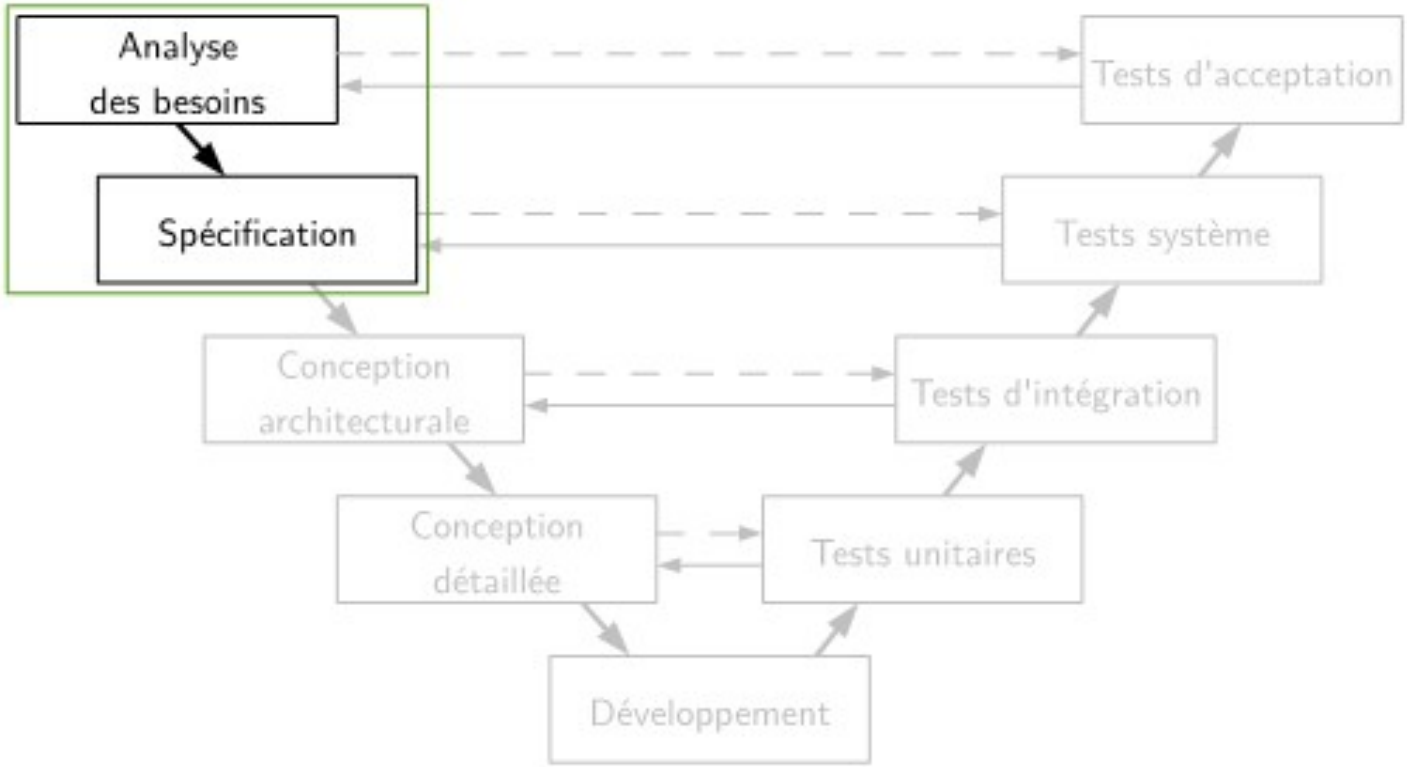
Chap 3:

Diagrammes de Cas d'Utilisation et de Séquences

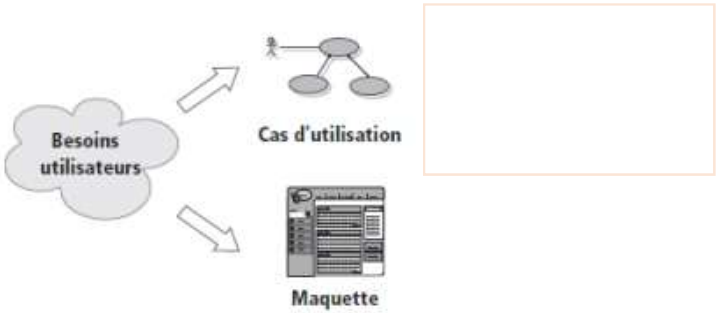
Processus de développement logiciel



Diagrammes de cas d'utilisation

Comment passer des besoins au code ?

L'expression préliminaire des besoins donne lieu à une modélisation par les cas d'utilisation et à une maquette d'Interface Homme-Machine (IHM)



Objectif : Comprendre les *besoins du client* pour rédiger le *cahier des charges fonctionnel*

Les diagrammes de cas d'utilisation sont utilisés pour recueillir les exigences d'utilisation d'un système.

Diagrammes de cas d'utilisation

Comment passer des besoins au code ?

QUOI ?

Avant tout développement, il convient de répondre à la question : *"A quoi va servir le logiciel ?"*

sous peine de fournir des efforts considérables en vain.

En UML, on établit des *Diagrammes de Cas d'Utilisation* pour répondre à cette question.

Les diagrammes de cas d'utilisation sont utilisés pour recueillir les exigences d'utilisation d'un système.

Questions :

- à quoi sert le système ? (*les fonctionnalités principales*)
- qui va utiliser ou interagir avec le système ? (*les acteurs*)
- où s'arrête la responsabilité du système ? (*les limites*)

• **Utiles** pour la discussion avec le client (intuitifs et concis)

• **Pas suffisant** pour l'équipe de développement

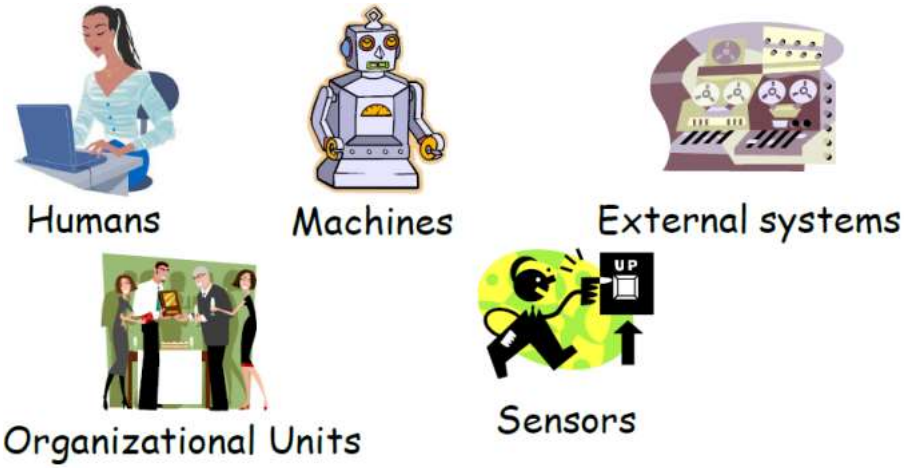
Principaux éléments :

• Acteurs (bonhommes) 

• Cas d'utilisation (ellipses) 

Acteurs ?

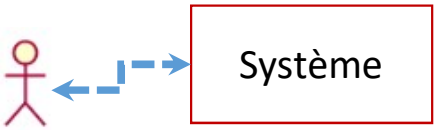
Un acteur est une entité extérieure au système modélisé, et qui interagit *directement* avec lui.



Acteurs non humains

- Les principaux acteurs sont les utilisateurs du système.
- En plus des utilisateurs, les acteurs peuvent être :
  - ✓ Des logiciels déjà disponibles à intégrer dans le projet ;
  - ✓ Des systèmes informatiques *externes* au système mais qui interagissent avec lui ;
  - ✓ tout élément *extérieur* au système et avec lequel il interagit

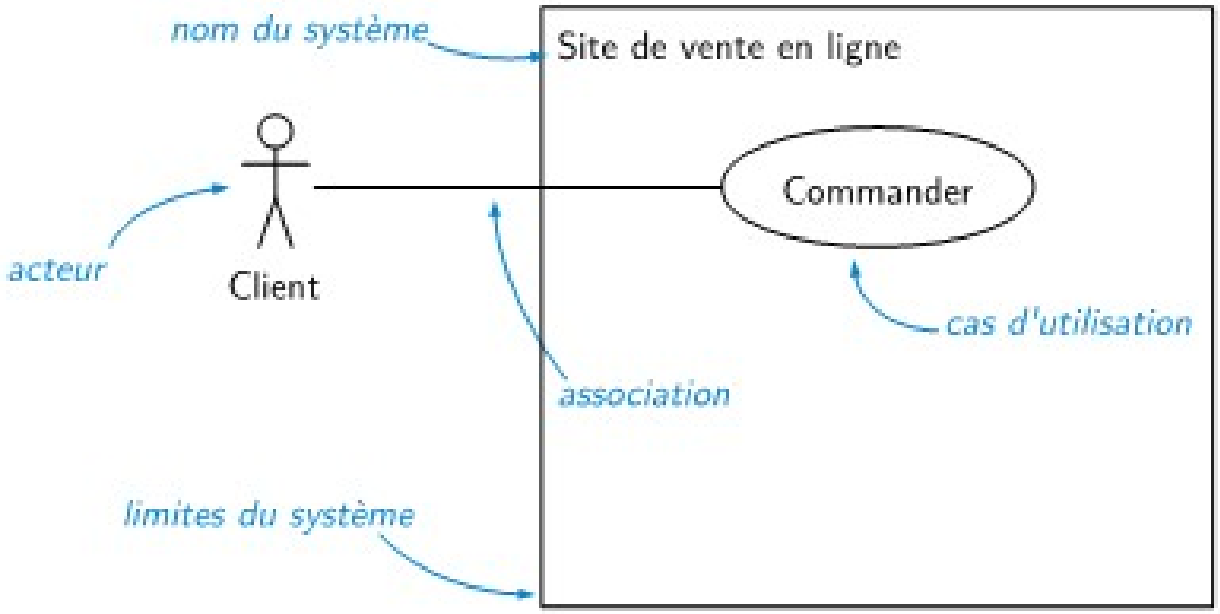
Pour identifier les acteurs, on se fonde sur les frontières du système.



C’est quoi un cas d’utilisation ?

- ✓ représente **une fonction ou une action** au sein du système
- ✓ Un cas d’utilisation du système
- ✓ Spécifie un aspect du comportement d’un système
- ✓ Besoin d’un utilisateur que le système doit satisfaire

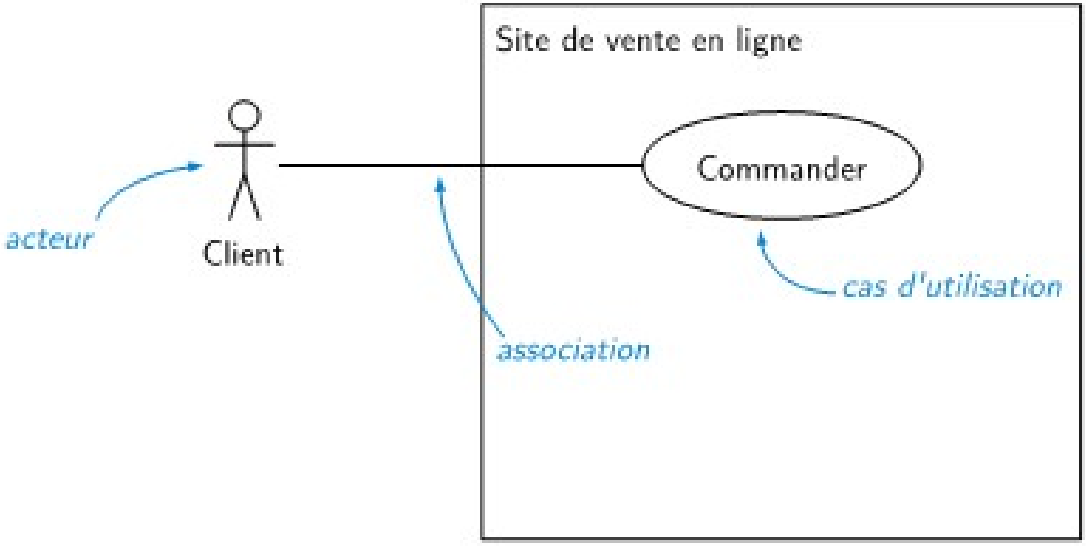
Diagramme des cas d'utilisation



Association :

- Relation entre acteurs et cas d'utilisation
- Représente la possibilité pour l'acteur de déclencher le cas

Associations



Association :

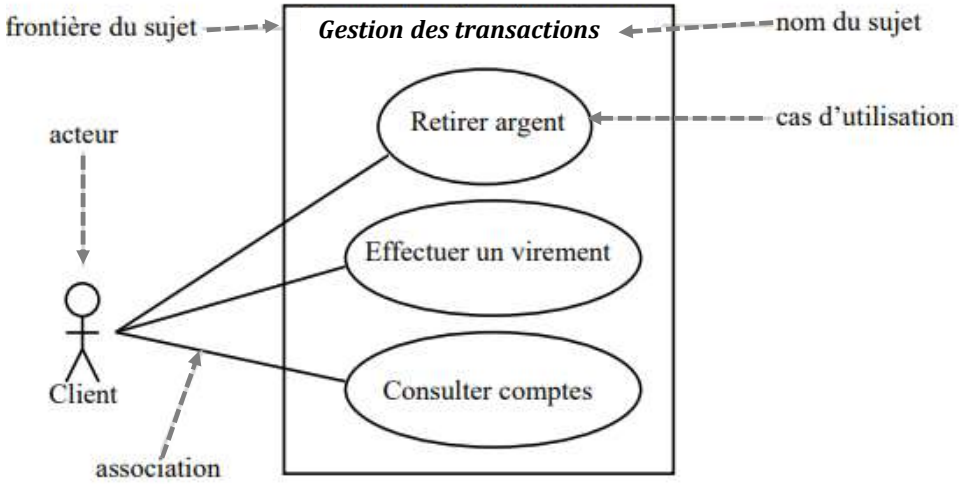
- Relation entre acteurs et cas d'utilisation
- Représente la possibilité pour l'acteur de déclencher le cas

Identification des cas d'utilisation (Use Case ou UC)

Une bonne façon de le faire est d'identifier ce dont les acteurs ont besoin de la part du système.

Exemple:

Dans un système bancaire, un client devra *ouvrir des comptes*, *déposer* et *retirer* des fonds, *demander des carnets de chèques, etc*. Tous ces éléments peuvent donc être considérés comme des UC.



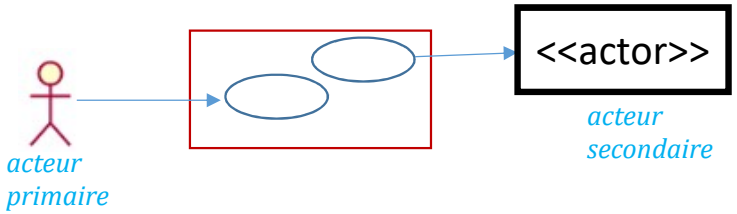
Associations entre cas et acteurs

Les acteurs demandant des services au système, ils sont le plus souvent à l'initiative des échanges avec le système :

- ils sont dits *acteurs primaires*.

Lorsqu'ils sont sollicités par le système (dans le cas de serveurs externes par exemple),

- ils sont dits *acteurs secondaires*.



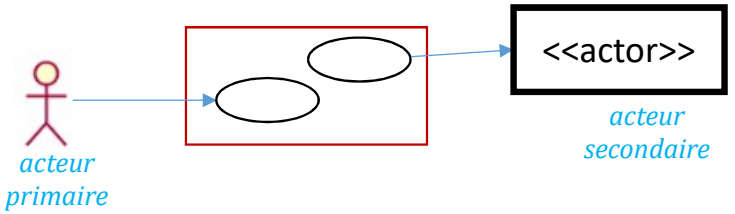
Associations entre cas et acteurs

Les acteurs demandant des services au système, ils sont le plus souvent à l'initiative des échanges avec le système :

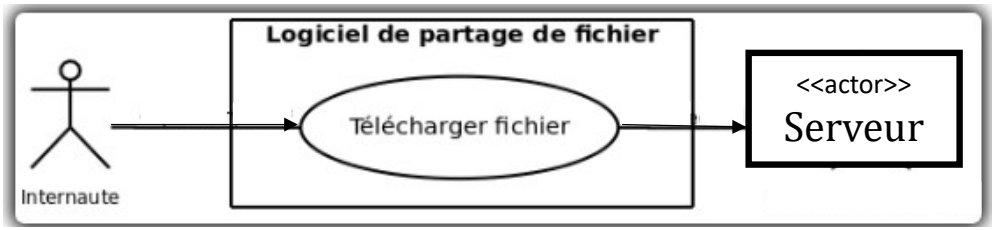
- ils sont dits *acteurs primaires*.

Lorsqu'ils sont sollicités par le système (dans le cas de serveurs externes par exemple),

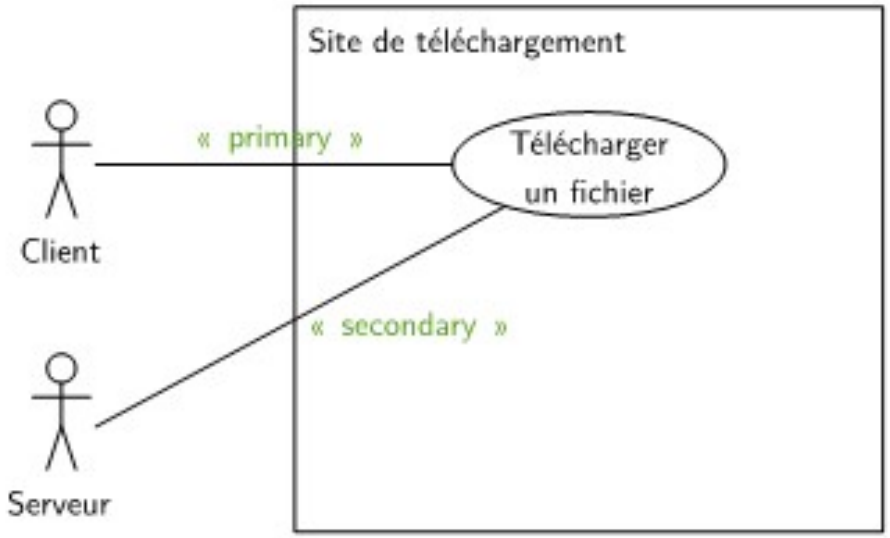
- ils sont dits *acteurs secondaires*.



un acteur secondaire est sollicité pour des informations complémentaires



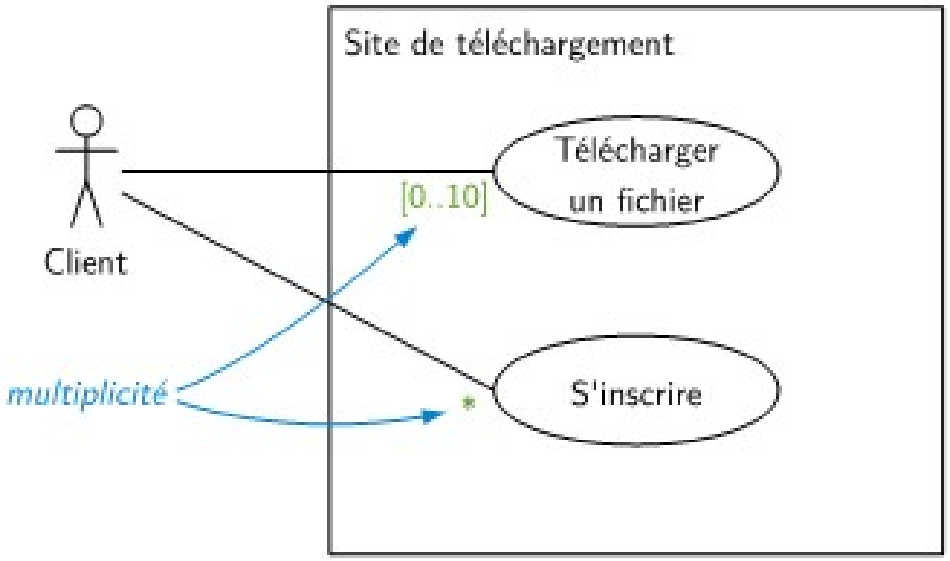
Associations



Acteurs primaires et secondaires :

- Acteur primaire « primary » : acteur déclenchant le cas
- Acteur secondaire « secondary » : acteur sollicité par le cas

Associations



Multiplicité : Nombre de fois où l'acteur peut déclencher le cas

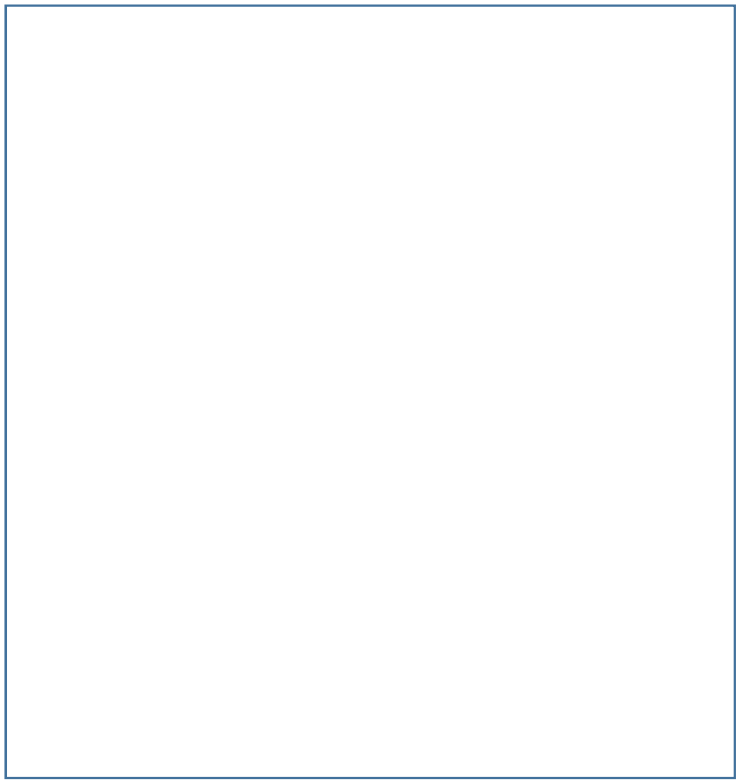
- * : une infinité de fois (pas représenté en général)
- $[n..m]$: entre n et m fois
- n : exactement n fois

Exemple 1 :

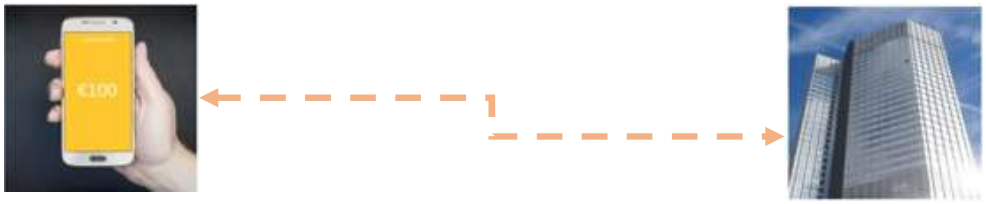
Donner un diagramme UC qui représente un système nommé ***Système de gestion des étudiants*** qui offre les fonctionnalités suivantes.

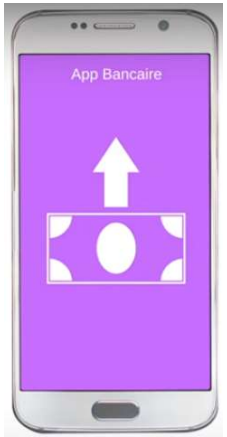
Les enseignants et les étudiants peuvent consulter les emplois du temps, les notes des étudiants et la liste de présence des étudiants.

Seuls les enseignants peuvent mettre à jour la présence ou les notes des étudiants.



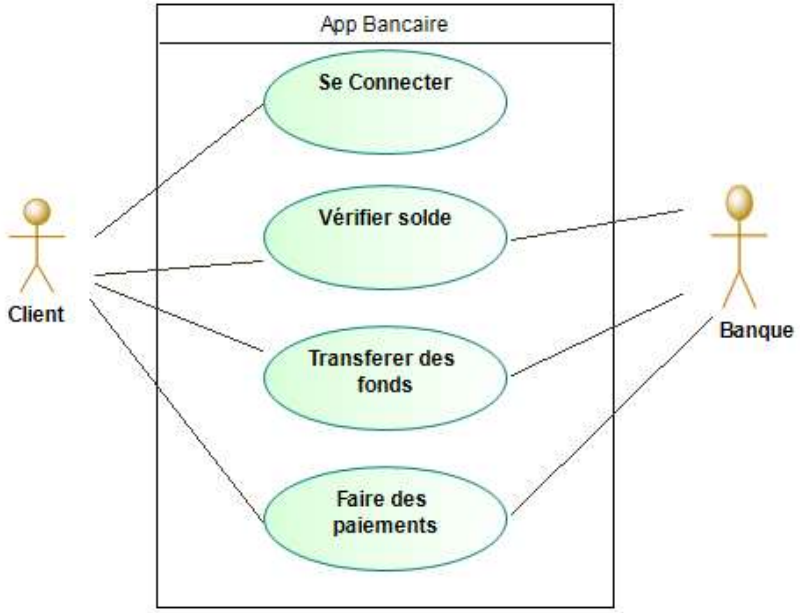
Exemple 2 : Application mobile bancaire





Le client, avec son smartyphone, peut:

- Se connecter
- Vérifier son solde
- Transférer des fonds entre comptes
- Faire des paiements



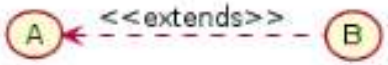
Etapes a suivre :

- ✓ Comprendre le problème,
- ✓ identifier les acteurs,
- ✓ identifier les cas d’utilisations
- ✓ dresser un premier diagramme de cas d’utilisation

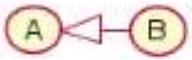
Types de relations possibles

- *Inclusion* : B est une partie obligatoire de A et on lit *A inclut B* (dans le sens de la flèche).

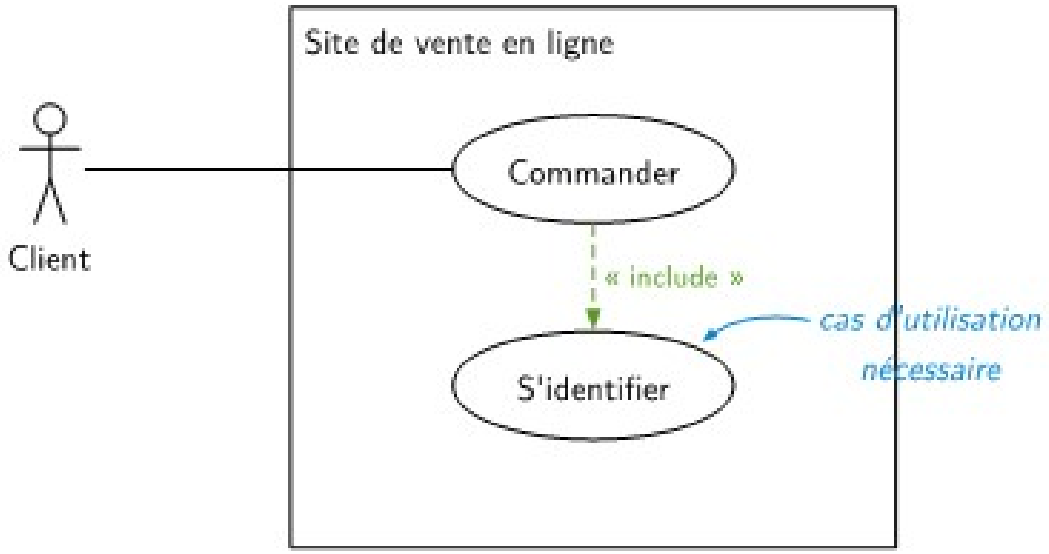
- *Extension* : B est une partie optionelle de A et on lit *B étend A* (dans le sens de la flèche).



- *Généralisation* : le cas A est une généralisation du cas du cas B et on lit *B est une sorte de A*.



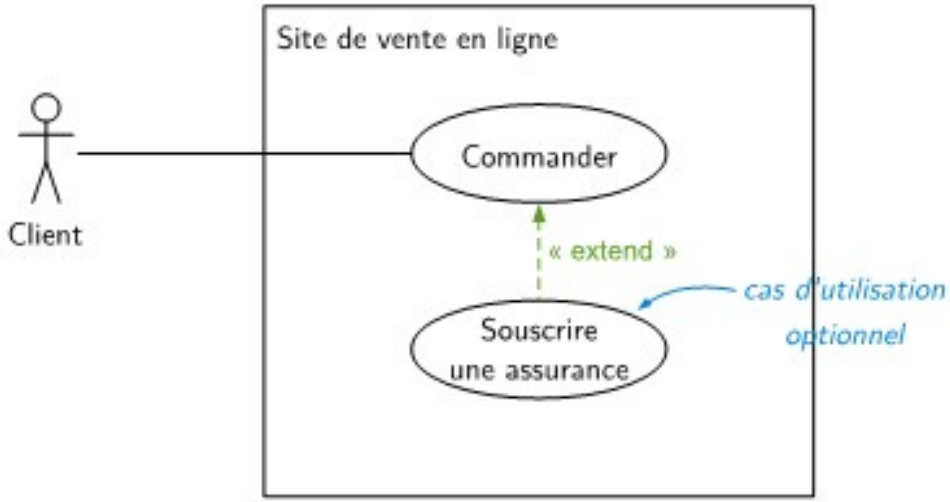
Entre cas d'utilisation



Relations entre cas d'utilisation

- Inclusion : X « include » Y ⇔ X implique Y
  - ↳ Y est nécessaire pour X

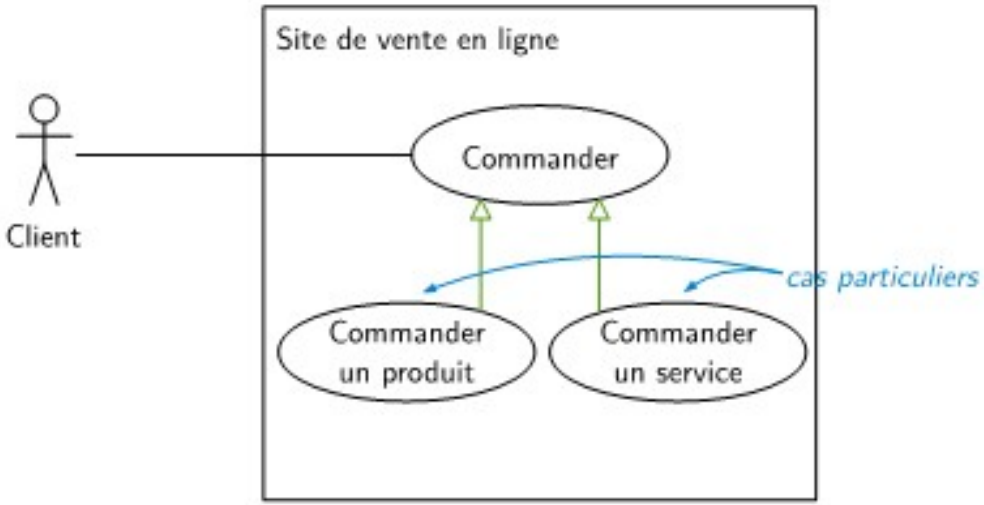
Entre cas d'utilisation

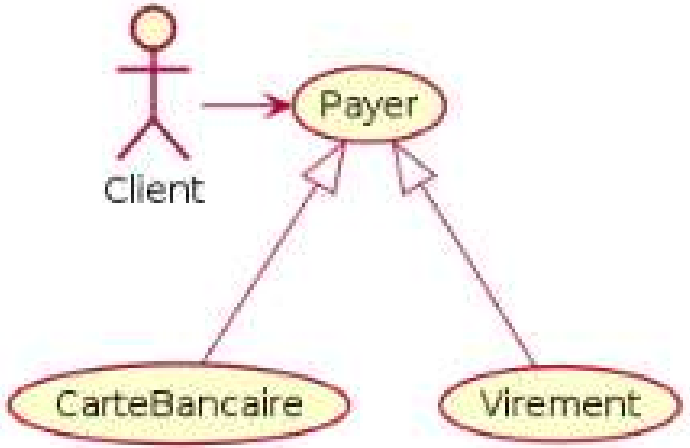


Relations entre cas d'utilisation

- Inclusion : X « include » Y ⇔ X implique Y
- Extension : X « extend » Y ⇔ X peut être provoqué par Y
  ↳ X est optionnel pour Y

Entre cas d'utilisation

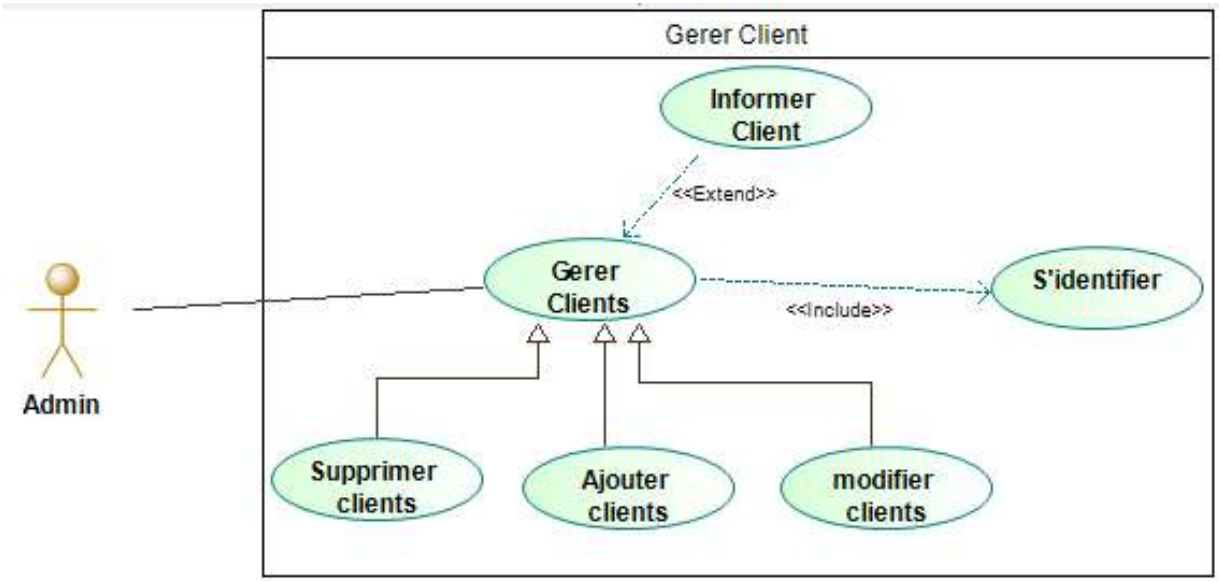




Relations entre cas d'utilisation

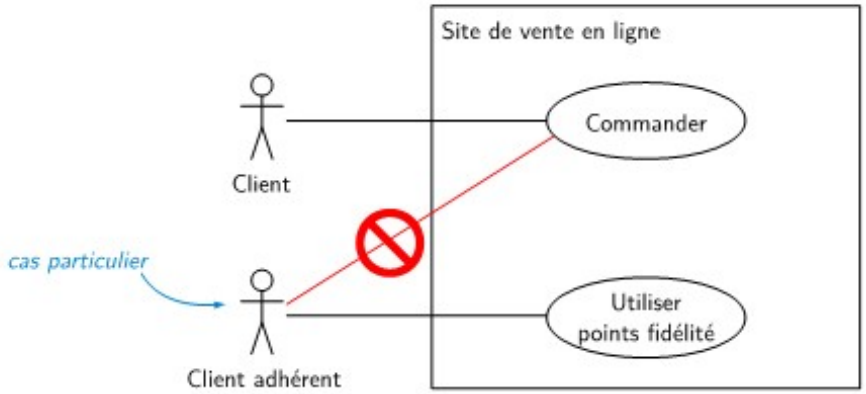
- Inclusion : X « include » Y ⇔ X implique Y
- Extension : X « extend » Y ⇔ X peut être provoqué par Y
- Généralisation : X est un cas particulier de Y

Exemple



- ✓ *Supprimer*, *ajouter* ou *modifier* un client c'est *gérer* un client.
- ✓ En gérant un client on peut lui *informer* mais pas obligatoire
- ✓ Pour gérer les clients on est obligé de *s'authentifier*.

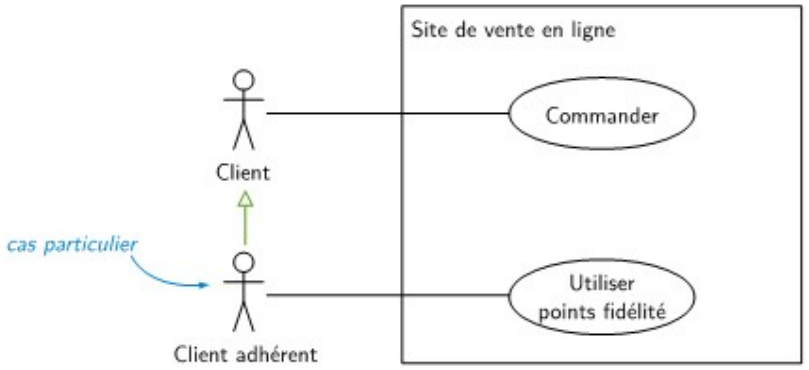
Entre les acteurs



Relations entre acteurs

- Généralisation : X peut faire tout ce que fait Y

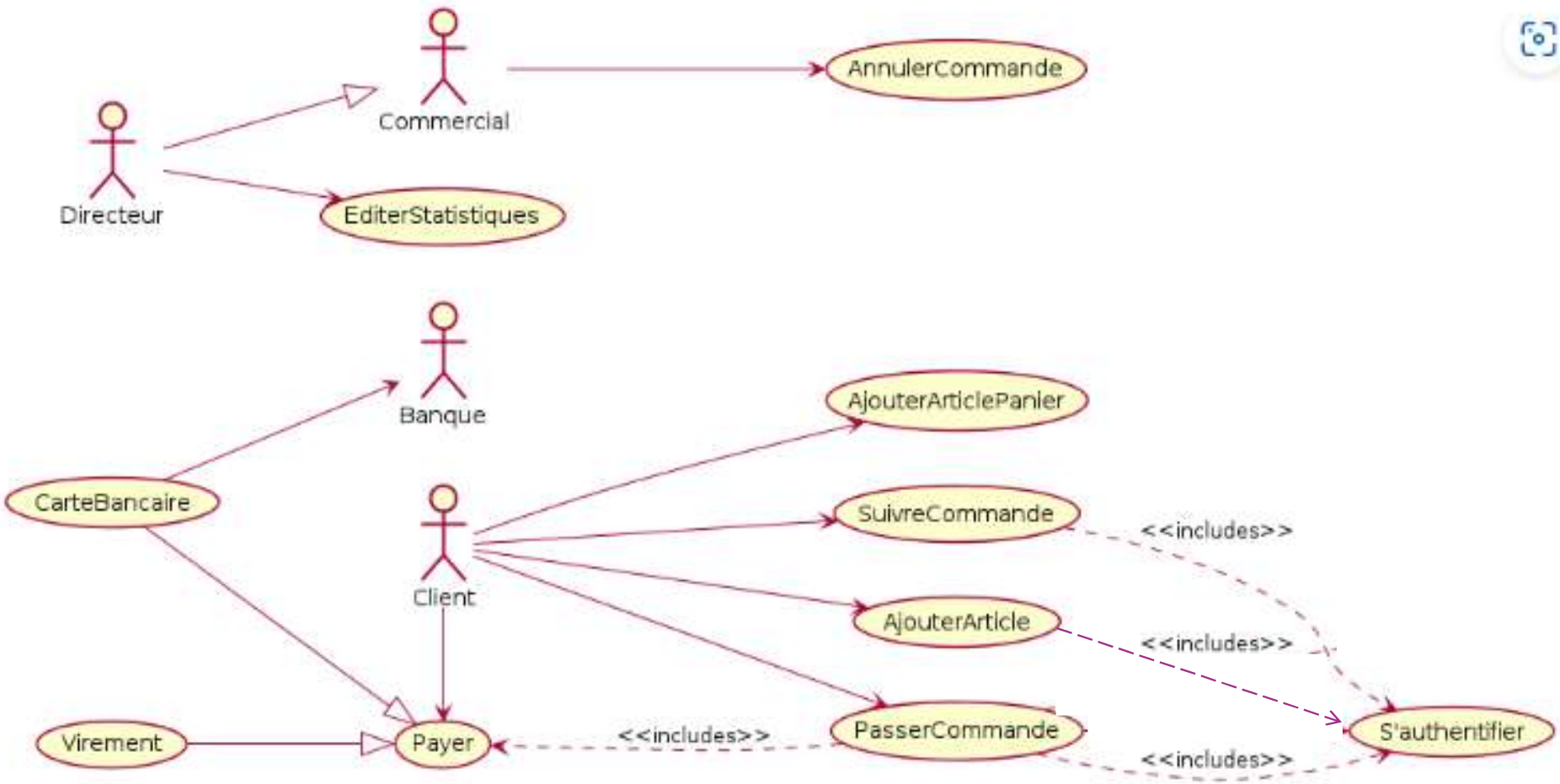
Entre les acteurs



Relations entre acteurs

- Généralisation : X peut faire tout ce que fait Y

Exemple complet



Exemple

On considère le système suivant de gestion des clients d'une banque :

- ✓ Un client de la banque doit pouvoir retirer, déposer (chèque ou numéraire) de l'argent ou consulter son solde directement à la banque.
- ✓ le DAB (Distributeur automatique de billets) délivre également de l'argent à tout porteur de carte (carte Visa ou carte Gimtel)
- ✓ toute transaction doit être sécurisée et nécessitée par conséquent une authentification
- ✓ dans le cas où une carte est avalée par le DAB, un opérateur de maintenance se charge de la récupérer. C'est la même personne qui collecte également les dépôts d'argent et qui recharge le DAB.

Travail à Faire :

Modéliser cette situation par un diagramme de cas d'utilisation.

Réponse

Scénarios d'utilisation

Exemple

L'exemple de la figure ci-dessus ne permet pas de savoir ce qui entre et ce qui sort du logiciel bancaire :

- le retrait d'argent se fait-il en quelle monnaie ?
- Dans quel ordre les opérations sont-elles effectuées ?
- Faut-il choisir le montant du retrait avant de choisir le compte à débiter, ou bien l'inverse ?

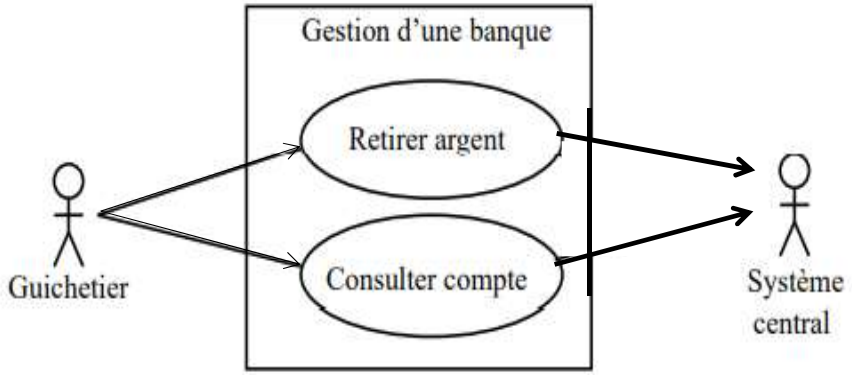
Tous ces détails sont des éléments de spécification.

Spécifier un produit, c'est le décrire de la façon la plus précise possible.

Séquences d'étapes

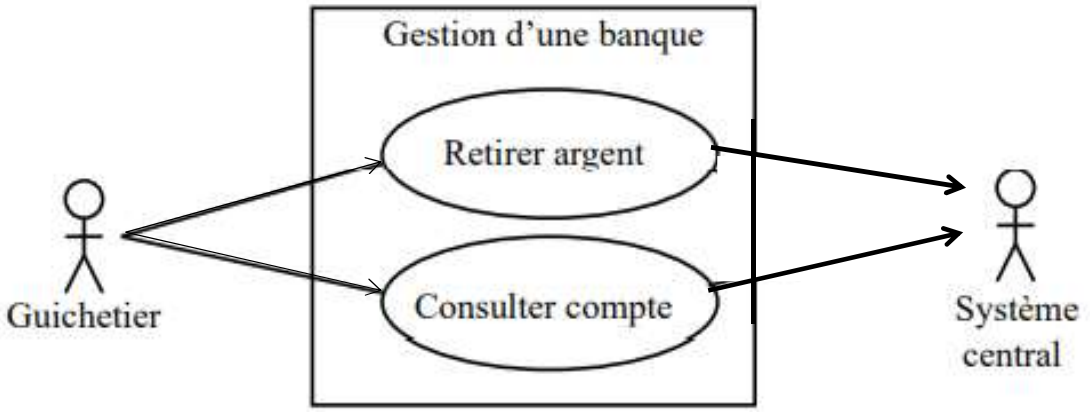
- décrivant une interaction entre l'utilisateur et le système
- permettant à l'utilisateur de réaliser un objectif

Intérêt : Base de discussion avec le client pour l'analyse des besoins



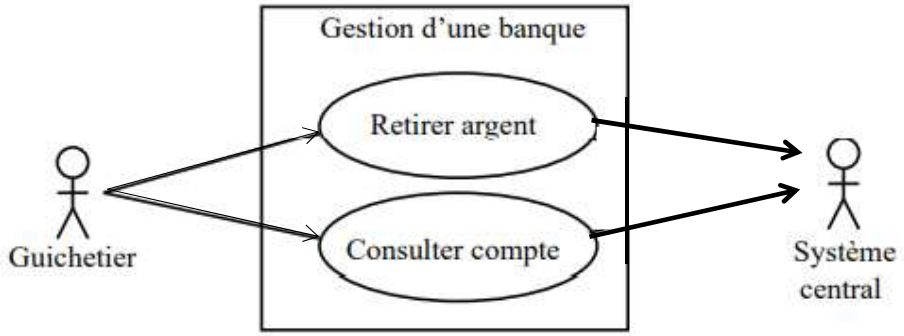
Scénarios d'utilisation

Il faut décrire le fonctionnement du système sous la forme d'une ***séquence de messages*** échangés entre les acteurs et le système.



- **Les pré-conditions :** *indiquent dans quel état est le système avant que se déroule la séquence (le distributeur est alimenté en billets par exemple).*
- **L'enchaînement des messages.**
- **Les post-conditions :** *indiquent dans quel état se trouve le système après le déroulement de la séquence nominale (une transaction a été enregistrée par la banque par exemple)*

Exemple (suite)



Le cas d'utilisation commence lorsqu'un client demande le retrait d'espèces en ouguiya.

Pré-conditions

Le client possède un compte (donne son numéro de compte).

Enchaînement nominal

1. Le guichetier saisit le numéro de compte client. 2. L'application valide le compte auprès du système central. 3. L'application demande le type d'opération au guichetier. 4. Le guichetier sélectionne un retrait d'espèces de 200 MRO. 5. L'application demande au système central de débiter le compte. 6. Le système notifie au guichetier qu'il peut délivrer le montant demandé.

Post-conditions

1. Le guichetier ferme le compte. 2. Le client récupère l'argent.

Exemple

Scénarios d'utilisation

Séquences d'étapes

- décrivant une interaction entre l'utilisateur et le système
- permettant à l'utilisateur de réaliser un objectif

Intérêt : Base de discussion avec le client pour l'analyse des besoins

Système : Site de vente en ligne

Scénario : Effectuer une commande

Le client s'authentifie dans le système puis choisit une adresse et un mode de livraison. Le système indique le montant total de sa commande au client. Le client donne ses informations de paiement. La transaction est effectuée et le système en informe le client par e-mail.

Scénarios d'utilisation

Séquences d'étapes

- décrivant une interaction entre l'utilisateur et le système
- permettant à l'utilisateur de réaliser un objectif

Intérêt : Base de discussion avec le client pour l'analyse des besoins

Système : Site de vente en ligne

Scénario : Effectuer une commande

Le client s'authentifie dans le système puis choisit une adresse et un mode de livraison. Le système indique le montant total de sa commande au client. Le client donne ses informations de paiement. La transaction n'est pas autorisée, le système invite le client à changer de mode de paiement. Le client modifie ses informations. La transaction est effectuée et le système en informe le client par e-mail.

Et si le mode de paiement est rejete'

Cas d'utilisation

Ensemble de scénarios réalisant un objectif de l'utilisateur

Cas d'utilisation : Effectuer une commande

Scénario principal :

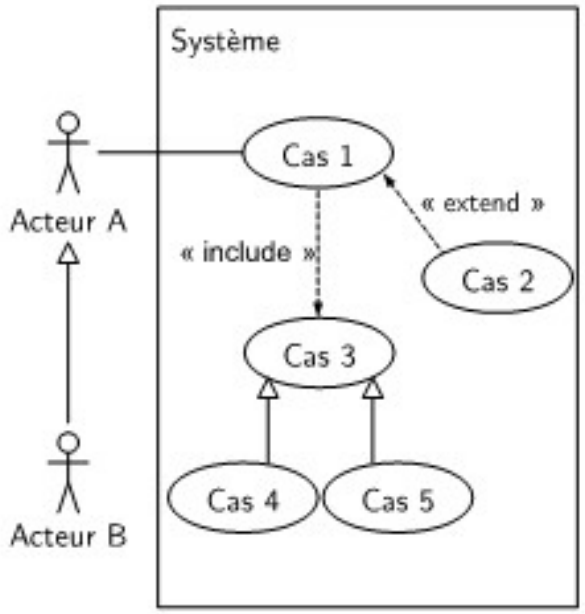
1. Le client s'authentifie dans le système
2. Le client choisit une adresse et un mode de livraison.
3. Le système indique le montant total de sa commande au client.
4. Le client donne ses informations de paiement.
5. La transaction est effectuée et le système en informe le client par e-mail.

Cas particulier :

5a. La transaction n'est pas autorisée, le système invite le client à changer de mode de paiement. Retour à l'étape 4.

Spécification des cas d'utilisation

Diagramme des cas d'utilisation + Description textuelle



Cas 1
Acteur : Acteur A
Contexte :
Entrées :
Sorties :
Scénario principal :
1.
2.
3.
Variantes :
1a.
1b.
3a.

Cas d'utilisation détaillé

Description textuelle d'un cas d'utilisation

- Nom du cas d'utilisation

  Brève description
- Acteurs
- Contexte

  Données en entrée et pré-conditions

  Données en sortie et post-conditions
- Scénario principal pour ce cas d'utilisation

  Étapes à suivre pour réaliser ce cas
- Variantes, cas d'erreur

  Déviations des étapes du scénario principal, scénarios alternatifs, scénarios d'erreur

Exemple : Retirer de l'argent du distributeur :

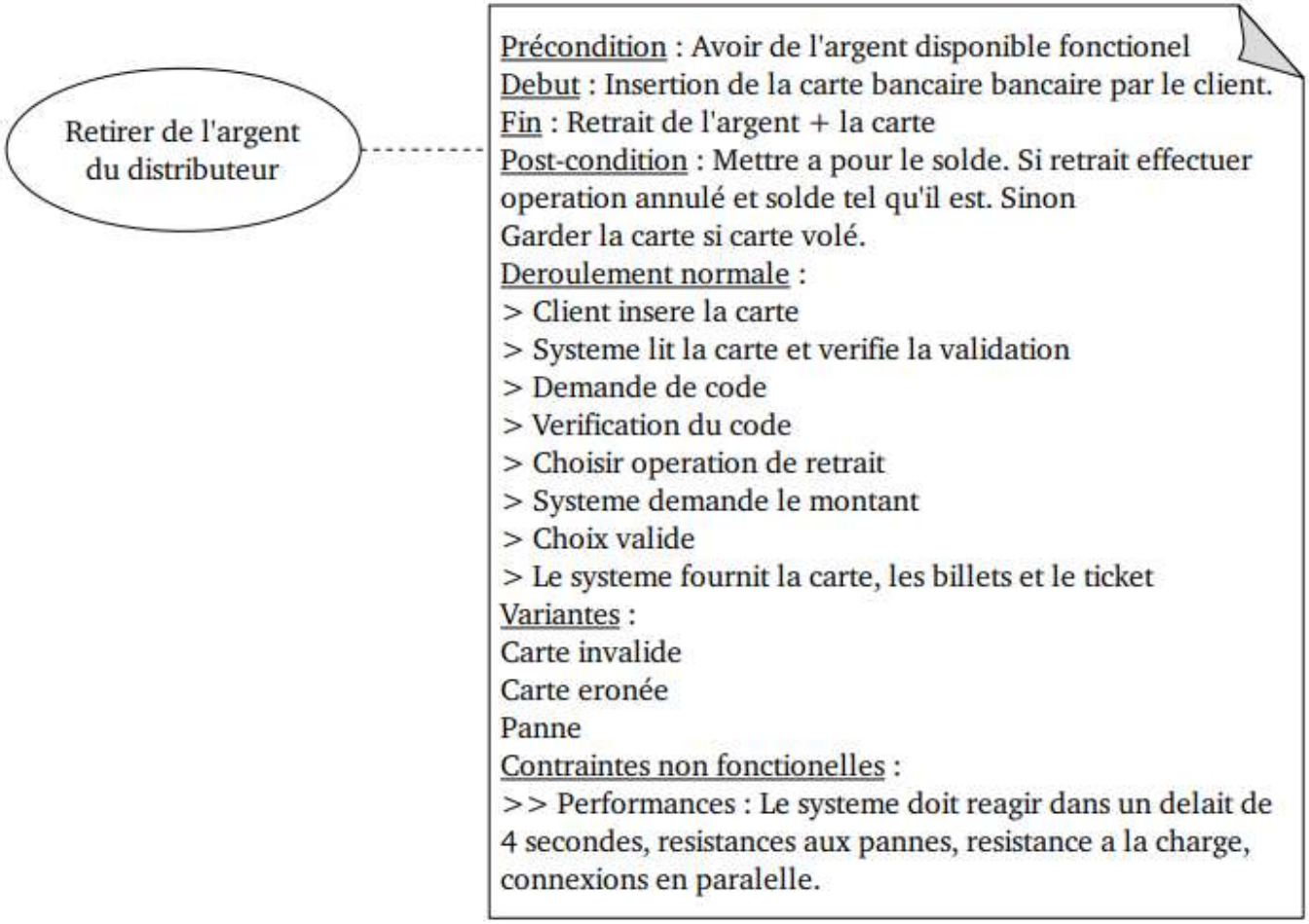


Diagramme de séquences

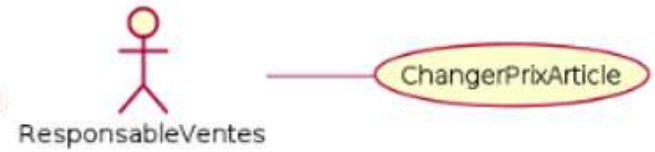
Diag de cas d'utilisation *vs.* Diag séquence vs. Diag classe

- Les ***diagrammes de cas d'utilisation*** modélisent à **QUOI** sert le système, en organisant les interactions possibles avec les acteurs.
- Les ***diagrammes de séquences*** permettent de décrire **COMMENT** les éléments du système interagissent entre eux et avec les acteurs :
  - ✓ Les objets au cœur d'un système interagissent en s'échangeant des messages.
  - ✓ Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).
- Les ***diagrammes de classes*** permettent de spécifier la structure et les liens entre les objets dont le système est composé : ils spécifie **QUI** sera à l'oeuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation.

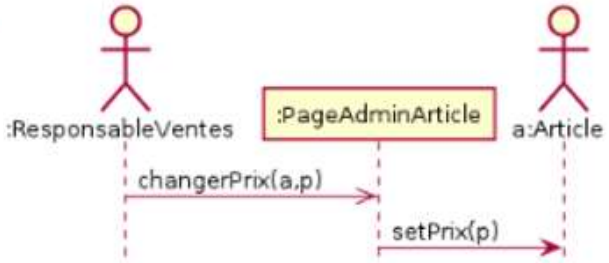
Interaction

Pour être complètement spécifiée, une interaction doit être décrite dans plusieurs diagrammes UML :

- Cas d'utilisation



- Séquences



- Classes pour spécifier les opérations nécessaires

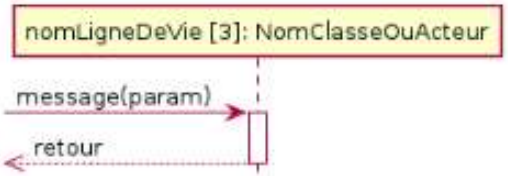
Ligne de vie

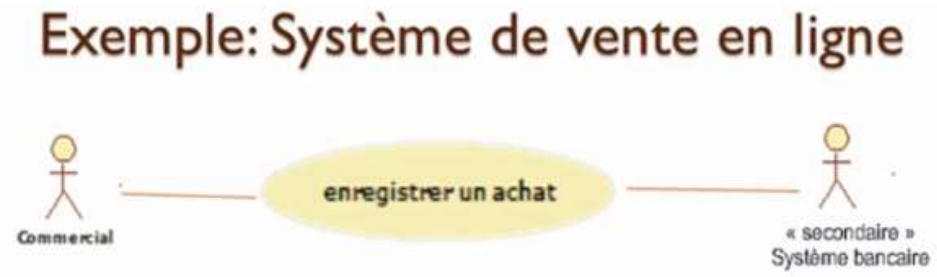
Une ligne de vie représente un participant à une interaction (objet ou acteur). La syntaxe de son libellé est :

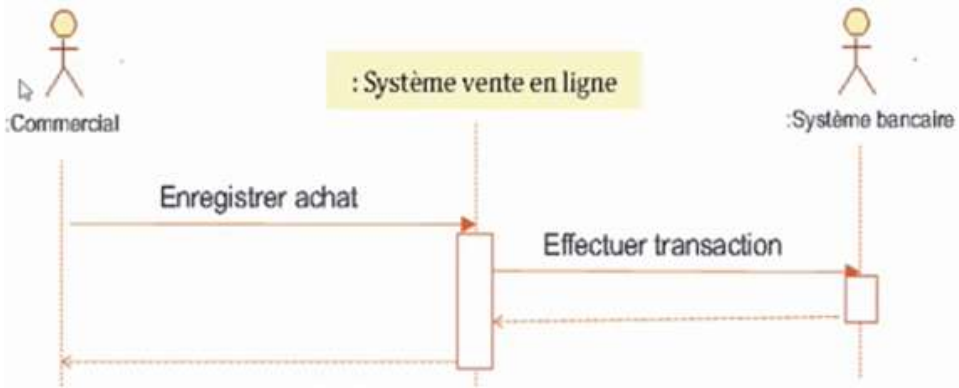
```
nomLigneDeVie {[selecteur]}: NomClasseOuActeur
```

Une ligne de vie est une instance, donc il y a nécessairement les *deux points* (:) dans son libellé.

Dans le cas d'une collection de participants, un sélecteur permet de choisir un objet parmi n (par exemple `objets[2]`).







Messages

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie :

- Ils sont représentés par des flèches
- Ils sont présentés du haut vers le bas le long des lignes de vie, dans un ordre chronologique

Un message définit une communication particulière entre des lignes de vie (objets ou acteurs).

Plusieurs types de messages existent, dont les plus courants :

- l'envoi d'un signal ;
- l'invocation d'une opération (appel de méthode) ;
- la création ou la destruction d'un objet.

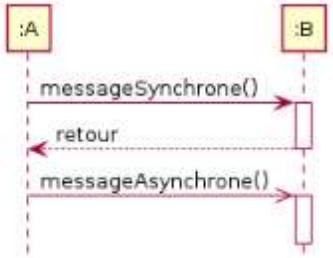
La réception des messages provoque une période d'activité (rectangle vertical sur la ligne de vie) marquant le traitement du message (spécification d'exécution dans le cas d'un appel de méthode).

Messages synchrones et asynchrones

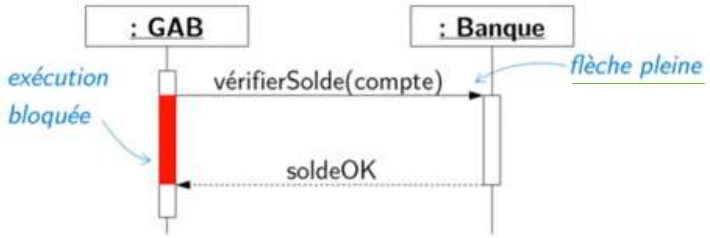
Un message *synchrone* bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.

- Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé.
- On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone.

Un message *asynchrone* n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.



Message synchrone : Émetteur bloqué en attente du retour



Message asynchrone : Émetteur non bloqué, continue son exécution

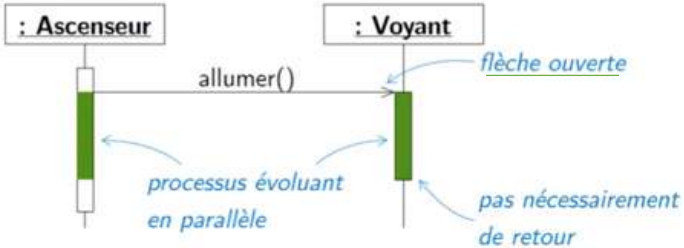
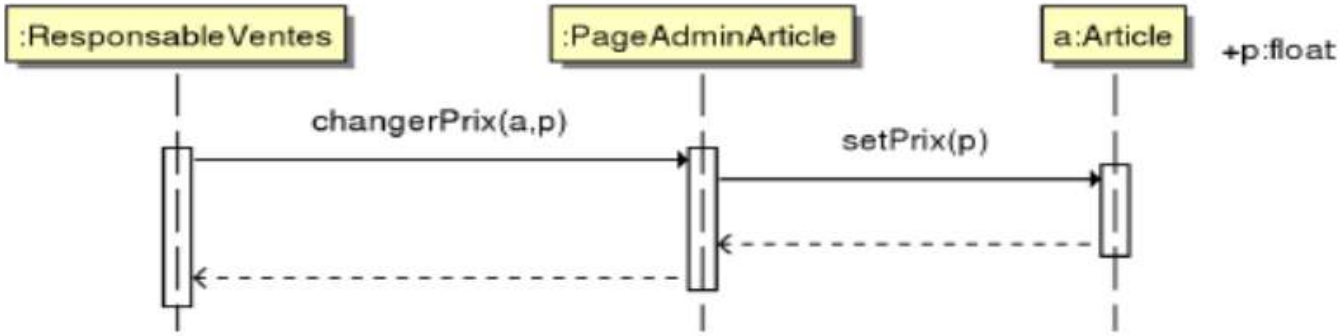


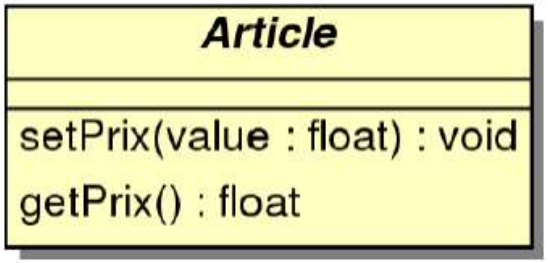
Diagramme de séquence – Exemple d'interaction

- Cas d'utilisation :

- Diagramme de séquences correspondant :



  - Opérations nécessaires dans le diagramme de classe:



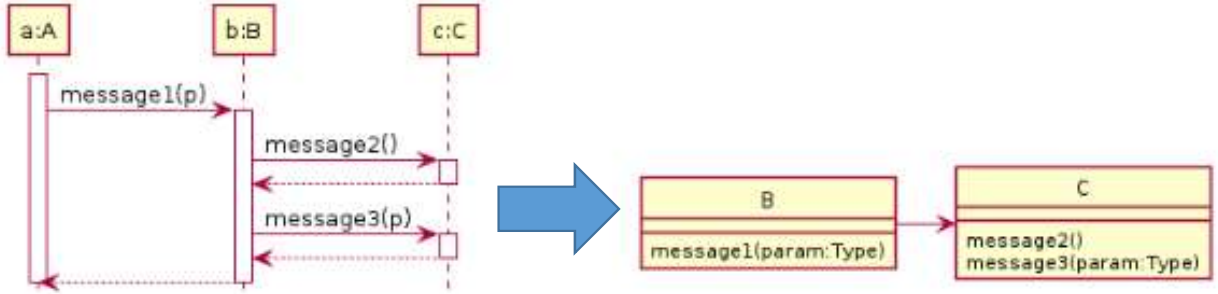
Messages synchrones et diagramme de classe

Les messages synchrones correspondent le plus souvent à une opération :

- A l'invocation, le flux contrôle passe de l'émetteur au récepteur
- L'émetteur attend la fin de l'exécution, et reprend après le retour

Les méthodes correspondant aux messages synchrones doivent être définies dans un diagramme de classes.

Les méthodes sont définies dans la classe du récepteur, et pas de l'émetteur du message.



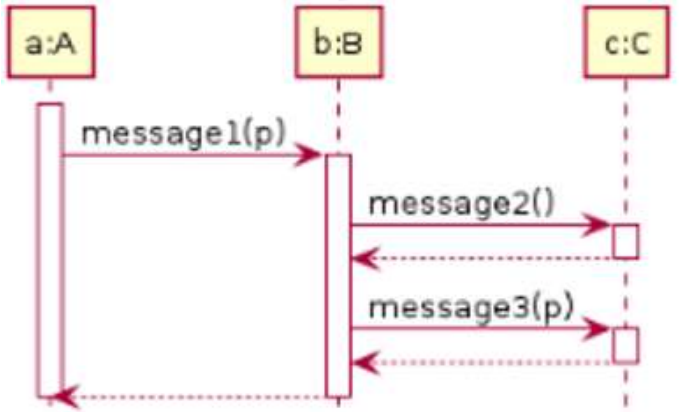
La flèche dans le diagramme de classes correspond à une association unidirectionnelle, et pas à un message : la notion de message n'a aucun sens dans le contexte d'un diagramme de classes.

Echange de messages et code Java

```
class B {

    message1 (p:Type) {


    }
}
class C {
    message2 () {
        ...
    }
    message3 (p:Type) {
        ...
    }
```



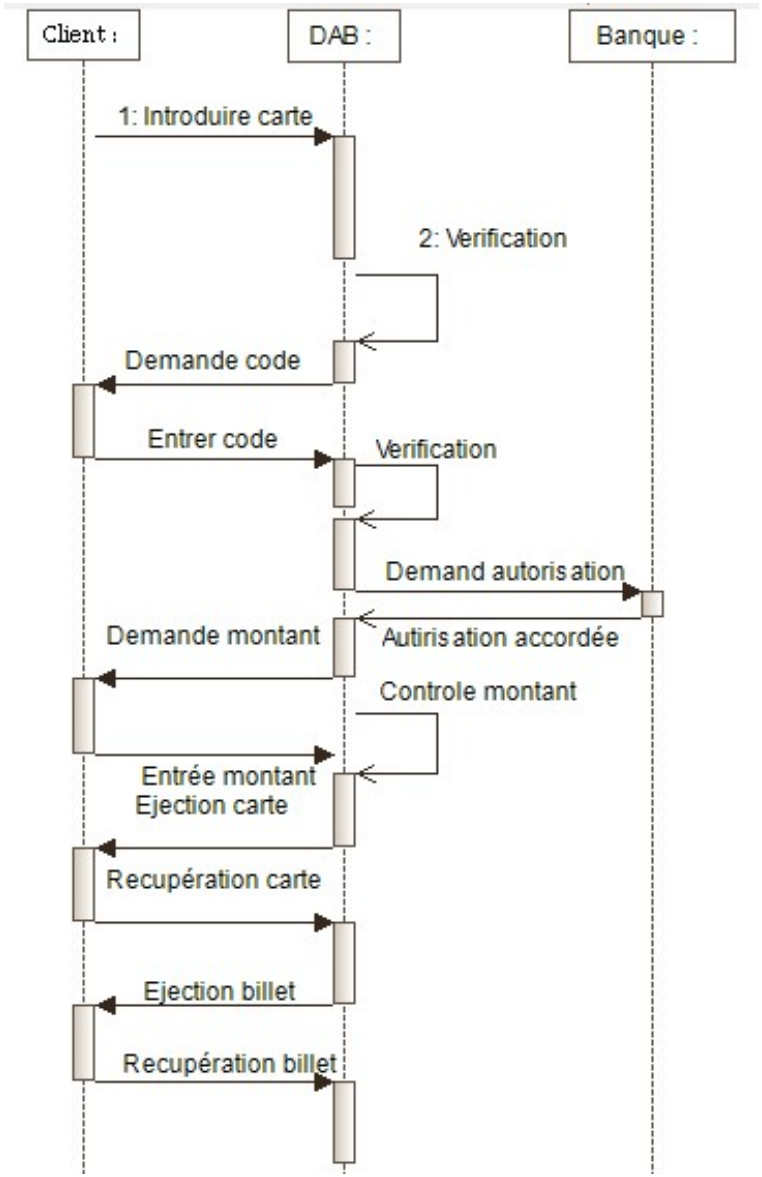

```
class B {
    C c;
    message1 (p:Type) {
        c.message2 ();
        c.message3 (p);
    }
}
class C {
    message2 () {
        ...
    }
    message3 (p:Type) {
        ...
    }
```

Le déroulement normal d'utilisation d'un DAB est le suivant

:

- le client introduit sa carte bancaire
- la machine vérifie alors la validité de la carte et demande le code au client
- si le code est correct, elle envoie une demande d'autorisation de prélèvement au groupement de banques. Ce dernier renvoie le solde autorisé à prélever.
- le distributeur propose alors plusieurs montants à prélever
- le client saisit le montant à retirer
- après contrôle du montant par rapport au solde autorisé, le distributeur demande au client s'il désire un ticket
- Après la réponse du client, la carte est éjectée et récupérée par le client
- les billets sont alors délivrés (ainsi que le ticket)
- le client récupère enfin les billets et son ticket

Travail à faire : Faire le diagramme de séquence correspondant


Client :
DAB :
Banque :
1: Introduire carte
2: Verification
Demande code
Entrer code
Verification
Demand autoris ation
Autiris ation accordée
Demande montant
Controle montant
Entrée montant
Ejection carte
Recupération carte
Ejection billet
Recupération billet