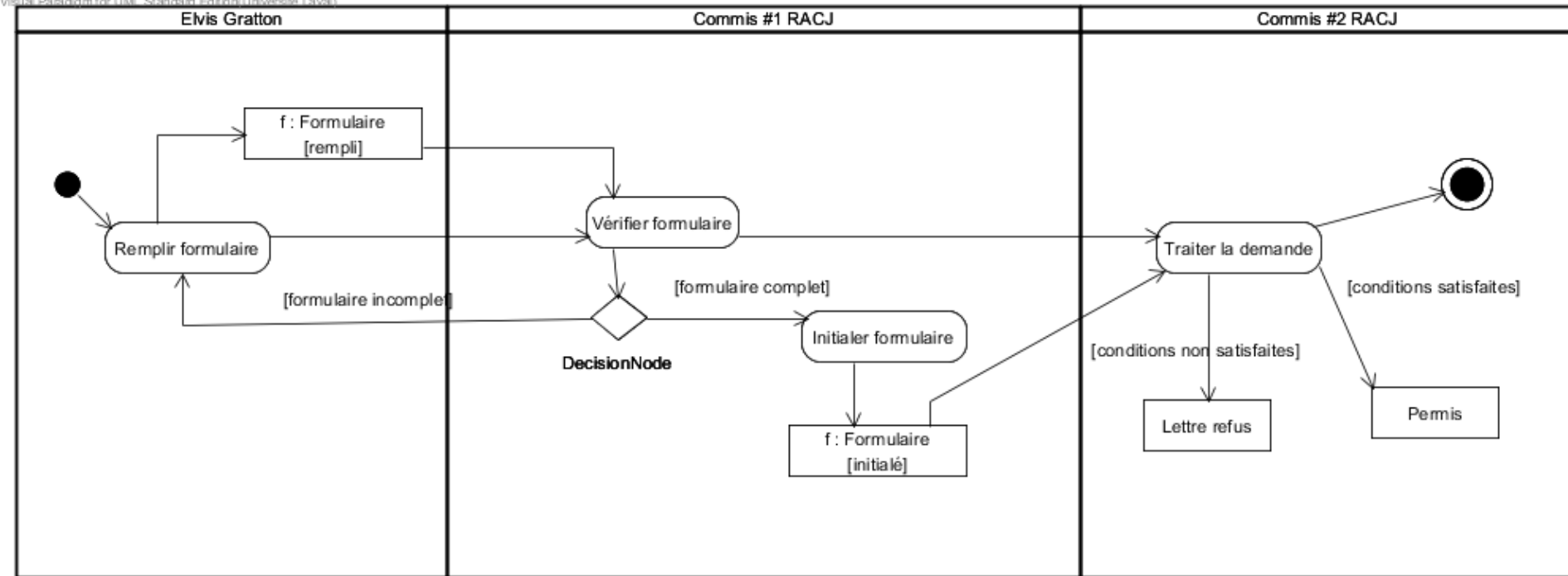
Exercice - diagramme d'activité

- « Elvis Gratton désire obtenir un permis de bière.
 - Il doit d'abord remplir le formulaire de demande.
 - Un premier commis de la *Régie des alcools, des courses et des jeux* vérifie le formulaire et s'il est bien rempli il y appose ses initiales.
 - Un second commis traite la demande.
 - Si Elvis Gratton remplit toutes les conditions, la demande est approuvée et le permis est accordé. Sinon, une lettre de refus est émise.
- Tracez le diagramme d'activité. Identifiez les trois acteurs et les actions respectives. Représentez bien les objets impliqués (Formulaire de demande et Permis) de même que les différents états s'il y a lieu. Représentez les flux de données et les flux de contrôle. »

- Elvis doit d'abord remplir le formulaire de demande.
- Un premier commis vérifie le formulaire, et s'il est bien rempli il y appose ses initiales.
- Un second commis traite la demande.
- Si les conditions sont remplies la demande est approuvée et le permis est accordé.
Sinon, une lettre de refus est émise.

- Elvis doit d'abord remplir le formulaire de demande.
- Un premier commis vérifie le formulaire, et s'il est bien rempli il y appose ses initiales.
- Un second commis traite la demande.
- Si les conditions sont remplies la demande est approuvée et le permis est accordé. Sinon, une lettre de refus est émise.

Visual Paradigm for UML - Standard Edition (Université Laval)



À faire en lien avec ce module

- Lecture des chapitres 28 (version anglaise)
ou 25.1 à 25.4 (version française)
- Références complémentaires (optionnelles) sur les diagrammes d'activités et diagrammes d'état: chapitres 20 et 25 de [Booch 2005] (disponible en version électronique à la bibliothèque)
- Comprendre les concepts suivants:
 - Diagrammes d'activité (action, activité, ...)
 - Relation avec le diagramme d'états
 - Flux des traitements vs flux des données

