

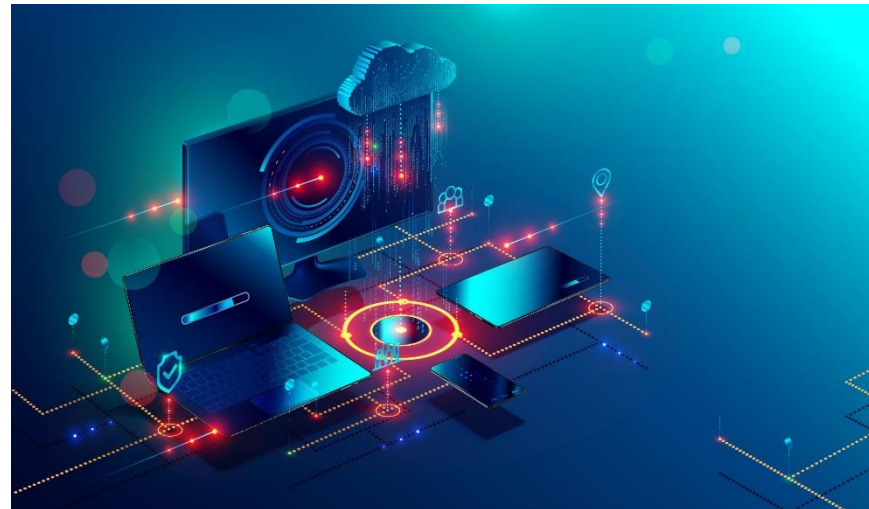
# Analyse Orientée Objet

I. Introduction

II. Approche objet et système d'information.

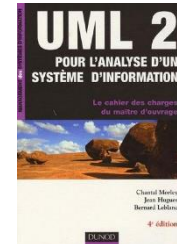
III. Principes Objet

IV. UML



# Analyse Orientée Objet

- Objectifs
- Organisation du cours
- Supports
- Evaluation



# Analyse Orientée Objet

## ➤ Objectifs:



- *Introduire la modélisation orientée objet.*
- *Acquérir d'une manière pédagogique et rigoureuse les bases du concept objet.*
- *Introduire le langage UML.*
- *Présenter des éléments méthodologiques d'utilisation des différents types de diagrammes dans un processus de développement.*

# Analyse Orientée Objet

## ➤ Organisation du cours:

*Bloc 2 toutes les options:*

*Théorie : 15 h. 2h par semaine*

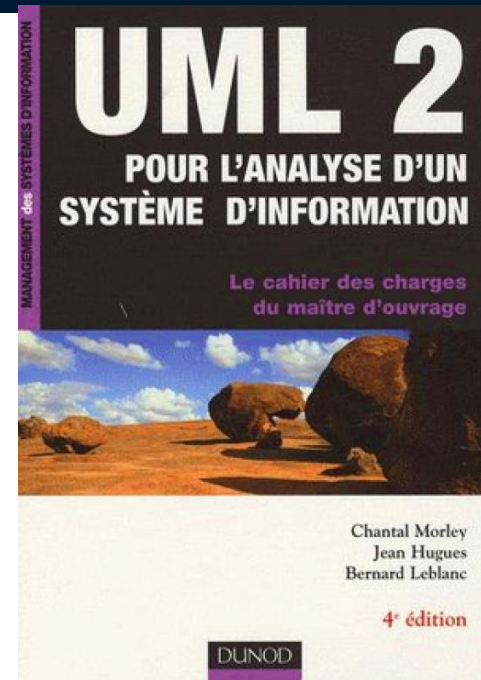
*Laboratoire: 30h. 2h par semaine en un quadrimestre.*



# Analyse Orientée Objet

## ➤ Supports:

- *UML 2 Pour l'analyse d'un système d'information.*  
Chantal Morley  
Jean Hugues  
Bernard Leblanc
- *Documents dans mon centre de ressources*  
[souad.serrhini@hepl.be](mailto:souad.serrhini@hepl.be)



# Analyse Orientée Objet

## ➤ Evaluation

### ➤ *Théorie et Laboratoire:* *Evaluation continue*

*Interrogations écrites/orales*  
*Devoirs/préparations*  
*Participation au cours*

*Examen en septembre.*



➡ 100%

# Analyse Orientée Objet

I. Introduction

II. Approche objet et systèmes d'information.

III. Les principes et concepts objets.

IV. UML.

# III. Les principes et concepts objets



Les trois gourous « *the Amigos* » se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs.

Naissance d'UML (Unified Modeling Language)  
Ou Langage de Modélisation Unifié.

**UML est bien un langage et non une méthode**



# IV. UML

1. Historique.
2. Standard de modélisation UML.
3. Les diagrammes UML.

# IV. UML

1. Historique.
2. Standard de modélisation UML.
3. Les diagrammes UML.

# IV. UML

## 1. Historique

- Juin 1996 UML 0.9
- Janvier 1997 UML 1.0
- Fin 1997 UML 1.1 par IBM, Microsoft, Unisys, Rational,...  
soumis à l'OMG



**OMG:** *Object Management Group.*

*Association américaine à but non lucratif créé en 1989 dont l'objectif est de standardiser et promouvoir le modèle objet.*

# IV. UML

## 1. Historique.

- Début 1998 UML 1.2
- En 1998 UML 1.3
- En 2001 UML 1.4
- En 2003 UML 1.5
- En 2005 UML 2.0
- En 2008 UML 2.2
- La dernière version diffusée par l'OMG est UML 2.5.1 depuis décembre 2017.

# IV. UML

1. Historique.
2. **Standard de modélisation UML.**
3. Les diagrammes UML.

# IV. UML

## 2. Standard de modélisation UML.

UML est un ensemble de *notations* destinées à représenter divers aspects d'une application orientée objet.

UML est un standard de modélisation pour:

- **visualiser**: chaque symbole graphique a une sémantique.
- **Spécifier**: de manière précise et complète sans ambiguïté.
- **Construire**: les classes peuvent être générées automatiquement.
- **Documenter**: les différents diagrammes, contraintes seront présentés dans un document.

# IV. UML

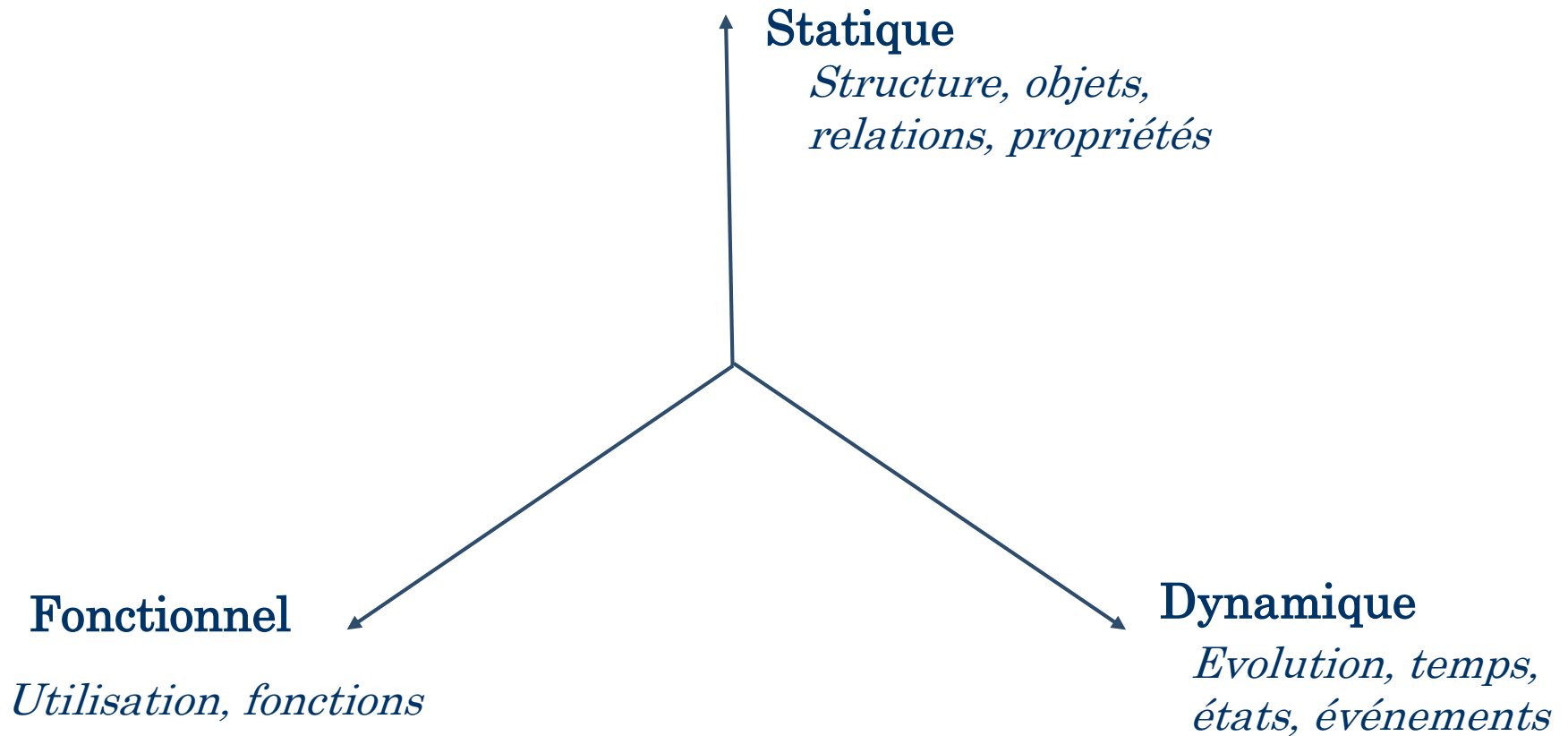
## 2. Standard de modélisation UML.

UML permet la création de **plusieurs diagrammes** d'un même système, chacun de ces diagrammes concerne un **aspect différent** du système(*fonctionnel, dynamique, statique*).

Les diagrammes **UML** sont indépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

# IV. UML

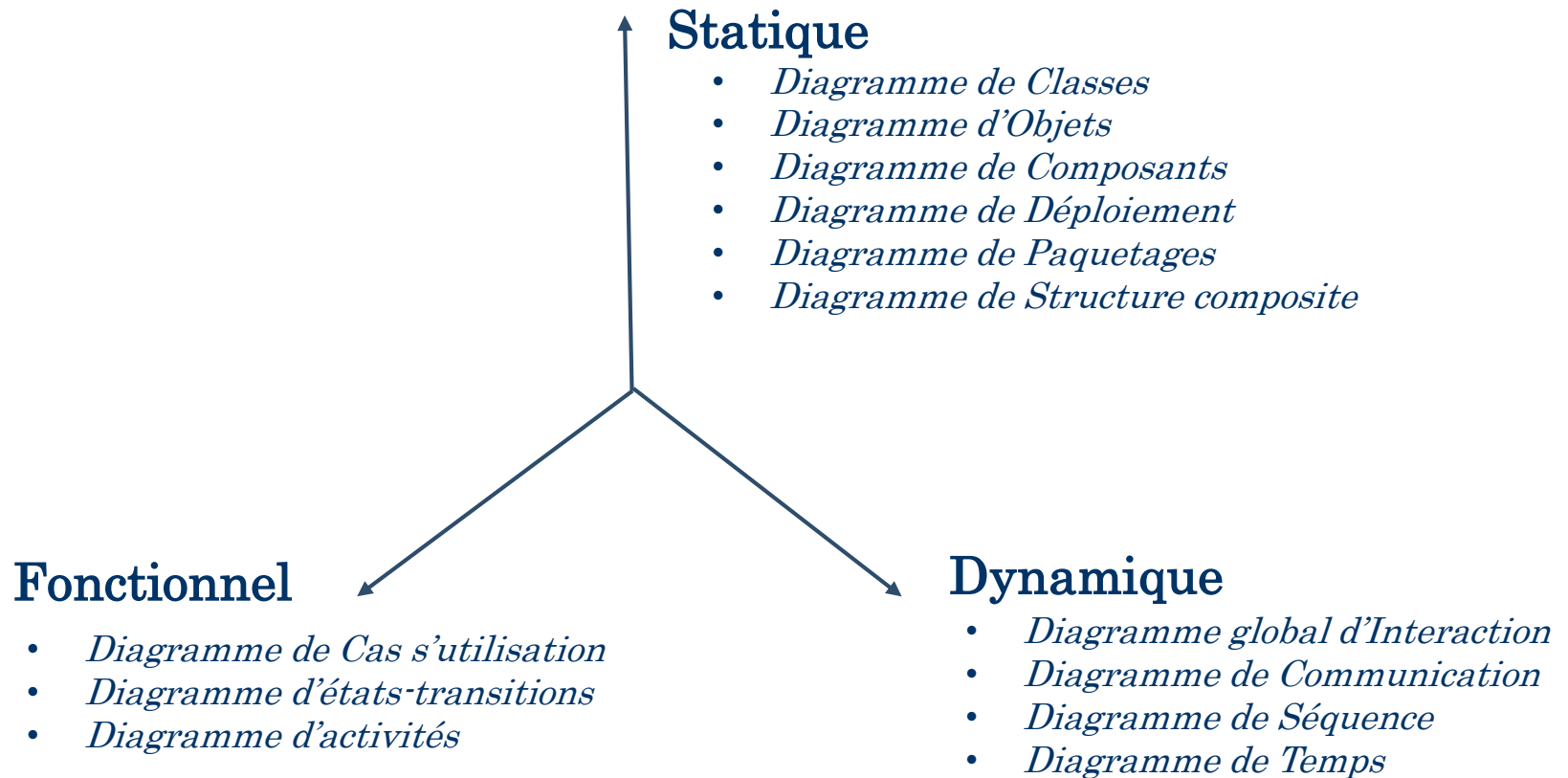
## 2. Standard de modélisation UML.





# IV. UML

## 2. Standard de modélisation UML.



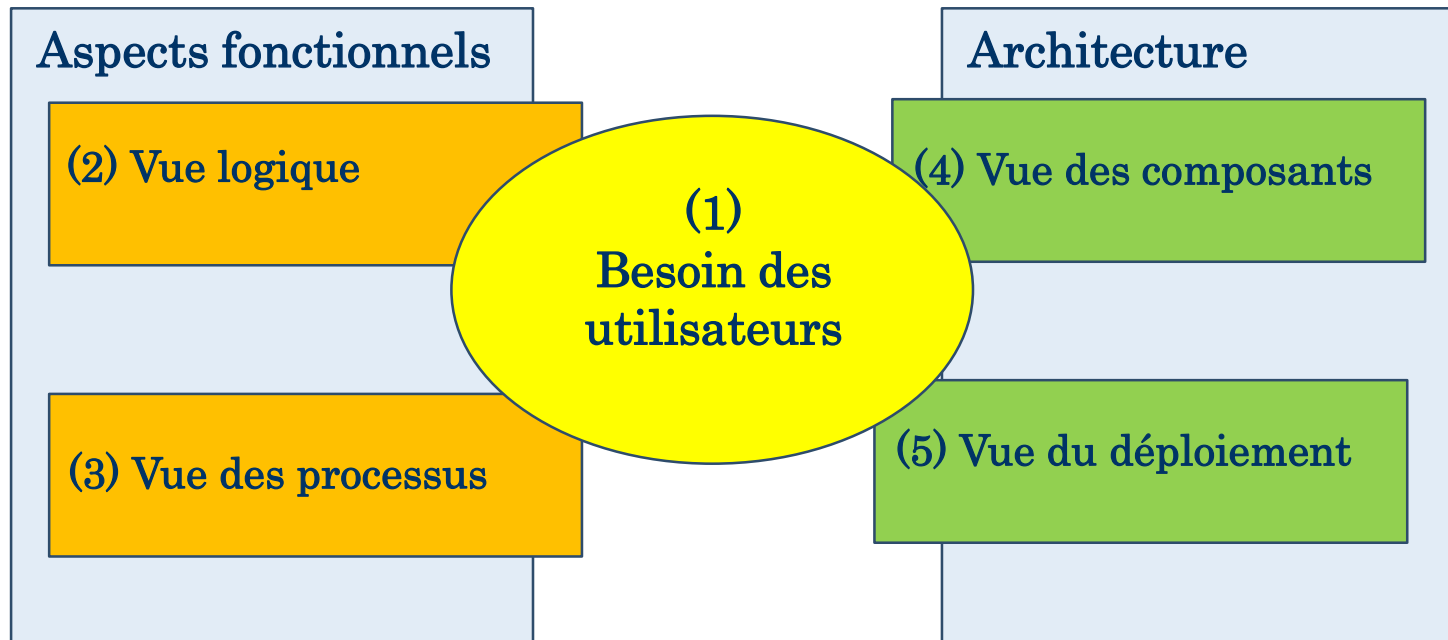
# IV. UML

1. Historique.
2. Standard de modélisation UML.
3. Les diagrammes UML.

# IV. UML

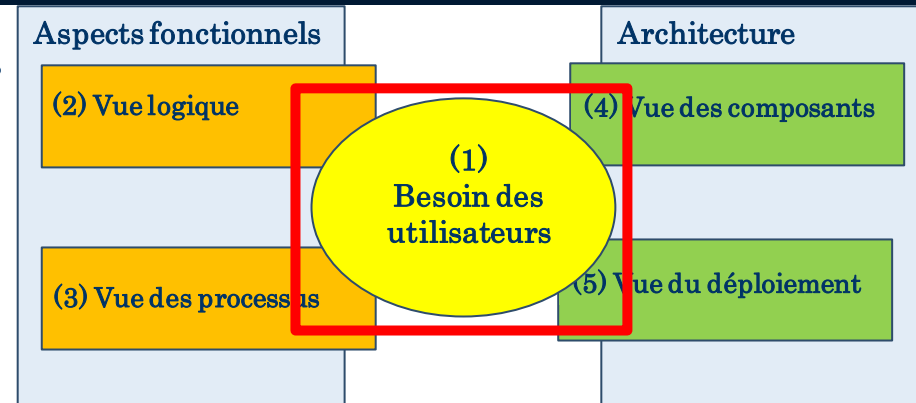
## 3. Les diagrammes UML.

*Pour mettre en œuvre UML, il faut considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système.*



# IV. UML

## 3. Les diagrammes UML.



### (1) Besoin des utilisateurs:

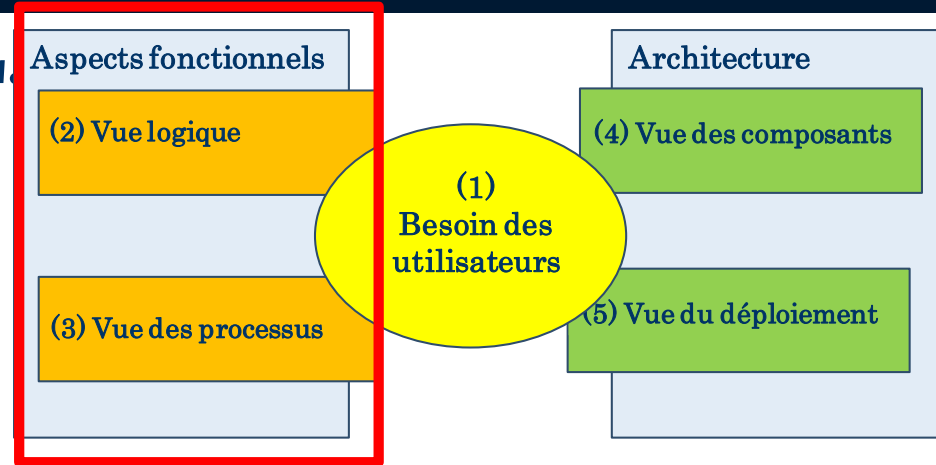
Décrit le contexte, les acteurs ou les utilisateurs du projet, les fonctionnalités du logiciel mais aussi les interactions entre ces acteurs et ces fonctionnalités.

### Diagrammes concernés:

- Diagramme de cas d'utilisation.
- Diagramme de paquetage.

# IV. UML

## 3. Les diagrammes UML

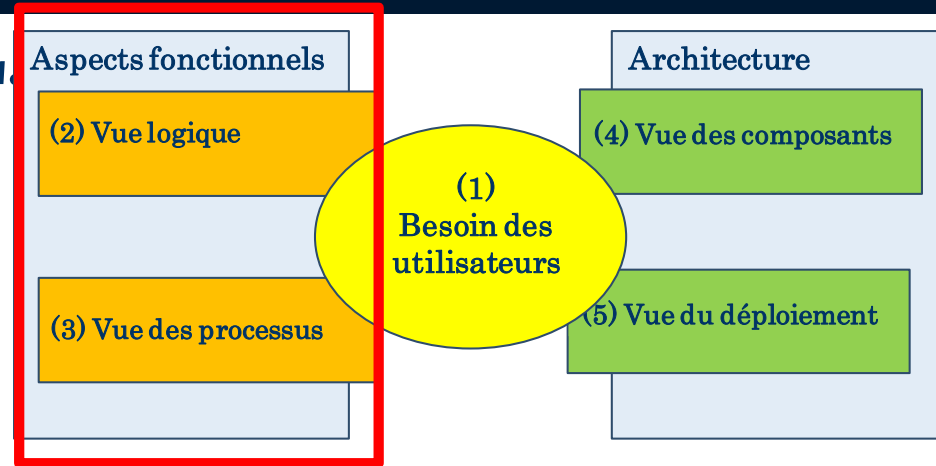


### Aspects fonctionnels:

Permet de définir qui utilisera le logiciel, pour quoi faire et comment les fonctionnalités vont se dérouler?

# IV. UML

## 3. Les diagrammes UML



### (2) Vue logique:

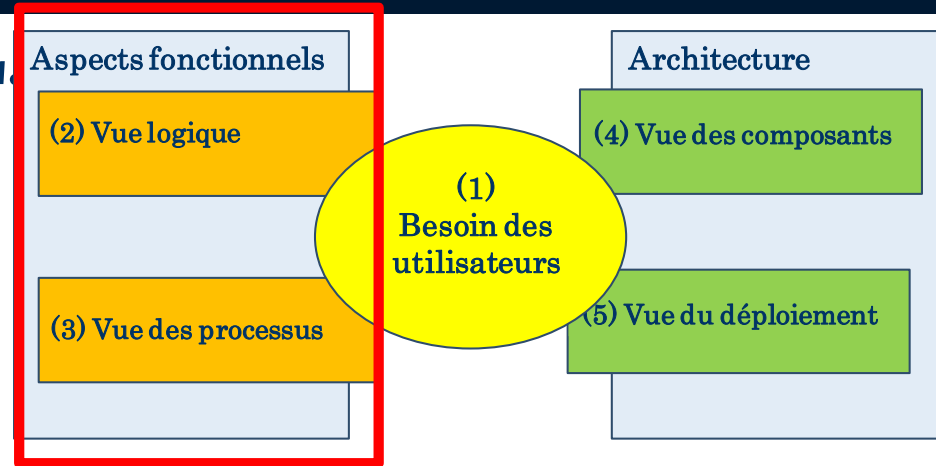
Permet d'identifier les éléments du domaine, les relations et interactions entre ces éléments.

### Diagrammes concernés:

- Diagramme de Classes.
- Diagramme d'objets.

# IV. UML

## 3. Les diagrammes UML



### (3) Vue des processus:

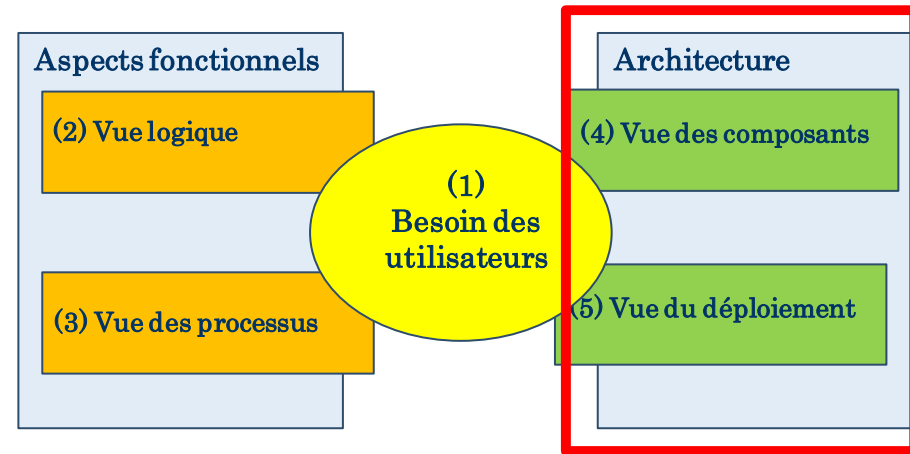
Permet de démontrer, la composition du système en processus et actions, et les interactions entre les processus.

### Diagrammes concernés:

- Diagramme d'interaction (communication et séquence).
- Diagramme d'activité.
- Diagramme d'état-transition.
- Diagramme de temps.

# IV. UML

## 3. Les diagrammes UML.



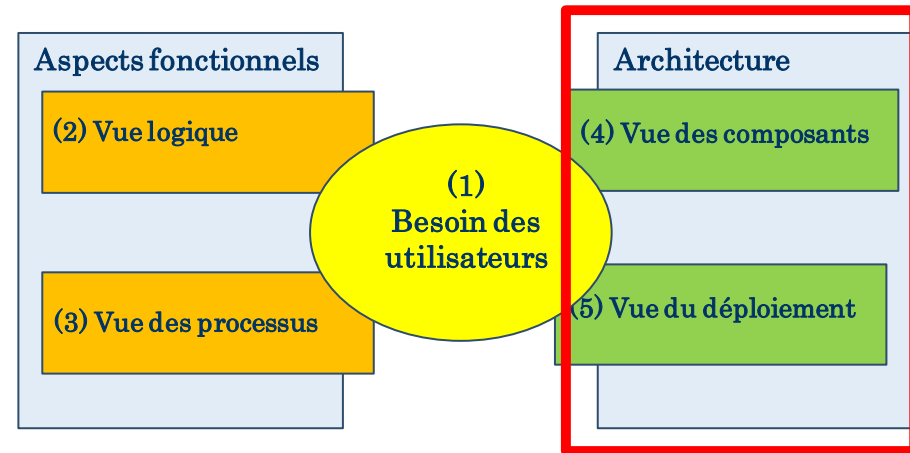
### Architecture:

Permet de définir les composantes à utiliser (interfaces, base de données, librairies de fonctions, etc.) et les matériels nécessaires.



# IV. UML

## 3. Les diagrammes UML.



### (4) Vue des composants:

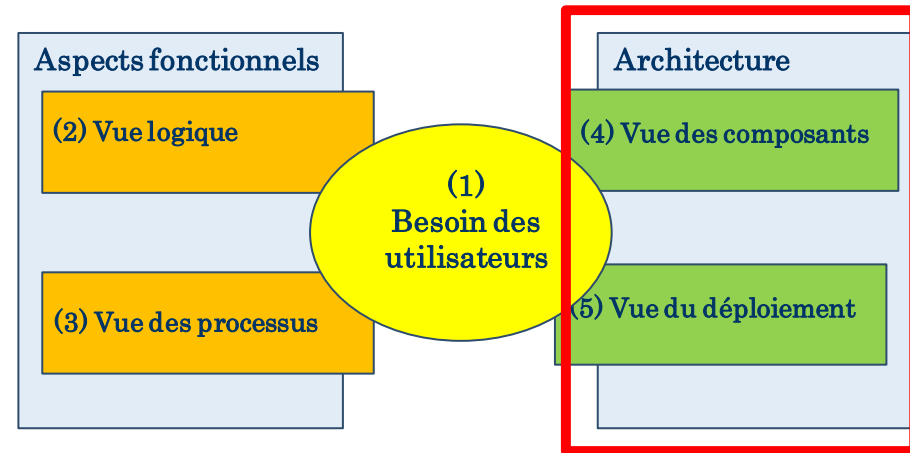
Permet de mettre en évidence les différentes parties qui composent le futur système (fichiers sources, bibliothèques, base de données, exécutables, etc.).

### Diagrammes concernés:

- Diagramme de structure composite.
- Diagramme des composants.

# IV. UML

## 3. Les diagrammes UML.



### (5) Vue du déploiement:

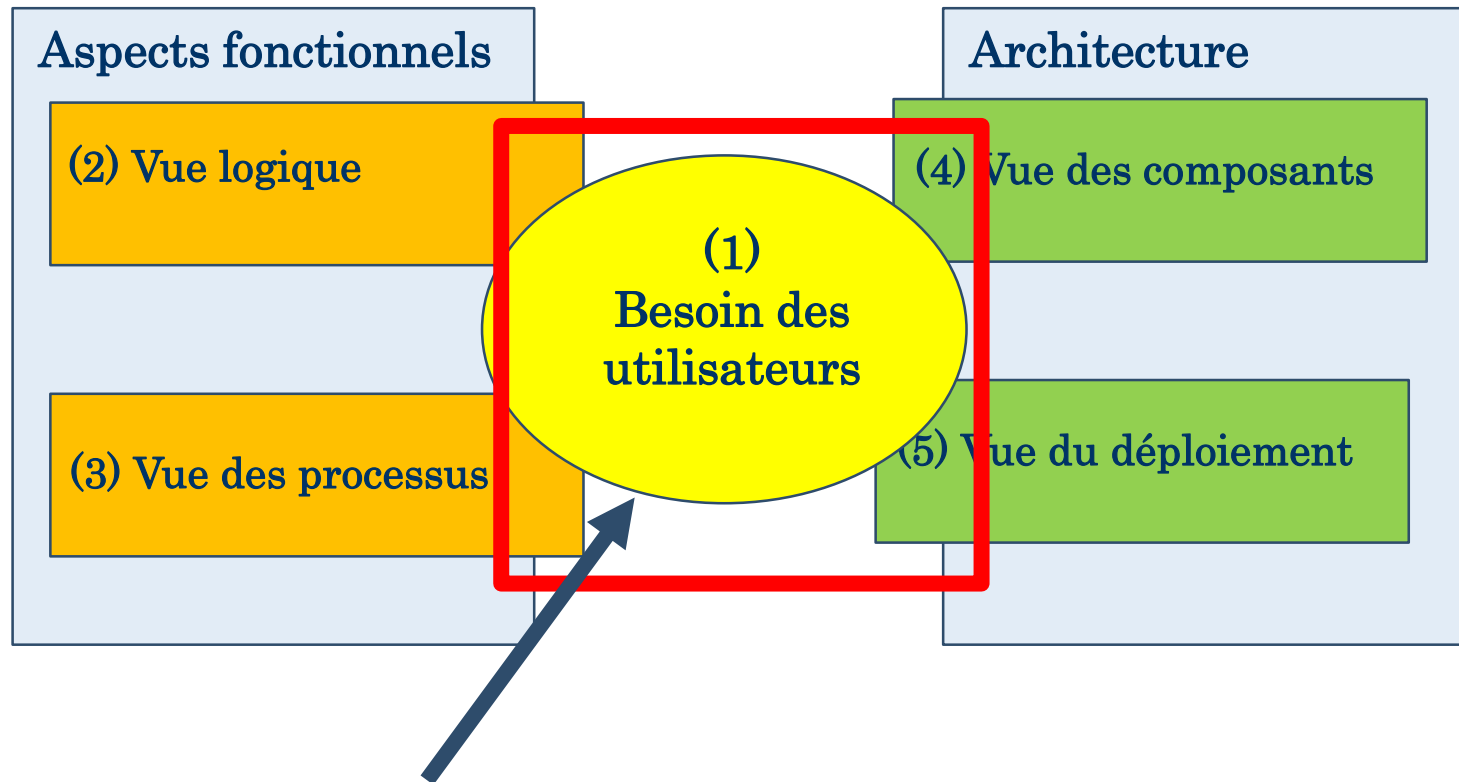
Permet de représenter le système et son environnement d'exécution.  
Décrit les ressources matériels nécessaires à la réalisation du système.

### Diagrammes concernés:

- Diagramme de déploiement.

# IV. UML

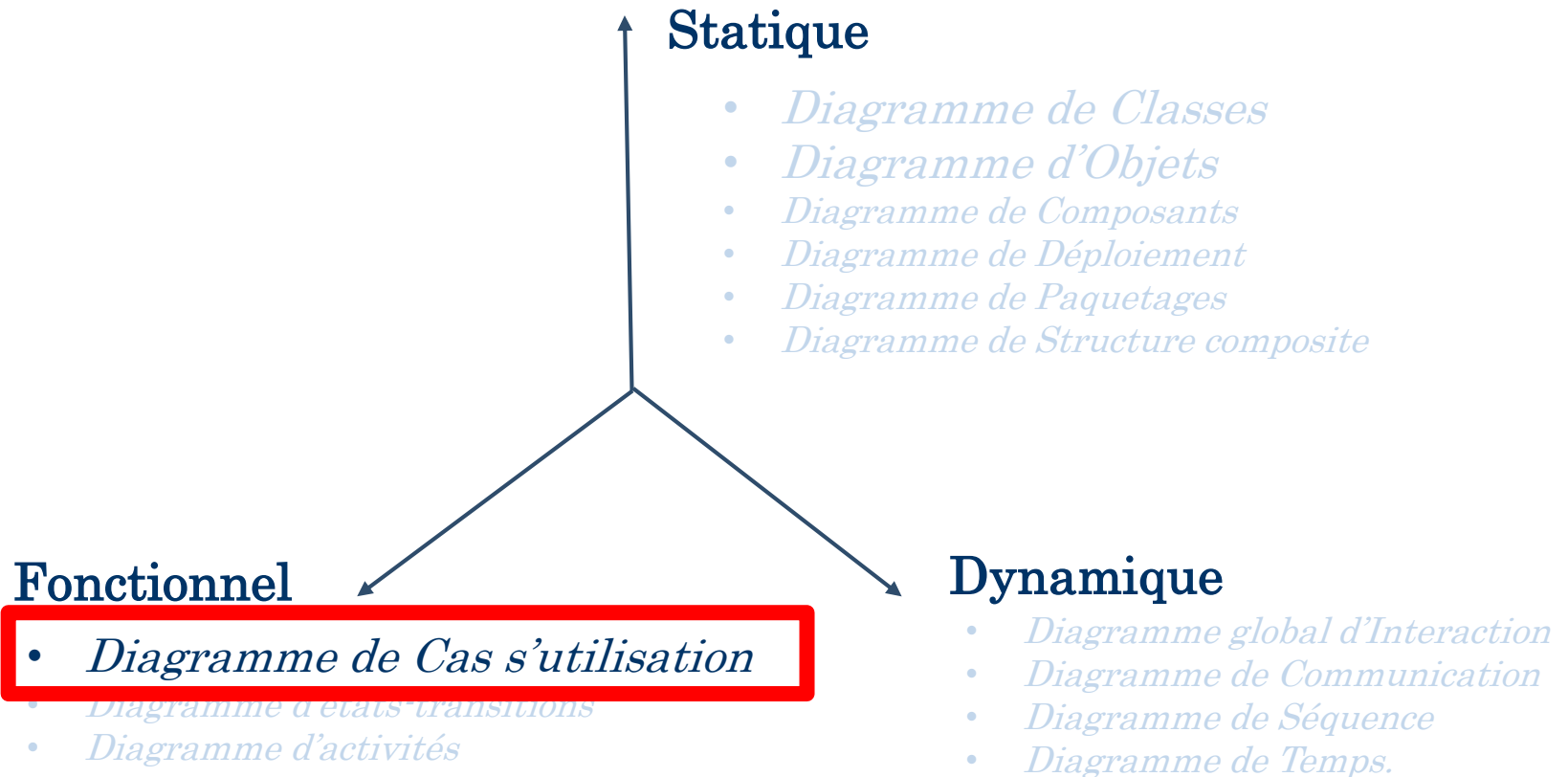
## 3. Les diagrammes UML.



Le premier diagramme du système à concevoir = Use Case

# IV. UML

## 3. Les diagrammes UML.



# IV. UML

## Diagramme de cas d'utilisation

1. Objectif.
2. Les éléments des diagrammes de cas d'utilisation.
3. Relations dans les diagrammes de cas d'utilisation.
4. Modélisation des besoins et diagrammes de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

1. **Objectif.**
2. Les éléments des diagrammes de cas d'utilisation.
3. Relations dans les diagrammes de cas d'utilisation.
4. Modélisation des besoins et diagrammes de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### 1. Objectif.

Les utilisateurs(non informaticiens) ont besoin d'un moyen simple pour exprimer leurs besoins.

Le rôle principale du diagramme de cas d'utilisation:

- **Recueillir, analyser et organiser** les besoins des utilisateurs.
- **Représenter** sous forme **graphique simple** ses besoins pour être compréhensible par toutes les personnes impliquées dans le projet.
- **Construire un référentiel de dialogue** entre les informaticiens et les utilisateurs.

# IV. UML

## Diagramme de cas d'utilisation

### 1. Objectif.

Avec le diagramme de cas d'utilisation débute l'étape de l'analyse d'un système d'information.



# IV. UML

## Diagramme de cas d'utilisation

1. Objectif.
2. Les éléments des diagrammes de cas d'utilisation.
3. Relations dans les diagrammes de cas d'utilisation.
4. Modélisation des besoins et diagrammes de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

2. Les éléments des diagrammes de cas d'utilisation.
  - a) Rôle de diagramme de cas d'utilisation.
  - b) Acteur.
  - c) Cas d'utilisation.
  - d) Représentation d'un diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### 2. Les éléments des diagrammes de cas d'utilisation.

- a) Rôle de diagramme de cas d'utilisation.
- b) Acteur.
- c) Cas d'utilisation.
- d) Représentation d'un diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### a) Rôle de diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation est donc parfait pour:

- Représenter les **objectifs des interactions** entre le système et les utilisateurs.
- Définir et organiser les **exigences fonctionnelles** dans un système.
- Préciser le **contexte** et les exigences d'un système.
- Modéliser le **flux** de base des **événements** dans un cas d'utilisation.

*Le système à modéliser*

ET

*Les différents acteurs qui vont interagir avec le système.*



*Deux notions*

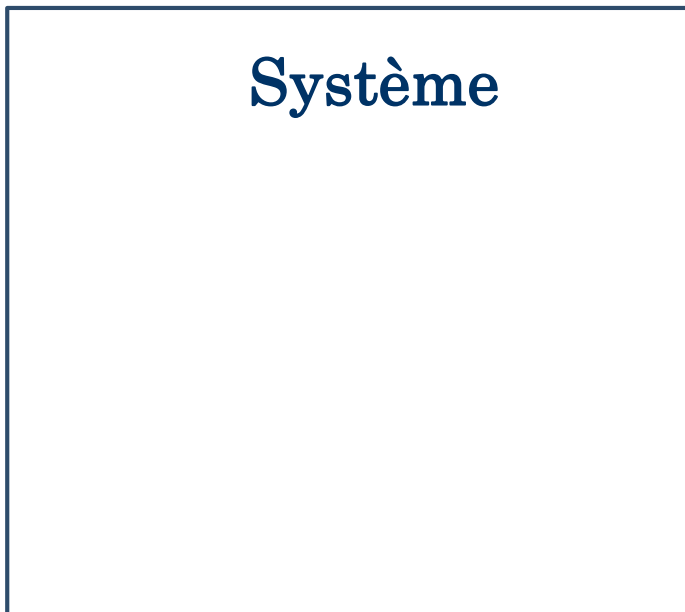
# IV. UML

## Diagramme de cas d'utilisation

### C'est quoi un système?

- Futur logiciel (ou application).
- Ensemble de services ou de fonctionnalités.

### Comment représenter un système?



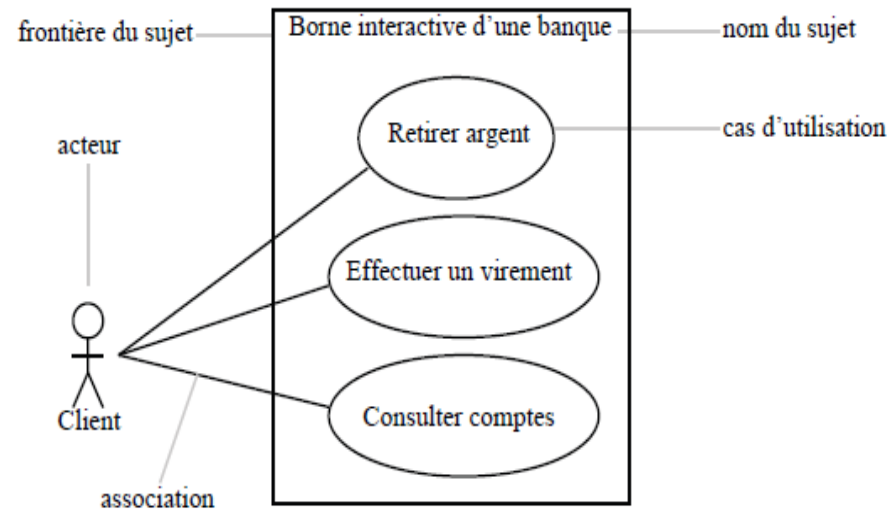
# IV. UML

## Diagramme de cas d'utilisation

### Un système?



Modélisation d'un distributeur de billet



Le système = une borne interactive qui permet d'accéder à une banque

- Il apparaît dans un cadre.
- Les utilisateurs sont représentés par des petits bonshommes.
- Les fonctionnalités sont représentées par des ellipses.

# IV. UML

## Diagramme de cas d'utilisation

2. Les éléments des diagrammes de cas d'utilisation.
  - a) Rôle de diagramme de cas d'utilisation.
  - b) Acteur.
  - c) Cas d'utilisation.
  - d) Représentation d'un diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### b) Acteur.

#### Définition:

Un acteur est l'idéalisation d'un **RÔLE** joué par une entité externe au système.

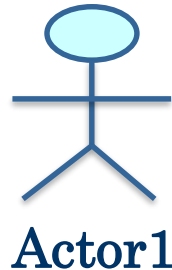
Un acteur est une entité qui interagit avec le système.



# IV. UML

## Diagramme de cas d'utilisation

### b) Acteur.



Stéréotype

Les **acteurs** sont représentés sous la forme de **petit personnage** (stick man) ou sous la forme d'une **case rectangulaire** (appelée classeur) avec le mot clé « Actor ».

Chaque **acteur** porte un **nom**.

#### Remarque importante:

En UML, une annotation entre guillemet est appelée « **stéréotype** ». Cela permet de préciser et de mieux caractériser l'élément à qui il s'adresse.

# IV. UML

## Diagramme de cas d'utilisation

### b) Acteur.

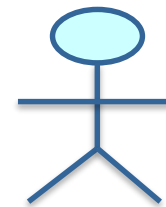
Un **acteur** est un utilisateur externe au système.

Cela peut-être:

- Une personne.
- Un matériel (capteurs, moteurs, etc.).
- Un autre système.

Nous utiliserons:

- Le **stick man** si l'acteur est humain.



Actor1

- Le **classeur** si l'acteur est du matériel ou autre système.

<<actor>>  
Actor2

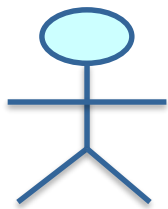
# IV. UML

## Diagramme de cas d'utilisation

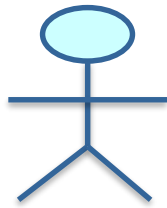
### b) Acteur.

Quels sont les acteurs pour un « système d'achat et de vente en ligne »?

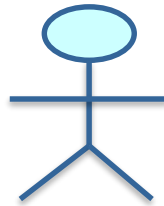
- Client.
- Vendeur.
- Administrateur.
- Banque.



Client



Vendeur



Administrateur

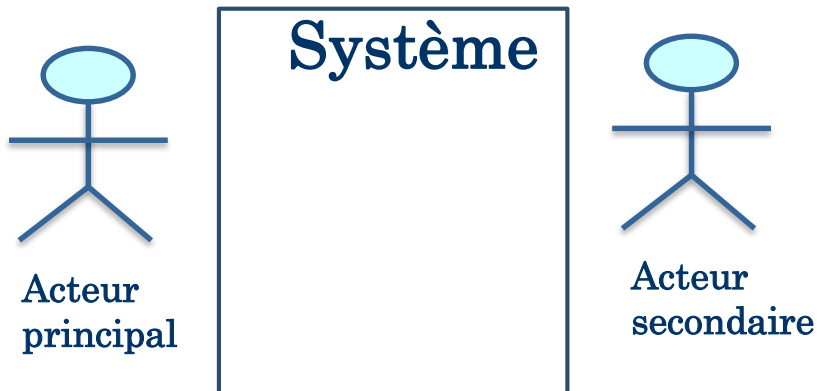
`<<actor>>`  
Banque

# IV. UML

## Diagramme de cas d'utilisation

### b) Acteur. Deux types d'acteurs:

1. Acteur primaire (ou acteur principal). À gauche du système.  
Directement **concerné** par le cas d'utilisation décrit.  
Le **cas d'utilisation rend service** à cet acteur.
2. Acteur secondaire. À droite du système  
Est **sollicité** pour des informations complémentaires nécessaires au déroulement du cas d'utilisation décrit.  
L'**acteur secondaire rend service** au cas d'utilisation.



#### Remarques:

Un acteur peut-être principal pour un cas d'utilisation et secondaire pour un autre cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### b) Acteur.

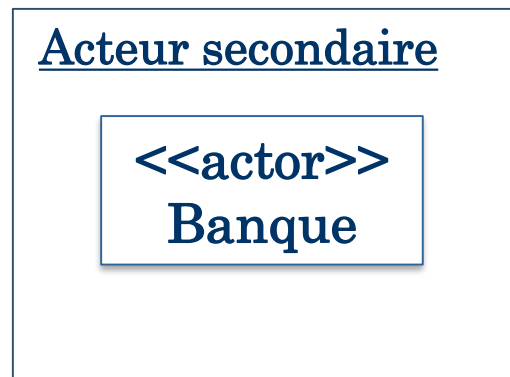
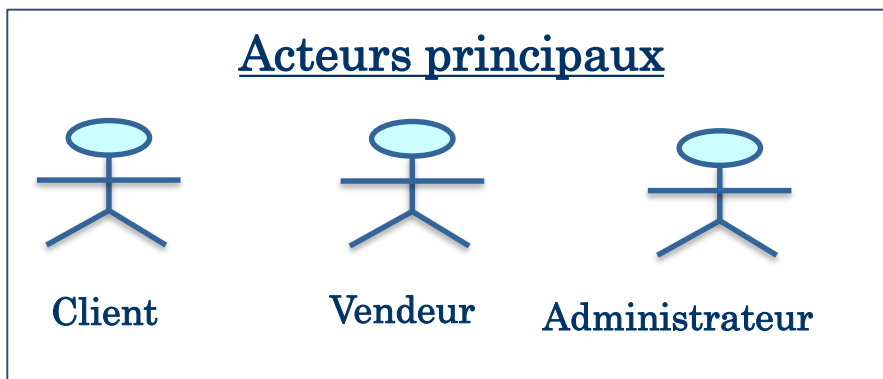
Quels sont les acteurs pour un « système d'achat et de vente en ligne »?

Acteurs principaux:

- Client.
- Vendeur
- Administrateur.

Acteur secondaire:

- Banque.



# IV. UML

## Diagramme de cas d'utilisation

### b) Acteur.



Client



Vendeur



Administrateur

système d'achat et de vente en ligne

<<actor>>  
Banque

# IV. UML

## Diagramme de cas d'utilisation

2. Les éléments des diagrammes de cas d'utilisation.
  - a) Rôle de diagramme de cas d'utilisation.
  - b) Acteur.
  - c) Cas d'utilisation.
  - d) Représentation d'un diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### c) Cas d'utilisation.

#### Définition:

Un **cas d'utilisation** représente une **fonctionnalité** du système (**visible de l'extérieur du système**).

Un **cas d'utilisation** modélise donc un **service** rendu par le système sans imposer le mode de réalisation de ce service.



# IV. UML

## Diagramme de cas d'utilisation

### c) Cas d'utilisation.



Les **cas d'utilisation** sont représentés sous forme d'une **ellipse** contenant le nom du cas d'utilisation (**phrase commençant par un verbe à l'infinitif**) et optionnellement un stéréotype au dessus du nom.

**CAS = Faire quelque chose**

#### Remarque importante:

Les différents cas peuvent être représentés à l'intérieur d'un même rectangle représentant les limites du système.

# IV. UML

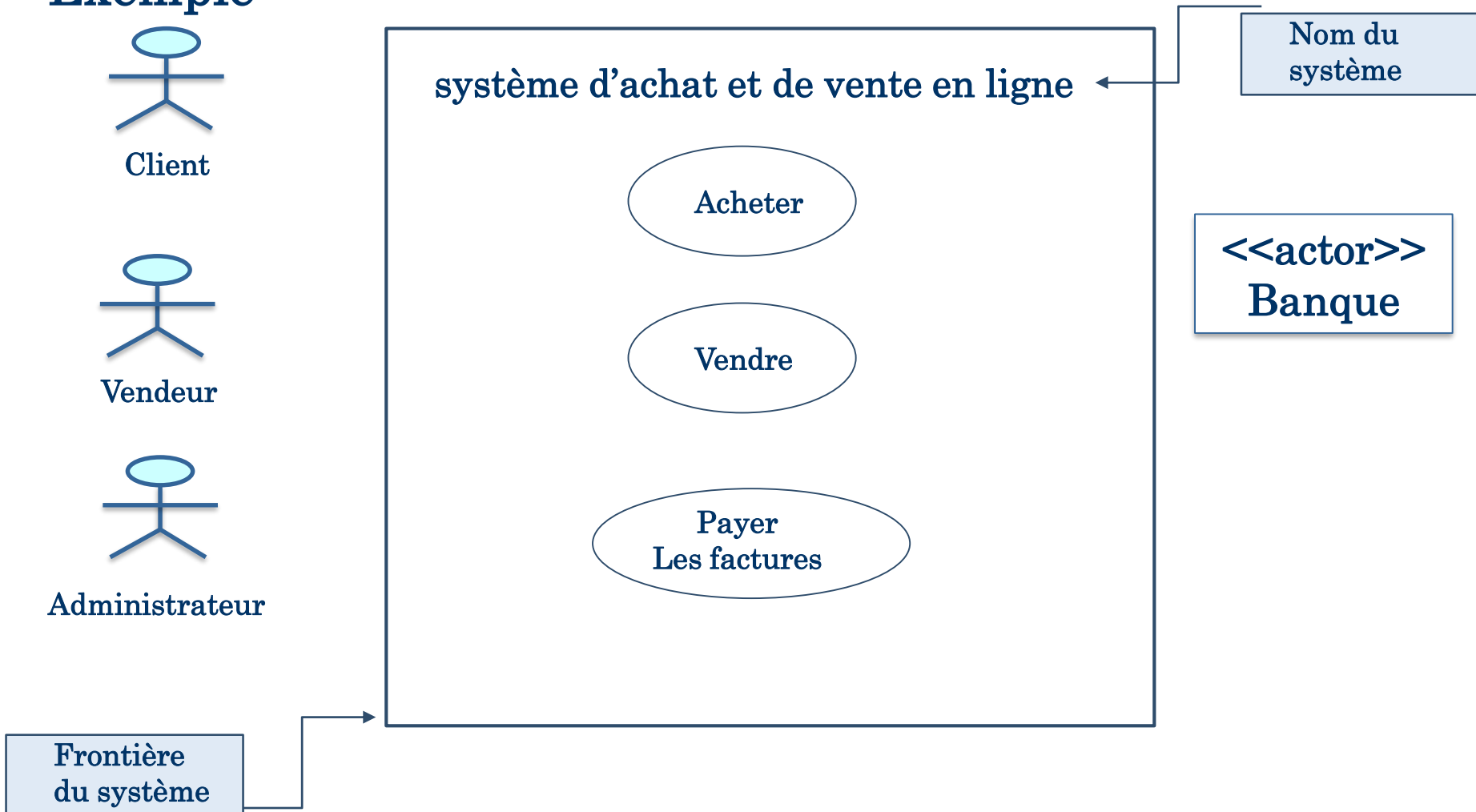
## Diagramme de cas d'utilisation

2. Les éléments des diagrammes de cas d'utilisation.
  - a) Rôle de diagramme de cas d'utilisation.
  - b) Acteur.
  - c) Cas d'utilisation.
  - d) Représentation d'un diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### Exemple



# IV. UML

## Diagramme de cas d'utilisation

1. Objectif.
2. Les éléments des diagrammes de cas d'utilisation.
3. Relations dans les diagrammes de cas d'utilisation.
4. Modélisation des besoins et diagrammes de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### 3. Relations dans les diagrammes de cas d'utilisation.

- a) Relations entre acteurs et cas d'utilisation.
- b) Relations entre cas d'utilisation.
- c) Relations entre acteurs.
- d) Exemple de diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### 3. Relations dans les diagrammes de cas d'utilisation.

- a) Relations entre acteurs et cas d'utilisation.
- b) Relations entre cas d'utilisation.
- c) Relations entre acteurs.
- d) Exemple de diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### a) Relations entre acteurs et cas d'utilisation.

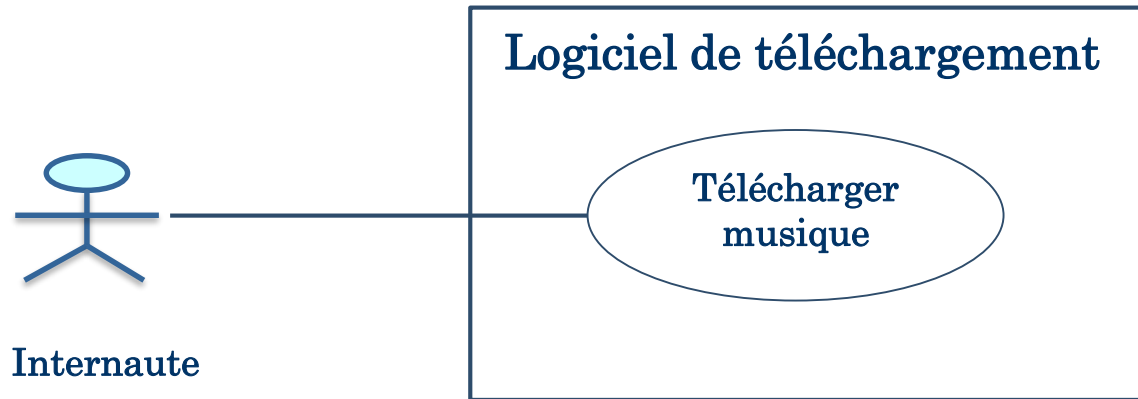
#### Relation d'association:

- Une **relation d'association** est un chemin de communication entre un acteur et un cas d'utilisation.
- Interaction entre acteur et cas d'utilisation:
  - \* l'acteur **initie** le cas d'utilisation
  - \* le cas d'utilisation interagit ou **rend service** à l'acteur.

# IV. UML

## Diagramme de cas d'utilisation

### Relation d'association.



La relation d'association est représenté par un **trait continu**.

#### Remarque importante:

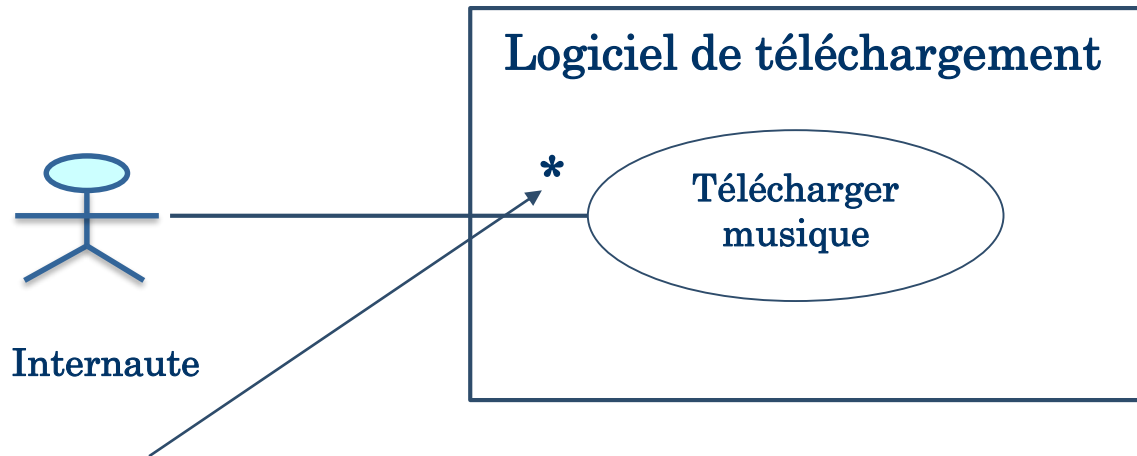
Comme pour les associations entre classes (diagramme des classes) on peut préciser la multiplicité.



# IV. UML

## Diagramme de cas d'utilisation

### Multiplicité.

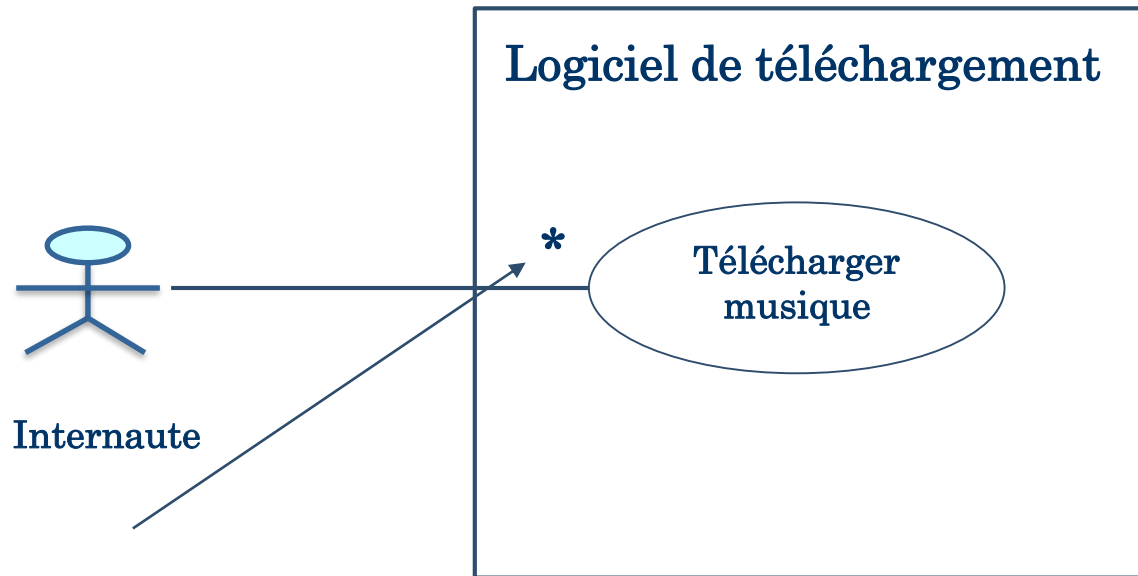


- Si un acteur peut **interagir plusieurs** fois avec un cas d'utilisation. On ajoute alors une multiplicité du côté du cas d'utilisation.
- Le symbole « \* » qui signifie « **plusieurs** » est ajouté à l'extrémité du cas. Plusieurs valeurs sont possibles pour la multiplicité.
- **On** signifie **exactement n**, **n...m** signifie **entre n et m**.
- Préciser une multiplicité sur la relation n'implique pas nécessairement que les **cas sont utilisés en même temps**.

# IV. UML

## Diagramme de cas d'utilisation

Multiplicité.

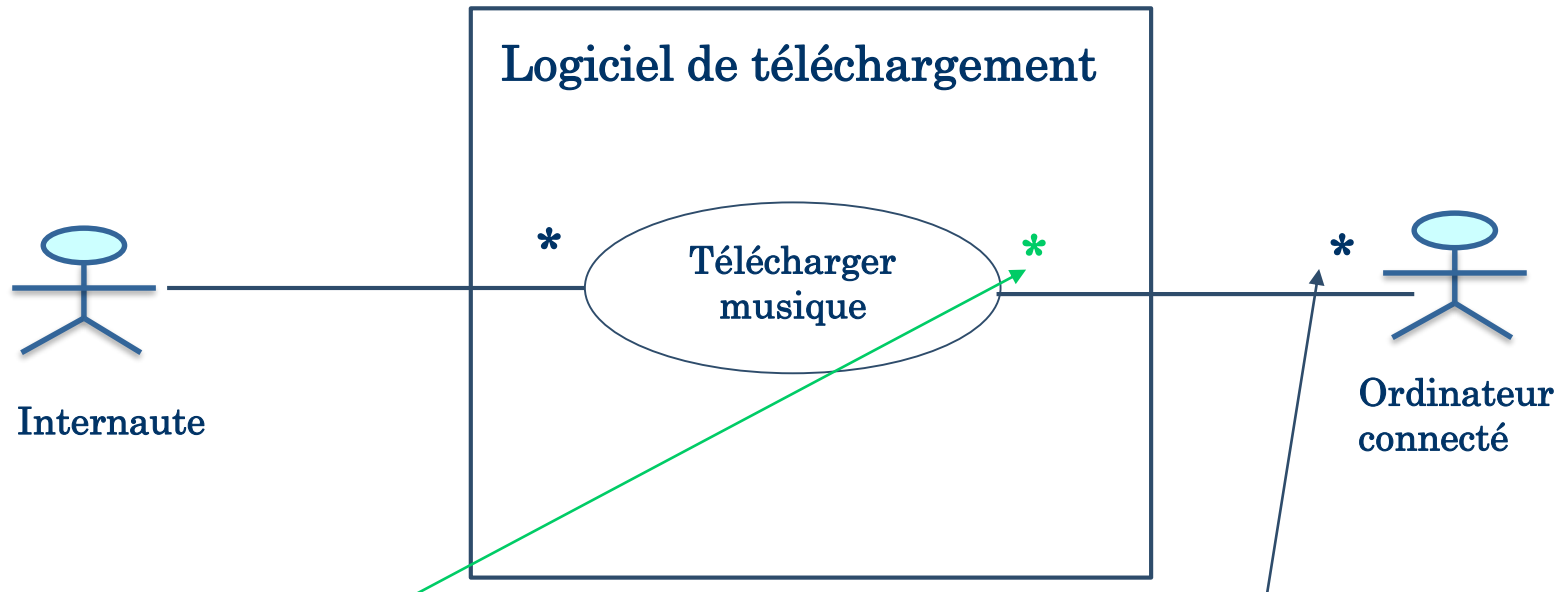


*La figure ci-dessus montre un internaute qui télécharge plusieurs morceaux de musique sur internet.*

# IV. UML

## Diagramme de cas d'utilisation

### Multiplicité.



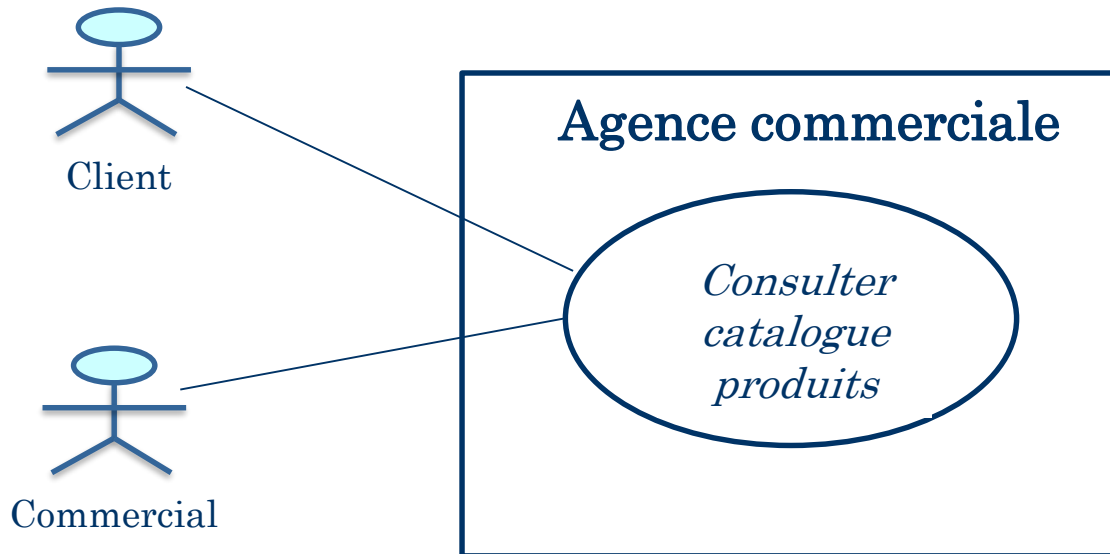
*Un ordinateur connecté peut être associé à plusieurs cas d'utilisation « télécharger musique ».*

*Plusieurs ordinateurs connectés peuvent être sollicités pour un même cas d'utilisation « Télécharger musique »*

# IV. UML

## Diagramme de cas d'utilisation

### a) Relations entre acteurs et cas d'utilisation: Exemples



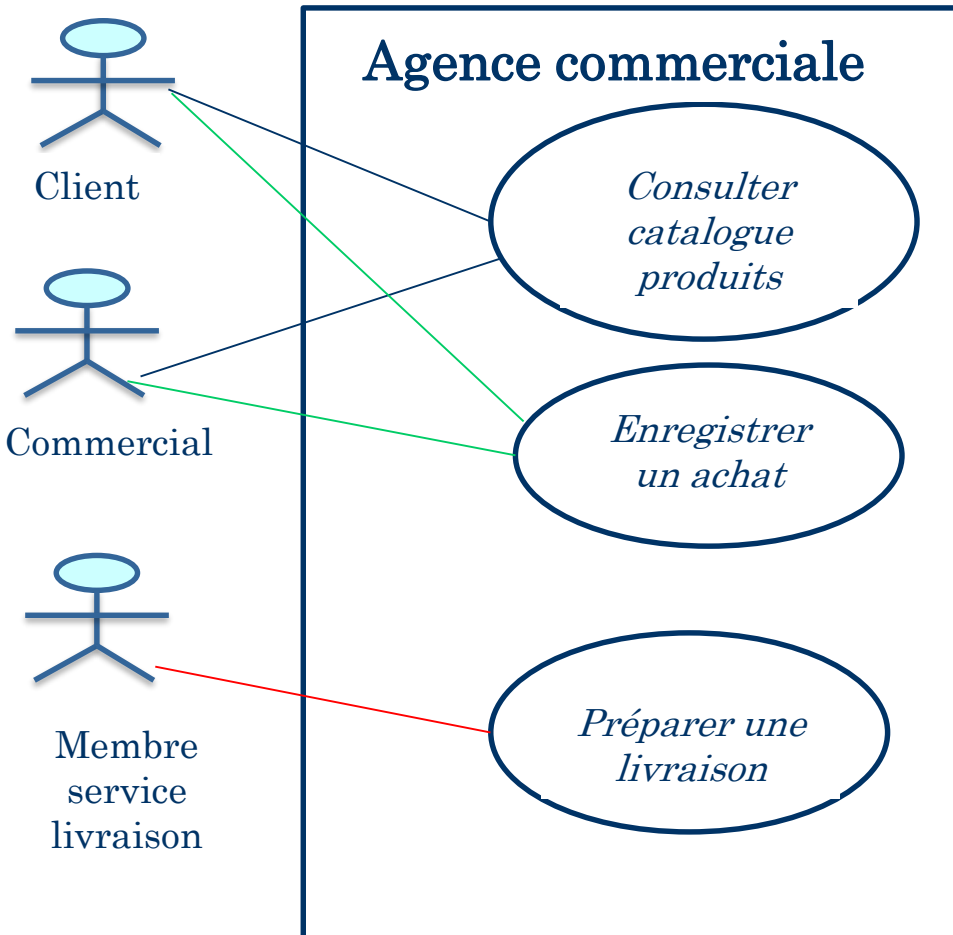
*Un cas n'est pas forcément lié à un seul acteur.*

*Le cas d'utilisation « Consulter catalogue produit » rend service à deux acteurs (client et commercial).*

# IV. UML

## Diagramme de cas d'utilisation

### a) Relations entre acteurs et cas d'utilisation: Exemples

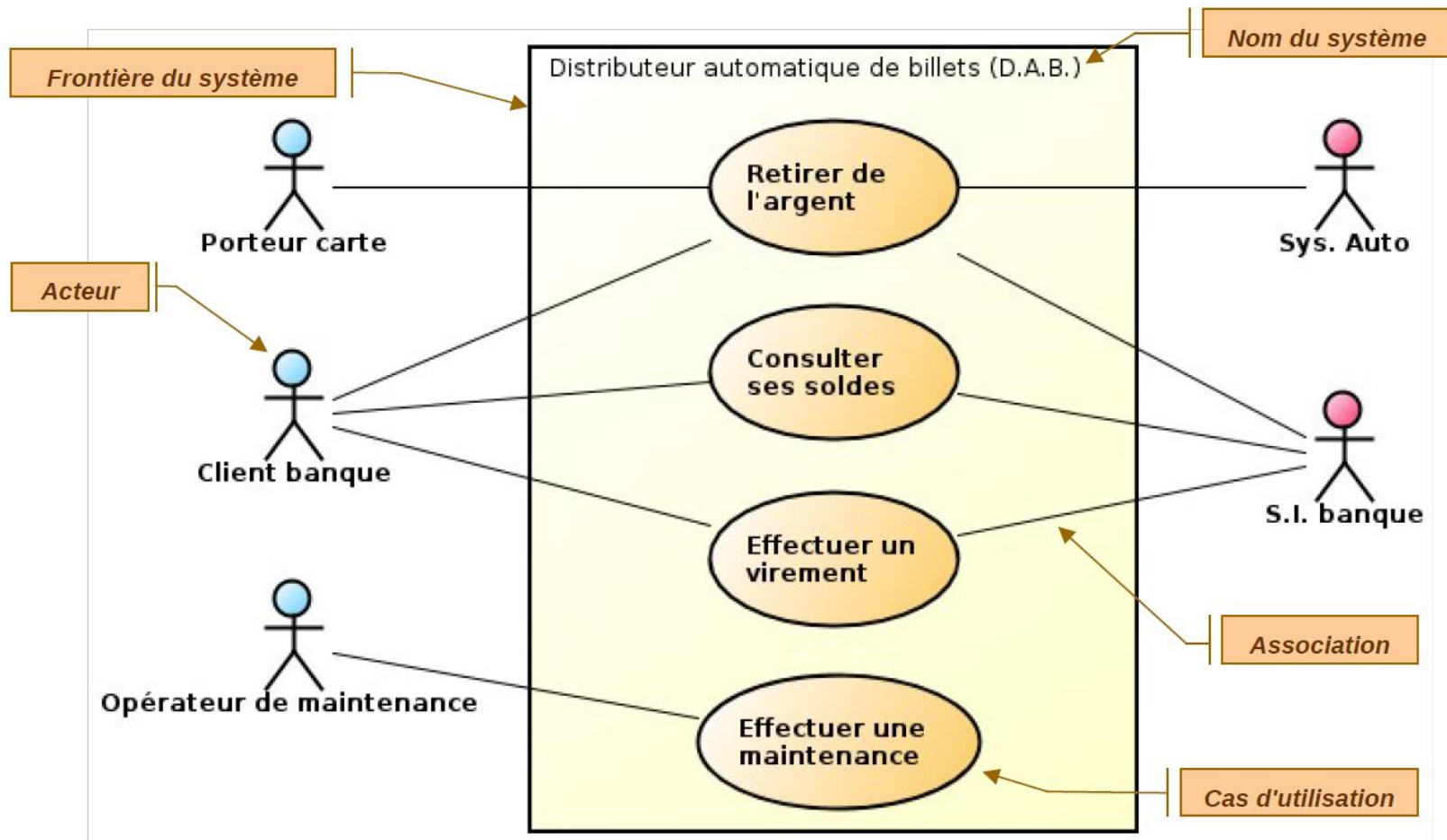


<i>Le client</i>	<ul style="list-style-type: none"><li>• Consulter le catalogue des produits</li><li>• Enregistrer un achat</li></ul>
<i>Le commercial</i>	<ul style="list-style-type: none"><li>• Consulter le catalogue des produits</li><li>• Enregistrer un achat</li></ul>
<i>Le membre service de livraison</i>	<ul style="list-style-type: none"><li>• Préparer une livraison</li></ul>

# IV. UML

## Diagramme de cas d'utilisation

### Exemple de diagramme de cas d'utilisation:



# IV. UML

## Diagramme de cas d'utilisation

### 3. Relations dans les diagrammes de cas d'utilisation.

- a) Relations entre acteurs et cas d'utilisation.
- b) Relations entre cas d'utilisation.
- c) Relations entre acteurs.
- d) Exemple de diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### b) Relations entre cas d'utilisation.

#### Relations entre cas d'utilisation:

Pour clarifier un diagramme, UML permet d'établir des relations entre cas d'utilisation.

Il existe principalement deux types de relations:

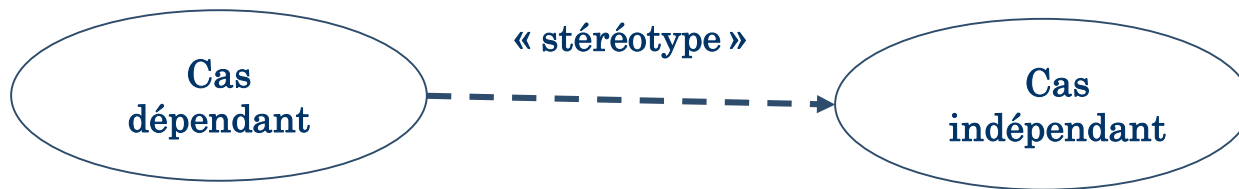
1. Les dépendances stéréotypées, qui sont des dépendances explicitées par un stéréotype.
2. La généralisation/spécialisation.



# IV. UML

## Diagramme de cas d'utilisation

### 1. Les dépendances stéréotypées:



Une dépendance est représentée par une **flèche avec un trait pointillé** issu du cas dépendant et pointant le cas indépendant.

Les stéréotypes les plus utilisés sont:

L'inclusion (**Include**)

ET

L'extension (**Extend**)

# IV. UML

## Diagramme de cas d'utilisation

### 1. Les dépendances stéréotypées: INCLUDE

#### Définition: « Include »

Un cas A est **inclus** dans un cas B, si le comportement décrit par le cas A **inclut** le comportement du cas B.  
Le cas A dépend de B.

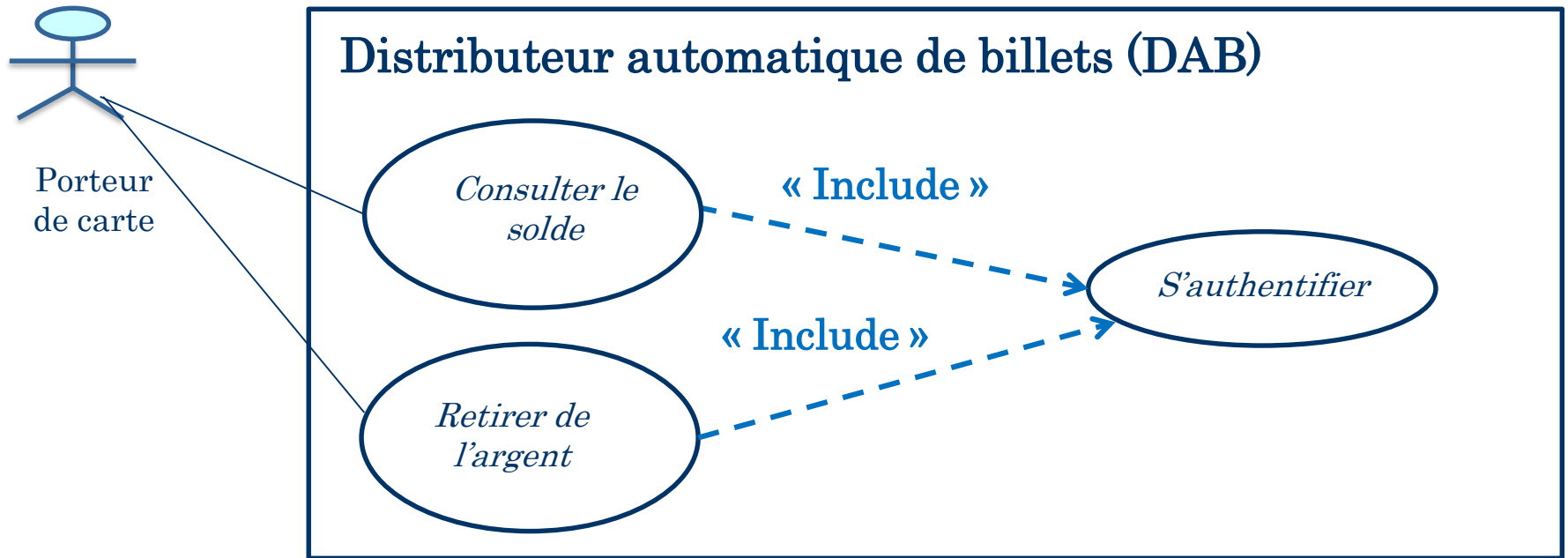


Remarque importante:  
Lorsque A est sollicité, B l'est obligatoirement.

# IV. UML

## Diagramme de cas d'utilisation

### Exemple : « Include »:

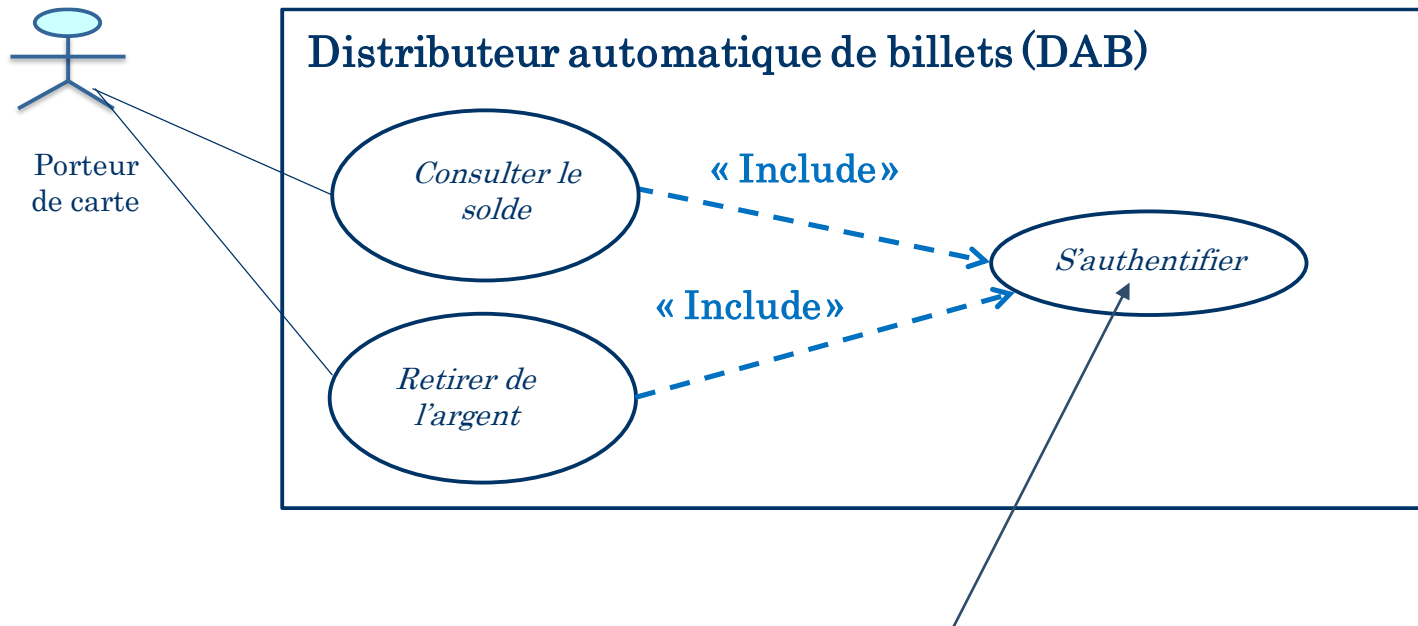


*L'accès aux informations du compte bancaire inclut nécessairement une **phase d'authentification** avec un identifiant et un mot de passe.*

# IV. UML

## Diagramme de cas d'utilisation

### Exemple : « Include »:



*Un cas d'utilisation est dit « **interne** » s'il n'est pas relié directement à un acteur.*

# IV. UML

## Diagramme de cas d'utilisation

### Intérêt de la relation « Include »:



- ❑ Partager une fonction commune entre plusieurs cas d'utilisation.
- ❑ Décomposer un cas d'utilisation en sous cas plus simples.

# IV. UML

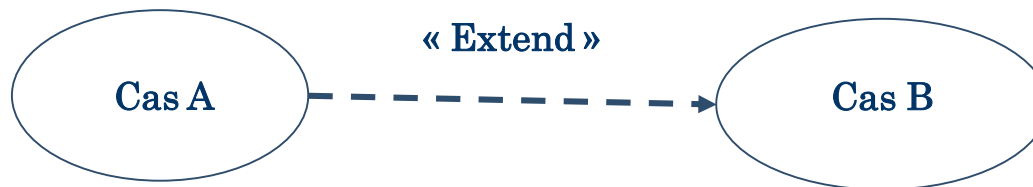
## Diagramme de cas d'utilisation

### 1. Les dépendances stéréotypées: EXTEND

#### Définition: « Extend »

Un cas A **étend** un cas B, lorsque le cas A **peut-être appelé** au cours de l'exécution du cas B.

Exécuter **B** peut éventuellement entraîner l'exécution de A.

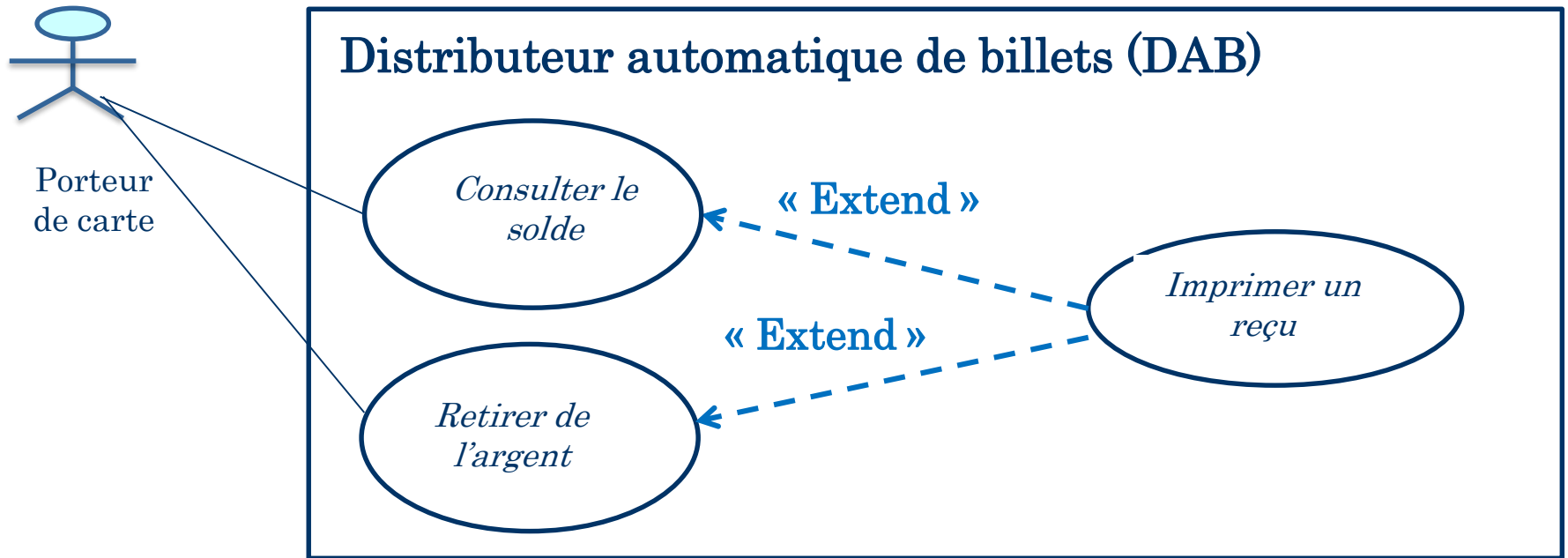


Remarque importante:  
Contrairement à l'inclusion, l'extension est optionnelle.

# IV. UML

## Diagramme de cas d'utilisation

### Exemple : « Extend »:



*Le porteur de carte peut « consulter le solde » et « retirer de l'argent ».*

*Et peut éventuellement (s'il le désire) imprimer un reçu.*

# IV. UML

## Diagramme de cas d'utilisation

### Exemple : « Extend »: Point d'extension

#### Définition: Point d'extension

L'extension peut intervenir à un **point** précis du cas étendu. C'est le **point d'extension**.

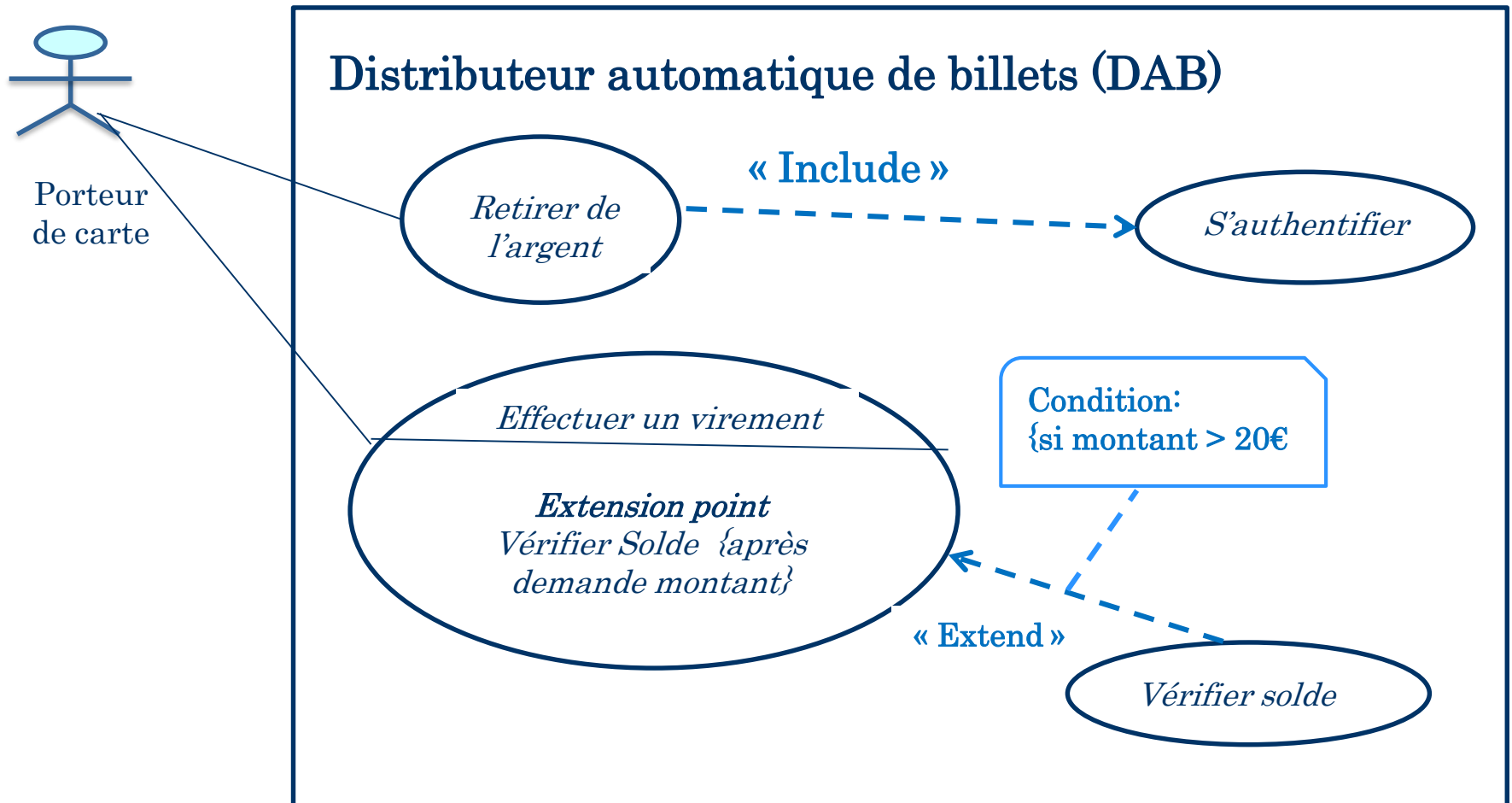
Il porte un nom, qui figure dans le cas étendu sous la rubrique '**extension points**' et est éventuellement associé à une contrainte indiquant le moment où l'extension intervient.



# IV. UML

## Diagramme de cas d'utilisation

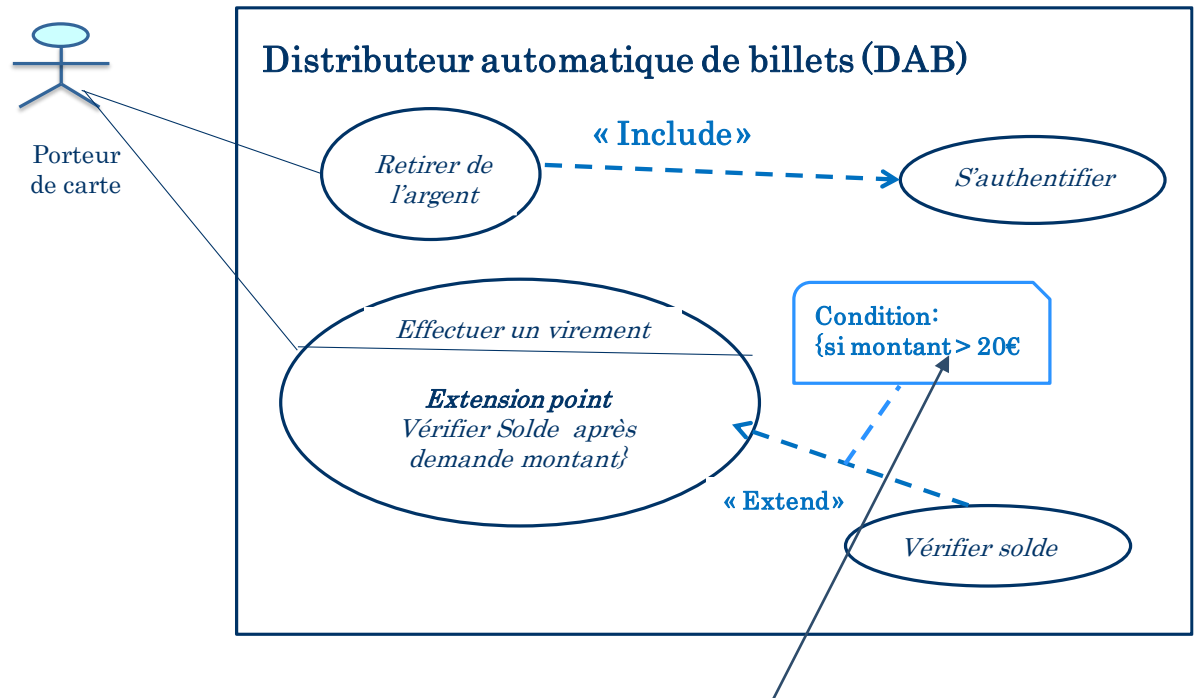
### Exemple : point d'extension



# IV. UML

## Diagramme de cas d'utilisation

### Point d'extension



Graphiquement, la condition est exprimée sous la forme d'une **note**. L'exemple ci-dessus présente un distributeur automatique de billets dans une banque où la vérification du solde '**vérifier solde**' n'étend le cas '**effectuer un virement**' que si la demande du virement **dépasse 20€**.

# IV. UML

## Diagramme de cas d'utilisation

### Synthèse:

Relation « Include »	Relation « Extend »
<b>Sens de la relation:</b> Le cas de base vers le cas inclus	<b>Sens de la relation:</b> Le cas d'extension vers le cas de base
<b>Exécution:</b> Inconditionnelle, le cas inclus est toujours exécuté par le cas de base. (Scénario principal)	<b>Exécution:</b> Le cas d'extension n'est pas nécessairement exécuté par le cas de base. Peut-être conditionnelle
<b>Dépendance:</b> Le cas de base n'est pas autonome sans le cas inclus	<b>Dépendance:</b> Le cas de base est nécessairement autonome et s'exécute en l'absence du cas d'extension.

# IV. UML

## Diagramme de cas d'utilisation

### 2. Généralisation/spécialisation:

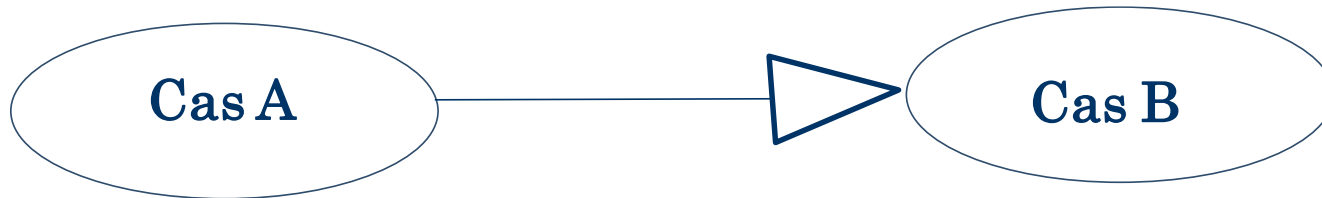
#### Définition: Généralisation/spécialisation.

- ❑ Un cas A est en généralisation d'un cas B, si B est un cas particulier de A.
- ❑ La relation généralisation/spécialisation est présente dans la plupart des diagrammes UML et se traduit par le concept d'héritage dans les langages de programmation orientés objets.

# IV. UML

## Diagramme de cas d'utilisation

### 2. Généralisation/spécialisation:



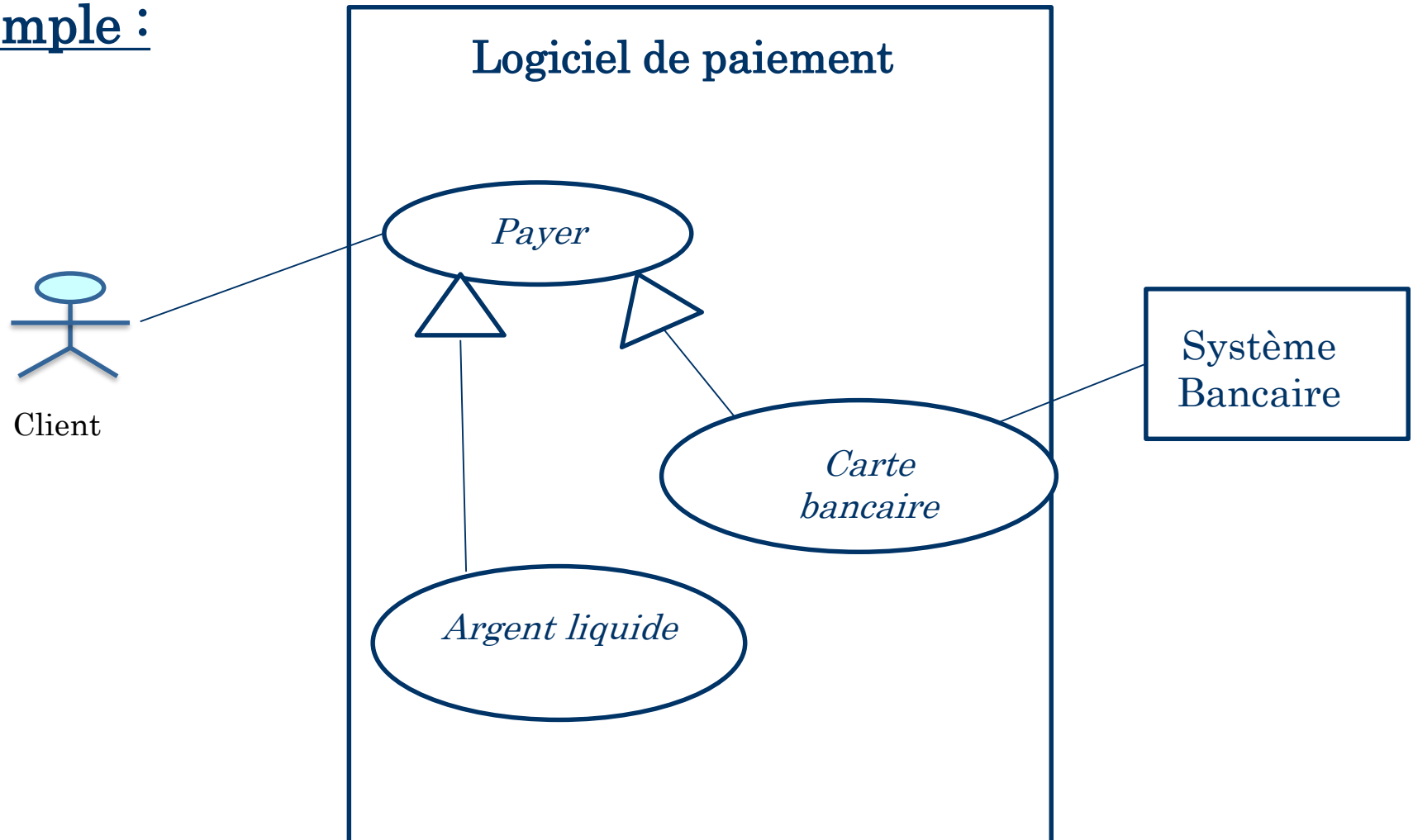
La relation de généralisation est représentée par une **flèche avec une extrémité triangulaire**.

- ☐ Comme pour les classes, le **sous cas** d'utilisation **hérite** du **comportement** du **sur cas** d'utilisation.
- ☐ Le **sous cas** hérite aussi de **toutes les associations** du **sur cas** d'utilisation.  
(relations d'associations avec les acteurs, relations d'inclusions et relations d'extensions).

# IV. UML

## Diagramme de cas d'utilisation

### Exemple :



# IV. UML

## Diagramme de cas d'utilisation

### 3. Relations dans les diagrammes de cas d'utilisation.

- a) Relations entre acteurs et cas d'utilisation.
- b) Relations entre cas d'utilisation.
- c) Relations entre acteurs.
- d) Exemple de diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### c) Relations entre acteurs:

#### Relations entre acteurs:

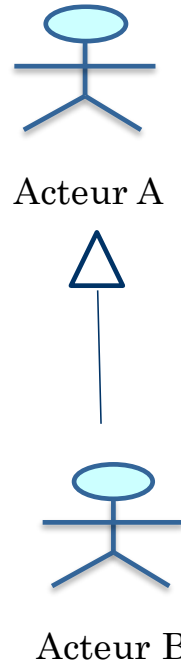
- ☐ La seule **relation** entre deux acteurs est la **généralisation**.
- ☐ Un **acteur A** est une **généralisation** d'un **acteur B** si l'acteur A peut être **substitué** par l'acteur B.
- ☐ Tous les cas d'utilisations **accessibles à A** le sont aussi à **B**, mais pas inversement.



# IV. UML

## Diagramme de cas d'utilisation

### c) Relations entre acteurs:



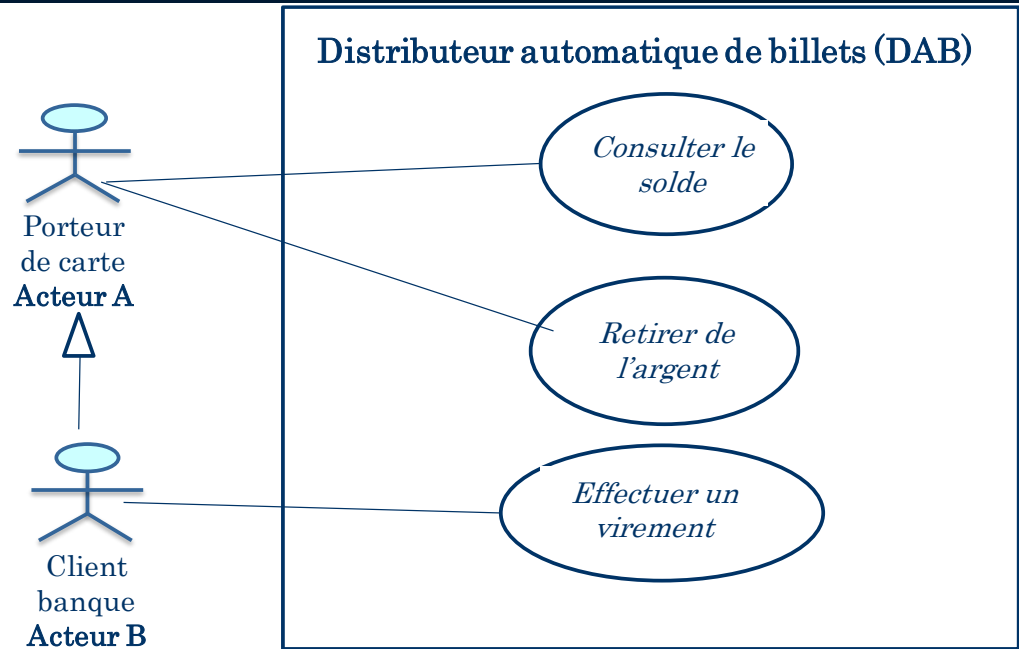
La relation de généralisation est représentée par une **flèche avec une extrémité triangulaire**.

Même comportement et même représentation graphique que la relation de généralisation entre deux cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### Exemple:



L'exemple montre que l'**acteur B** (client de la banque) est un acteur (porteur de carte) qui a une **capacité supplémentaire**. Celle d'*effectuer un virement*.

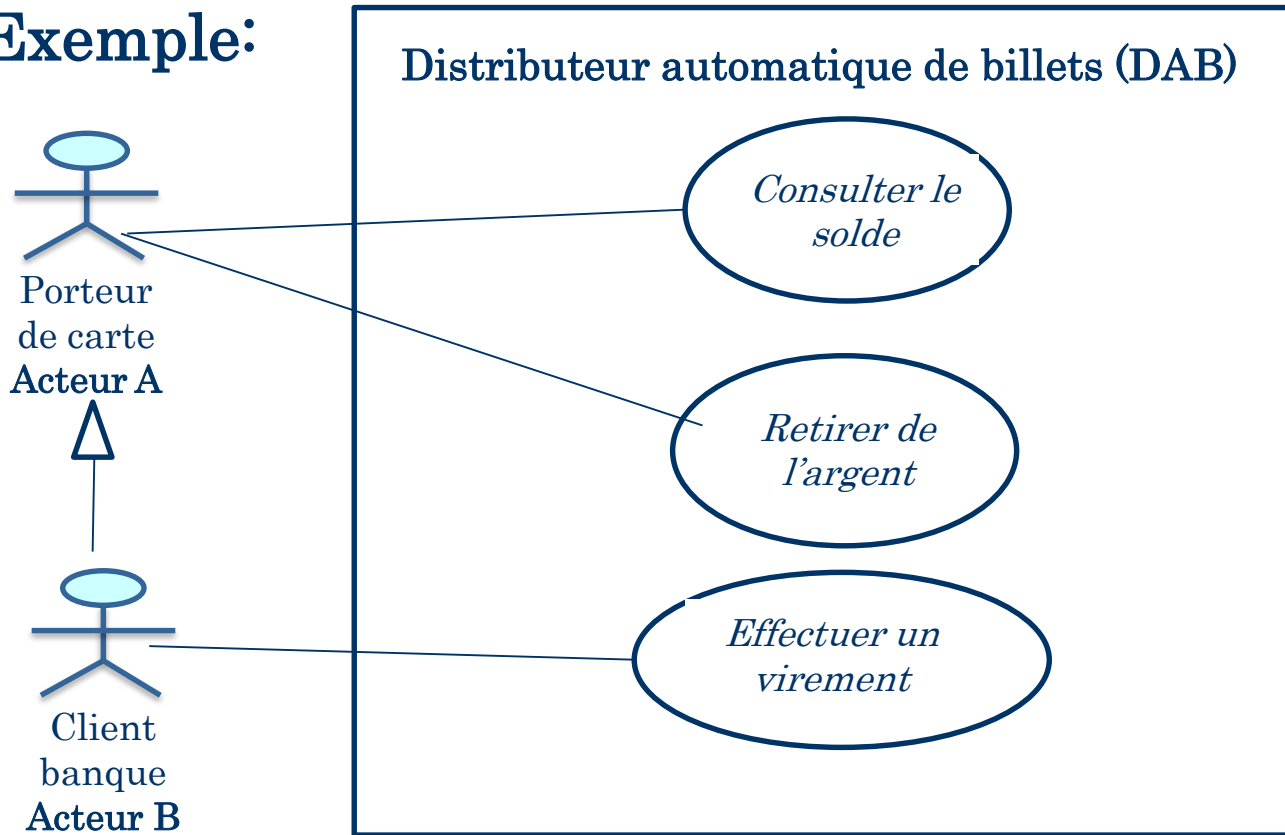
Puisque le *client banque* **hérite** de *porteur de carte bancaire*, il est également associé aux cas d'utilisation '*consulter le solde*', '*retirer de l'argent*' et '*effectuer un virement*'.

Par contre, l'*acteur porteur de carte* ne peut pas *effectuer de virement*.

# IV. UML

## Diagramme de cas d'utilisation

### Exemple:



L'exemple montre que l'**acteur B** (client de la banque) est un acteur (porteur de carte) qui a une **capacité supplémentaire**: celle d'effectuer un virement.

# IV. UML

## Diagramme de cas d'utilisation

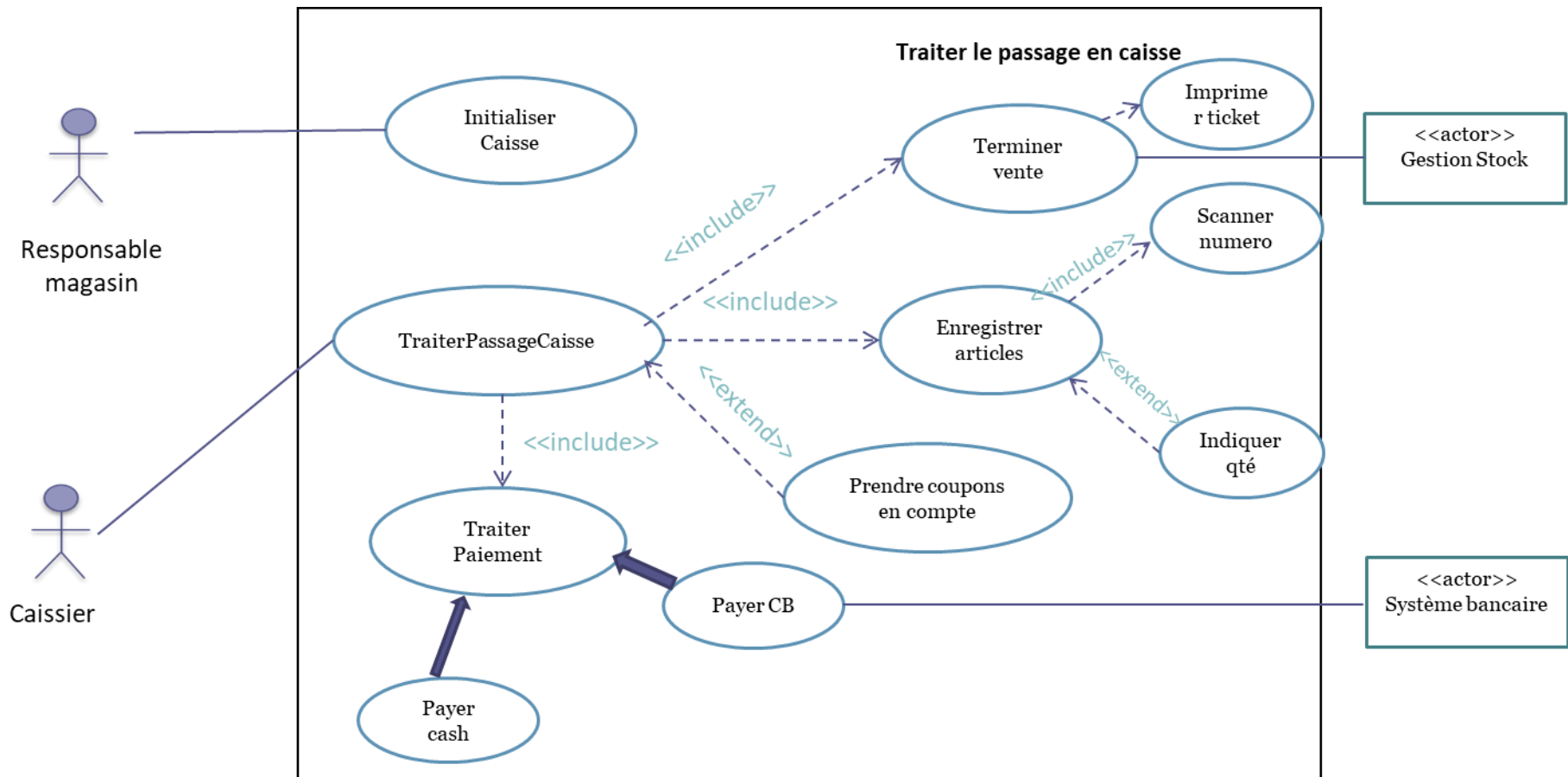
### 3. Relations dans les diagrammes de cas d'utilisation.

- a) Relations entre acteurs et cas d'utilisation.
- b) Relations entre cas d'utilisation.
- c) Relations entre acteurs.
- d) Exemple de diagramme de cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### d) Exemple de diagramme de cas d'utilisation.



# IV. UML

## Diagramme de cas d'utilisation

**Exemple:** Modélisez à l'aide d'un diagramme de cas d'utilisation un système informatique d'une agence de voyage.

Une agence de voyages organise des voyages où l'hébergement se fait en hôtel.

Le client doit disposer d'un taxi quand il arrive à la gare (le voyage se fait en train) pour se rendre à l'hôtel.



1. Identifier les acteurs.
2. Trouver les fonctionnalités.

# IV. UML

## Diagramme de cas d'utilisation

### 1. Identifier les acteurs:



*Utilisateur du logiciel.*

### 2. Trouver les fonctionnalités



Trois tâches = trois cas d'utilisations.



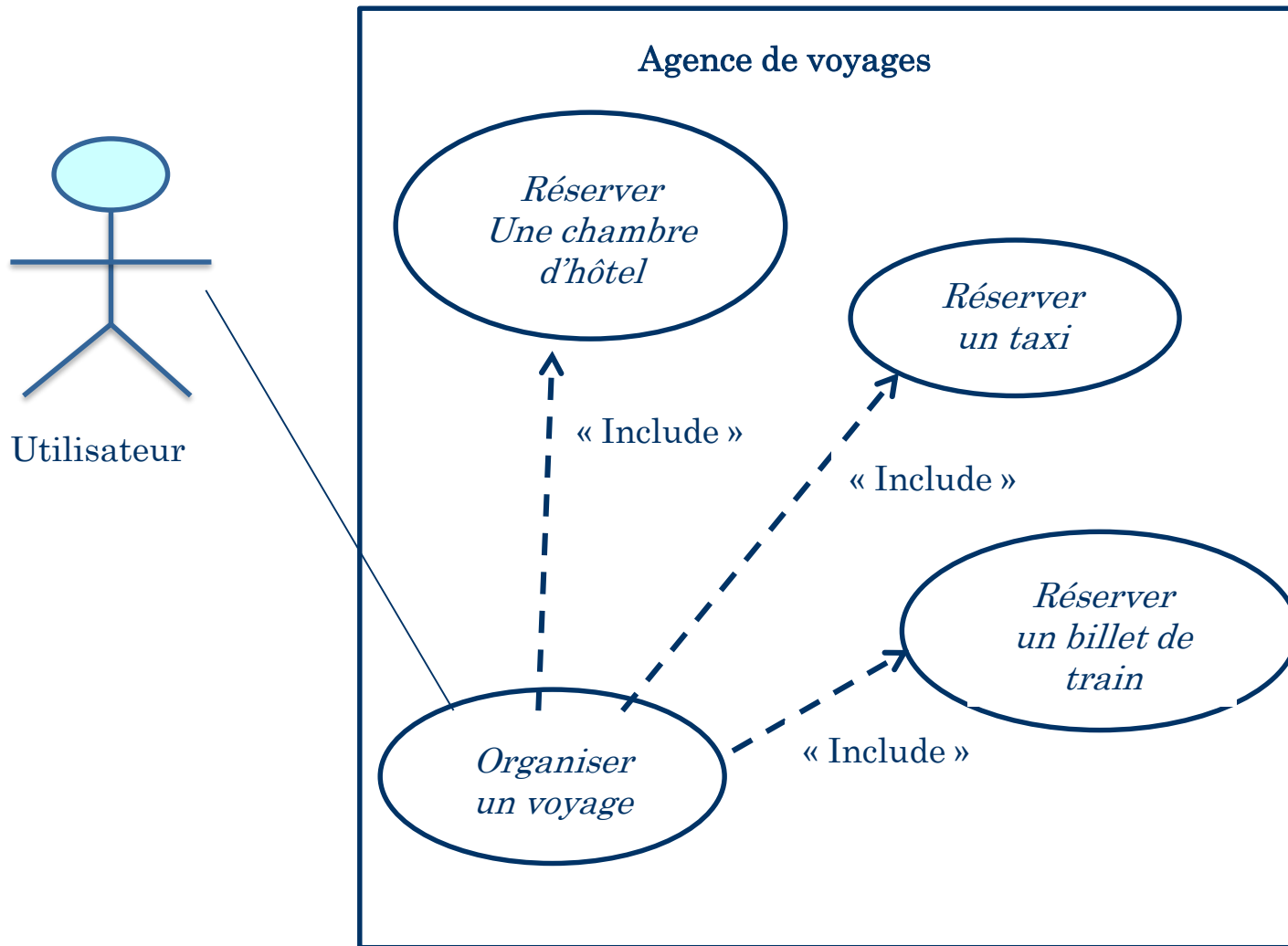
Pourquoi les 3 tâches sont des cas d'utilisations ?



Car ils *forment des transactions.*

# IV. UML

## Diagramme de cas d'utilisation





# IV. UML

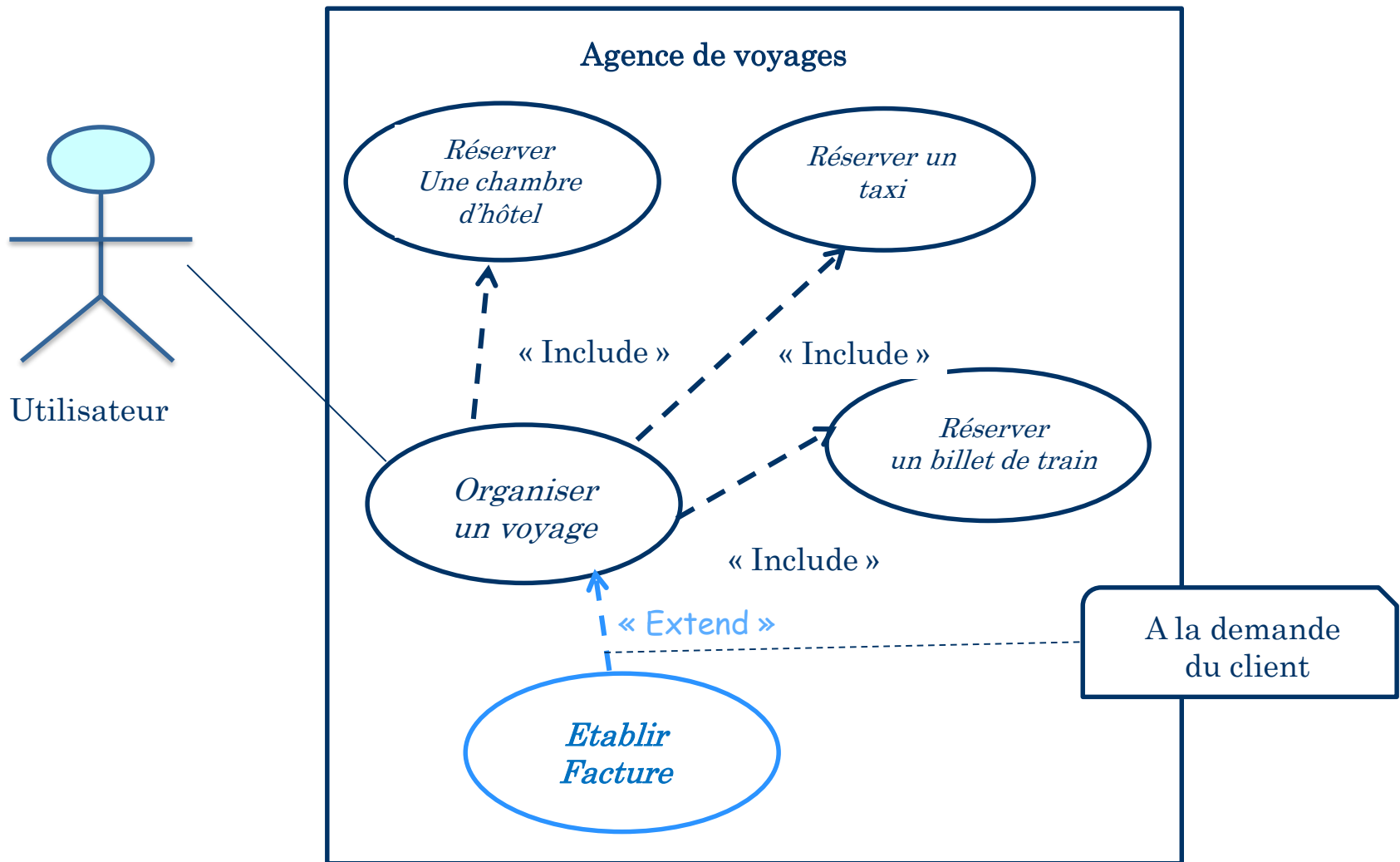
## Diagramme de cas d'utilisation

### **Exemple suite:**

Certains clients demandent à l'agence de voyage une facture détaillée. Modifier le diagramme de cas d'utilisation précédent et ajouter cette nouvelle information.

# IV. UML

## Diagramme de cas d'utilisation



# IV. UML

## Diagramme de cas d'utilisation

### **Exemple suite:**

Le voyage peut se faire soit en train soit en avion.  
Modélisez cette nouvelle fonctionnalité.

# IV. UML

## Diagramme de cas d'utilisation

### Exemple suite:

Le voyage peut se faire soit en train soit en avion.  
Modélisez cette nouvelle fonctionnalité.

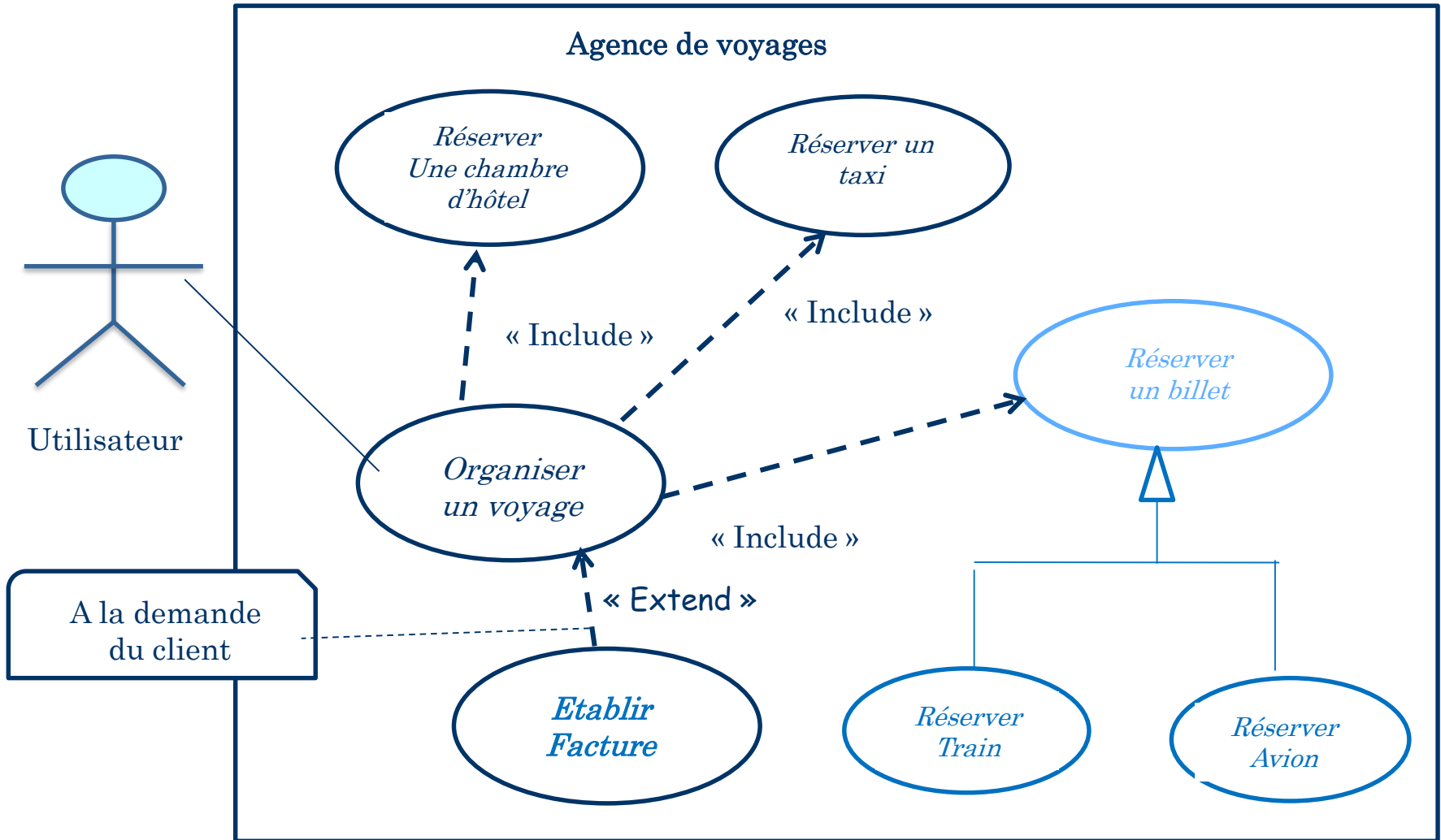


Deux cas particuliers qui **dérivent** du cas  
« *réserver un billet* » en général.

1. « réserver un billet de train ».
2. « réserver un billet d'avion ».

# IV. UML

## Diagramme de cas d'utilisation



# IV. UML

## Diagramme de cas d'utilisation

1. Objectif.
2. Les éléments des diagrammes de cas d'utilisation.
3. Relations dans les diagrammes de cas d'utilisation.
4. **Modélisation des besoins et diagrammes de cas d'utilisation.**

# IV. UML

## Diagramme de cas d'utilisation

### 4. Modélisation des besoins et diagrammes de cas d'utilisation.

- a) Comment identifier les acteurs?
- b) Comment recenser les cas d'utilisation?
- c) Description textuelle des cas d'utilisation.
- d) Exercices.

# IV. UML

## Diagramme de cas d'utilisation

### a) Comment identifier les acteurs?

Pour trouver les acteurs du système il faut:

- ☐ Identifier quels sont les différents rôles que vont devoir jouer ses utilisateurs.  
(porteur de carte bancaire, client de la banque, internaute, etc.)
- ☐ Identifier les autres systèmes avec lesquels le système va devoir communiquer.  
(les périphéries : imprimantes, des logiciels à intégrer dans le projet, etc.)



# IV. UML

## Diagramme de cas d'utilisation

### a) Comment identifier les acteurs?

**QUI** (extérieur au système) **interagit avec le système?**

**Acteur = ensemble cohérent de rôles joués vis-à-vis du système**



#### Remarque importante:

Un cas d'utilisation (s'il n'est pas interne) a toujours **au moins un acteur principale** pour qui le système produit un résultat observable.

**Un seul acteur**  
**= les clients de la banque**

# IV. UML

## Diagramme de cas d'utilisation

### 4. Modélisation des besoins et diagrammes de cas d'utilisation.

- a) Comment identifier les acteurs?
- b) Comment recenser les cas d'utilisation?
- c) Description textuelle des cas d'utilisation.
- d) Exercices.

# IV. UML

## Diagramme de cas d'utilisation

### b) Comment recenser les cas d'utilisation?

Pour recenser les cas d'utilisation il faut:

Se placer au point de vue de chaque acteur et déterminer:

- Comment il se sert du système?
- Dans quels cas il l'utilise?
- A quelles fonctionnalités il doit avoir accès?

# IV. UML

## Diagramme de cas d'utilisation

### b) Comment recenser les cas d'utilisation?

**Exemple:** Système de réservation de billets de train via des bornes situées dans les gares.

**Acteur** = personne qui souhaite obtenir un billet.



**Liste des cas d'utilisation possibles:**

- Rechercher un voyage.
- Réserver une place dans un train.
- Acheter le billet.

# IV. UML

## Diagramme de cas d'utilisation

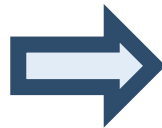
### b) Comment recenser les cas d'utilisation?



- Eviter les redondances.
- Limiter le nombre de cas en se situant au bon niveau d'abstraction.

#### Remarque importante:

Il n'y a pas de notion temporelle dans un diagramme de cas d'utilisation.



Description textuelle des cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### 4. Modélisation des besoins et diagrammes de cas d'utilisation.

- a) Comment identifier les acteurs?
- b) Comment recenser les cas d'utilisation?
- c) Description textuelle des cas d'utilisation.
- d) Exercices.

# IV. UML

## Diagramme de cas d'utilisation

### c) Description textuelle des cas d'utilisation.

Le diagramme de cas d'utilisation décrit les **grandes fonctions** d'un système du point de **vue des acteurs**.

**MAIS pas le détail** au niveau du dialogue entre les acteurs et les cas d'utilisation.



Description textuelle des cas d'utilisation.

# IV. UML

## Diagramme de cas d'utilisation

### c) Description textuelle des cas d'utilisation.

#### POURQUOI?

- Identifier les **interactions** du système avec son environnement humains et autres systèmes.
- Identifier les **besoins** c'est-à-dire les fonctionnalités du système.
- Identifier les **dépendances** entre les fonctionnalités?

#### OUI MAIS

Un diagramme de cas d'utilisation ne donne pas d'information sur le temps.

*Grâce aux scénarios on va pouvoir spécifier l'enchaînement temporel des cas d'utilisation et des interactions entre les acteurs et le système.*



# IV. UML

## Diagramme de cas d'utilisation

PAS SEULEMENT

Abstrait

Modèle des besoins

Modèle d'analyse

Modèle de conception

Modèle de déploiement

Concret

Aspects  
fonctionnels

Aspects  
statiques

Aspects  
dynamiques

Diag. de cas  
d'utilisation

Scénarios

Diag.  
d'activités

Diag. de  
séquence

### Remarque importante:

Une des forces de la notation UML est de proposer de nombreux types de diagrammes qui mettent en avant des aspects différents d'une description.

# IV. UML

## Diagramme de cas d'utilisation

### COMMENT?

I: <u>Identification du cas</u>	
Numéro	<i>Numéro du cas</i>
Nom	<i>Utiliser une tournure à l'infinitif(ex: réceptionner un colis)</i>
Objectif	<i>Une description résumée permettant de comprendre l'intention principale du cas d'utilisation.</i>
Acteurs principaux	<i>Ceux qui vont réaliser le cas d'utilisation.</i>
Acteurs secondaires	<i>Ceux qui ne font que recevoir des informations à l'issue de la réalisation de cas d'utilisation.</i>
Dates	<i>Les dates de création et de mise à jour de la description courante.</i>
Responsable	<i>Le nom des responsables.</i>
Version	<i>Le numéro de la version</i>
Les préconditions	<i>Elles décrivent dans quel état doit être le système avant que ce cas d'utilisation puisse être déclenché.</i>

# IV. UML

## Diagramme de cas d'utilisation

### COMMENT?

#### II: Scénarios

Un scénario est une instance d'un cas d'utilisation dans lequel tous les paramètres ont été fixés.

Scénario nominal	<i>Correspond à un déroulement classique et sans erreur. Permet d'arriver au résultat souhaité. Donne en français la suite des étapes du scénario sous la forme d'une liste numérotée.</i>
Scénarios alternatifs	<i>Sont des variantes du scénario nominal. C'est un embranchement dans une séquence nominale (et y revient).</i>
Scénarios d'exceptions	<i>Intervient quand une erreur se produit. Le séquençement nominal s'interrompt, sans retour à la séquence nominale.</i>
Les post conditions	<i>Elles indiquent dans quel état se trouve le système après le déroulement de la séquence nominale.</i>

# IV. UML

## Diagramme de cas d'utilisation

COMMENT?

*Description [optionnelle] des cas d'utilisation:  
spécifications non fonctionnelles  
(techniques,...).*

# IV. UML

## Diagramme de Cas d'utilisation



*Pour élaborer un diagramme de cas d'utilisation il faut:*

- *Identifier les acteurs et les classifier.*
- *Trouver les cas d'utilisation .*
- *Établir les relations entre les cas d'utilisation et les acteurs.*
- *Identifier les relations de dépendances entre les cas d'utilisation.*
- *Vérifier s'il est possible de simplifier le diagramme avec la relation d'héritage.*
- *Identifier les cas d'utilisation les plus complexes pour les détailler avec les descriptions textuelles des cas d'utilisation.*

# IV. UML

## Diagramme de cas d'utilisation

### 4. Modélisation des besoins et diagrammes de cas d'utilisation.

- a) Comment identifier les acteurs?
- b) Comment recenser les cas d'utilisation?
- c) Description textuelle des cas d'utilisation.
- d) Exercices.

# Bibliographie

1. Benoît CHARROUX, Aomar OSMANI, Yann THIERRY-MIEG. UML2 Pratique de la modélisation. 3<sup>ème</sup> édition. PEARSON.
2. Laurent AUDIBERT. UML2 de l'apprentissage à la pratique. 2<sup>ème</sup> édition. ELLIPSES.
3. Christian SOUTOU. Modélisation des bases de données (UML et les modèles entité-association). 3<sup>ème</sup> édition. EYROLLES.
4. Chantal MORLEY, Jean HUGUES, Bernard LEBLANC. 4<sup>ème</sup> édition. UML 2 pour l'analyse d'un système d'information. DUNOD.
5. Hugues BERSINI. L'orienté objet. 3<sup>ème</sup> édition. EYROLLES.
6. Laurent DEBRAUWER, Fien VAN DER HEYDE. UML 2.5. 4<sup>ème</sup> édition. ENI Editions.
7. Jean-Luc HAINAUT. Bases de données concepts, utilisation et développement. DUNOD.
8. Gilles ROY. Conception de bases de données avec UML. Presses de l'université du Québec.
9. Craig LARMAN. UML2 et les design patterns. 3<sup>ème</sup> édition. PEARSON Education.
10. Frank BARBIER. UML 2 et MDE. DUNOD.
11. Laurent DEBRAUWER, Naouel KARAM. UML 2 entraînez-vous à la modélisation. Seconde édition. ENI Editions.
12. Corine COSTA. Cours Projets et bureau d'études.