



# *Diagramme de cas d'utilisation*

Dame Samb  
UCAD/FST/DMI

— ■ — ■ — ■ — ■ — ■ — ■ —

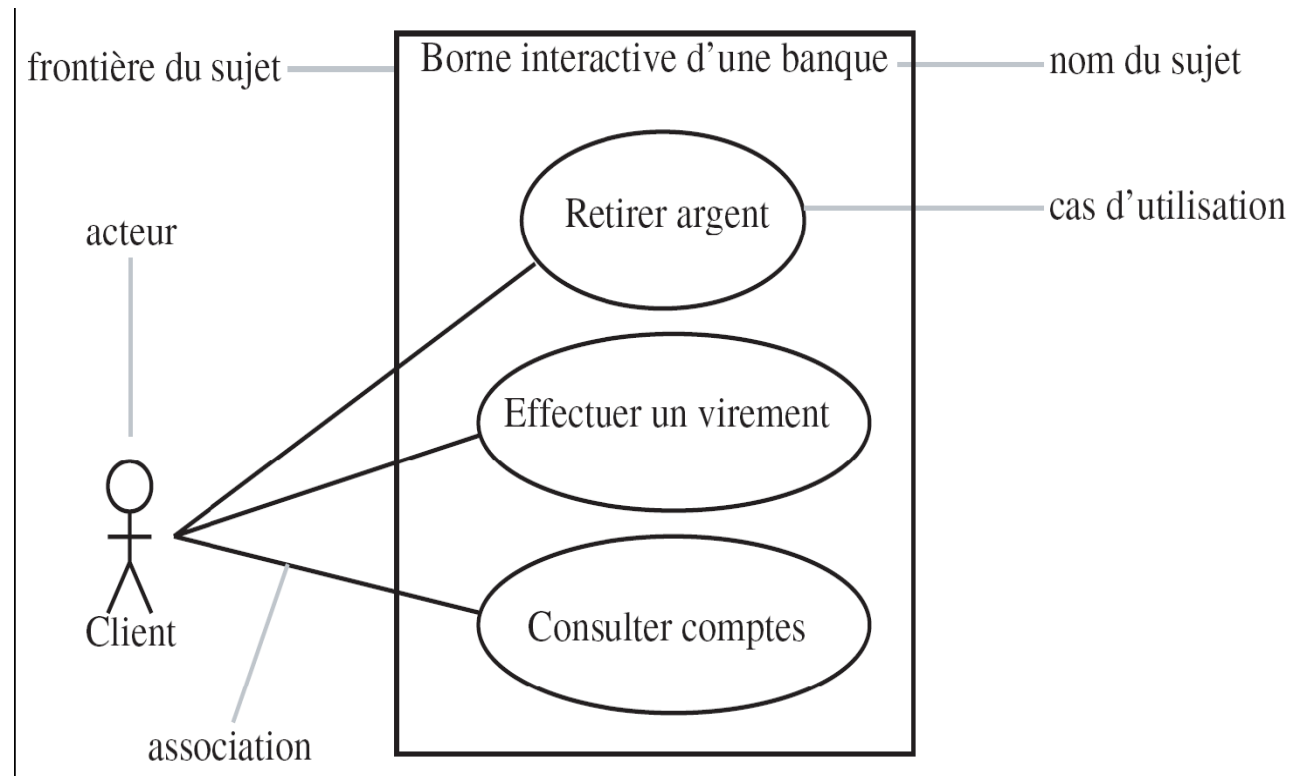
# *Diagramme de cas d'utilisation*

- ✦ Décrit le système étudié en privilégiant le point de vue de l'utilisateur.
- ✦ Permet de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système.
- ✦ S'intéresse à ce que doit faire le système sans spécifier comment il le fait.
- ✦ Constitue un instrument de validation et de test du système en cours et en fin de construction.

# *Diagramme de cas d'utilisation*

- ✦ Un diagramme de cas d'utilisation définit:
- **Le système:** définit l'application informatique
  - **Les acteurs:** entités externes qui agissent sur le système.
  - **Les cas d'utilisation:** services rendus par le système
  - **Les liens entre acteurs et cas d'utilisation**

# *Diagramme de cas d'utilisation*



**Exemple de DCU modélisant une borne d'accès a une banque**



# *DCU: Le système*

✧ La détermination de la frontière du système à développer est une tâche fondamentale.

✧ Un système peut être:

- Un logiciel
- Un processus d'affaires
- Une machine
- Etc.

✧ Exemples:

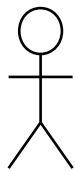
- Gestion d'une bibliothèque
- GAB

# *DCU: Acteur*

- ✧ Un acteur représente une entité externe qui interagit (saisie de données, réception d'information, etc.) avec le système.
- ✧ Il peut être:
  - Un utilisateur qui communique avec le système étudié.
  - Un autre système qui interagit avec le système étudié.
  - Un opérateur chargé de la maintenance
  - Du matériel mécanique, électronique ou autre
  - etc.
- ✧ Un acteur idéalise un rôle joué et non pas une personne physique.

# *DCU: Acteur*

- ✦ Un acteur est représenté soit par l'icône d'un humain soit sous la forme d'un classeur pour les acteurs non humains.
- ✦ Il faut attribuer un nom qui reflète le rôle de l'acteur.
  - Exemple : Une personne qui achète un livre est modélisée par un acteur «acheteur».



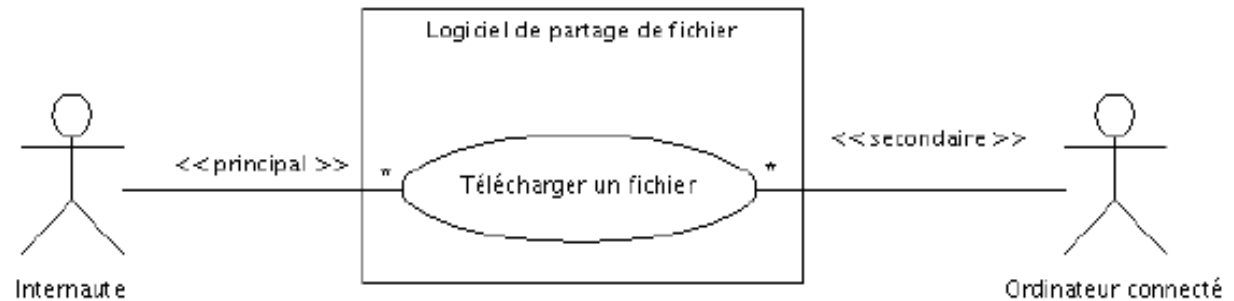
Client

<<acteur>>  
Bibliothécaire

# *DCU: Acteur*

✧ Un acteur peut être:

- **Principal** : Il initie le cas d'utilisation par ses sollicitations et obtient un résultat observable du système.
- **Secondaire** : Il est sollicité pour des informations complémentaires.

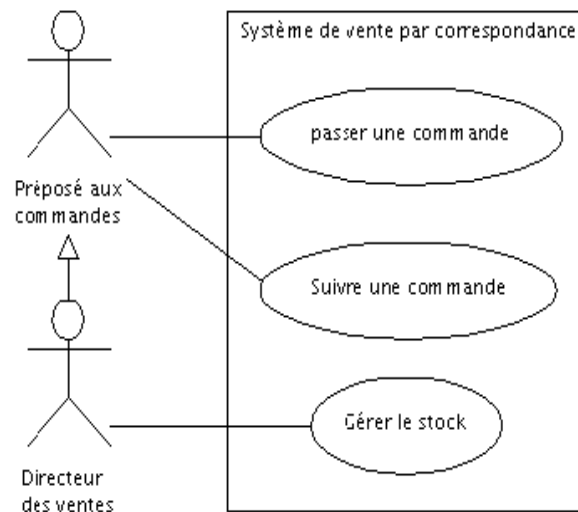




# *DCU: Acteur*

✦ **Relation entre acteurs:** un acteur A peut être une généralisation d'un acteur B. Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai.

✦ **Exemple:**



# *DCU: Acteur*

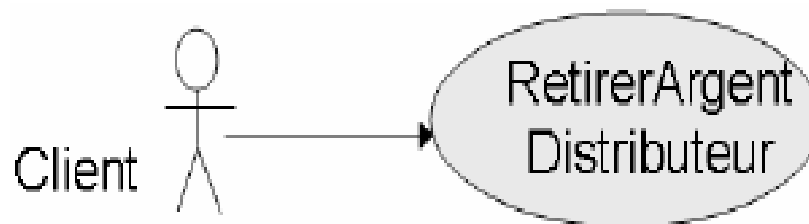
- ✦ Pour trouver les acteurs d'un système, il faut identifier les différents rôles que vont devoir jouer ses utilisateurs.
- ✦ Il faut se poser les questions suivantes:
  - qui utilisera les fonctionnalités principales du système ?
  - qui aura besoin du système pour achever les tâches qui lui sont dédiées ?
  - qui aura besoin de maintenir, administrer et laisser le système fonctionner ?
  - Avec quels systèmes le système interagit ?
  - qui est intéressé par les résultats retournés par le système ?

# *DCU: cas d'utilisation*

- ✦ Fonctionnalité complète telle que perçue par un acteur externe au système.
- ✦ Ensemble d'actions réalisées par le système en réponse à une action d'un acteur.
- ✦ Décrit la fonctionnalité de façon globale
  - Inclut les différentes alternatives
  - Inclut les traitements d'erreurs
  - Inclut les exceptions qui peuvent se produire durant l'exécution du cas d'utilisation
- ✦ L'ensemble des cas d'utilisation décrit les objectifs du système.

# *DCU: cas d'utilisation*

- ✦ Un cas d'utilisation se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif).
- ✦ Un cas d'utilisation est connecté à un acteur par une association de communication.
- ✦ Exemple:





# *DCU: cas d'utilisation*

✦ Les cas d'utilisation peuvent être structurés par les relations suivantes:

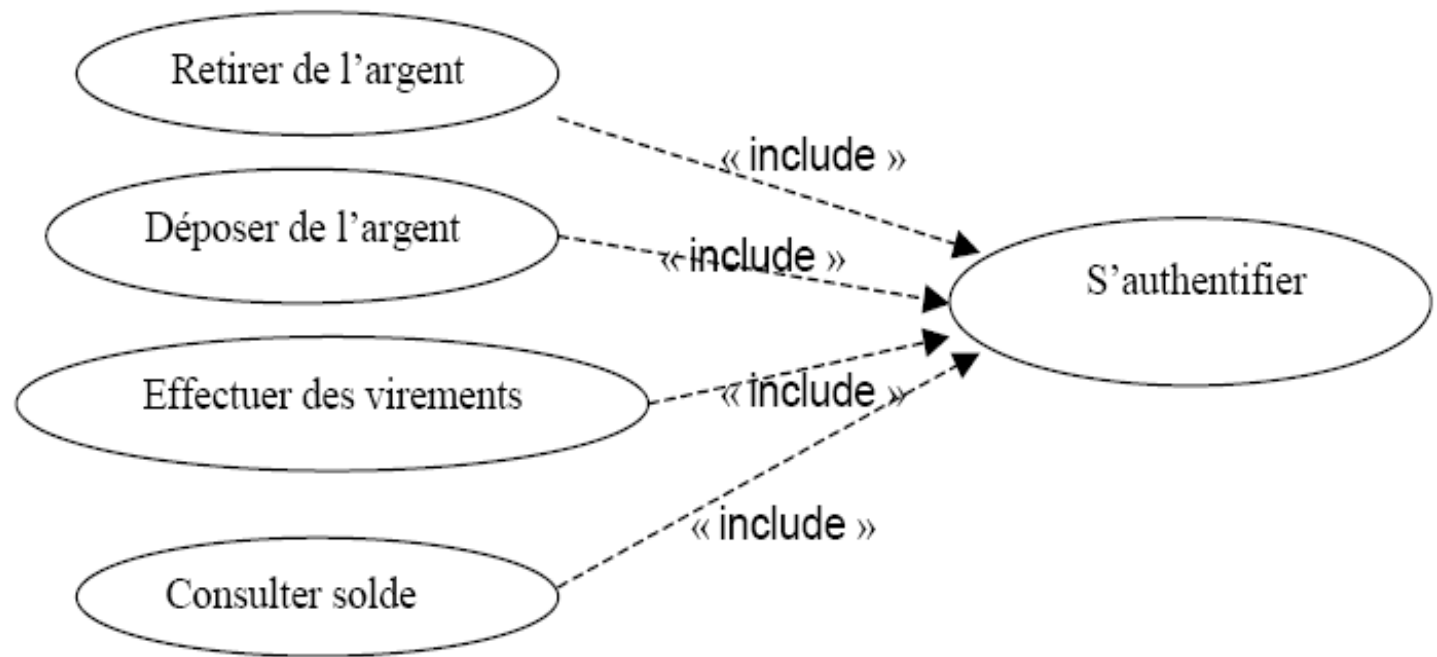
- **Une relation d'inclusion** : formalisée par une flèche pointillée stéréotypée par « **include** »
- **Une relation d'extension** : formalisée par une flèche pointillée stéréotypée par « **extend** »
- **Une relation de généralisation/spécialisation**: formalisée par une flèche à trait plein dont la pointe est un triangle fermé.

# *DCU: cas d'utilisation*

- ✦ **Relation d'inclusion:** permet de factoriser une partie de la description d'un cas d'utilisation qui serait commune à d'autres cas d'utilisation.
- ✦ Un cas d'utilisation A inclut (utilise) un cas d'utilisation B signifie que :
  - une instance de A va engendrer une instance de B et l'exécuter,
  - A dépend de B,
  - B n'existe pas tout seul et A n'existe pas sans B

# *DCU: cas d'utilisation*

## ✧ Exemples de relations d'inclusion



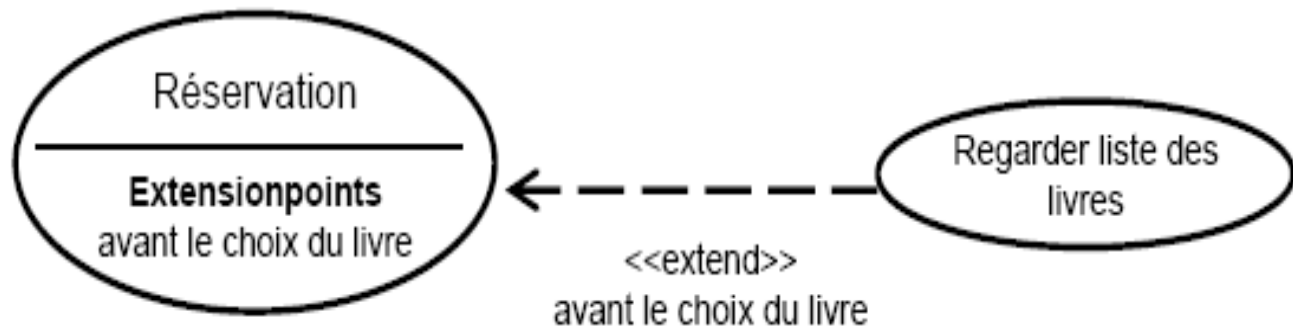
# *DCU: cas d'utilisation*

- ✦ **Relation d'extension:** précise qu'un cas d'utilisation peut dans certains cas augmenter le comportement d'un autre cas d'utilisation.
- ✦ L'extension peut intervenir à un point précis du cas étendu appelé "point d'extension" peut être soumise à une condition.
- ✦ Le cas d'utilisation B étend le cas d'utilisation A signifie que:
  - Une instance de A va engendrer une instance de B et l'exécuter sous certaines conditions.
  - B dépend de A.
  - B n'existe pas tout seul et A existe sans B.



# *DCU: cas d'utilisation*

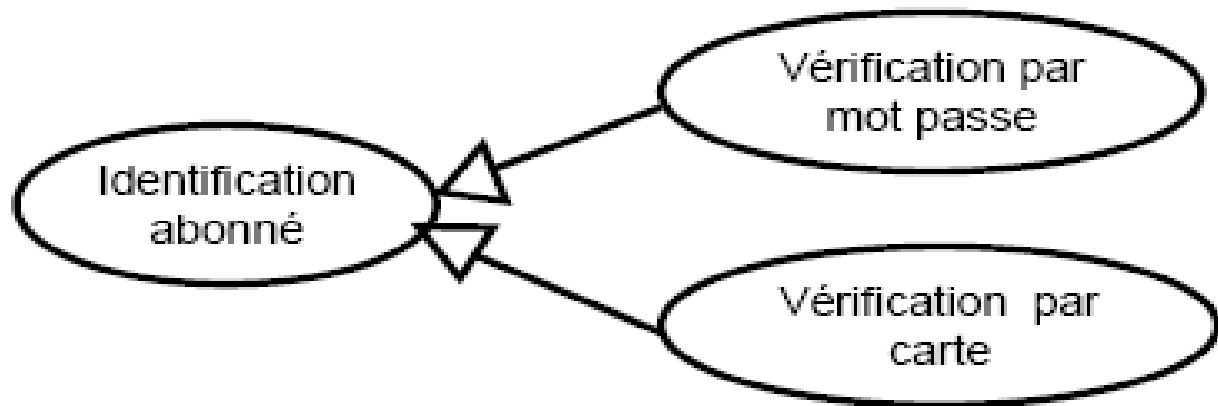
✦ **Exemple de relations d'extension:** le cas d'utilisation "*Regarder la liste des livres*" augmente le cas d'utilisation d'une réservation, avant le choix du livre si l'utilisateur en fait la demande.



# *DCU: cas d'utilisation*

✧ **Relation de généralisation/spécialisation:**  
précise qu'un cas d'utilisation est un cas particulier d'un autre cas d'utilisation.

✧ Exemple:



# *DCU: cas d'utilisation*

- ✦ Pour identifier les cas d'utilisation, il faut se placer du point de vue de chaque acteur et déterminer comment et surtout pourquoi il se sert du système.
- ✦ Pour chaque acteur identifié, on pose les questions suivantes :
  - Quelles sont les fonctions principales que l'acteur exige du système ? (trouver les cas d'utilisation de base).
  - Si le cas d'utilisation de base est complexe, on peut se questionner sur les sous-fonctionnalités. Cette question permet d'identifier les cas d'utilisation associés à ceux de base par les relations «include» ou «extend»

# *DCU: cas d'utilisation*

✦ Pour trouver les cas d'utilisation sous-jacents aux cas d'utilisation de base (les acteurs ne sont pas concernés directement), on se pose les questions suivantes :

- Quelles sont les entrées et les sorties dont le système a besoin ?
- Où partent ces entrées et ces sorties ?



# *DCU: cas d'utilisation*

✦ Un cas d'utilisation est incomplet sans une description textuelle précisant.

- Le nom du cas
- L'objectif du cas
- Les acteurs principaux
- Les acteurs secondaires
- Les préconditions
- Les postconditions
- Les scénarios nominaux, alternatifs et d'exception
- Les informations échangées entre acteurs et système.

# *DCU: Exemple*

✧ Nom: réserver livre

✧ Objectif: réservation d'un livre dans une bibliothèque

✧ Acteur principal: client

✧ Preconditions

- Le client doit être inscrit à la bibliothèque
- Le client ne doit pas avoir atteint le nombre maximum de réservation
- Un exemplaire du livre doit être enregistré

✧ Postconditions (Si l'opération s'est bien déroulée)

- Le client a une réservation supplémentaire
- Le nombre d'exemplaires disponibles du livre est décrémenté de 1

# *DCU: Exemple*

## ✧ Scénario nominal

Le client se présente devant un terminal:

- (1) Le système affiche un message d'accueil.
- (2) Le client choisit l'opération réservation parmi les différentes opérations proposées.
- (3) Le système lui demande de s'authentifier.
- (4) Le client donne son identification (nom, mot de passe).
- (5) Le système lui demande de choisir un livre.
- (6) Le client précise le livre qu'il désire.
- (7) Le système lui précise si un exemplaire du livre lui est réservé.

# *DCU: Exemple*

## ✦ Scénarios alternatifs

- **variante 1:** en (6) le client demande à connaître les livres présents
- **variante 2:** en (4) le client n'est pas reconnu, dans ce cas, tant qu'il n'est pas reconnu, on lui redemande de s'authentifier
- **variante 3:** en (4) le client n'a plus le droit de réserver
- **variante 4:** en (7) le livre n'est pas libre
- **variante 5:** en (4) le client est reconnu, mais le mot de passe est incorrect, il peut avoir 5 essais, par la suite le client sera interdit pendant 1 heure.