R3.04 cours n°3

I.Borne

- Diagramme d'activité
- Couplage et cohésion

Hiérarchie des diagrammes UML

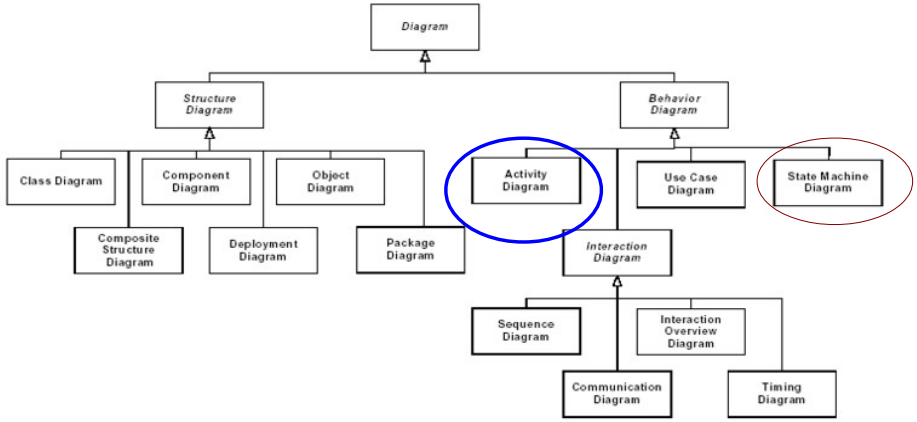


Diagramme d'activités (modèle dynamique)

- Un des 8 diagrammes UML pour la modélisation des aspects dynamiques.
- Il fait partie de la vue fonctionnelle.
- Une technique pour décrire une logique procédurale, un processus métier ou un flot de contrôle.
- Chaque processus décrit une séquence de tâches et les décisions indiquant quand et comment elles sont réalisées.

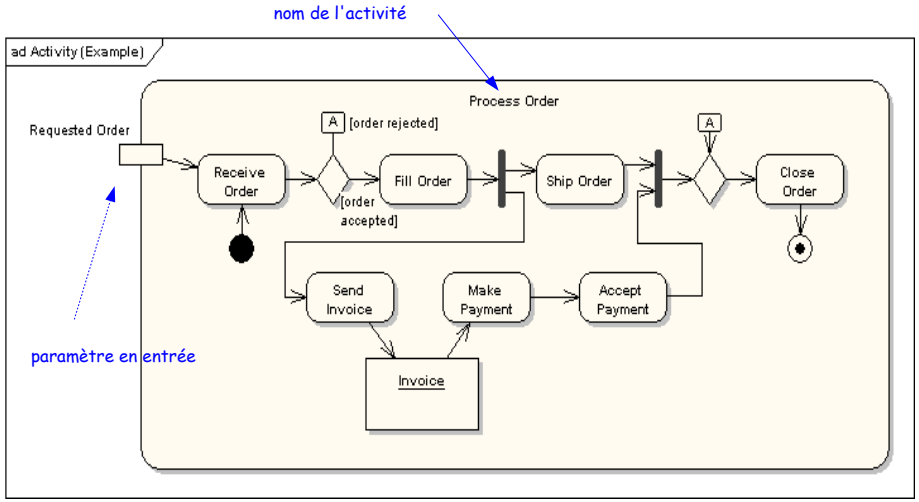
Diagramme d'activités : objectifs

- Modéliser le flot de contrôle d'une opération
- Visualiser, spécifier, construire, documenter la dynamique d'une société d'objets
- Ordonnancement et dépendance des activités
- Comportements parallèles et synchronisation d'activités
- Modéliser les aspects dynamiques d'un système, mais également pour construire des systèmes exécutables .

Diagramme d'activités : quand les utiliser ?

- Pour décrire des comportements qui dépendent des résultats de processus (traitements) internes.
- Dans des situations où presque tous les événements représentent l'achèvement d'actions générées de façon interne.
- Pour définir des opérations, des cas d'utilisation, des *workflows* qui relient des séries de cas d'utilisation.

Un premier exemple de diagramme d'activité



Modélisation des activités et des actions

Une activité fait référence à une séquence d'actions.

Une activité est constituée d'un ensemble de noeuds et de connecteurs.

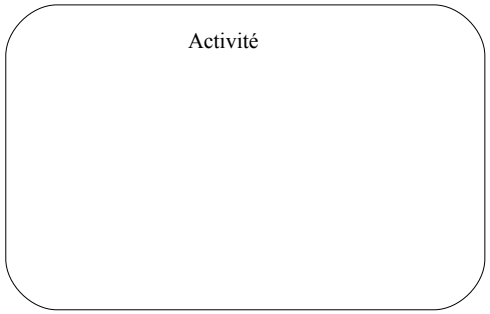
Les noeuds sont appelés des actions.

Un noeud d'activité peut être :

- Un noeud exécutable ou plus spécifiquement une action
- Un noeud de contrôle
- Un noeud d'objet (une instance d'une classe)

Les activités

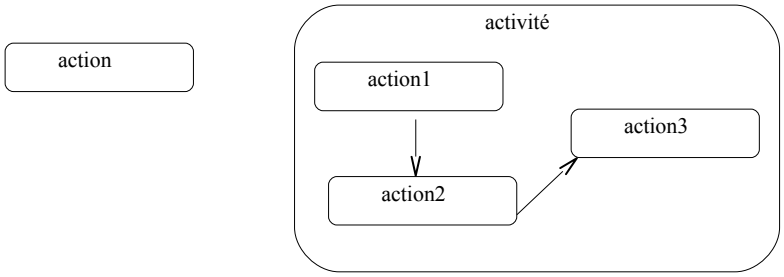
Une activité se présente dans un rectangle arrondi contenant toutes les actions, flots de contrôle et autres éléments



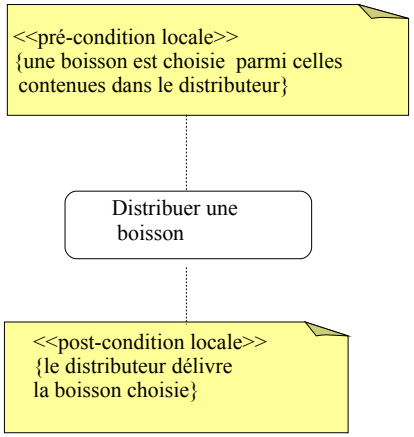
Les actions

Une action représente une sous-activité ou une méthode sur une classe.

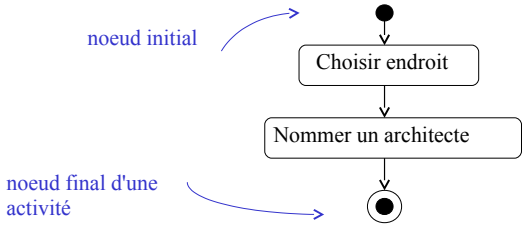
Elles sont généralement une unité atomique de comportement exécutée dans le contexte d'une activité.



Contraintes attachées à une action



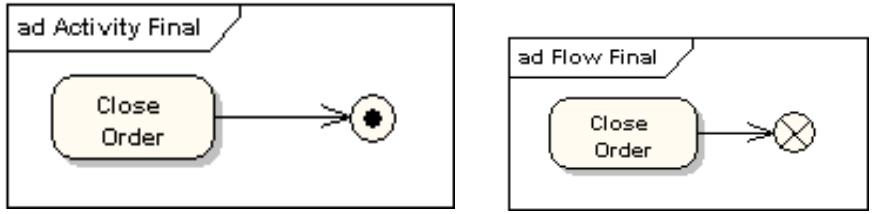
Noeud initial et noeud final



Noeud final d'activité et noeud final de flot

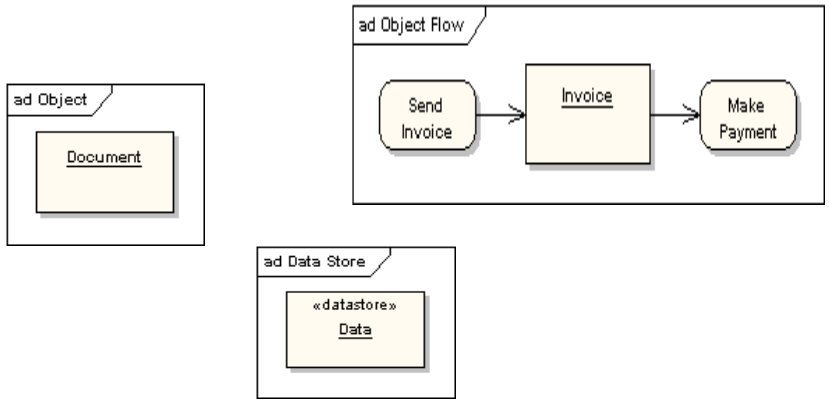
Le noeud final de flot dénote la fin d'un flot de contrôle

Le noeud final d'activité dénote la fin de tous les flots de contrôle de l'activité

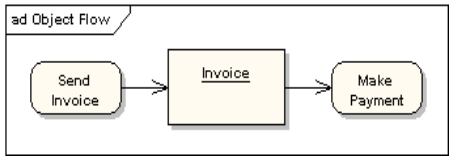


Objet et flot d'objets

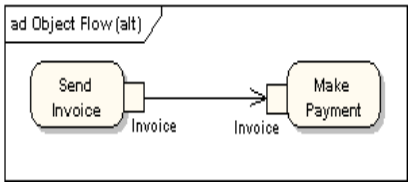
Un flot d'objets est un chemin avec des objets ou des données
Il est dénoté comme un connecteur avec une flèche montrant la direction dans laquelle l'objet est passé



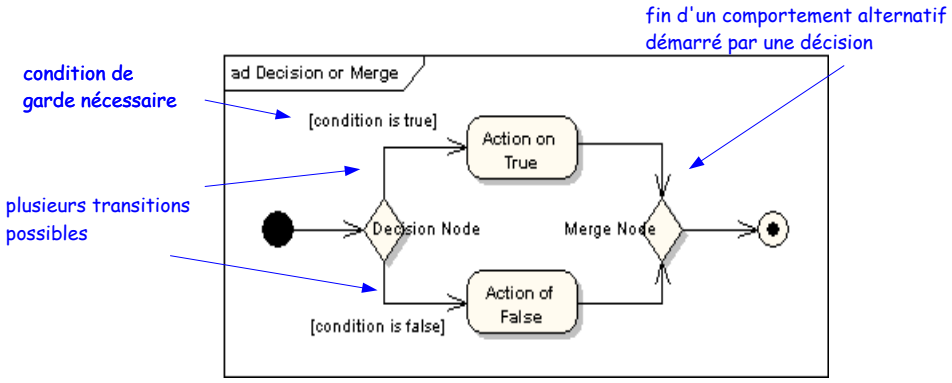
Flot d'objets



Notation abrégée pour les flots d'objets

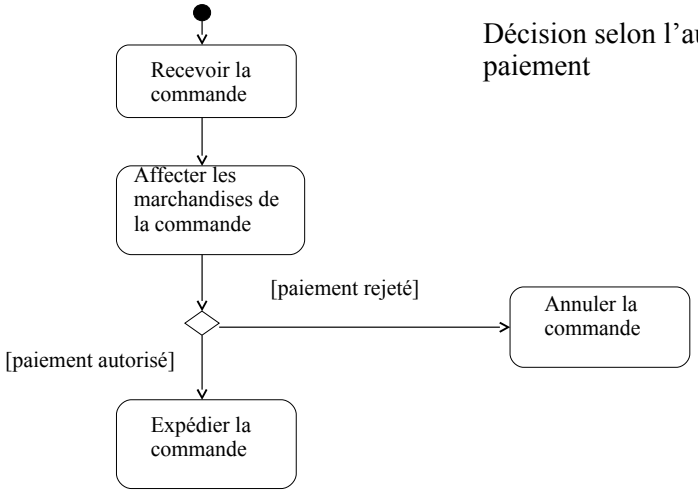


Noeud de décision et noeud de fusion



Exemple de réception de commande

Décision selon l'autorisation de paiement



Synchronisation de flots concurrents

La représentation des synchronisations entre les flots de contrôle se fait au moyen de barres de synchronisation :

division et regroupement de flots de contrôle parallèles

Une barre de synchronisation permet d'ouvrir (*fork*) ou de fermer (*join*) des branches parallèles.

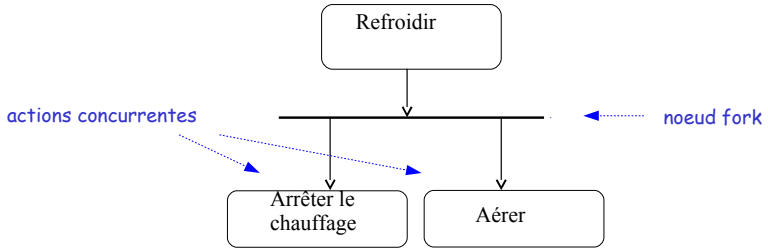
Les transitions au départ d'un *fork* sont déclenchées simultanément.

Noeud fork

Division d'un flot de contrôle :

- Pour refroidir il faut simultanément arrêter le chauffage et ouvrir les fenêtres.

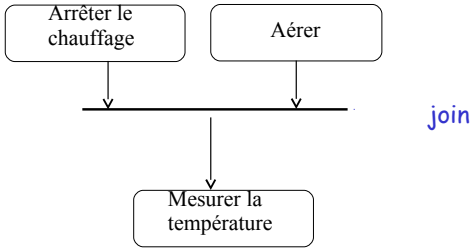
Sous la fourche, les activités continuent en parallèle.



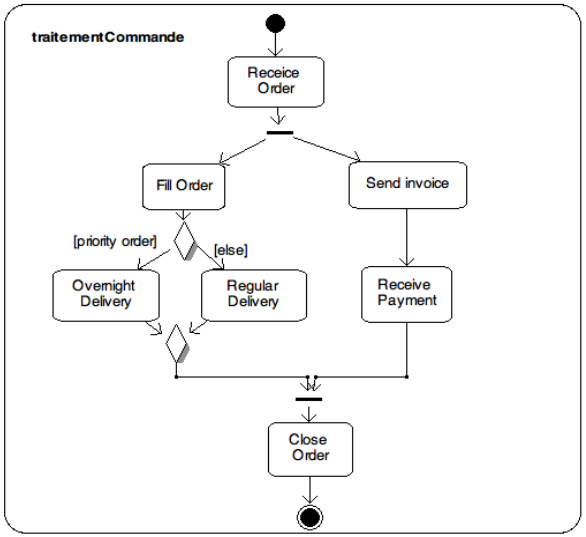
Noeud join

Une barre de synchronisation ne pourra être franchie que lorsque toutes les transitions en entrée auront été déclenchées.

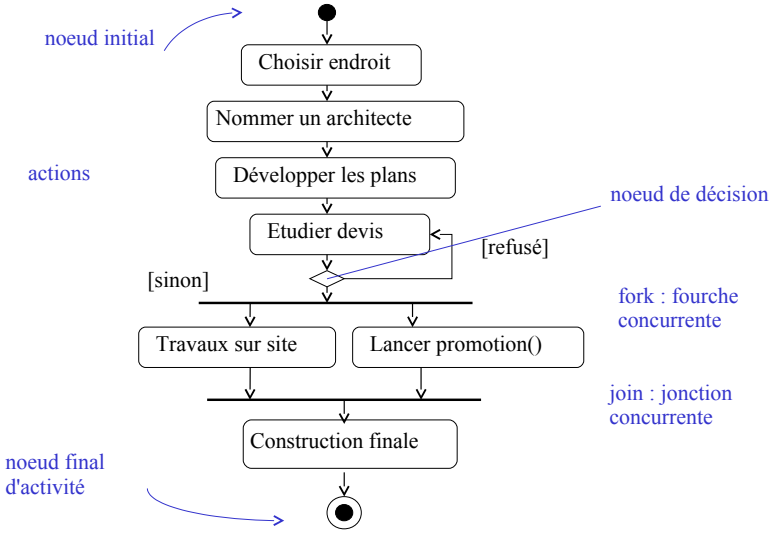
La mesure de la température est effectuée une fois que le chauffage est arrêté et la pièce aérée.



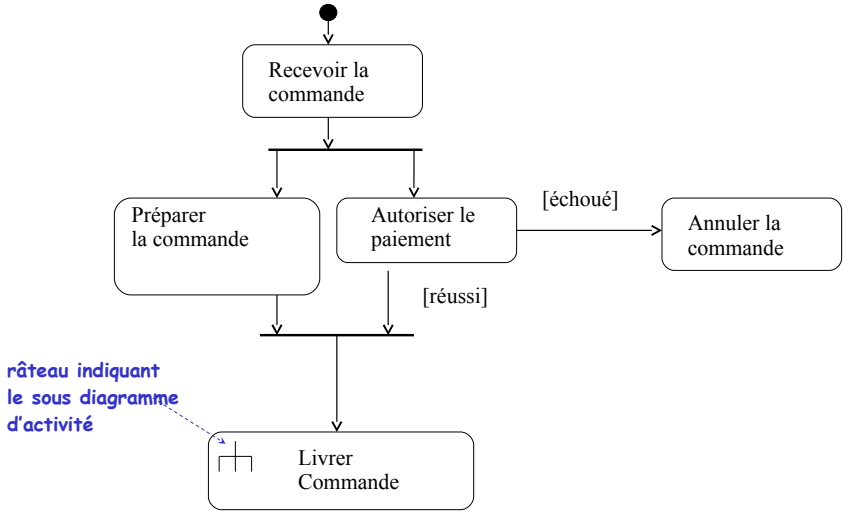
Exemple de traitement d'une commande



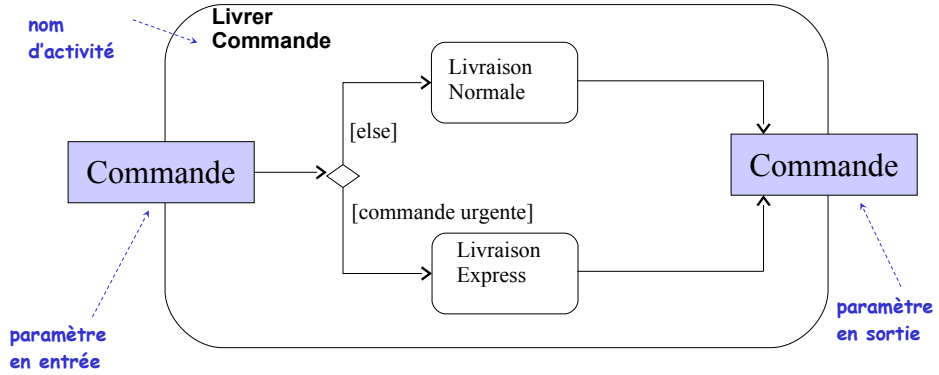
Un autre exemple proche d'un graphique Pert



Invocation de l'activité

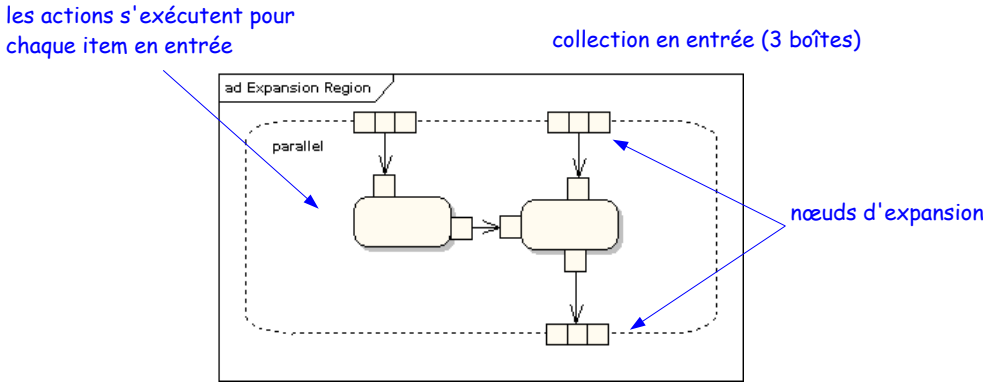


Détails de l'activité « livrer commande »

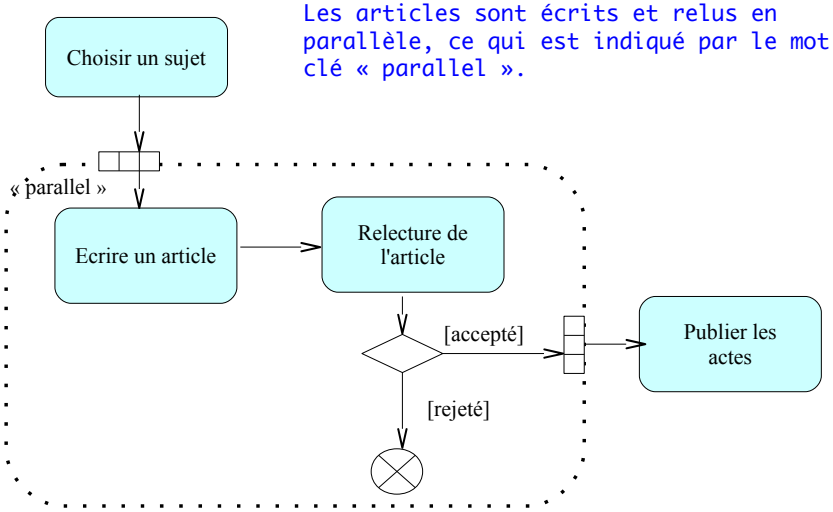


Région d'expansion

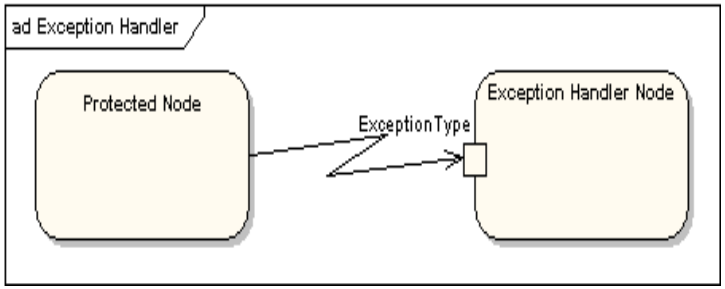
Les mots clés *iterative*, *parallel* ou *stream* sont à indiquer en haut à gauche



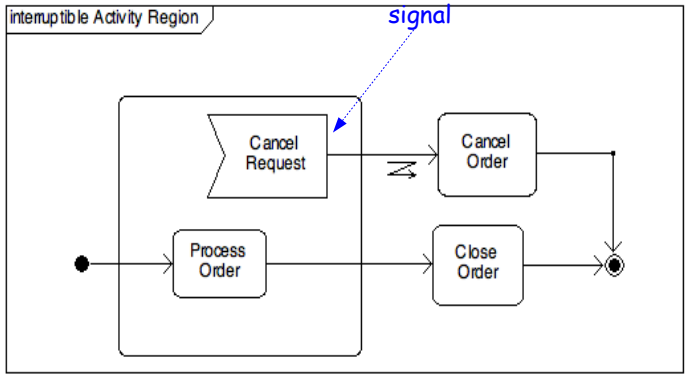
Exemple



Gestion des exceptions



Région d'activité interruptible



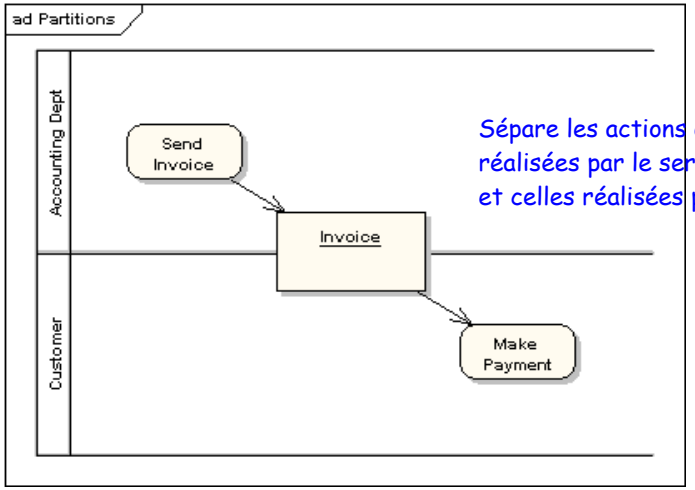
Couloir d'activités ou partition d'activité (*swimlane*)

Les diagrammes d'activités peuvent être découpés en partitions d'activités afin de montrer quelles actions sont exécutées par une classe, ou une entité organisationnelle :

- chaque partition représente la responsabilité pour une partie ou la totalité de l'activité qui peut être assurée par un ou plusieurs objets
- chaque activité est affectée à une partition donnée.

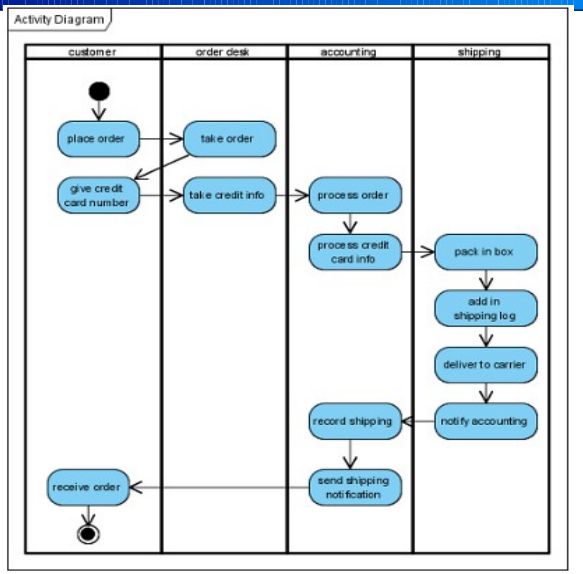
La position relative des partitions n'est pas significative.

Notation des partitions

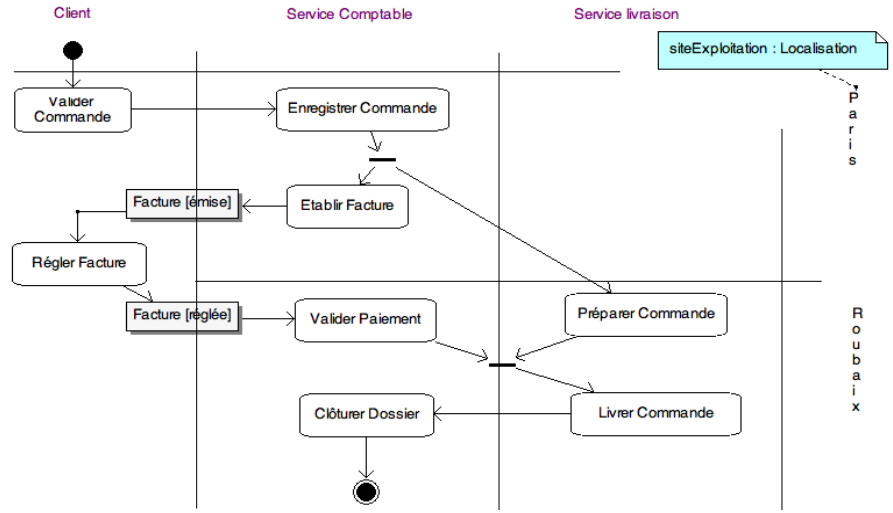


Sépare les actions entre celles réalisées par le service de comptabilité et celles réalisées par le client

Exemple de partitions (VP-UML)



Exemple de partitions multidimensionnelles



Partition explicite

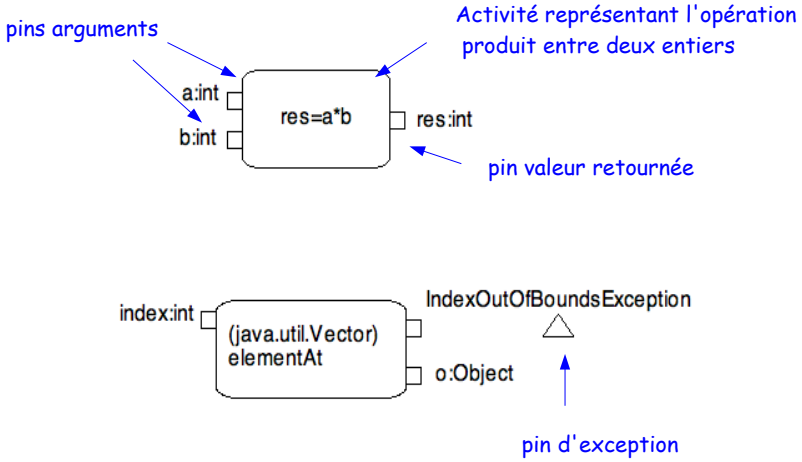
↳ On indique la partition dans les actions.

« external »
(Client)
Régler Facture

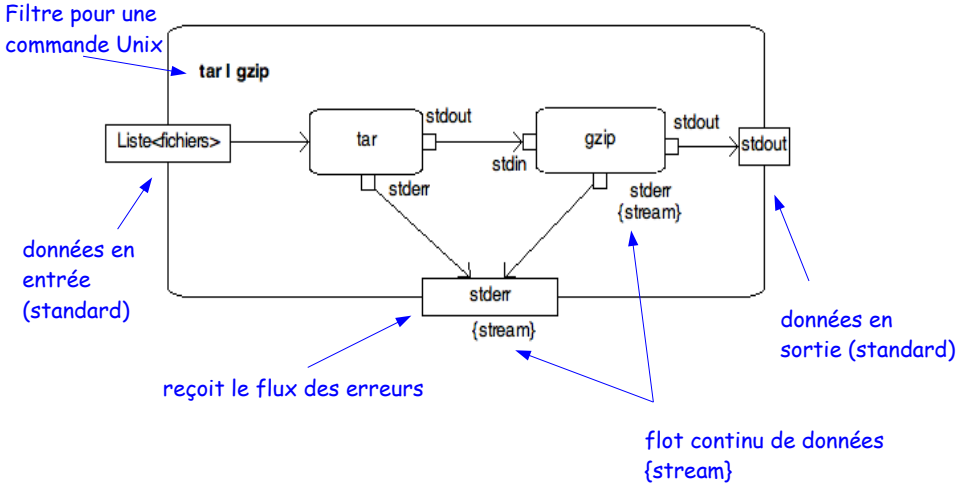
(Service Comptable)
Etablir Facture

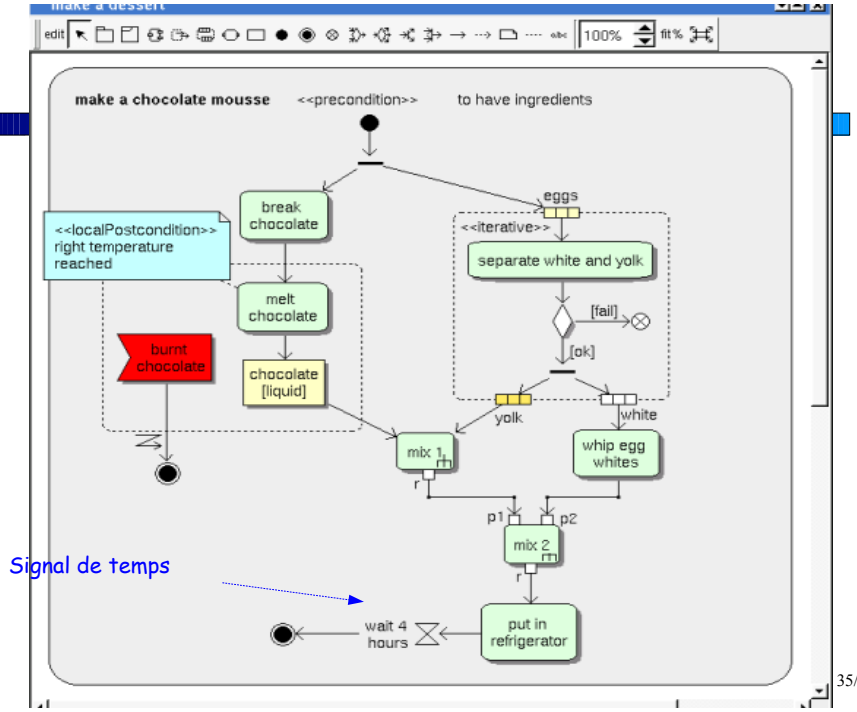
(Service Livraison)
Livrer Commande

Les flots de données : argument et valeur retournée



Utilisation de pin : flot de données





Résumé

Pour modéliser les aspects dynamiques d'un système, d'une classe ou d'un cas d'utilisation, on utilise les diagrammes d'états-transitions d'une seule et unique manière : *pour modéliser les objets réactifs*

La grande force des diagrammes d'activités réside dans le fait de supporter les comportements en parallèle.

C'est un bon outil pour décrire des flots de tâches et modéliser des traitements.

Les diagrammes d'activités conviennent mieux à la modélisation du flot d'activités dans le temps tel qu'on peut le représenter dans un organigramme.

A méditer

Pour voir comment les objets collaborent, j'utilise ....

Pour voir comment un objet se comporte pendant son cycle de vie, j'utilise ...

Pour représenter une logique conditionnelle complexe, j'utilise ...

Couplage et cohésion

Couplage

- C'est la mesure du degré d'interdépendance entre des entités
- Deux entités sont couplées si une modification d'une de ces entités demande une modification dans l'autre
- Les interactions entre deux objets se produisent parce qu'il y a un couplage

- Les programmes faiblement couplés offrent une grande flexibilité et extensibilité.
- Un couplage fort conduit toujours à des programmes avec une flexibilité faible et une évolutivité / extensibilité moindre.
- Cependant, dans des langages de programmation tels que Java, il est impossible d'éviter tout couplage.
- Recommandation : réduire le plus possible le couplage

Les différents types de couplage

niveau	Type de couplage	Description
bon	pas de couplage direct	Les méthodes ne sont pas reliées, pas d'appel entre elles
	de données	Les composants sont indépendants et communiquent par le biais de données
	Tampon (Stamp)	Un objet est transmis d'une méthode à un autre mais seul une portion de l'objet est utilisé dans le traitement
	de contrôle	La méthode appelante passe une variable de contrôle dont la valeur contrôlera l'exécution de la méthode appelée
	externe	Les classes dépendant d'autre classes externes
	commun ou global	Les classes ont des données partagées telles que des structures de données globales
mauvais	de contenu ou pathologique	Une méthode d'un objet fait référence au contenu d'un autre objet. Violation du principe d'encapsulation

Cohésion

- Mesure la relation entre chacune des fonctions dans un programme
- Concerne un composant logiciel individuel.

- Un composant a une seule responsabilité, donc une seule raison de changer : cohésion forte
- Un composant a beaucoup de responsabilité , donc beaucoup de raisons de changer : cohésion faible

- Cohésion de méthode, classe, généralisation/spécialisation
- Mesure la compréhensibilité des classes.

- Recommandation : forte cohésion

Cohésion des méthodes

- Une méthode cohésive est responsable d'une tâche précise.
- Chaque méthode doit être raisonnablement courte et facile à comprendre avec un nom qui précise clairement son rôle.
- Plusieurs types de cohésion ont été identifiés :
  - Fonctionnel
  - Séquentiel
  - ….

  (voir tableau suivant)

Types de cohésion (méthodes)

niveau	type de cohésion	description
bon	fonctionnelle	Un seul calcul dans chaque méthode
↓	séquentielle	Des données générées d'une méthode deviennent l'entrée d'une autre
↓	communicationnelle	Deux méthodes fonctionnent sur les même données d'entrée ou contribuent aux mêmes données de sortie
↓	procédurale	La méthode contient plusieurs fonctions faiblement reliées
↓	temporelle	Des tâches doivent être exécutées dans le même laps de temps (initialisation des toutes les parties)
↓	logique	La méthode contient plusieurs fonctions liées de manière logique et non fonctionnelle
mauvais	fortuite	La méthode a plusieurs fonctions qui ne sont pas liées

Cohésion des classes

- Une classe cohésive représente une entité précise.
- Niveau de cohésion parmi les attributs et les méthodes d'une classe.
- Il ne doit pas y avoir d'attribut ou de méthode jamais utilisé
- Une classe cohésive :
  - doit contenir plusieurs méthodes visibles à l'extérieur ;
  - chaque méthode visible effectue une seule fonction ;
  - toutes les méthodes font référence seulement aux attributs ou aux autres méthodes définies dans la classe ou les superclasses ;
  - pas de couplage de contrôle entre les méthodes visibles.

Types de cohésion (classes)

niveau	Type de cohésion	description
Bon	idéale	La classe n'a pas de cohésions mixtes
	mixed-role	La classe a des attributs qui relient les objets de la classe à d'autres objets sur la même couche, mais les attributs n'ont rien à voir avec la sémantique sous-jacente de la classe
	mixed-domain	Idem mais dans ce cas, les attributs incriminés appartiennent à une autre classe située sur l'une des autres couches
mauvais	mixed-instance	La classe représente deux types différents d'objets. Elle doit être décomposée en deux classes séparées

Cohésion de généralisation/spécialisation

- Comment les classes sont reliées dans la hiérarchie d'héritage ?
- Sémantique « sorte-de »
- Liaison via une association, agrégation ou liaison de type pour de la réutilisation ?
- Attention aux mauvais usages de l'héritage basés sur des mises en commun abusives.