

La notation UML

Unified Modeling Language

Zoubida Kedad

UML

Use Cases
(Cas d'utilisations)

Diagramme de Use Cases

- Sert à donner une vue intuitive des interactions entre le système et ses utilisateurs potentiels.
- Diagramme représentant :
 - les acteurs du système
 - les cas d'utilisation du système
 - les relations :
 - entre acteurs
 - acteurs/cas d'utilisation
 - entre cas d'utilisation

Acteur

- Entité externe agissant sur le système
 - être humain
 - Machine
 - autre système
- L'acteur peut consulter et/ou modifier l'état du système
- Le système répond à l'acteur en lui fournissant des informations, et le prévient éventuellement lors de changements d'états

Acteur

- ❑ Défini en fonction du rôle qu'il joue
- ❑ Représente un nombre quelconque d'entités externes utilisant le système
- ❑ Notations graphiques



Use Cases

- ❑ Séquence d'actions réalisées par le système produisant un résultat observable à un acteur particulier
- ❑ L'exécution du Use Case est contrôlée par des événements externes envoyés au système par les acteurs
- ❑ Les Use Cases peuvent être organisés en packages
- ❑ L'ensemble des Use Cases décrit les exigences fonctionnelles du système

Use Cases

- ▣ Notation graphique



Diagrammes de Use Cases

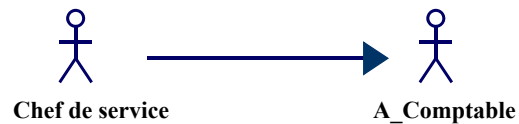
- Permettent de reformuler les besoins et constituent un moyen de communication entre utilisateurs et équipes de développement
- Un Use Case décrit le système du point de vue de son utilisation :
 - ▣ les interactions entre le système et les acteurs
 - ▣ les réactions du système aux événements externes
 - ▣ permet de développer un système orienté utilisateur

Représentation des relations dans un diagramme de Use Cases

■ Association Use Case/acteur

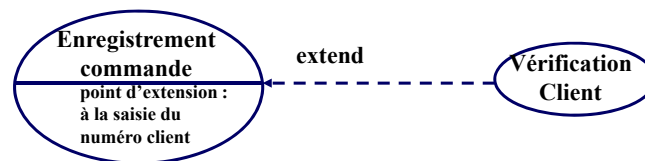


■ Association entre acteurs

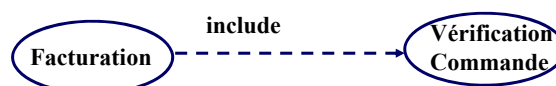


Représentation des relations dans un diagramme de Use Cases

■ « extend »



■ « include »



Use Cases

- Comment décrire des Use Cases :
 - rédiger une spécification textuelle
 - faire un diagramme global montrant les relations entre les acteurs et les Use Cases
 - Construire un diagramme d'objets participants pour chaque Use Case

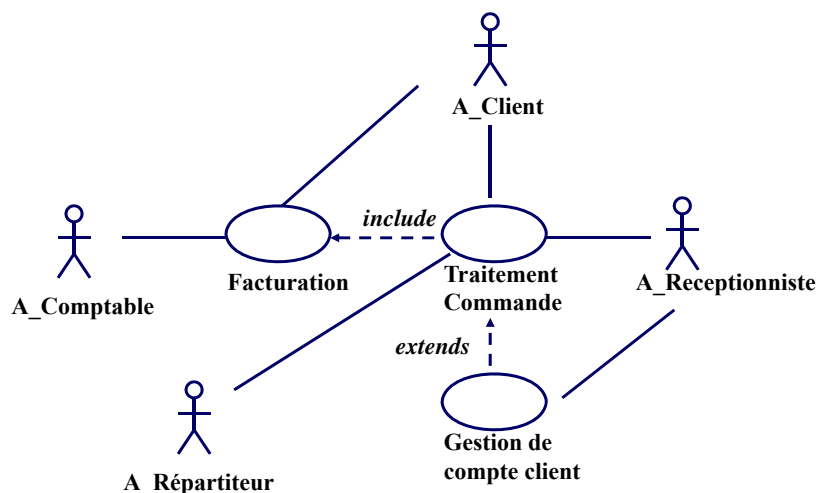
Use Cases : description textuelle

- Use Case : nom et numéro
- Auteur :
- Date de mise à jour :
- Résumé : (description sommaire)
- Contexte d'utilisation : (fréquence,)
- Acteurs : (au moins 1)
- Pré-conditions : (initialisation, restrictions)

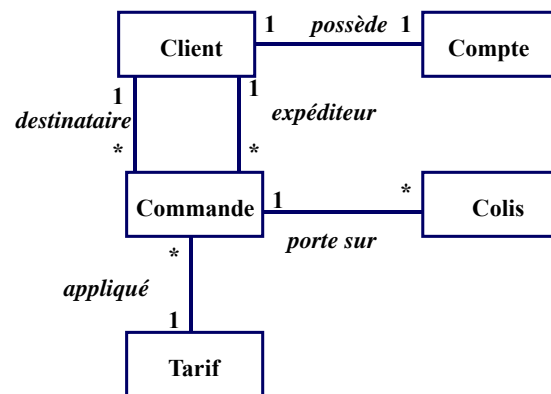
..... Use Cases : description textuelle

- Description :
 - ce Use Case débute lorsque l'acteur X.....le système répond par.....
 - [séquences d'interactions]
 -les données manipulées, enregistrées, modifiées,..
 -description de toutes les options du comportement
 -règles de choix
 -événements envoyés aux acteurs
- Exceptions envoyées aux acteurs : (ponctuelles et globales)
- Post-conditions : (ce qui a changé dans le système à la fin du Use Case)

Use Cases : diagramme



Use Cases : diagramme d'objets participants



UML

Diagrammes dynamiques :
Le diagramme de séquence

Modélisation de la dynamique d'un système d'information

- Un modèle dynamique représente les séquences d'événements, d'états et de réactions qui doivent survenir dans le système.
- Le modèle dynamique est intimement lié au modèle des objets et décrit les aspects de contrôle d'un système, en prenant en compte le temps, le séquençement des opérations et les interactions entre les objets

Objectifs des diagrammes dynamiques

- Représentation du cycle de vie des objets
 - liste des états possibles
- Représentation des événements
 - provoquent la transition d'un état à un autre
- Représentation des opérations
 - décrivent les transformations entre deux états

Diagrammes de séquence

- Permettent de décrire des scénarios
- Un scénario est une exécution particulière du système

Scénarios

- Un scénario montre une séquence particulière d'interactions entre objets, dans un seul contexte d'exécution du système
- Un scénario peut être vu comme une des instances possibles d'un des Uses Cases

Exemple

■ Scénario :

- deux usagers U1 et U2 situés respectivement au rez-de-chaussée et au deuxième étage empruntent un ascenseur pour se rendre au troisième étage

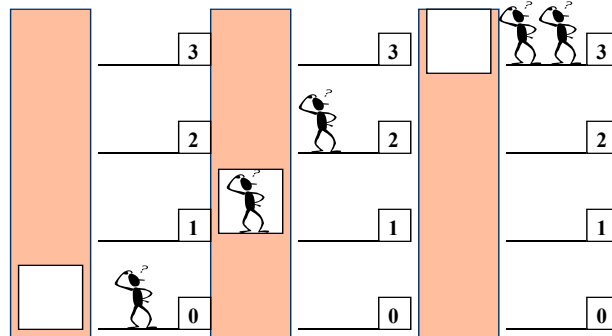


Diagramme de séquence

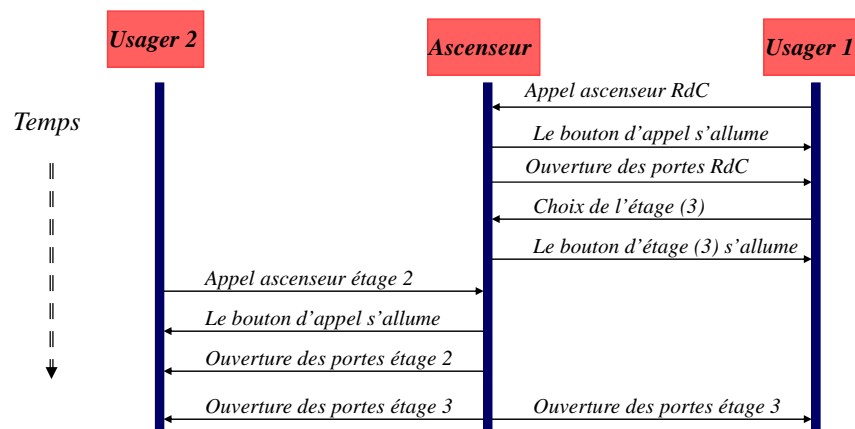


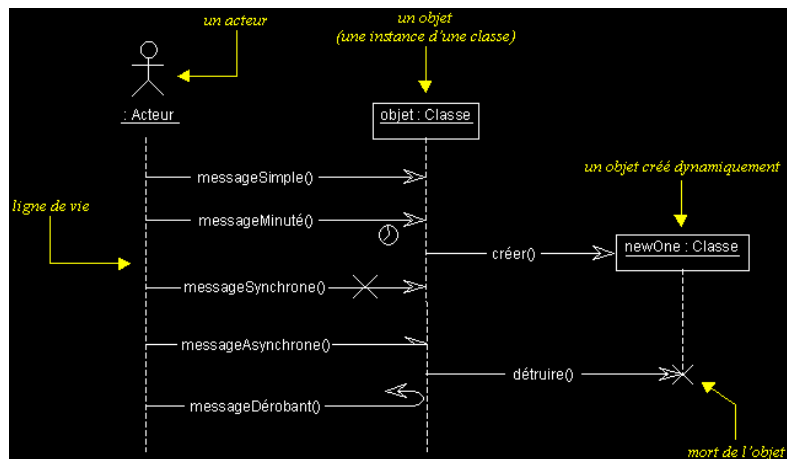
Diagramme de séquence

- Dans l'exemple précédent :
 - Chaque ligne verticale représente un objet :
 - «Usager 1»
 - «Usager 2»
 - «Ascenseur»
 - Chaque flèche horizontale représente un événement entre deux objets :
 - «appel ascenseur»
 -

Evénements

- Transmission d'information d'un objet à un autre
- Sans durée
- Liens de causalité entre événements
- Concurrence entre événements
- Caractérisés par :
 - nom, source, destination, description, attributs
 - Exemple :
 - click souris, décrocher ligne téléphonique

Diagramme de séquence



Types de messages

- **Message simple**
 - Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.
- **Message minuté (timeout)**
 - Bloque l'expéditeur pendant un temps donné, en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.
- **Message synchrone**
 - Bloque l'expéditeur jusqu'à prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise en compte du message.

Types de messages

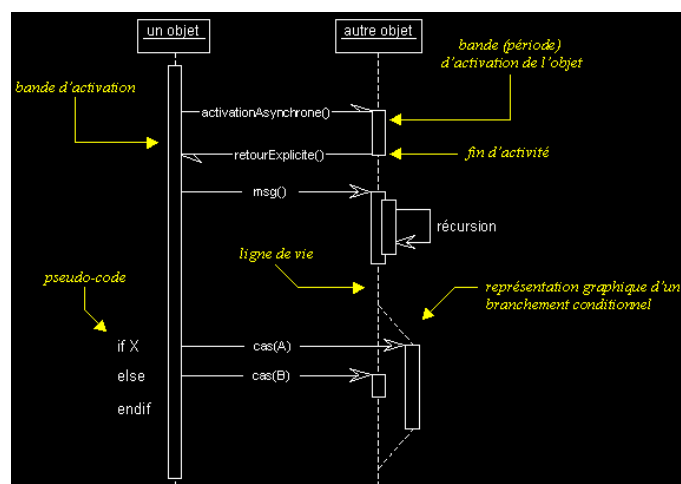
■ Message asynchrone

- N'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

■ Message déroband

- N'interrompt pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message.

Activation d'un objet



UML

Diagrammes dynamiques :
Le diagramme d'état/transition

Diagrammes d'états/transitions

- Un diagramme d'états/transitions :
 - ❑ décrit l'évolution au cours du temps d'une instance d'une classe en réponse aux interactions avec d'autres objets
 - ❑ est associé à une classe (mais seules certaines classes seront décrites par un diagramme d'états/transitions)
 - ❑ est un graphe orienté d'états (noeuds) connectés par des transitions (arcs orientés)

Etats d'un objet

- Valeur d'un objet durant un intervalle de temps séparant deux événements
- Caractérisé par une durée
- Caractérisé par des valeurs d'attributs de l'objet

Différence entre état et événement

- Un état correspond à la valeur d'un objet, persistante sur une période de temps
- Un événement correspond au changement de cette valeur à un instant donné

Changement d'état d'un objet

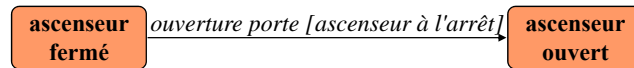
- Transition

- passage d'un état à un autre en réponse à un événement

- Condition

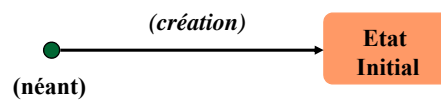
- expression booléenne devant être vraie pour qu'une transition ait lieu

- Exemple :



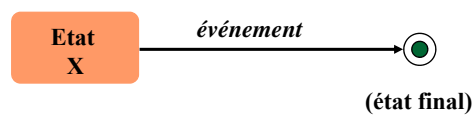
Etat initial

- L'état initial décrit l'état d'une instance de classe juste après sa création



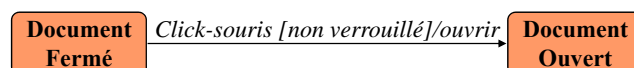
Etat final

- L'état final précise les conditions de destruction de l'instance de classe



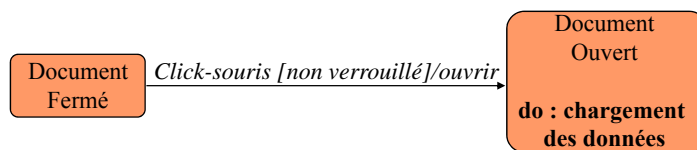
Action

- Méthode faisant passer un objet d'un état à un autre
 - ininterrompible
 - sans durée
 - déclenchée par une transition
- Avec l'événement et la condition, elle contribue à spécifier une transition



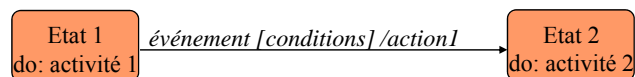
Activité

- Enchaînement d'un ensemble d'actions
 - ❑ réalisée par un objet dans un certain état
 - ❑ peut être interrompue par un ou plusieurs événements



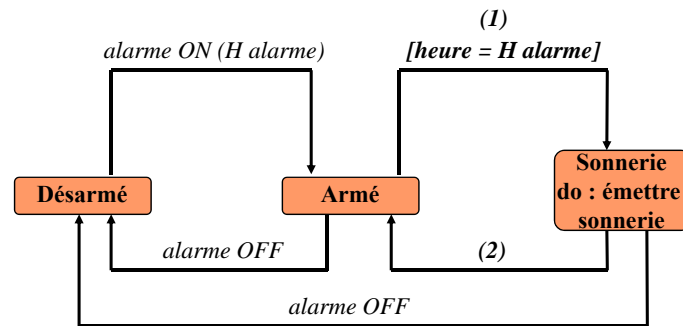
Opérations

- ❑ Action :
 - opération instantanée, non interruptible
 - exécutée durant une transition
- ❑ Activité :
 - opération caractérisée par une durée, pouvant être interrompue par un événement
 - exécutée durant un état



Transition sans événement

- Exemple : diagramme d'états/transitions de l'objet « réveil »



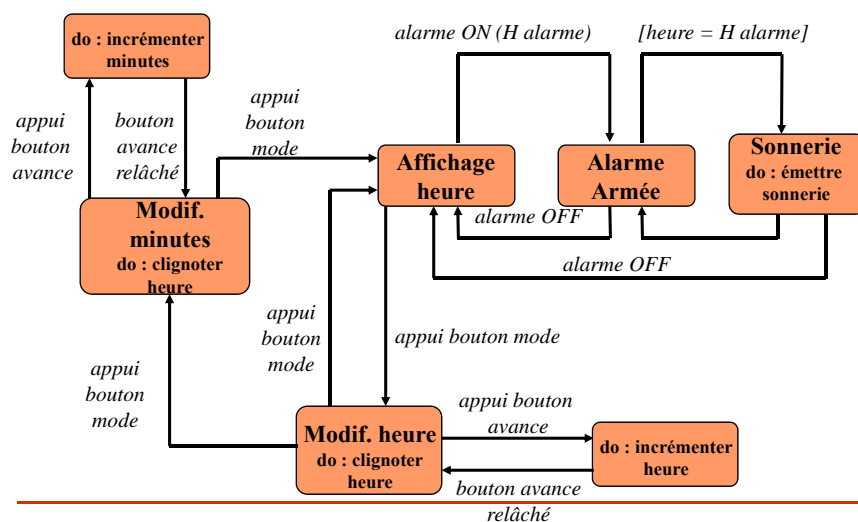
Transition sans événement

- Événement interne : la transition est déclenchée lorsque la condition devient vraie (1)
- L'activité associée à l'état source est terminée (2)

Diagramme d'états/transitions

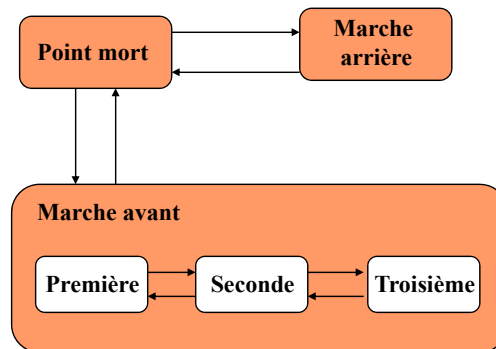
- Représentation des liens entre états et événements
- Graphe d'états connectés par des transitions labellées par des événements
- Abstraction de tous les scénarios possibles

Exemple de diagramme d'états/transitions



Diagrammes d'états/transitions structurés

- Diagramme d'états/transitions d'une classe d'objets « voiture »



UML

Diagrammes dynamiques :
Le diagramme d'activités

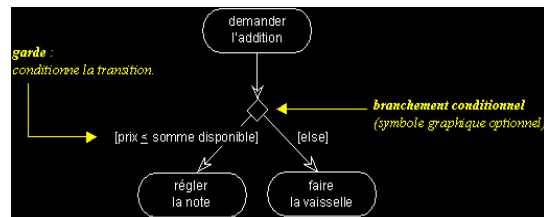
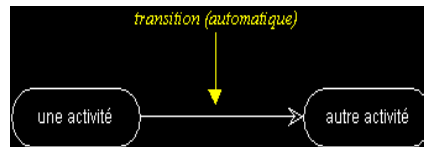
Diagramme d'activités

- Variante des diagrammes d'états-transitions
- Permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation

Diagramme d'activités

- Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles.
- Le passage d'une activité vers une autre est matérialisé par une transition.
- Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre (elles sont automatiques).
- En théorie, tous les mécanismes dynamiques pourraient être décrits par un diagramme d'activités, mais seuls les mécanismes complexes ou intéressants méritent d'être représentés.

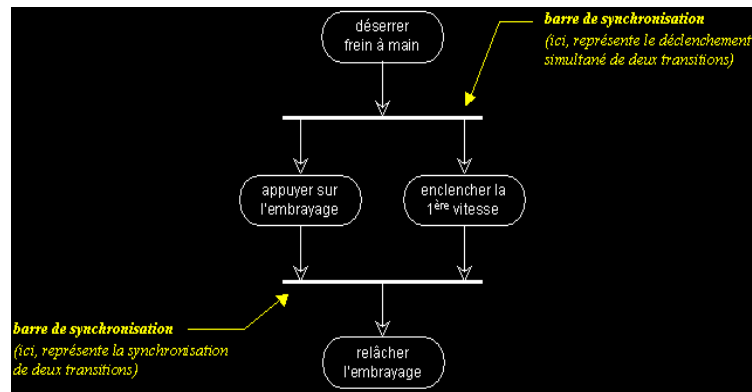
Notations graphiques



Synchronisation

- Une barre de synchronisation permet d'ouvrir et de fermer des branches parallèles au sein d'un flot d'exécution :
 - Les transitions qui partent d'une barre de synchronisation ont lieu en même temps.
 - On ne franchit une barre de synchronisation qu'après réalisation de toutes les transitions qui s'y rattachent.

Exemple de synchronisation



Couloirs d'activités

- Permettent d'organiser un diagramme d'activités selon les différents responsables des actions représentées
- Il est possible de faire apparaître sur le diagramme les états des objets mis en oeuvre

Exemple

