

Module B9-1 : sensibilisation à l'UML

Session 4 : Vue fonctionnelle

Olivier Habart : habart.olivier@free.fr



Session 3 : Vue fonctionnelle Sommaire

- **Diagramme d'interaction** :
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de timing

- Diagramme d'activités

- Diagramme d'états-transitions

- TD



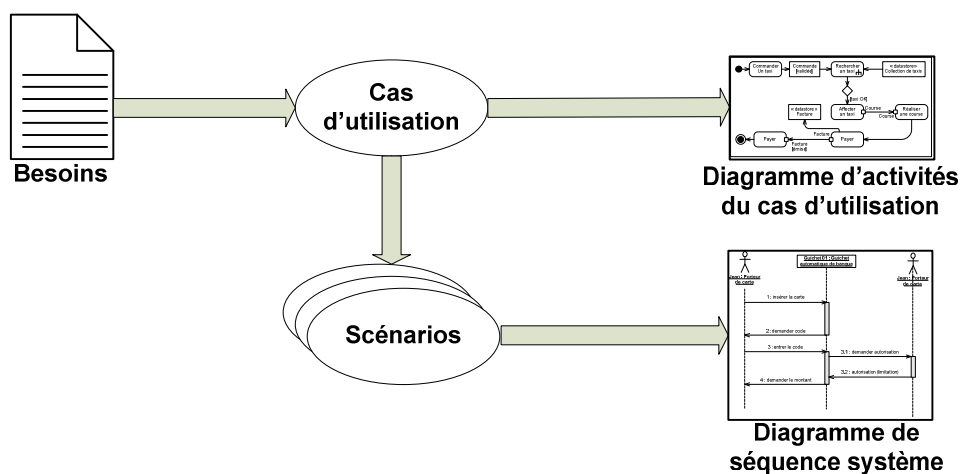


Session 3 : Diagrammes d'interaction *Par qui ?*

- Concepteur :
 - Détailler les cas d'utilisation : représenter la succession chronologique des opérations réalisées par un acteur
 - Identifier les cas de validation
 - Identifier les opérations
 - Distribuer le comportement des cas d'utilisation sur les classes d'analyse
 - Distribuer le comportement sur les sous-systèmes



Session 3 : Diagrammes d'interaction *Par qui ? – Détailler les cas d'utilisation*



Session 3 : Diagrammes d'interaction

Par qui ? – Distribuer le comportement des cas d'utilisation sur les classes d'analyse

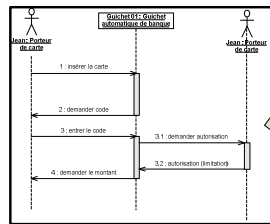
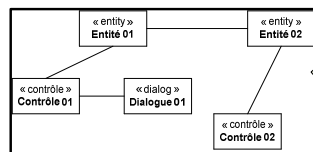
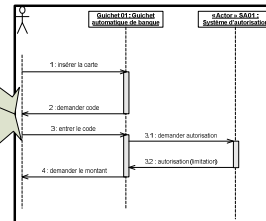


Diagramme de séquence système



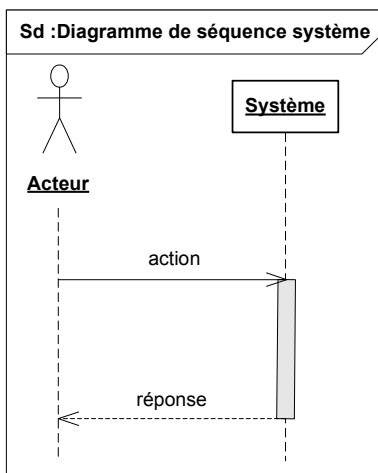
Modèle des classes participantes



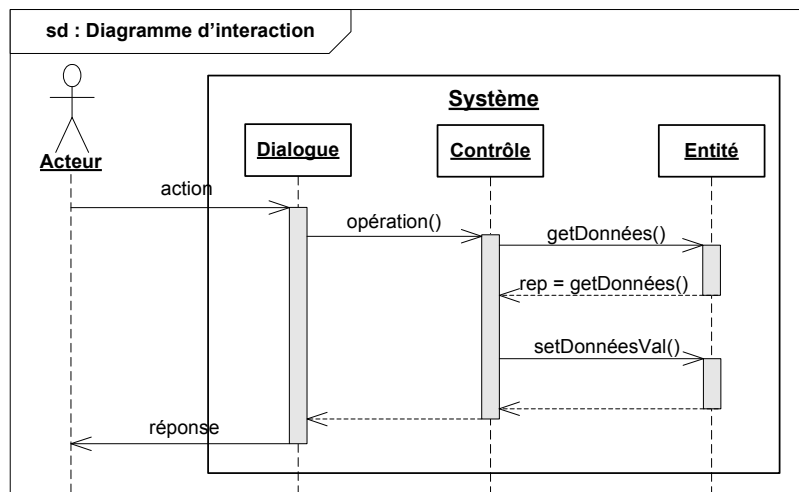
Diagrammes d'interaction

Session 3 : Diagrammes d'interaction

Par qui ? – Distribuer le comportement des cas d'utilisation sur les classes d'analyse



Analyse

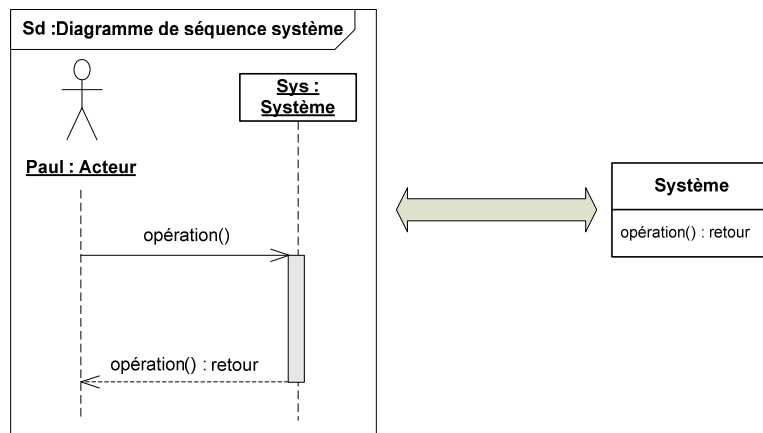


Conception



Session 3 : Diagrammes d'interaction

Par qui ? – Identifier les opérations



Session 3 : Diagrammes d'interaction

Concepts

- Montre le comportement d'un classeur structuré ou d'une collaboration en se focalisant sur l'échange d'informations entre les éléments du classeur ou de la collaboration
- Diagrammes pour représenter une interaction :
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme de timing
 - Diagramme d'ensemble d'interactions
- Concepts communs aux interactions :
 - Ligne de vie : instance d'un classeur
 - Message :
 - ✓ Matérialisation d'une communication avec transmission d'information entre un émetteur (source) et un récepteur (destination)
 - ✓ Un message déclenche une opération, l'émission d'un signal, la création/destruction d'un objet
 - ✓ Un message est synchrone ou asynchrone



Session 3 : Diagrammes d'interaction

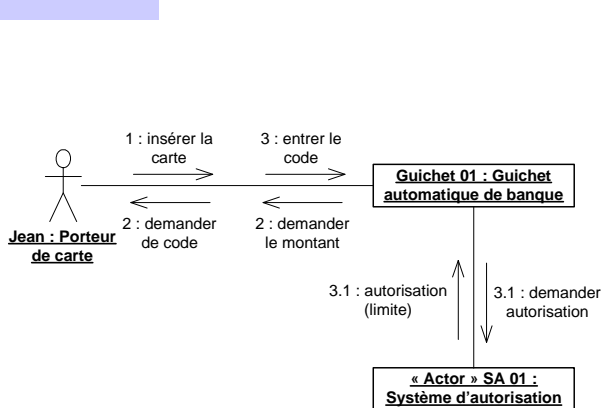
Concepts – Message (syntaxe)

- Syntaxe : « **séquence message (paramètres) : retour** »
- Signification :
 - **séquence** :
 - ✓ numéro de séquence du message
 - ✓ x.y : numéro des messages étant la conséquence de la réception du message de séquence x
 - ✓ L'envoi du message 1.4.4 est postérieur à celui du message 1.4.3
 - **retour** : valeur du retour du message (syntaxe : « **nom du paramètre = valeur** »)
 - **message** : nom du message
 - **paramètres** : valeurs des paramètres du message (syntaxe : « **nom du paramètre = valeur** »)
- Exemples :
 - traiter()
 - traiter(x=3)
 - traiter(3) : 4
 - 1.2.1 traiter(3) : 4

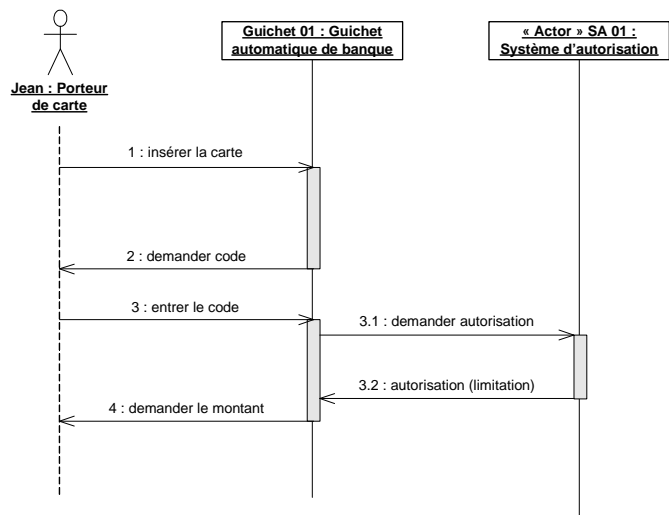


Session 3 : Diagrammes d'interaction

Relation entre les diagrammes



Représentation d'une interaction avec un diagramme de communication



Représentation d'une interaction avec un diagramme de séquence



Session 3 : Vue fonctionnelle

Sommaire

- Diagramme d'interaction :
 - **Diagramme de séquence**
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de timing
- Diagramme d'activités
- Diagramme d'états-transitions
- TD



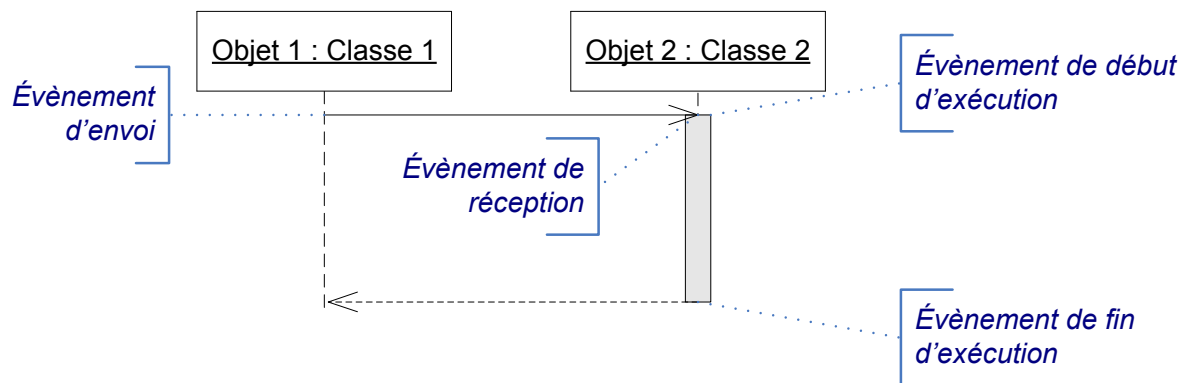
Session 3 : Diagramme de séquence

Concepts – Messages

	<p>Message asynchrone : n'attend pas de réponse et ne bloque pas l'émetteur qui ne sait pas si le message arrivera</p>
	<p>Message synchrone : l'émetteur reste bloqué le temps que dure l'invocation de l'opération</p>
	<p>Message de création : création d'un objet Message de destruction : destruction d'un objet</p>
	<p>Message trouvé : message dont l'évènement d'envoi est inconnu</p>
	<p>Message perdu : message dont l'évènement de réception est inconnu</p>



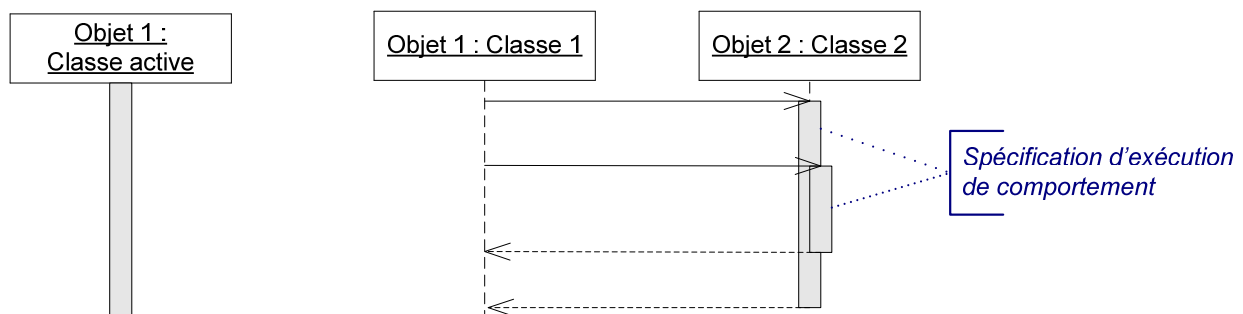
Session 3 : Diagramme de séquence Concepts – Evènements



- Évènement :
 - Peut être utilisé dans les diagrammes d'état-transition afin de caractériser le déclenchement des transitions



Session 3 : Diagramme de séquence Quoi? – Spécification d'exécution de comportement



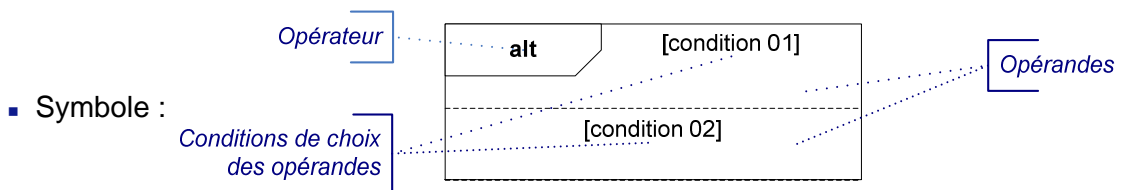
- Spécification d'exécution de comportement :
 - Objet actif : possède une spécification d'exécution de comportement sur toute sa ligne de vie
 - Objet passif : a besoin d'être sollicité pour déclencher une exécution.
 - Message synchrone :
 - ✓ La spécification d'exécution de comportement est bornée par les évènements de réception du message et de retour
 - Les exécutions simultanées sont représentées par un chevauchement des spécifications d'exécution



Session 3 : Diagramme de séquence

Concepts – Fragments d'interaction

- Pourquoi ? :
 - Représenter l'articulation des interactions
 - Permet de représenter les diagrammes de séquence de manière compacte
- Quoi ? :
 - Opérateur :
 - ✓ 13 opérateurs
 - ✓ Conditionne la signification du fragment
 - Opérande :
 - ✓ Portion, d'un fragment d'interaction conditionnée par une garde



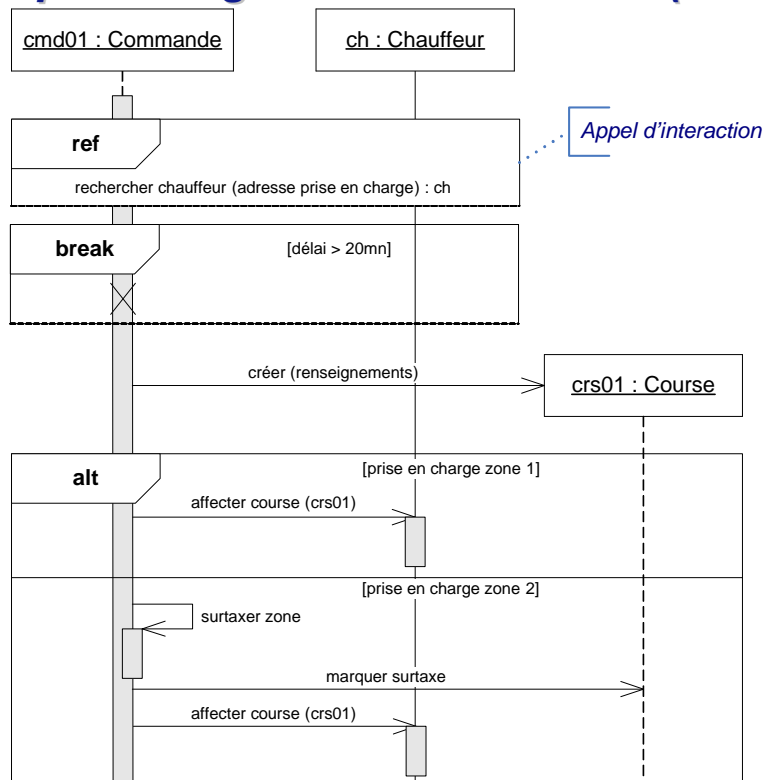
Session 3 : Diagramme de séquence

Concepts – Fragments d'interaction (opérateur)

- Choix de boucles :
 - Alternatif (*alt*) : plusieurs fragments possibles. Seul celui dont la condition est vraie s'exécute
 - Optionnel (*opt*) : ne s'exécute que si la condition est vraie
 - Exception (*break*) : la fin de ce fragment interrompt la séquence entière
 - Itération (*loop*) : le fragment peut s'exécuter plusieurs fois selon les conditions de la garde
- Contrôle de l'envoi de messages en parallèle :
 - Parallèle (*par*) : chaque fragment est exécuté en parallèle
 - Critique (*critical*) : le fragment ne peut avoir qu'un thread qui s'exécute à la fois
- Contrôle de l'envoi de messages :
 - Insignifiant (*ignore*) : les messages du fragment sont considérés comme insignifiants
 - Signifiant (*consider*) : seuls les messages du fragment sont considérés comme signifiants
 - Assertion (*assert*) : seul l'interaction du fragment est considérée comme valide
 - Invalide (*negative*) : le fragment représente une interaction invalide
- Fixe l'ordre d'envoi des messages :
 - Séquencement faible (*seq*) : les sous-fragment s'exécutent dans un ordre quelconque
 - Séquencement fort (*strict*) : les sous-fragments s'exécutent selon l'ordre d'apparition. Valeur par défaut
- Référence :
 - Référence (*ref*) : référencement d'une interaction
 - Diagramme de séquence (*sd*) : référencement d'un diagramme de séquence

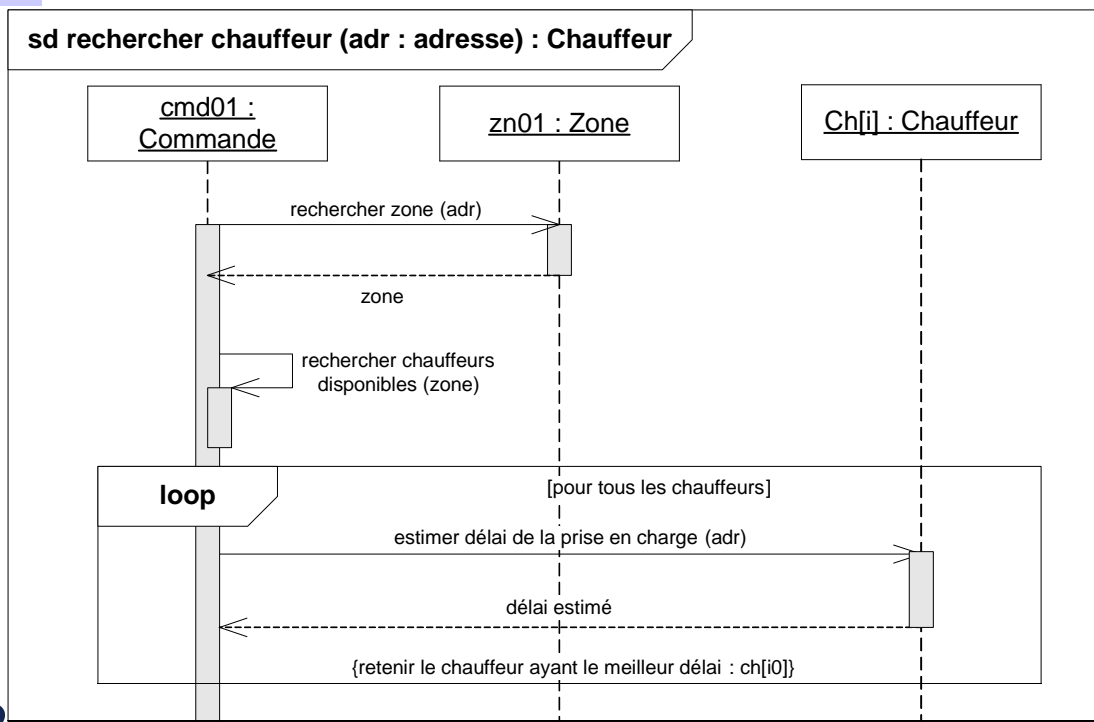
Session 3 : Diagramme de séquence

Concepts – Fragments d'interaction (exemple 1)



Session 3 : Diagramme de séquence

Concepts – Fragments d'interaction (exemple 2)





Session 3 : Diagramme de séquence

Exercices



- Exercice 1 :
 - Représenter la spécification détailler d'un cas d'utilisation
- Exercice 2 :
 - Modélisation des interactions entre classes
- Exercice 3 :
 - Correspondance entre le code JAVA et le diagramme de séquence



Session 3 : Vue fonctionnelle

Sommaire

- Diagramme d'interaction :
 - Diagramme de séquence
 - **Diagramme de communication**
 - Diagramme global d'interaction
 - Diagramme de timing
- Diagramme d'activités
- Diagramme d'états-transitions
- TD

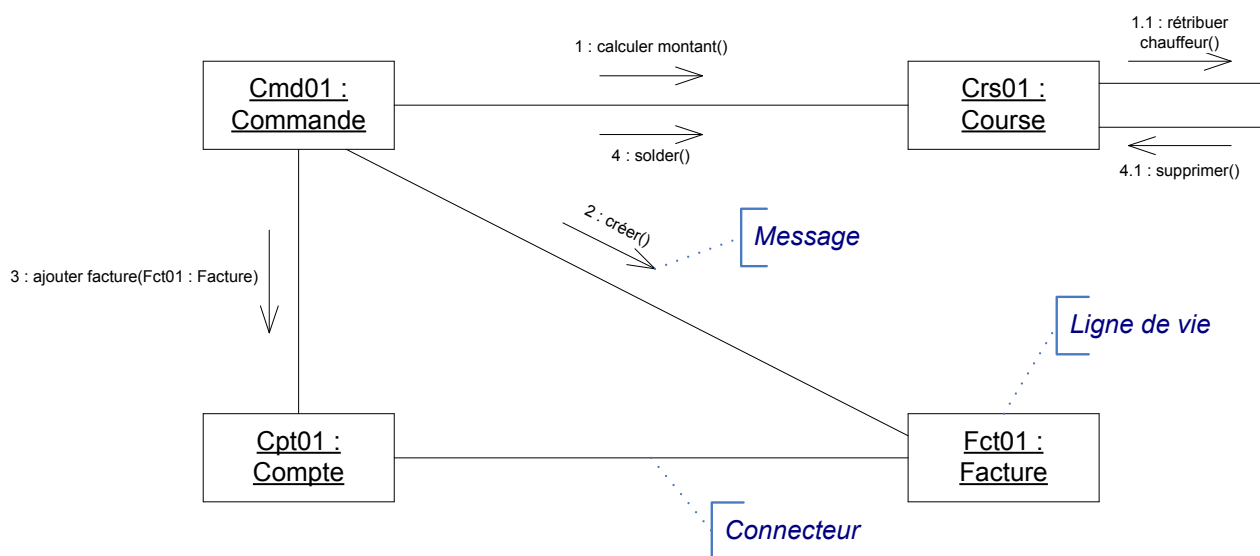


Session 3 : Diagramme de communication

- Pourquoi ? :
 - Concepteur :
 - ✓ Valider les associations du diagramme de classe en les utilisant comme support de transmission des messages
- Quoi ? :
 - Rend compte de l'organisation spatiale des participants à l'interaction
 - Connecteur : relations entre les lignes de vie



Session 3 : Diagramme de communication *Exemple*





Session 3 : Diagramme de communication

Exercices



- Exercice 4 :
 - Traduire un diagramme de séquence en diagramme de communication



Session 3 : Vue fonctionnelle

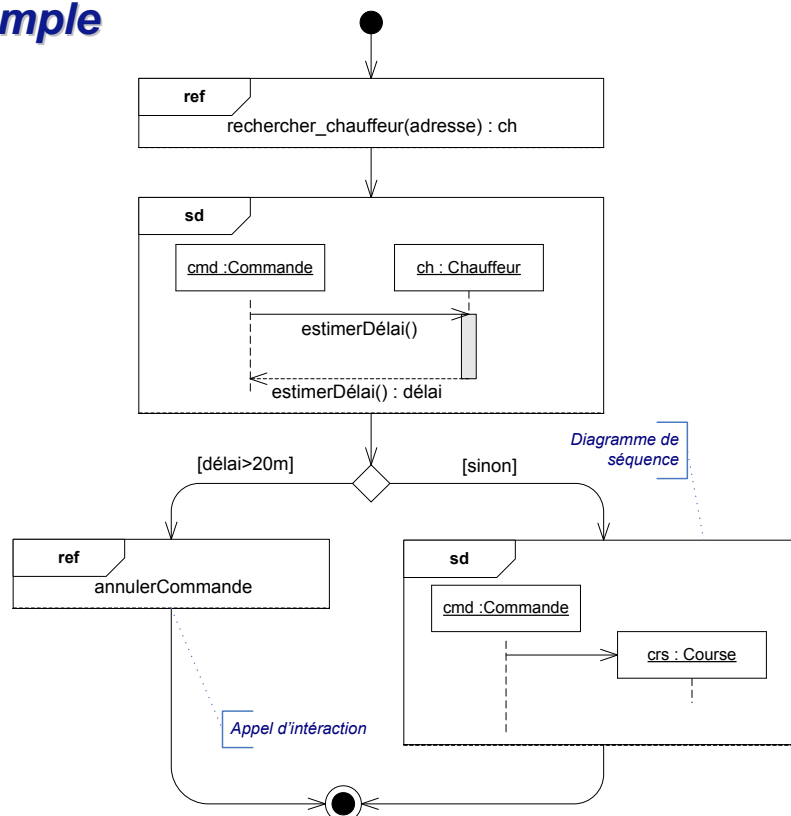
Sommaire

- Diagramme d'interaction :
 - Diagramme de séquence
 - Diagramme de communication
 - **Diagramme global d'interaction**
 - Diagramme de timing
- Diagramme d'activités
- Diagramme d'états-transitions
- TD

Session 2 : Diagramme global d'interaction

- Diagramme d'activité dont les activités sont exprimées par des appels d'interactions
- Nouveau diagramme d'UML 2
- Les flots d'objet ne sont pas représentés
- Permet d'éviter l'utilisation abusive des fragments d'interactions, dont le représentation peut devenir lourde

Session 2 : Diagramme global d'interaction *Exemple*





Session 3 : Vue fonctionnelle Sommaire

- Diagramme d'interaction :
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - **Diagramme de timing**
- Diagramme d'activités
- Diagramme d'états
- TD

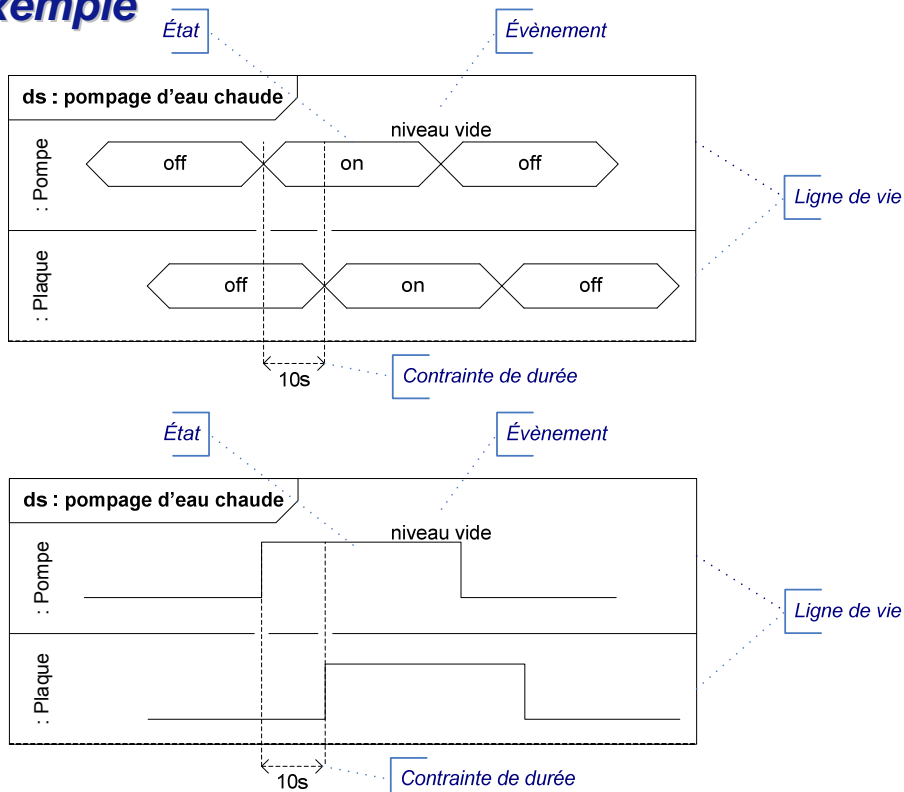


Session 3 : Diagramme de timing

- Pourquoi ? :
 - Illustrer les changements d'état d'un objet dans le temps suite à réponse à des évènements ou stimuli
- Quoi ? :
 - Nouveau diagramme d'UML 2.0
 - Concepts représentés :
 - ✓ Ligne de vie
 - ✓ Évènement
 - ✓ État
 - ✓ Contrainte de durée

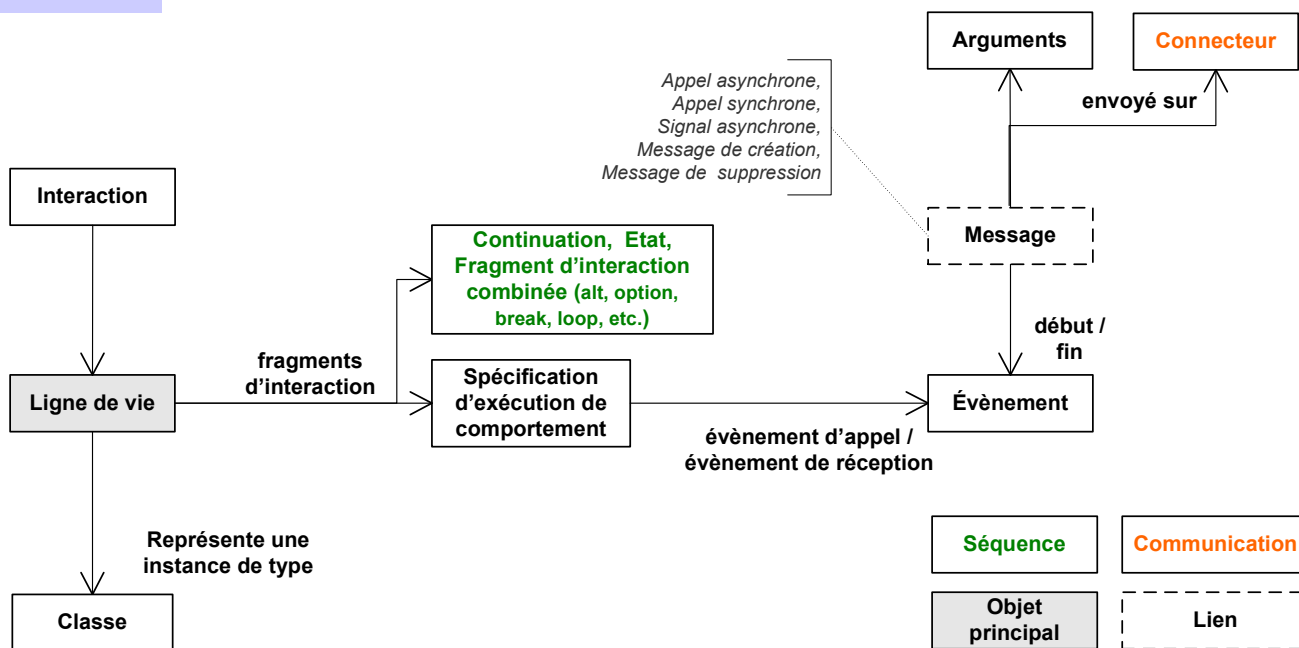
Session : Diagramme de timing

Exemple



Session 3 : Diagrammes d'interaction

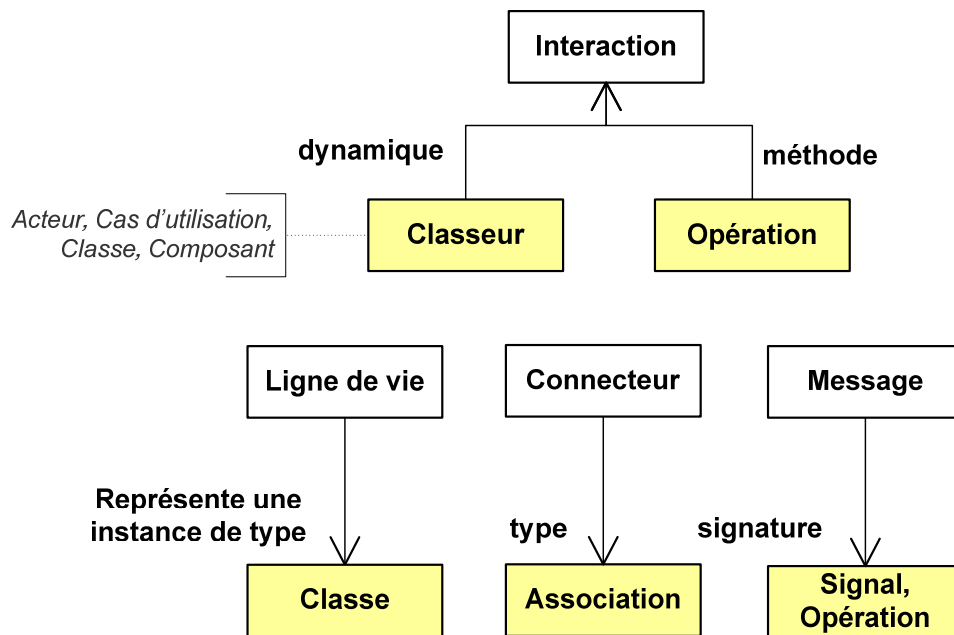
Synthèse des concepts manipulés





Session 3 : Diagrammes d'interaction

Synthèse des concepts externes



Session 3 : Vue fonctionnelle

Sommaire

- Diagramme d'interaction :
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de timing
- **Diagramme d'activités**
- Diagramme d'états-transitions
- TD



Session 3 : Diagramme d'activités

Par qui ?

- Concepteur :
 - Détailler les cas d'utilisation : représenter la succession chronologique des opérations réalisées par un acteur
 - Représenter les activités de navigation



Session 3 : Diagramme d'activités

Quoi? – Activité

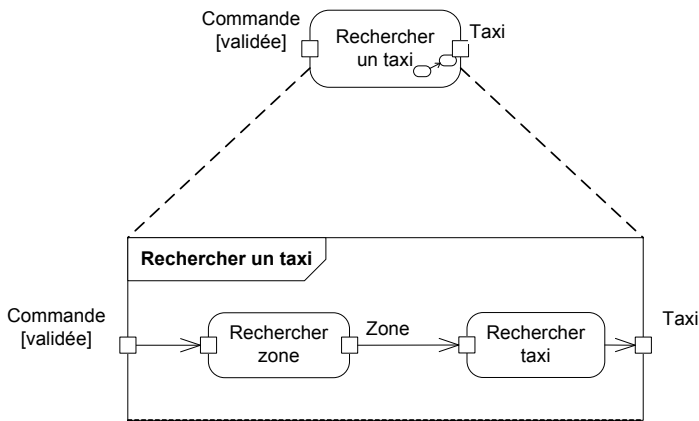
- Pourquoi ? :
 - Représenter l'enchaînement des activités qui concourent au processus
- Quoi ? :
 - Diagramme/Modèle composé de plusieurs actions.
 - Un activité peut avoir des paramètres d'entrées et de sorties qui peuvent correspondre à des interruptions et peuvent supporter les exceptions.
 - Jetons :
 - ✓ Les activités communiquent par des jetons offerts par des actions lorsque leur exécution est complétées, et utilisés en entrées d'autres actions
 - ✓ Les jetons sont échangées par des flots , ils peuvent former une file d'attente en entrée et sortie, et être sauvegardés
 - ✓ Deux types de flots :
 - Flot d'objet : échange d'objet d'action en action ;
 - Flot de contrôle : relaie les messages de contrôle.
 - Nœuds d'activité :
 - ✓ Nœud de contrôle :
 - ✓ Nœud d'objet : représente l'existence d'un objet généré par une action et utilisé par d'autres actions



Session 3 : Diagramme d'activités

Concepts – Paramètre d'activités

- Pourquoi ? :
 - Passer un objet en paramètre d'une activité
- Quoi ? :
 - **Parent** : Nœud d'objet



décomposition d'une activité



flot d'interruption



Session 3 : Diagramme d'activités

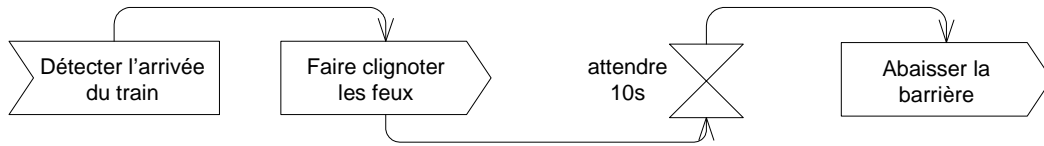
Concepts – Action

- Définition :
 - Plus petit traitement qui puisse être exprimé en UML
 - Étapes à partir desquelles se construisent les comportements des activités
- Types (principaux):
 - Action d'invocation (*invocation action*) :
 - ✓ Action d'envoi d'objet (*send object action*) :
 - ✓ Action d'envoi de signal (*send signal action*) :
 - ✓ Action d'appel (*call action*) :
 - Action d'appel de comportement (*call behavior action*) : invocation directe d'une activité
 - Action d'appel d'opération (*call operation action*) : invocation synchrone ou asynchrone d'une opération sur un objet
 - Répondre (*reply*) : transmission d'un message en réponse d'une action *accept call*
 - ✓ Action d'acceptation d'évènement (*accept event action*) : blocage de l'exécution jusqu'à réception d'un évènement :
 - Évènement d'appel (*accept call action*) : variante d'action d'acceptation pour les évènements d'appel synchrone
 - Évènement temporel (*time event*) : variante d'action d'acceptation pour les évènements temporels
 - Évènement de signal (*signal event*) : variante d'action d'acceptation pour les évènements de signal
 - Actions d'objets :
 - ✓ Créer (*create*) : instanciation d'un objet
 - ✓ Détruire (*destroy*) : destruction d'un objet
 - Actions d'exception :
 - ✓ Lever un exception (*raise exception*)



Session 3 : Diagramme d'activités

Concepts – Action (nœud d'action)



	Action quelconque
	Action d'acceptation d'évènement
	Action d'acceptation d'un évènement temporel
	Action d'envoi d'un signal



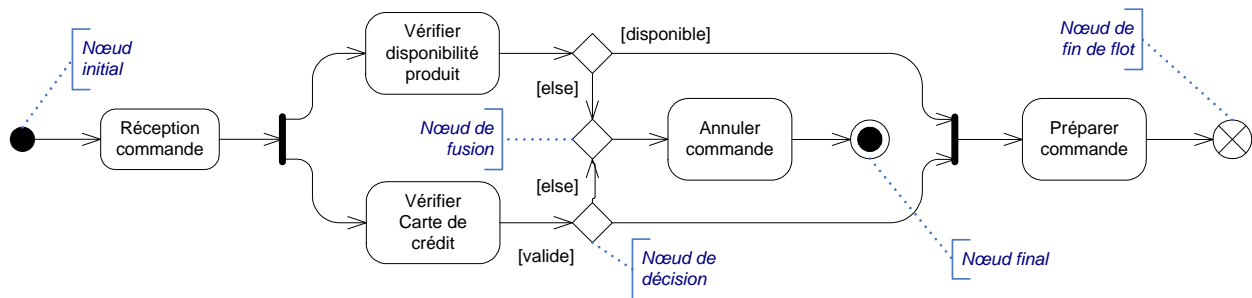
Session 3 : Diagramme d'activités

Concepts – Nœud de contrôle (1)

	Nœud initial : nœud de contrôle à partir duquel débute le flot d'activité
	Nœud de fin d'activité : lorsqu'un flot d'exécution atteint de nœud de fin d'activité, l'exécution de l'activité enveloppante s'achève
	Nœud de fin de flot : fin du flot qui atteint le nœud.
	nœud de décision : fait le choix entre plusieurs flots sortants
	nœud de fusion : rassemble plusieurs flots alternatifs



Session 3 : Diagramme d'activités Concepts – Nœud de contrôle (2)



	Nœud de bifurcation : sépare un flot en plusieurs flots concurrents
	Nœud d'union : synchronise des flots multiples



Session 3 : Diagramme d'activités Concepts – Nœud d'objet (1)

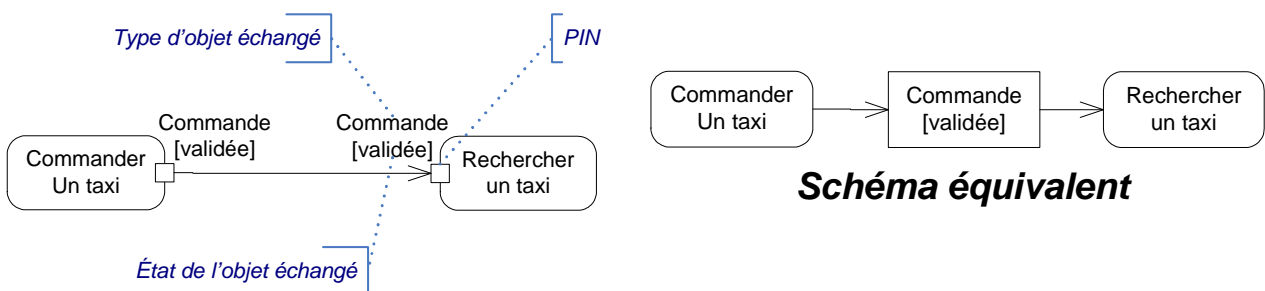
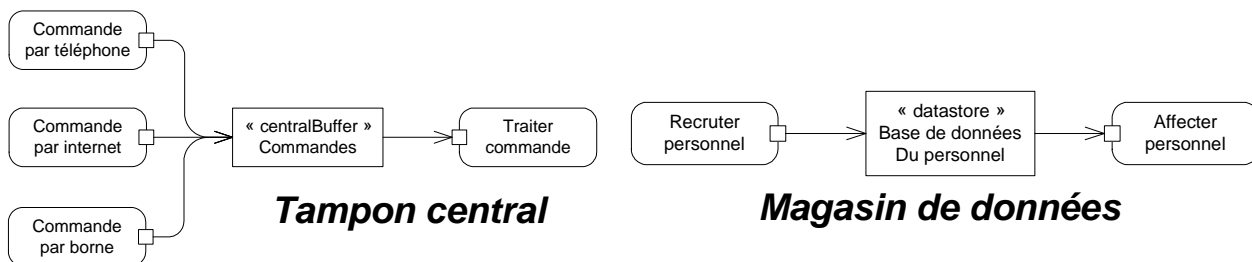


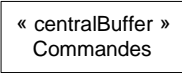
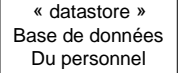
Schéma équivalent

	PIN : permet de spécifier les paramètres d'entrée et de sortie des actions
	Flot d'objet : permet de passer des données d'une action à une autre :

Session 3 : Diagramme d'activités

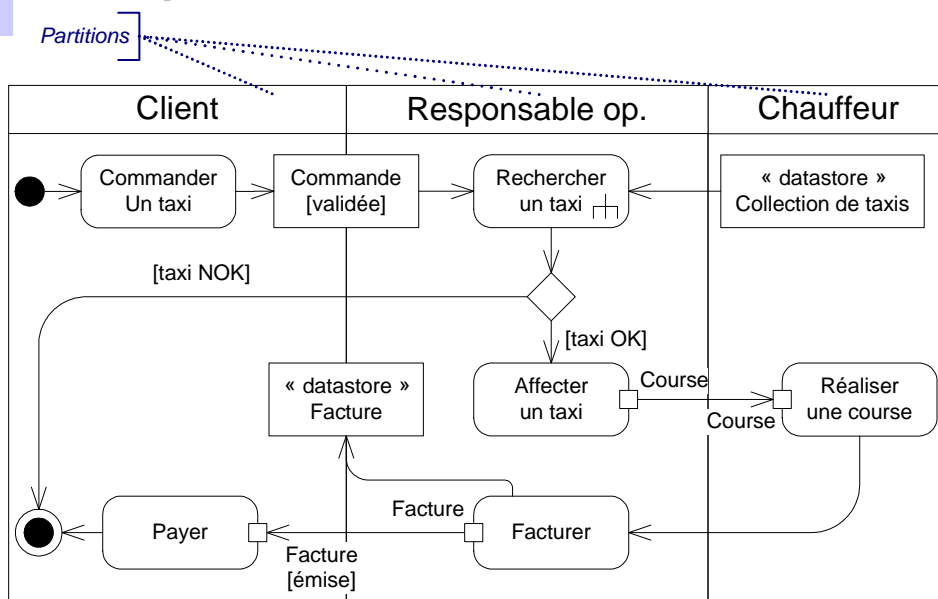
Concepts – Nœud d'objet (2)



	<p>Tampon central : nœud d'objet qui accepte les entrées de plusieurs nœud d'objet ou produit des sorties vers plusieurs nœuds d'objet. Modélise un tampon traditionnel qui peut contenir des valeurs en provenance de diverses sources et livrer des valeurs vers différentes destinations</p>
	<p>Magasin de données : tampon central particulier qui assure la persistance des données.</p>

Session 3 : Diagramme d'activités

Concepts – Partition



Permet d'allouer une collection d'activité à un élément du modèle (classe, composant, etc.)



Session 3 : Diagramme d'activités

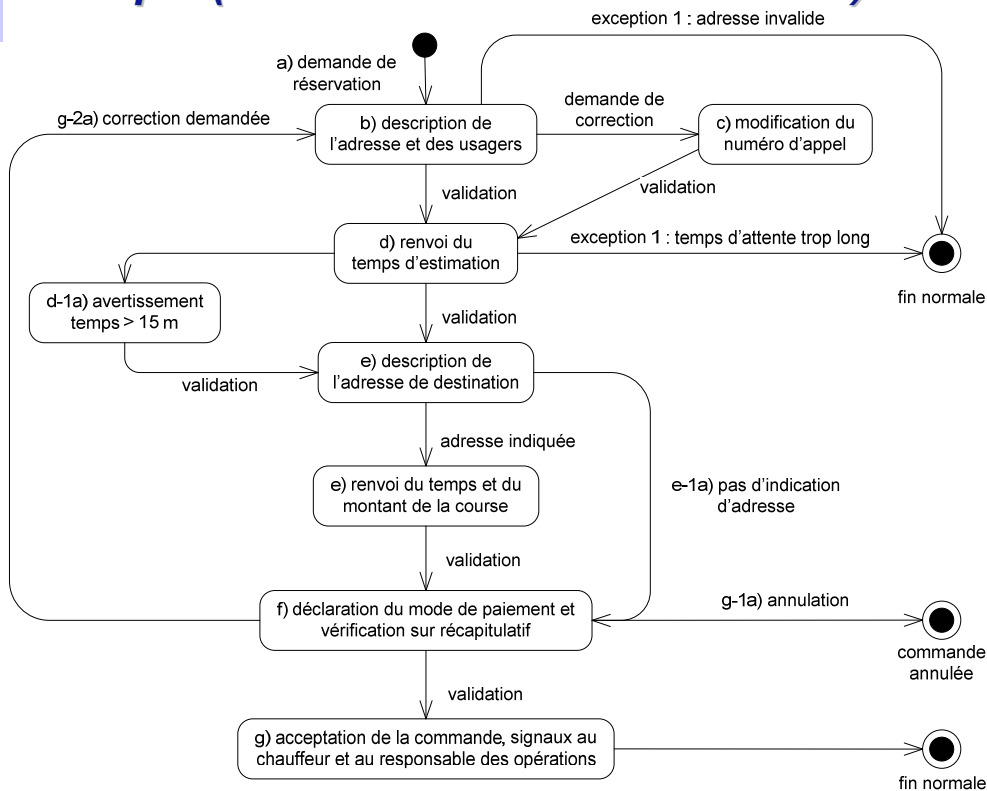
Exemple (traduction de cas d'utilisation)

- Scénario nominal :
 - Le client demande une réservation de taxi
 - Le client entre l'adresse de prise en charge [exception 1 : l'adresse n'est pas valide]
 - Le client modifie, s'il le désire le numéro téléphonique
 - Le client choisit de poursuivre sa commande, et le système lui renvoie une confirmation [exception 2 : temps d'attente trop long]
 - Le client entre son adresse de destination et le système lui renvoie une estimation de devis
 - Le client déclare soit payer par facture, soit faire payer les usagers. Le système lui propose un récapitulatif
 - Le client valide, le système alarme le chauffeur concerné et le responsable des opérations
- Scénarios alternatifs :
 - d-1a) si le temps d'attente dépasse 15m, un message prévient le client
 - d-1b) si le client décide de poursuivre, retour en e)
 - e-1a) si le client ne désire pas connaître le montant et le temps de la course, et peut directement passer à f)
 - g-1a) si le client refuse, il annule l'exécution du cas d'utilisation
 - g-2a) si le client désire modifier une information, il peut revenir à 2)
- Exceptions :
 - exception 1) l'adresse de prise en charge n'est pas valide
 - exception 2) le temps d'attente dépasse 45 min



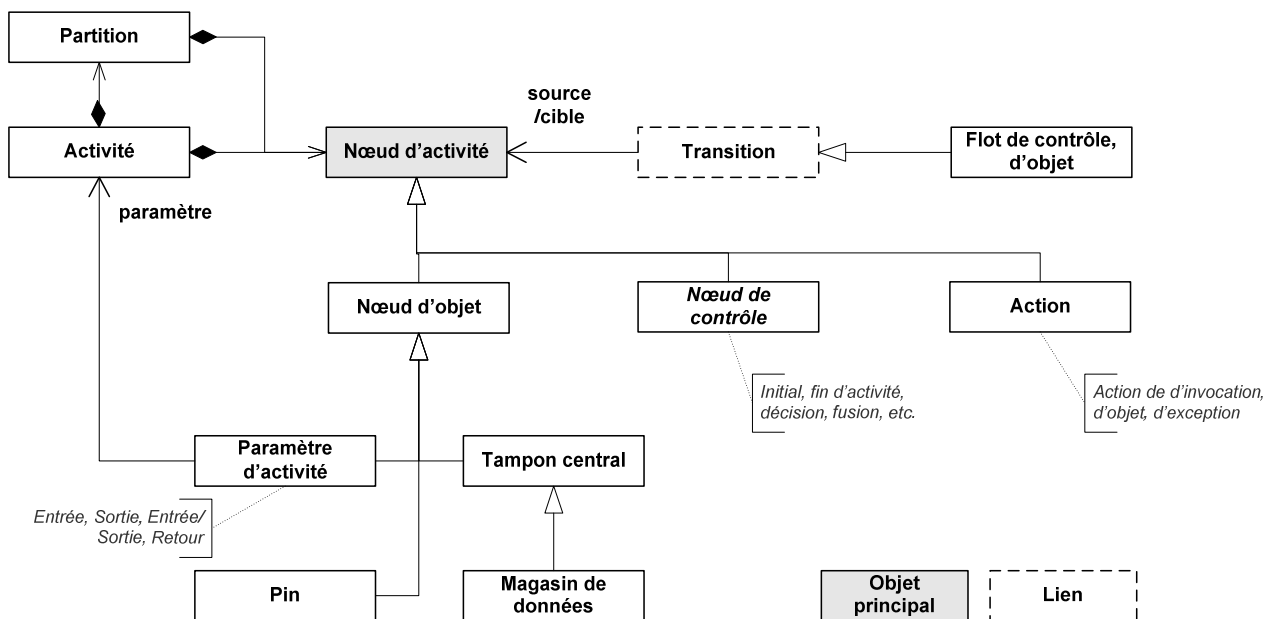
Session 3 : Diagramme d'activités

Exemple (traduction de cas d'utilisation)

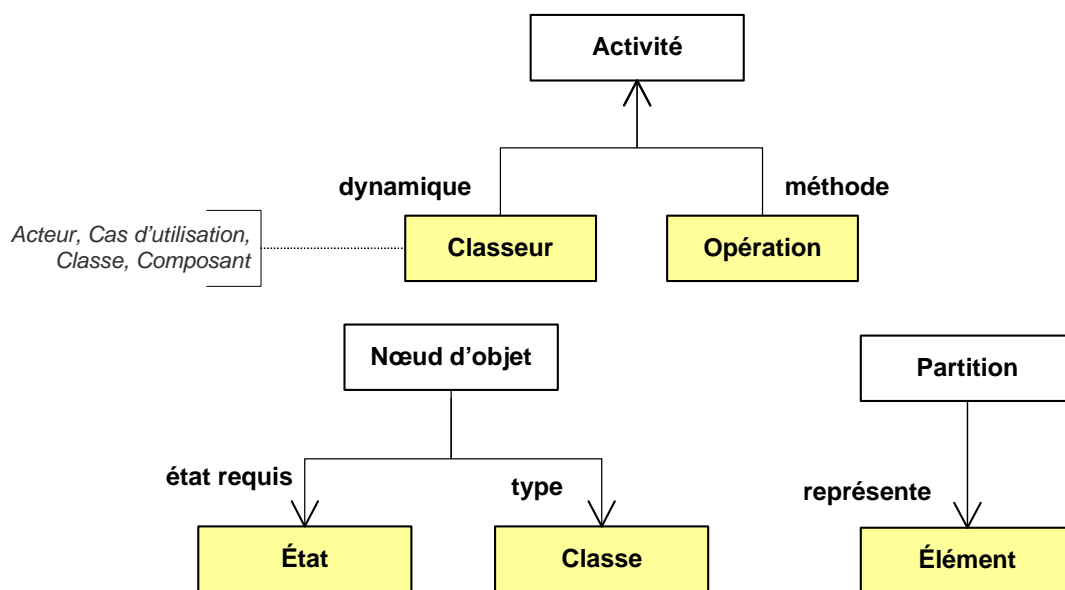




Session 3 : Diagramme d'activités Synthèse des concepts manipulés



Session 3 : Diagramme d'activités Synthèse des concepts externes





Session 3 : Diagramme d'activités

Exercices



- Exercice 5 :
 - Traduire un cas d'utilisation en diagramme d'activités
- Exercice 6 :
 - Représenter un diagramme d'activités avec couloirs



Session 3 : Vue fonctionnelle

Sommaire

- Diagramme d'interaction :
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de timing
- Diagramme d'activités
- **Diagramme d'états-transitions**
- TD



Session 3 : Diagramme d'états-transitions

Par qui ?

- Concepteur :
 - Définir le comportement d'une opération
 - Décrire les classes actives



Session 3 : Diagramme d'états-transitions


Pourquoi ?

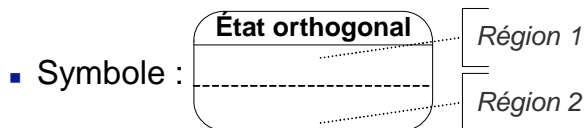
- Pourquoi ? :
 - Représenter la façon dont évoluent (cycle de vie) les objets appartenant à une même classe
- Quoi ? :
 - Présente les séquences possibles d'états et d'action en réaction à des évènements discrets (de type signaux, ou invocation de méthode)
 - Peut spécifier le comportement interne d'une instance de classe, ou de composant, de cas d'utilisation, ou de méthodes
 - Machine dont le comportement ne dépend pas seulement de l'état de ses entrées, mais aussi d'un historique des sollicitations passées



Session 3 : Diagramme d'états-transitions

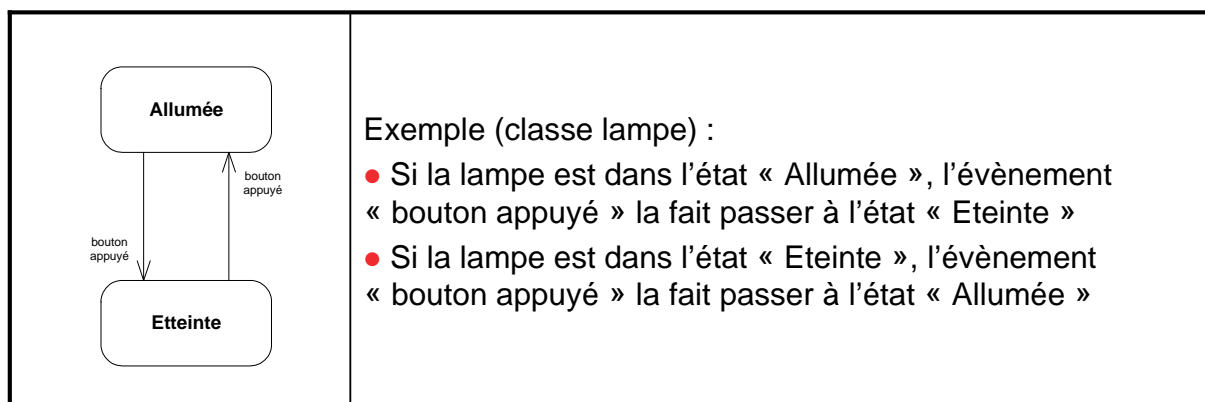
Concepts – Etat

- État global :
 - Jeu des état (élémentaires) actifs à un instant donné
 - Produit la même réponse face aux évènements
- État :
 - condition ou situation de vie d'un objet qui satisfait certaines conditions, réalise certaines activités et attend certains évènements
 - Types :
 - ✓ État composite : englobe des sous-états ()
 - ✓ État orthogonal : état composite comportant plus d'une région
- Région :
 - Représente un flot d'exécution d'un état orthogonal
 - Chaque région peut posséder un état initial et final



Session 3 : Diagramme d'états-transitions

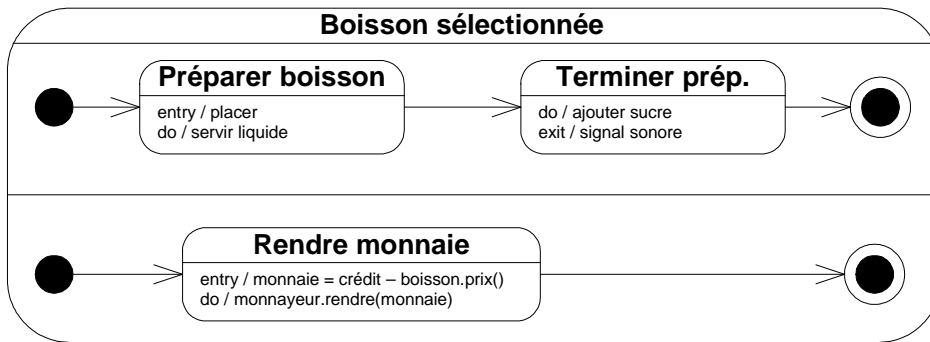
Concepts – Etat (exemple)





Session 3 : Diagramme d'états-transitions

Concepts – Etats concurrents



- États concurrents :
 - Sous états d'un état composite inclus dans des régions séparées
 - Un objet peut être simultanément dans deux états concurrents



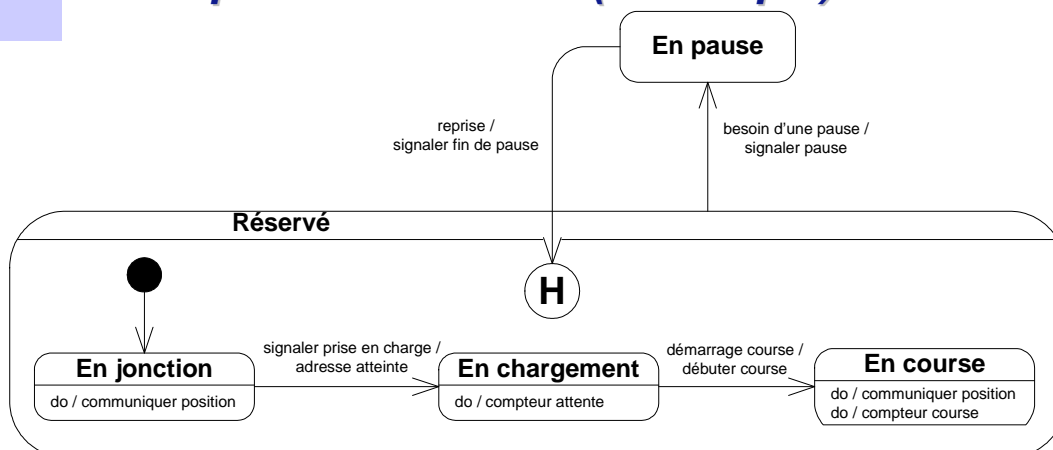
Session 3 : Diagramme d'états-transitions

Concepts – Pseudo-états

	État initial : indique l'état de départ
	État final : indique que le diagramme d'état-transitions ou l'état enveloppant, est terminé
	Point de jonction : <ul style="list-style-type: none"> • permet de partager les segments de transitions • état qui ne peut pas être actif, les gardes de sortie sont évaluées avant de permettre l'entrée
	Point de décision : les gardes de sortie sont évaluées un fois le point atteint.

Session 3 : Diagramme d'états-transitions

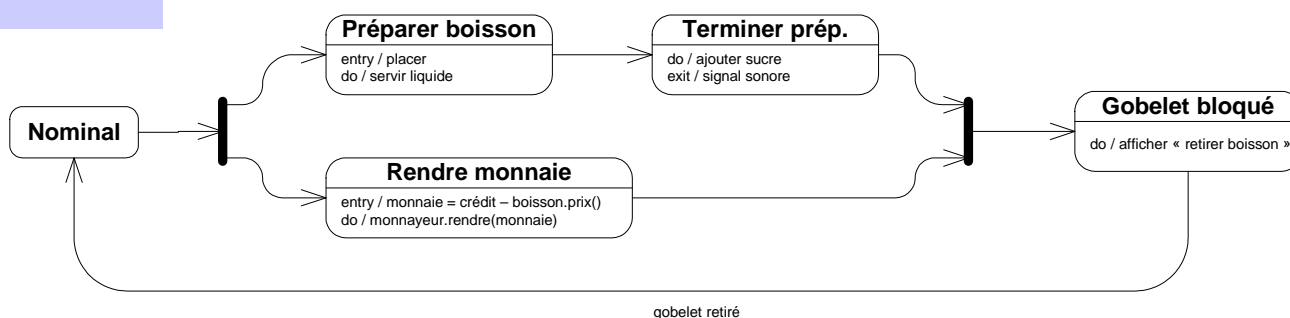
Concepts – Pseudo-état (historique)



(H)	Historique : une transition ayant pour cible le pseudo-état historique est équivalente à une transition qui a pour cible le dernier état visité dans la région le contenant
(H*)	Historique profond : historique valable pour tous les niveaux d'imbrication

Session 3 : Diagramme d'états-transitions

Concepts – Pseudo-état (synchronisation)

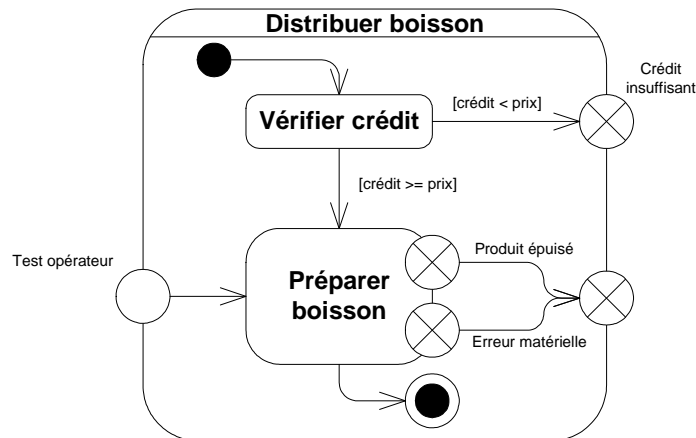




Transitions concurrentes	
	Fractionner : création de deux état concurrents
	Joindre : barrière de synchronisation qui supprime la concurrence



Session 3 : Diagramme d'états-transitions

Concepts – Pseudo-état (points de connexion)



Points de connexion	
 État	Point d'entrée : pointer un état interne de l'extérieur
 État	Point de sortie : pointer un état externe de l'intérieur



Session 3 : Diagramme d'états-transitions

Concepts – Evènement

- Pourquoi ? :
 - Spécifier les conditions de déclenchement des transitions
- Quoi ? :
 - Quelque chose qui se produit pendant l'exécution du système et qui mérite d'être modélisé
 - Un évènement se produit à un instant précis et est dépourvu de durée
 - Types d'évènements :
 - ✓ Signal : communication asynchrone à sens unique entre deux objets
 - ✓ Appel : réception de l'appel d'une opération par un objet
 - ✓ Changement : passage de « faux » à « vrai » d'une expression booléenne (syntaxe « **when (%condition booléenne%)** »)
 - ✓ Temporel : généré par le passage du temps (syntaxe « **when (%date%)** », ou « **after(%durée%)** »)



Session 3 : Diagramme d'états-transitions

Concepts – Transition

- Définition :
 - Réponse d'un objet à l'occurrence d'un évènement
 - Lie deux états et indique qu'un objet dans un « état 1 » peut entrer dans l'« état 2 » et exécuter certaines activités si une condition est vérifiée
- Caractéristiques :
 - Évènement : évènement déclencheur de la transition
 - Garde : expression logique sur les attributs de l'objet. Si la transition est fausse au moment de l'occurrence de l'évènement, la transition ne se déclenche pas
 - Effet :
 - ✓ Déclenché au passage de la transition
 - ✓ Le comportement peut être une activité ou une machine d'état

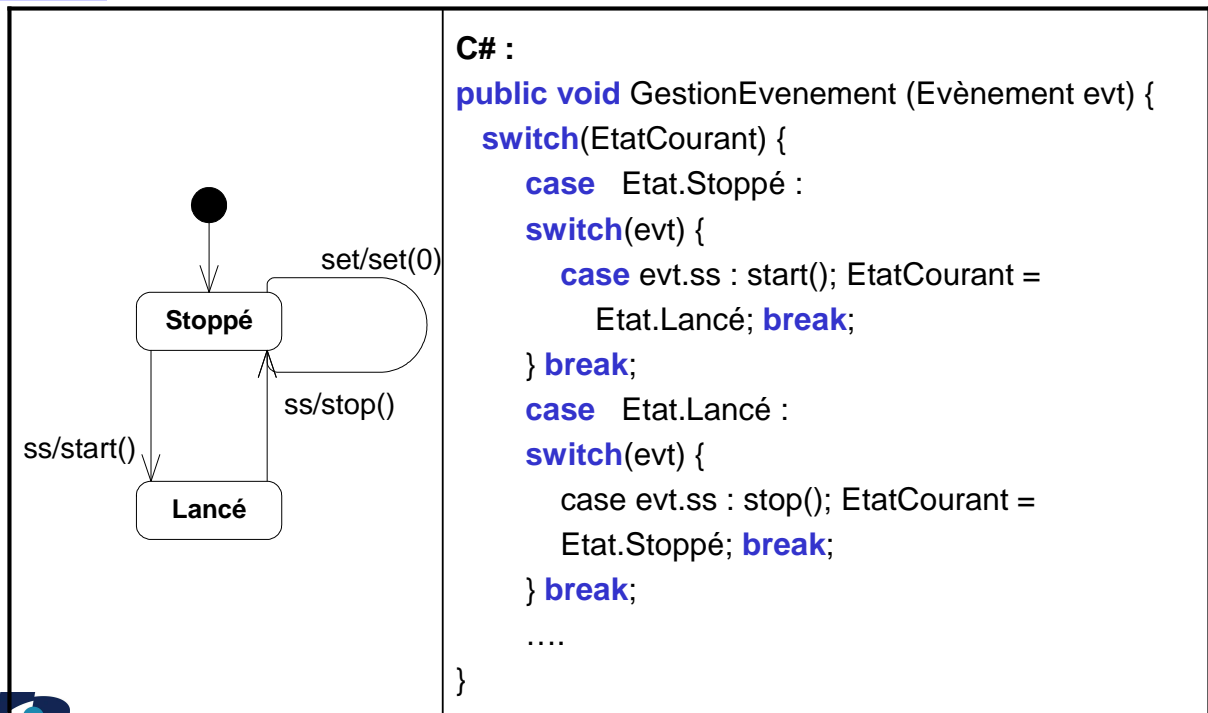


Session 3 : Diagramme d'états-transitions

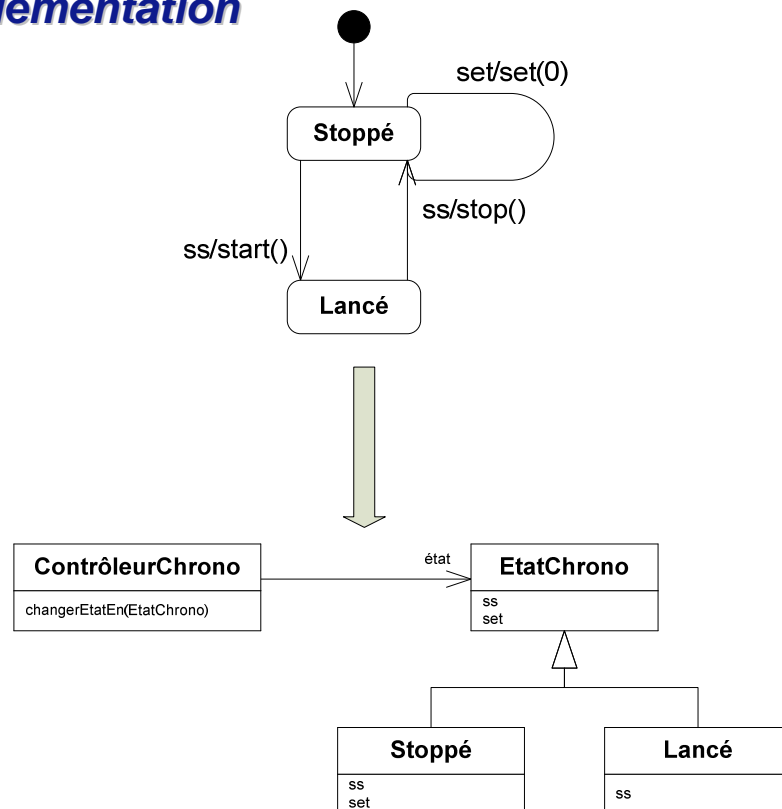
Concepts – Transition (types)

- Transition externe :
 - Transition qui modifie l'état actif
 - Symbole :
- Transition d'achèvement :
 - Transition dépourvue d'évènement déclencheur
 - Elle s'exécute à la fin des activités de l'état
 - Symbole :
- Transition interne :
 - Ne possède pas d'état cible, l'état actif reste l'état courant
 - Déclencheur particuliers :
 - ✓ **entry** : entrée dans l'état
 - ✓ **exit** : sortie de l'état
 - ✓ **do** : dès que l'activité « entry » est finie
 - ✓ **include** : inclus un sous-diagramme d'état-transitions
 - Symbole :

Session 3 : Diagramme d'états-transitions Implémentation - Switch



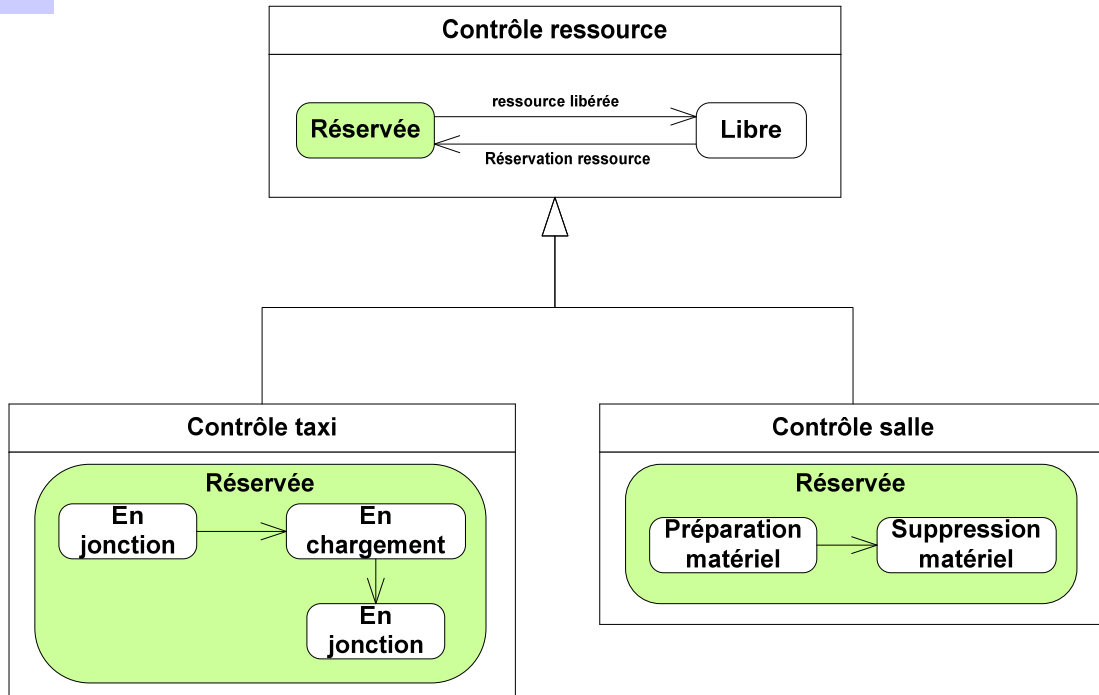
Session 3 : Diagramme d'états-transitions Implémentation





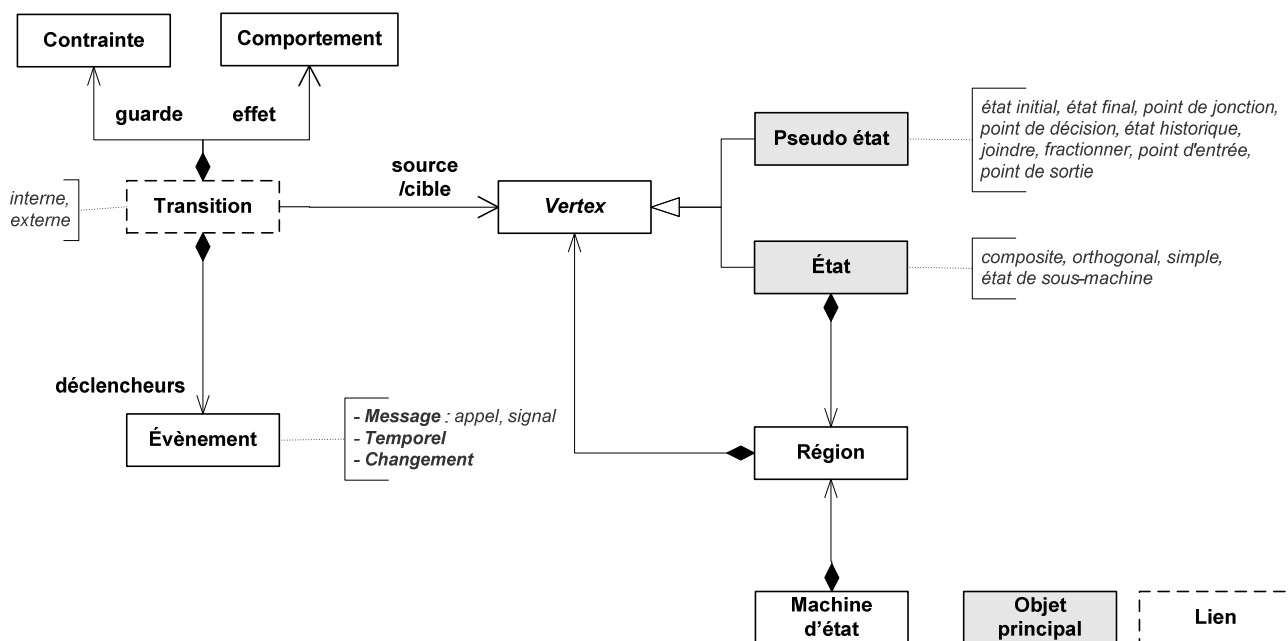
Session 3 : Diagramme d'états-transitions

Héritage



Session 3 : Diagramme d'états-transitions

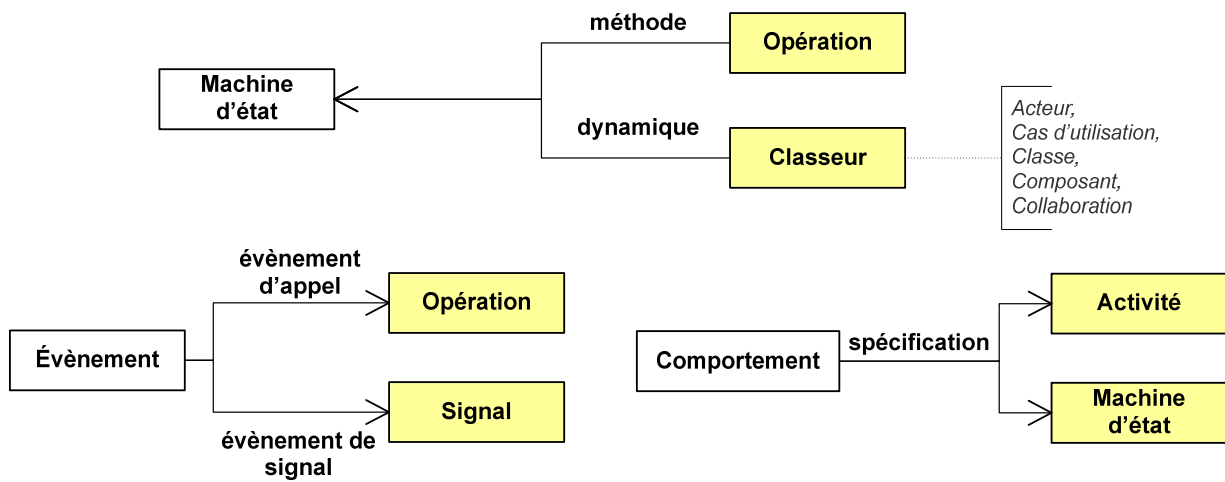
Synthèse des concepts manipulés





Session 3 : Diagramme d'états-transitions

Synthèse des concepts externes



Session 3 : Diagramme d'états-transitions

Exercices

- Exercice 7 :
 - Représenter un diagramme d'états-transition
- Exercice 8 :
 - Représenter des interactions entre classes



Session 3 : *Bonnes pratiques*

- Diagramme d'interaction :
 - Se limiter aux scénarios nominaux
- Diagramme d'états transitions :
 - Limiter l'utilisation des effets de transitions externes



Session 3 : *Vue fonctionnelle* *Sommaire*

- Diagramme d'interaction :
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de timing
- Diagramme d'activités
- Diagramme d'états-transitions
- **TD**

www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com