

---

# ***UML*** **(Cas d'utilisation)**

Unified Modeling Language



# Sommaire

---

- Introduction
- Objectifs
- Cas d'utilisation
- Acteur
- Diagramme de cas d'utilisation
- Dépendances entre cas d'utilisation
- Scénario

# Cas d'utilisation (Use Cases)

---

- Début des années 90, Ivar Jacobson invente la méthode **OOSE** (Object-Oriented Software Engineering) chez Ericsson
- Adaptation de la méthode au BPR (Business Process Reengineering)
- 1996, Jacobson rejoint Rumbaugh et Booch donnant ainsi naissance à **UML 0.9**
- **Cas d'utilisation** est la traduction française de **Use case**

# Cas d'utilisation (Use Cases)

---

## Objectifs

- Définir les **besoins** fonctionnels du système  
Les cas d'utilisation ont pour principal objectif la capture des fonctionnalités couvertes par le système
- Définir le **périmètre** fonctionnel du système  
Les cas d'utilisation permettent de définir les frontières du système avec son environnement
- Définir le **dialogue** entre l'utilisateur et le système  
Les cas d'utilisation recensent comment l'utilisateur interagit avec le système

# Cas d'utilisation (Use Cases)

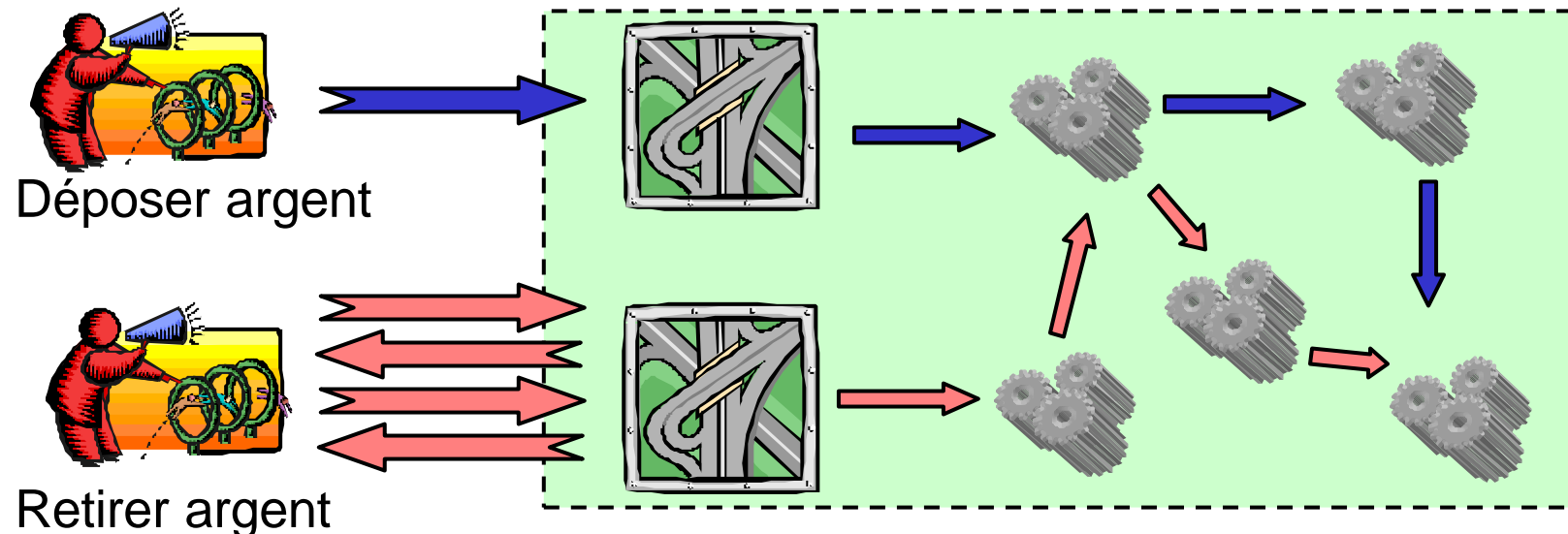
---

## Objectifs (suite)

- Etablir les **scénarios fonctionnels** qui seront utilisés pour la recette du système  
Les cas d'utilisation recensent et décrivent les principales fonctionnalités attendues du système
- Servir de **support de référence** tout au long des phases de développement du système  
Les cas d'utilisation seront consultés et référencés tout au long du processus de développement du système

## Cas d'utilisation

- Une interaction en provenance de l'extérieur déclenche un flot de contrôle (séquence d'activités) au sein du système
- Pendant l'exécution de ce flot de contrôle, plusieurs interactions avec son initiateur peuvent avoir lieu
- Chaque flot de contrôle correspond à une fonctionnalité ou un processus fonctionnel attendu du système



## Cas d'utilisation (Définition)

---

Un cas d'utilisation est une **séquence d'activités** ou d'actions organisées en **étapes distinctes**, et qu'un système effectue en réponse à une **sollicitation extérieure**

- Le cas d'utilisation est déclenché par un événement extérieur au système appelé **événement initiateur**
- Le cas d'utilisation possède un **nom** : celui de la fonctionnalité du système qu'il prend en charge
- Le cas d'utilisation met en œuvre un **dialogue** entre le système et l'entité à l'origine de l'événement initiateur

## Cas d'utilisation (Notation)

---

- Un cas d'utilisation est représenté par un ovale
- Le nom du cas d'utilisation apparaît à l'intérieur de l'ovale. Il est composé :
  - d'un nom optionnel de paquetage
  - du nom de la fonctionnalité qu'il prend en charge

### Notation



DAB::Retirer argent

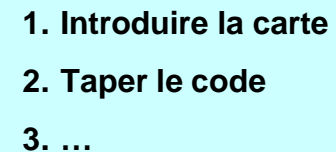
The diagram shows a rectangular box containing an oval. Inside the oval is the text 'DAB::Retirer argent'. This represents a use case in UML notation, where the text inside the oval is the use case name, which can include an optional package name followed by a double colon and the use case name itself.



# Description d'un cas d'utilisation

---

- Titre (commence par un verbe)
- Objectif (descriptif court : une phrase si possible)
- Acteurs
- Pré-conditions
  - conditions nécessaires pour que le cas d'utilisation s'exécute
- Scénario nominal
  - description pas à pas textuelle
  - chaque étape du cas d'utilisation est numérotée
- Exceptions
- Post-conditions
  - état d'une partie du système après l'exécution du cas d'utilisation
- Fréquence & performance requises



1. Introduire la carte
2. Taper le code
3. ...

# Comment déterminer les cas d'utilisation

---

**Se poser les questions suivantes :**

- Quelles sont les grandes fonctionnalités attendues du système ?
- Le système doit-il informer une personne ou un dispositif extérieur lorsque son état interne est modifié ?
- Le système doit-il être informé d'événements extérieurs se produisant dans son entourage ?
- Le système stocke-t-il des informations ? Comment sont-elles stockées, mises à jour, détruites ?
- ...

## Acteur (Définition)

---

Un acteur définit un **rôle** qu'une **entité extérieure** assume lors de son **interaction** avec le système

- L'acteur est à **l'origine** des événements initiateurs reçus par le système
- L'acteur **dialogue** par la suite avec le cas d'utilisation dont il est l'initiateur
- L'acteur possède un **nom** : celui du **rôle** qu'il joue lors de son interaction avec le système
- L'acteur **n'est pas forcément humain**. Il peut s'agir :
  - d'un autre système
  - d'un équipement

# Acteur

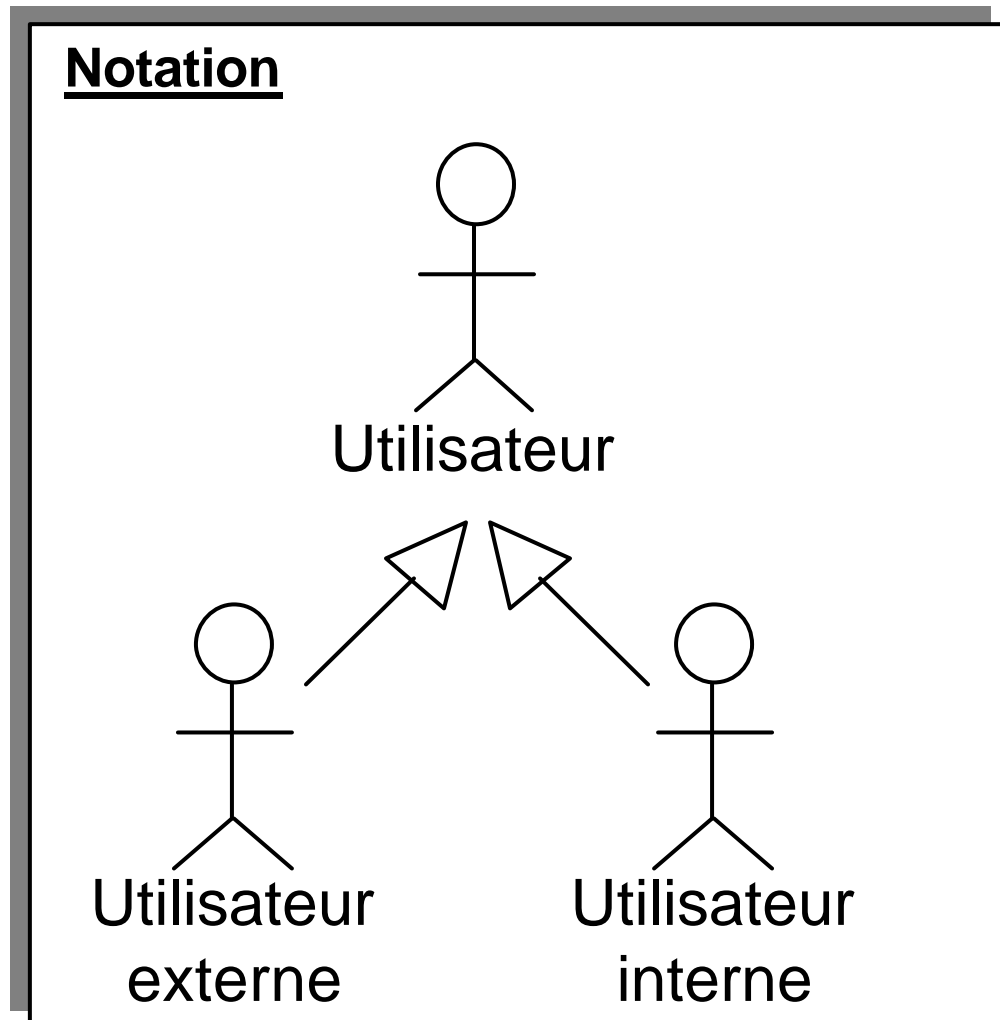
---

- « Pierre utilise le système pour gérer son agenda »
- « Philippe utilise aussi le système pour gérer son agenda. Mais Philippe est aussi autorisé à administrer le système »
- Pierre n'est pas un acteur du système, Philippe n'est pas un acteur du système
- Le rôle « Utilisateur » est un acteur du système
- Le rôle « Administrateur » est un acteur du système

**Ne pas confondre** personne physique et rôle  
Une personne peut très bien assumer plusieurs rôles  
et réciproquement

## Acteur (Notation)

- Un acteur est représenté par un **petit personnage**
- Le nom de l'acteur apparaît sous le petit personnage
- On peut définir des **catégories d'acteurs** plus générales  
ou  
au contraire spécialiser un type d'acteur



# Comment déterminer les acteurs

---

**Se poser les questions suivantes :**

- Qui installe le système ?
- Qui utilise le système ?
- Qui démarre le système ?
- Qui maintient le système ?
- Quels sont les autres systèmes qui utilisent le système ?
- Qui fournit de l'information au système ?
- Qui récupère de l'information à partir du système ?
- ...

# Diagramme de cas d'utilisation (Définition)

---

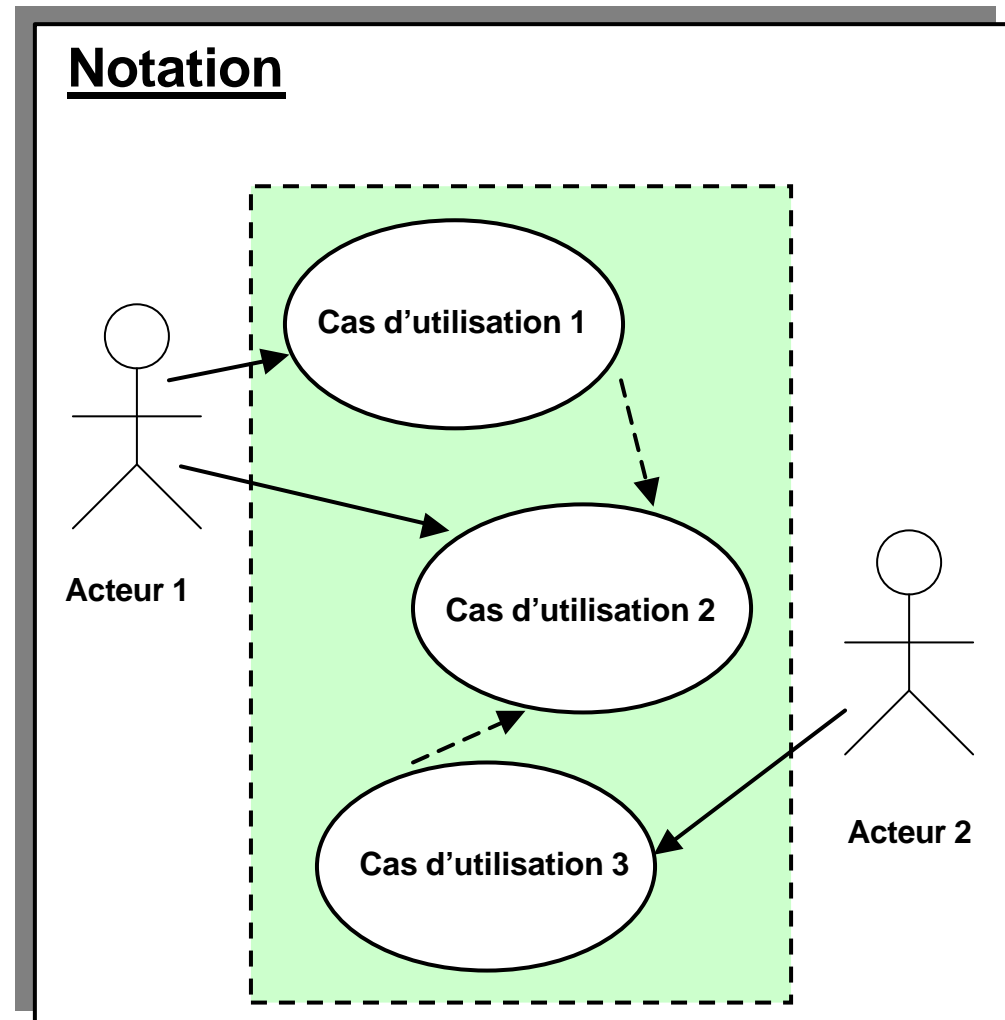
Le diagramme de cas d'utilisation est une **représentation contextuelle de haut niveau** du système modélisé

- Permet de définir de manière précise les **frontières** du système à modéliser
- Montre les **interactions** entre le système et son environnement extérieur
- Montre les **dépendances** existant entre les cas d'utilisation

# Diagramme de cas d'utilisation (Notation)

Le diagramme de cas d'utilisation met en scène :

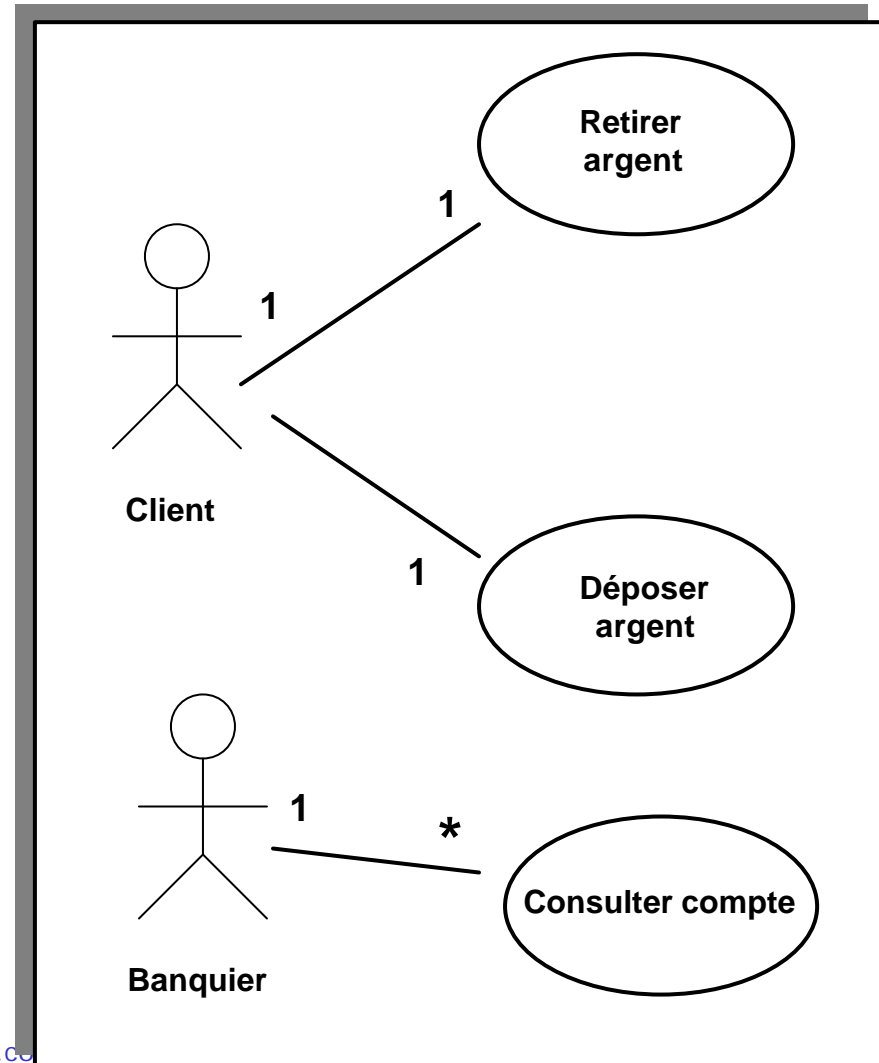
- les acteurs
- les cas d'utilisation
- les interactions entre acteurs et cas d'utilisation
- les dépendances entre cas d'utilisation





# Interaction entre acteur et cas d'utilisation

- Elle est représentée par une **association** sous la forme d'un lien éventuellement orienté dans le sens de l'interaction
- **Une seule** association est utilisée pour représenter l'ensemble des événements échangés
- L'association peut comporter des **cardinalités**



# Dépendances entre cas d'utilisation

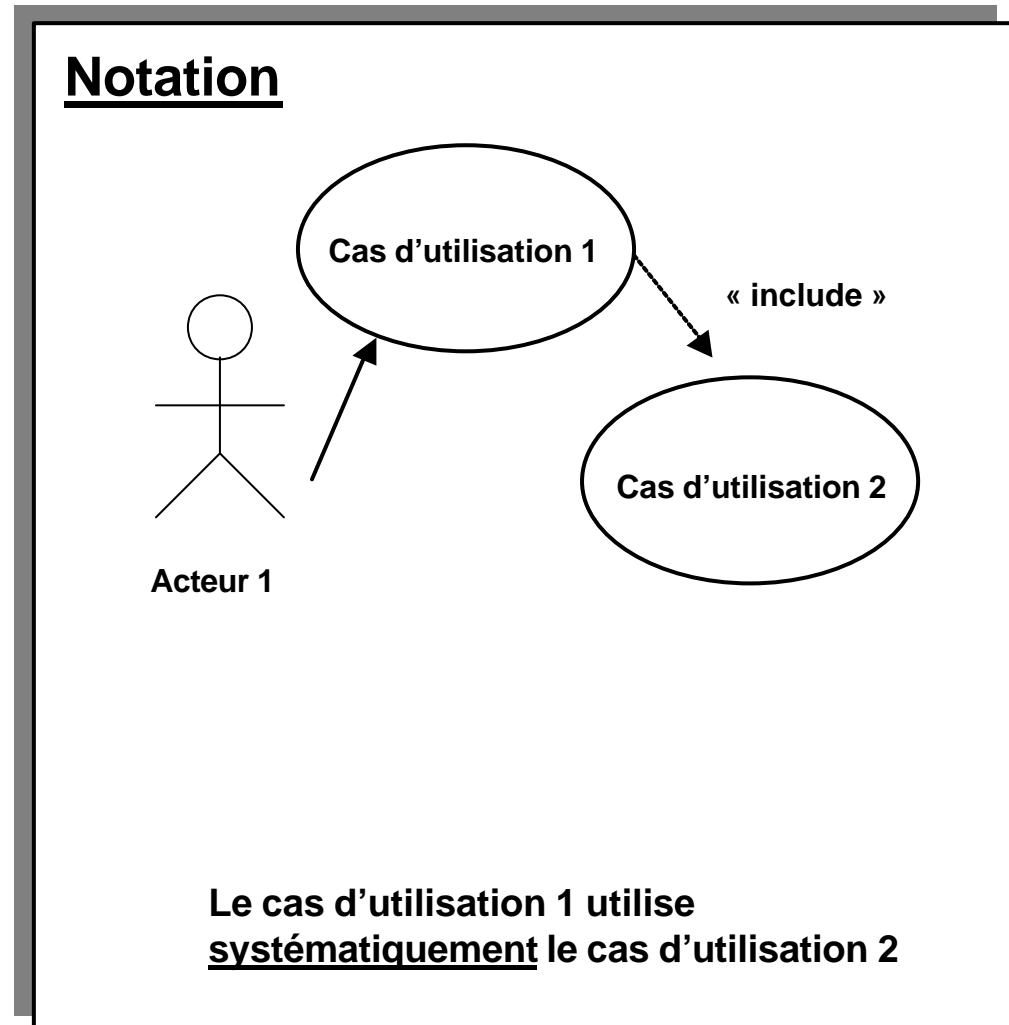
---

Il existe 3 types de dépendances entre use cases :

- **Les dépendances d'utilisation**  
Mise en facteur de séquences d'événement communes
- **Les dépendances d'extension**  
Externalisation de séquences d'événement exceptionnelles
- **Les dépendances de généralisation**  
Généralisation / spécialisation de cas d'utilisation

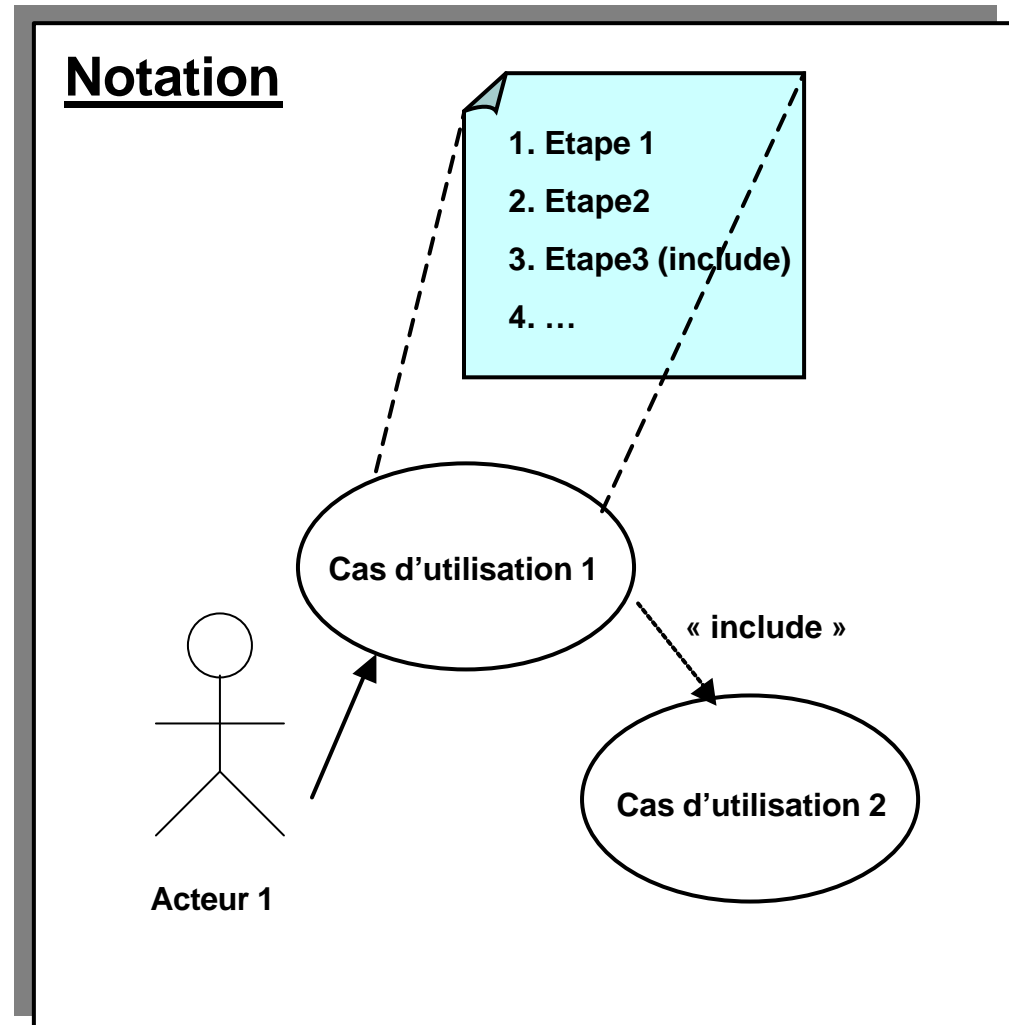
# Dépendance d'utilisation

- Indique qu'un cas d'utilisation utilise **systematiquement** et **intégralement** une séquence d'activités décrite dans un autre cas d'utilisation
- Est représentée par une flèche pointillée étiquetée « **include** », pointant vers le cas d'utilisation utilisé



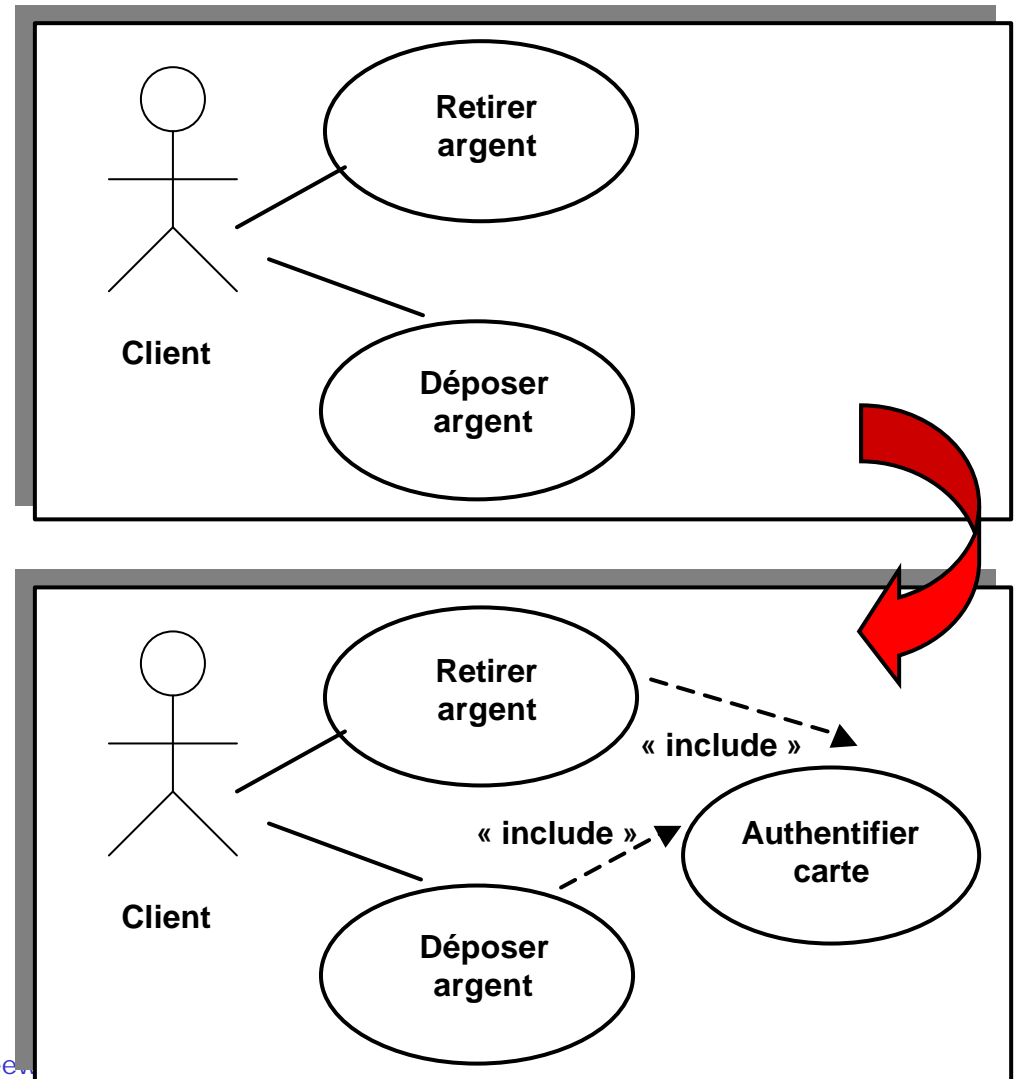
# Dépendance d'utilisation

- Le cas d'utilisation source de la dépendance d'utilisation contient une **référence** vers le cas d'utilisation inclus
- L'inclusion est signalée par le texte « **(include)** » à l'étape correspondante



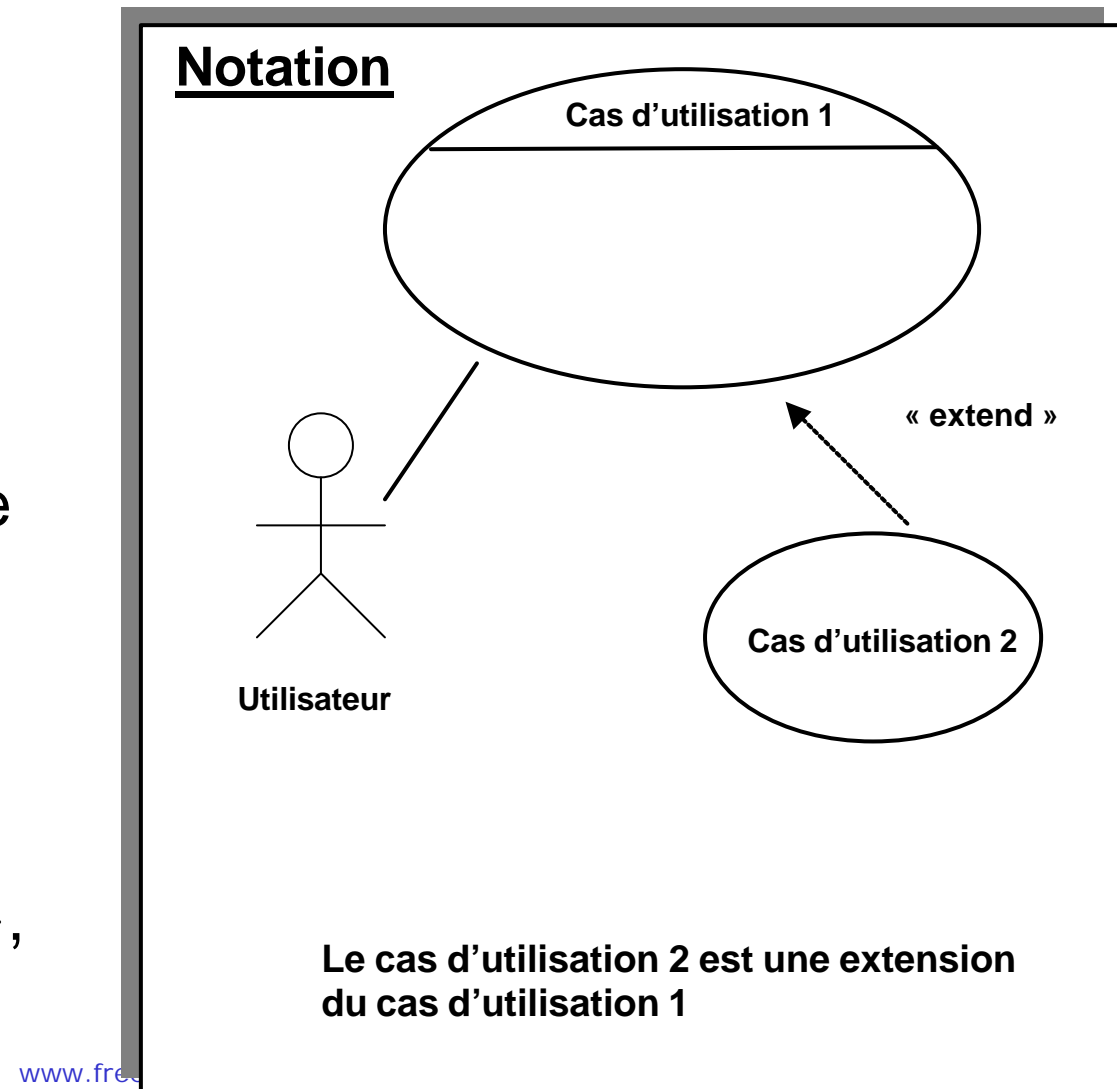
# Dépendance d'utilisation

- Permet de **décomposer** un cas d'utilisation complexe en cas d'utilisation plus simples
- Permet de **factoriser** des comportements utiles à plusieurs cas d'utilisation



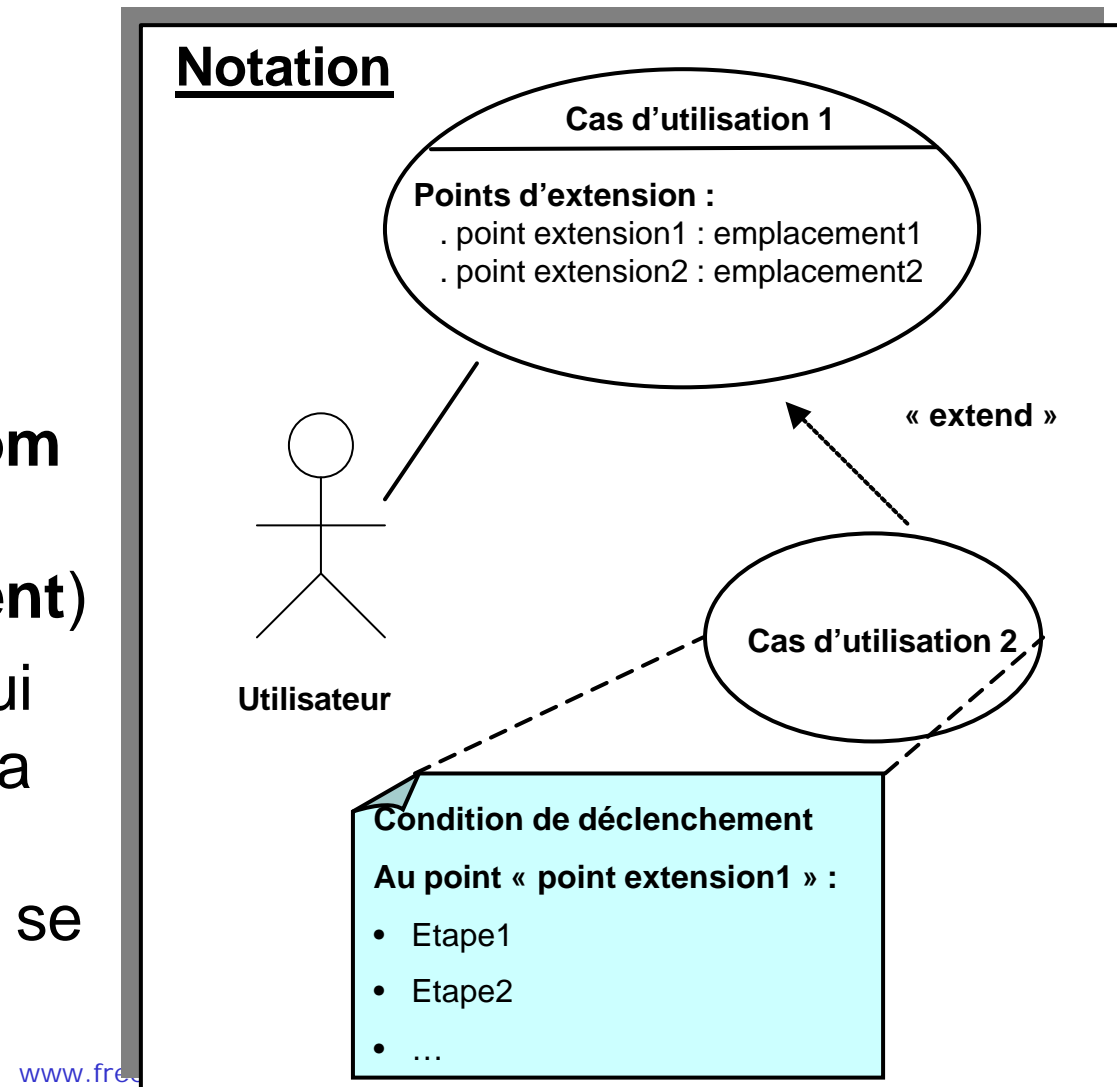
# Dépendance d'extension

- Indique qu'un cas d'utilisation utilise **facultativement** ou **sous certaines conditions** une séquence d'activités décrite dans un autre cas d'utilisation
- Est représentée par une flèche pointillée étiquetée « **extend** », pointant vers le cas d'utilisation étendu



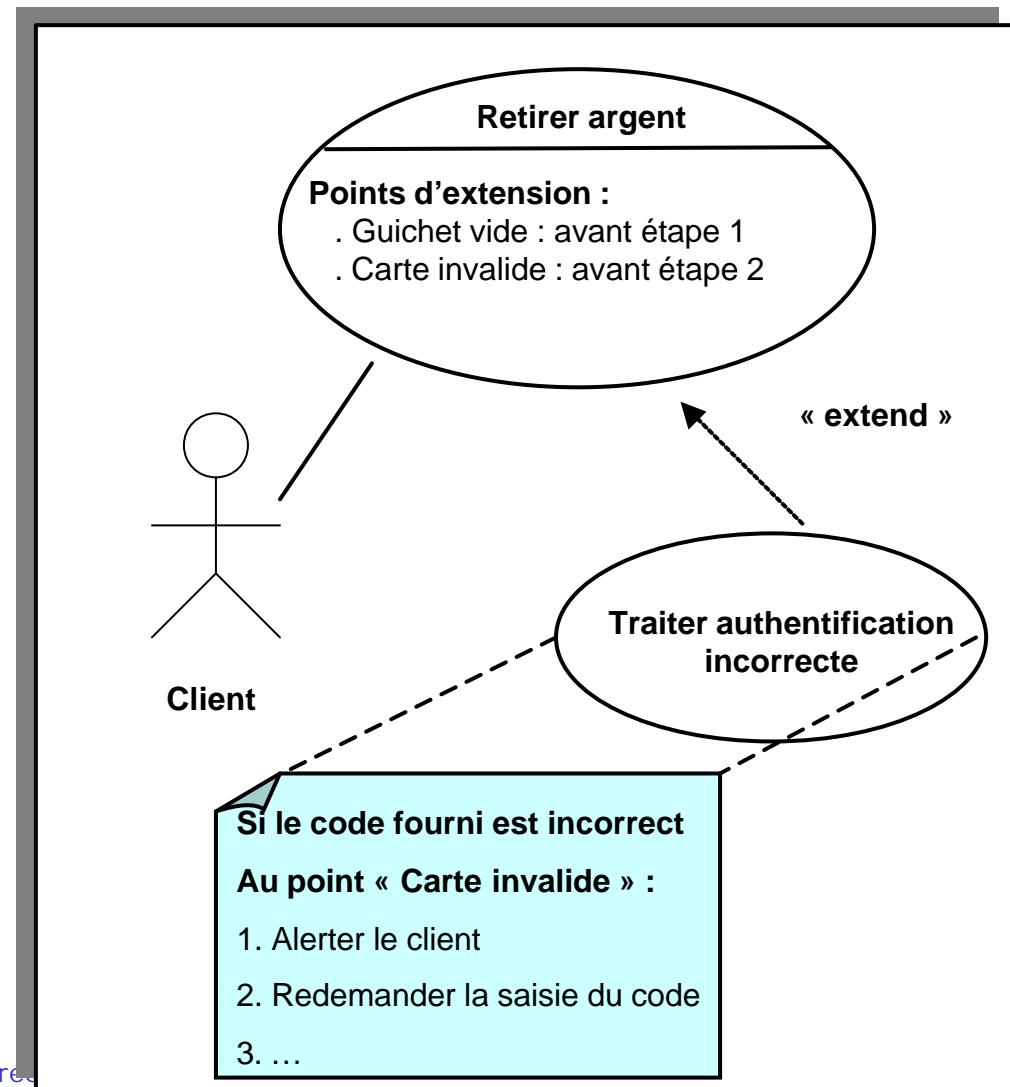
# Dépendance d'extension

- Le cas d'utilisation étendu contient une liste de **points d'extension**
- Un point d'extension est composé d'un **nom** suivi d'un numéro d'étape (**emplacement**)
- Le cas d'utilisation qui étend indique dans sa description sous quelles **conditions** il se déclenche



# Dépendance d'extension

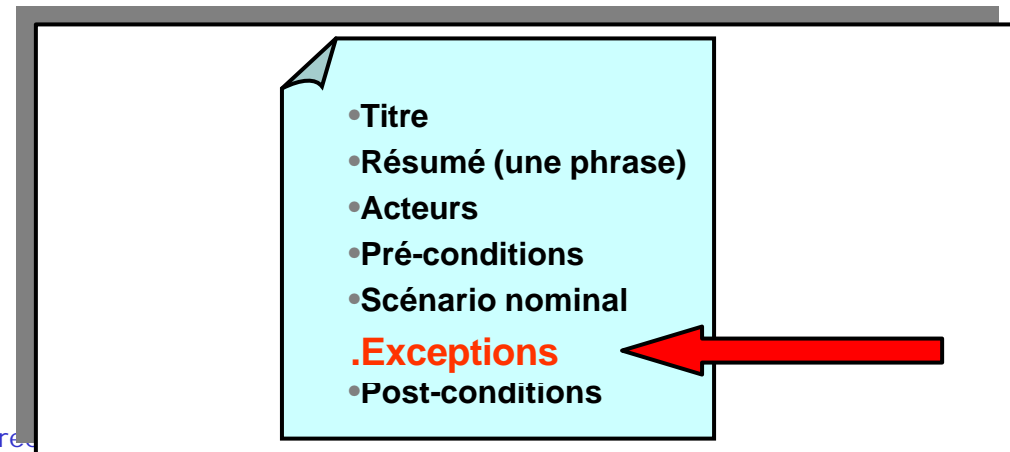
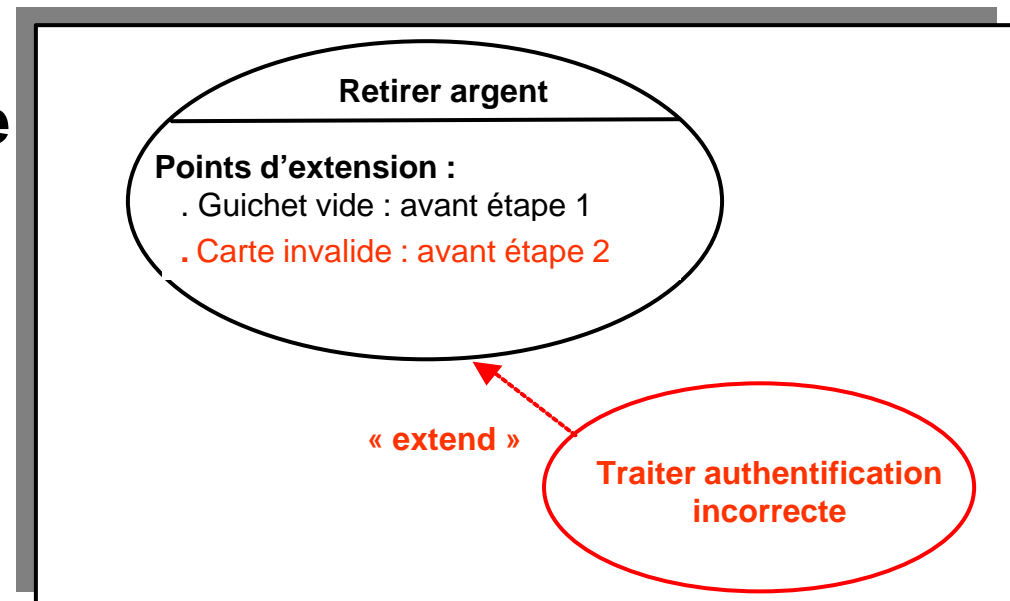
- Permet **d'ajouter** à un cas d'utilisation un comportement exceptionnel complexe (hors cas nominal)
- **Détermine les conditions** d'application d'un comportement alternatif au cas nominal
- **Déporte** la description d'une exception significative dans un cas d'utilisation





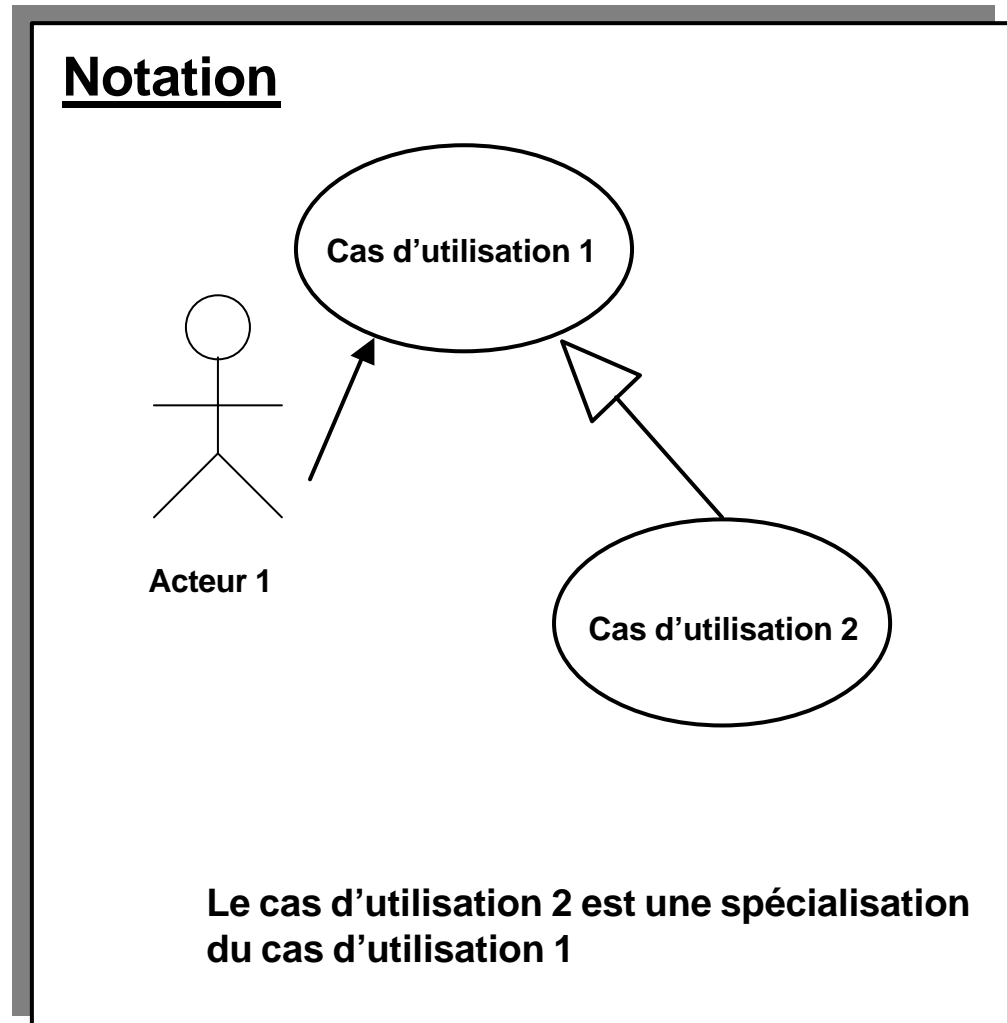
# Dépendance d'extension

- Un **comportement exceptionnel complexe** d'un cas d'utilisation doit être déporté dans un cas d'utilisation avec une dépendance « extend »
- Un **comportement exceptionnel simple** doit être explicité dans le paragraphe « Exceptions » de la description du cas d'utilisation



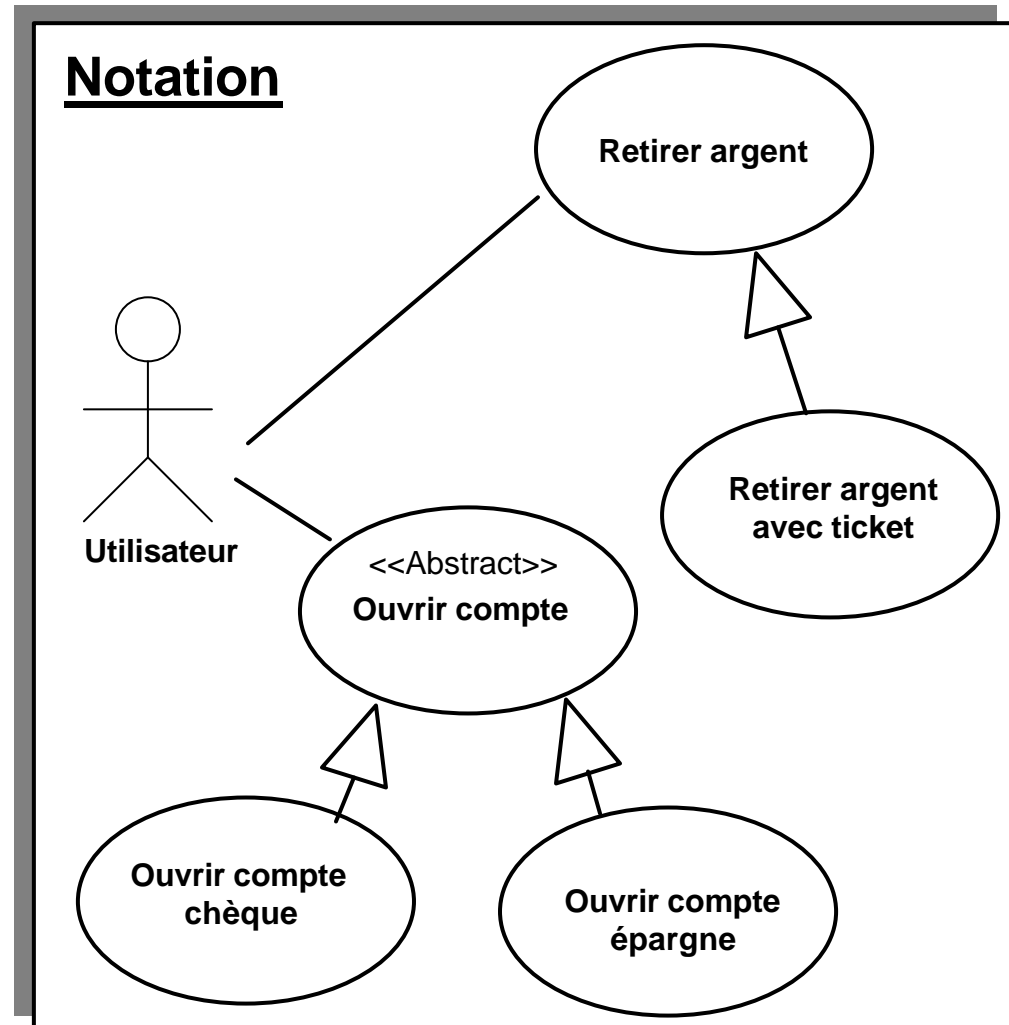
# Dépendance de généralisation

- Indique qu'un cas d'utilisation est une **spécialisation** d'un autre cas d'utilisation
- Est représentée par une flèche « **d'héritage** » pointant du cas d'utilisation spécialisé vers le cas d'utilisation le plus général



# Dépendance de généralisation

- Permet de **factoriser** un comportement commun à un ensemble de cas d'utilisation **proches**
- Le cas d'utilisation le plus général est dit **abstrait** si seuls les cas d'utilisation spécialisés sont exécutables



# Scénario

---

Le scénario est au cas d'utilisation  
ce que l'objet est à la classe

- Un cas d'utilisation propose un comportement nominal (**scénario nominal**)
- Un cas d'utilisation propose aussi un ou plusieurs comportements alternatifs (**scénario alternatif**) chacun représentant un cheminement particulier dans le cas d'utilisation
- Un cas d'utilisation décrit aussi des **situations exceptionnelles**
- L'idéal est de créer suffisamment de scénarios couvrant l'essentiel d'un cas d'utilisation
  - Il est inutile d'identifier tous les scénarios possibles

## Exemple de scénario (I)

---

- CAS D'UTILISATION # 2
  - Nom : prendre une commande
  - Contexte : système de prise de commande par téléphone pour un club de loisir...
  - Objectif : le membre, après avoir consulté son catalogue, téléphone pour commander
  - Scénario :
    - # 1 : le télévendeur identifie l'internaute (U
    - # 2 : le membre choisit les articles commandés
    - # 3 : le système accepte la commande
    - # 4 : le système envoie la commande à la logistique

## Exemple de scénario (II)

---

- Le scénario décrit est le scénario « idéal » !
- Que se passe-t-il si....
  - L'internaute n'est pas un membre du club ?
  - Il n'y a pas assez d'article en stock ?
  - Il n'y a plus d'article en stock ?
  - L'internaute a eu un incident de paiement par le passé ?
  - ...

## Exemple de scénario (III)

---

- CAS D'UTILISATION # 2
  - Nom : prendre une commande
  - Contexte : système de prise de commande par téléphone pour un club de loisir...
  - Objectif : le membre, après avoir consulté le catalogue, téléphone pour commander
  - Scénario :
    - # 1 : le télévendeur identifie l'internaute
    - # 2 : le membre choisit les articles commandés
    - # 3 : le système accepte la commande
    - # 4 : le système envoie la commande à la logistique
  - Extensions :
    - # 1a : l'internaute n'est pas membre...
    - # 2a : un des articles n'est pas en nombre suffisant : proposer une réduction du nombre d'article ?
    - # 3a : l'internaute a eu un incident de paiement : est-il VIP ?

## Exemple de scénario (IV)

- CAS D'UTILISATION # 2
  - Nom : prendre une commande
  - Contexte : système de prise de commande par téléphone pour un club de loisir...
  - Objectif : le membre, après avoir consulté le catalogue, téléphone pour commander
  - Scénario :
    - # 1 : le télévendeur identifie l'internaute (CU # 21)
    - # 2 : le membre choisit les articles commandés
    - # 3 : le système accepte la commande
    - # 4 : le système envoie la commande à la logistique
  - Extensions :
    - # 1a : l'internaute n'est pas membre...
    - # 2a : un des articles n'est pas en nombre suffisant : proposer une réduction du nombre d'article ?
    - # 3a : l'internaute a eu un incident de paiement : est-il VIP ?

On ne s'intéresse ici qu'au résultat du CU 21



# Recommandations

---

- Ne pas confondre cas d'utilisation et scénario
  - Chaque cas d'utilisation correspond à un objectif d'un acteur vis à vis du système
  - Un scénario décrit le déroulement d'un cas d'utilisation
  - Ne pas oublier les scénarios correspondant aux principaux cas d'échec du cas d'utilisation (mot de passe invalide, carte de crédit invalide, article du catalogue indisponible ou en nombre insuffisant...)
- Ne décrire que les principaux cas d'utilisation du système  
Un cas d'utilisation doit contenir une quantité « appréciable et tangible » de travail
- Ne pas décomposer trop finement les cas d'utilisation  
Les dépendances d'utilisation et d'extension ne doivent être utilisées que pour des comportements significatifs du système

## Recommandations (suite)

---

- Lorsque le système est complexe, il n'est pas anormal d'avoir de nombreux cas d'utilisation. Il faut alors les classer et les rassembler dans des bibliothèques de cas d'utilisation
  - ➔ on définit pour cela des paquetages (notion abordée dans ce cours)
- Lorsque le système est complexe, les cas d'utilisation peuvent s'emboîter les uns dans les autres. Veiller à ne pas mélanger un cas d'utilisation « générale » (exemple : prendre une commande) avec des cas d'utilisation plus fins (exemple : choisir son mode de paiement)
  - ➔ Utiliser les relations de dépendance entre cas (notamment les include)

## Recommandations (suite)

---

- Compléter le diagramme de cas d'utilisation par un diagramme de séquence ou un diagramme de collaboration
- Faire figurer uniquement les acteurs en interaction avec les cas d'utilisation
- Un diagramme de cas d'utilisation n'est pas un diagramme de flots de données

Les dépendances entre cas d'utilisation ne traduisent pas un échange de données ou un flot de contrôle (diagramme d'activité)

## Quelques références utiles...

---

- <http://www.usecases.org/>
- Writing effective Use Cases / Alistair CockBurn / éditions Addison Wesley / 2000
- UML en action / Roques & Vallée / éditions Eyrolles / 2000