

Analyse et Conception avec UML

# Les diagrammes de cas d'utilisation

[blay@unice.fr](mailto:blay@unice.fr)

[www.polytech.unice.fr/~blay](http://www.polytech.unice.fr/~blay)

IUT Nice-Sophia Antipolis

février 2012

Site web du module :

<http://anubis.polytech.unice.fr/iut/>

Résumé du cours de SI  
puis extension

# Bibliographie

Principalement :

- cours IBM : Writing Good Use Cases (2006)
- Voir sur le site web les autres cours.
- Méthodologie en Ingénierie du logiciel, Modélisation Orientée objet, M.Grimaldi – janvier 2010

# Plan du cours

- Utilisation d'un dictionnaire du domaine
- Des cas d'utilisations (Use-cases/UC)
  - Acteurs, use-cases, system UML
- Processus de construction des Uses-case
- Compléments

# UML au travail : Guichet automatique de banque

Le guichet automatique d'une banque (GAB) offre les services suivants :

- Distribution d'argent à partir d'une carte de la banque ou d'une carte Visa.
- Consultation de solde de compte, dépôt en numéraire et dépôt de chèques pour les clients de la banque porteurs d'une carte de la banque.

De plus,

- Toutes les transactions sont sécurisées.
- Il est parfois nécessaire de recharger le distributeur, .



# UML au travail : Une ludothèque

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
- (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- (3) Les jeux empruntés sont rendus à un conseiller....
- (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe
- (8) Un internaute peut consulter les jeux et s'inscrire.

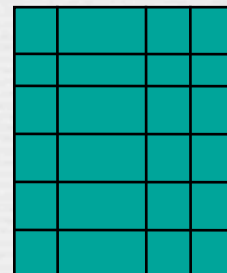
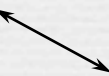
# Intérêt du dictionnaire

- Outil de dialogue
- Informel, évolutif, simple à réaliser
- Etablir et figer la terminologie
  - Permet de figer la terminologie du domaine d'application.
  - Constitue le point d'entrée et le référentiel initial de l'application ou du système.

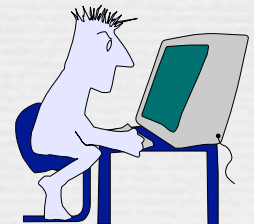
Homonymie

Synonymie

Polysémie



6



# UML au travail : Une ludothèque

- Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- Les jeux empruntés sont rendus à un conseiller....
- Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- Un adhérent peut payer sa cotisation en ligne par un système de paiement externe
- Un internaute peut consulter les jeux et s'inscrire.

# UML au travail : Une ludothèque

- Les **adhérents** peuvent emprunter des **jeux** en s'adressant à un **conseiller** qui enregistre l'**emprunt**.
- Les jeux empruntés sont rendus à un conseiller....
- Un adhérent peut réserver des jeux. Une **réservation** précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- Pour organiser un **événement** le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- Un adhérent peut s'inscrire pour participer à l'événement en en faisant la demande à un conseiller spécialisé, à condition qu'il y ait encore de la place.
- Un adhérent peut payer sa cotisation en ligne par un **système de paiement externe**
- Un internaute peut consulter les jeux et s'inscrire.



# UML au travail : Une ludothèque

**Adhérents** : Personne connue du système *par son nom, prénom, date de naissance, adresse postale, date du dernier paiement de la cotisation et adresse email éventuelle.*

**Conseiller** : Personne *identifiée* qui a des droits sur le système.

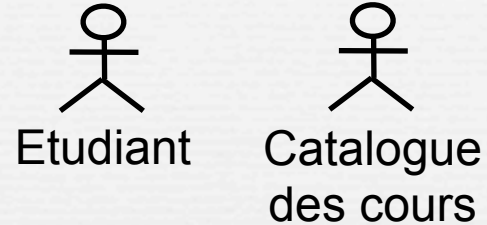
**Réservation** : Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.

**Événement** : temps de démonstration et de jeux organisé à l'avance. Il précise les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.

**Jeux** : Un jeu est caractérisé par un identifiant, un nombre de joueurs, ...

# Processus d'écriture des UC

Trouver les acteurs



Trouver les UC

Enregistrer des cours

**Brève description:** Ce UC permet à un étudiant d'enregistrer ses cours... Seuls les formations bien construites sont acceptés. Le catalogue des cours est notifié des inscriptions.

Décrire les UC

**Description de «Enregistrer des cours»**

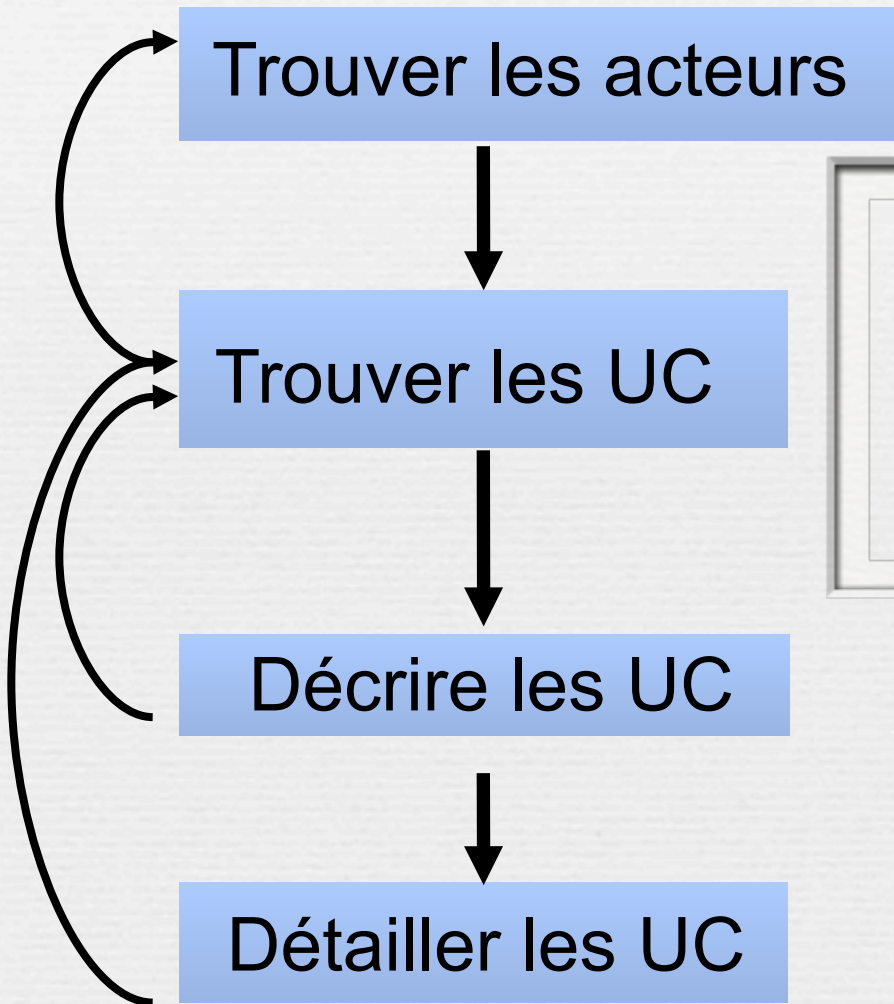
- Flot d'évènements
- Pas à pas

Détailler les UC

**Specification de «Enregistrer des cours»**

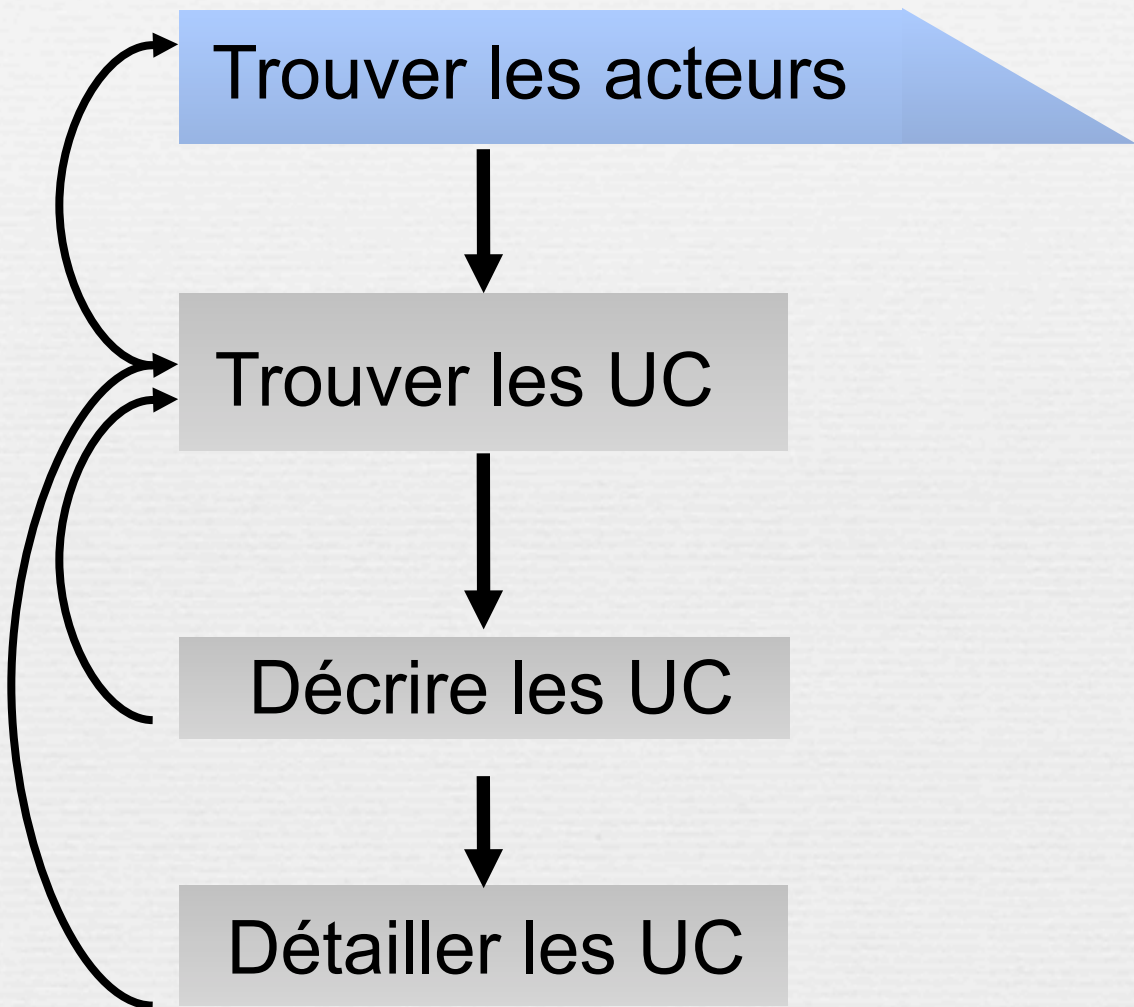
- Flot d'évènements détaillés
- Exigences spéciales
- Pre/Postconditions

# Processus d'écriture des UC



**Important**  
C'est un processus *itératif*

# Processus d'écriture des UC



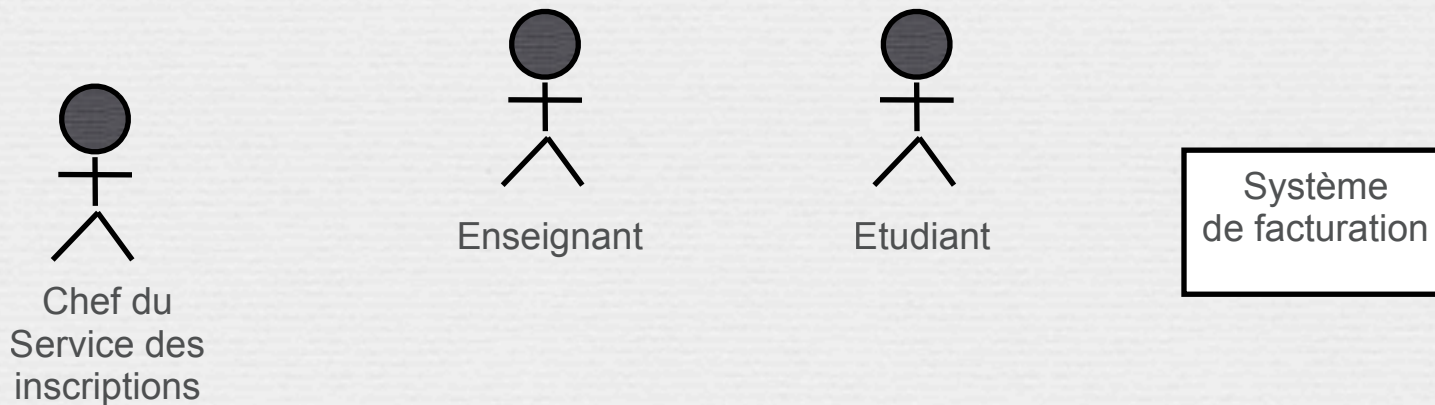
► Nommer et brièvement décrire les acteurs trouvés

# Définir le périmètre du SI : Acteurs

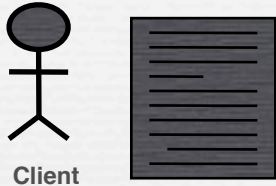
## Définir les acteurs **externes**

- physiques et logiques
- rôle et entité concrète

« Un acteur est une personne ou une chose qui va interagir avec le système »



# Description des acteurs



Client



Pour chaque acteur :

- choisir un identificateur représentatif de son **rôle** (*un bon nom décrit la responsabilité des acteurs*)
- donner une brève description textuelle



Guichetier

Un guichetier est un employé de la banque chargé de faire l'interface entre le système informatique et les clients qu'il reçoit au comptoir. Le guichetier peut réaliser les opérations courantes : création d'un compte, dépôt et retrait d'argent, etc.

# Trouver les acteurs

- Qui ou quoi utilise le système?
- Qui ou quoi obtient de l'information de ce système ?
- Qui ou quoi fournit des informations au système ?
- Où dans la compagnie le système est-il utilisé ?
- Qui ou quoi supporte et maintient le système?
- Quels autres systèmes utilisent ce système?

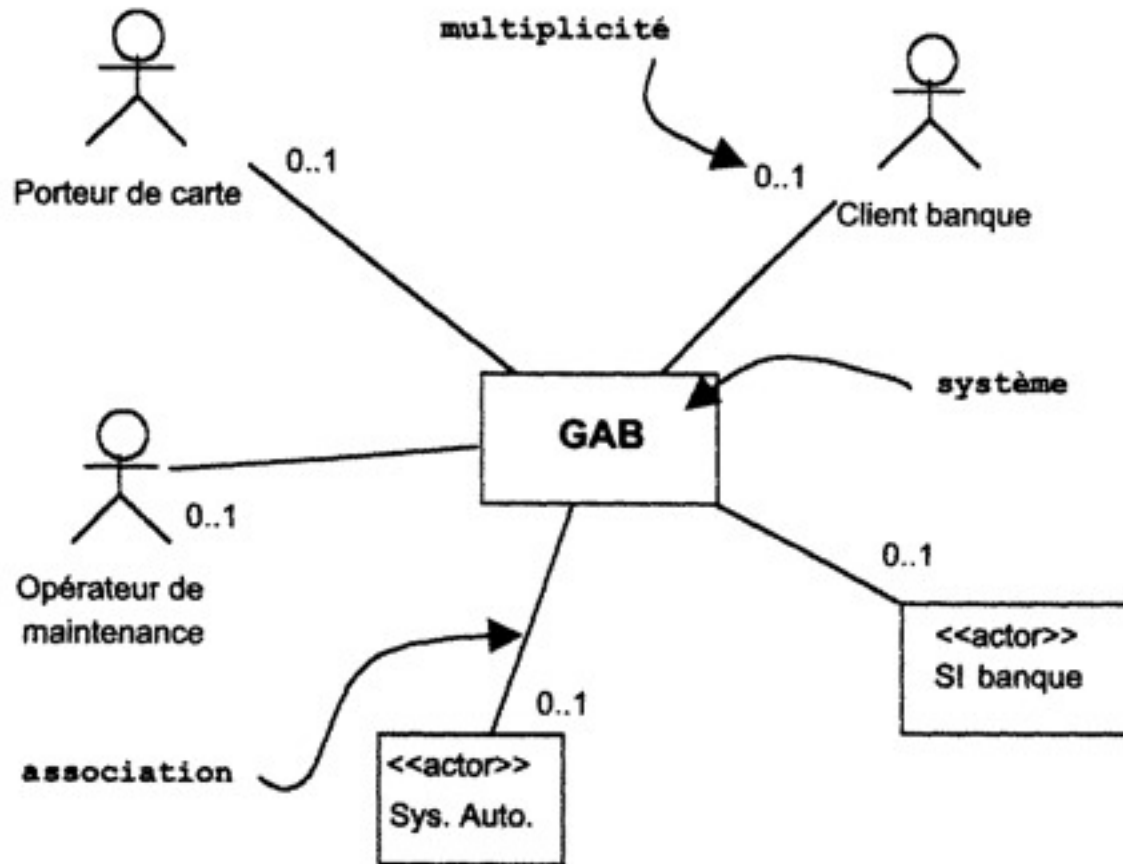
# UML au travail : Une ludothèque

- Les **adhérents** peuvent emprunter des **jeux** en s'adressant à un **conseiller** qui enregistre l'**emprunt**.
- Les jeux empruntés sont rendus à un conseiller....
- Un adhérent peut réserver des jeux. Une **réservation** précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- Pour organiser un **événement** le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- Un adhérent peut s'inscrire pour participer à l'événement en en faisant la demande à un conseiller spécialisé, à condition qu'il y ait encore de la place.
- Un adhérent peut payer sa cotisation en ligne par un **système de paiement externe**
- Un internaute peut consulter les **jeux** et s'inscrire.



# Diagramme de contexte

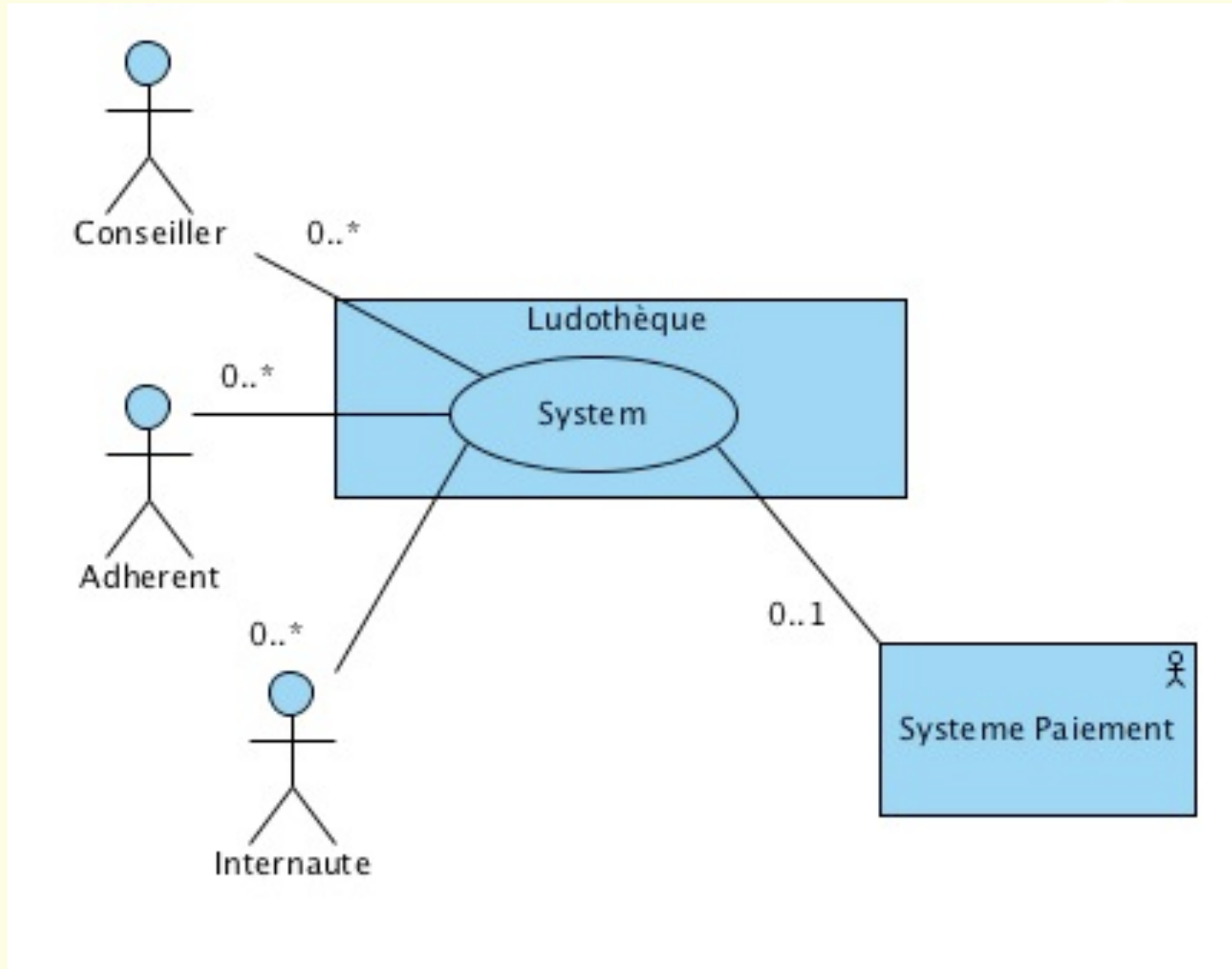
# Diagramme de contexte statique



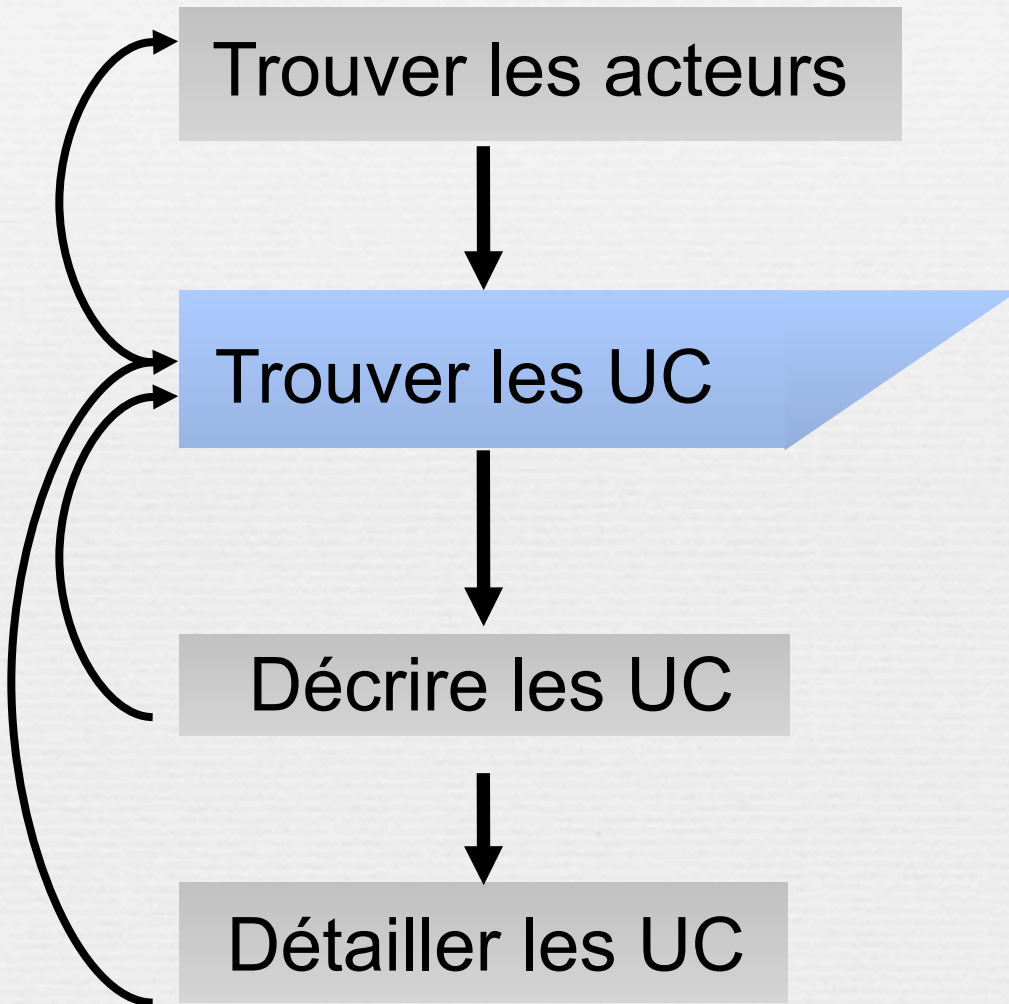
# UML au travail : Une ludothèque

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
- (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- (3) Les jeux empruntés sont rendus à un conseiller....
- (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe

# UML au travail : Une ludothèque



# Processus d'écriture des UC



- ▶ Nommer et brièvement décrire les UC trouvés
- ▶ Créer un diagramme de UC
- ▶ Etablir la plus-value métier et les risques techniques des UC

# Cas d'utilisation (UC)

- Un cas d'utilisation est un motif cohérent de comportement
  - réalisé par le système.
- Chaque cas d'utilisation est décrit par une séquence d'actions connectées, effectuées par un dialogue entre des acteurs et le système
  - qui produit un résultat observable
  - d'intérêt pour un ou plusieurs acteurs du système.
  - ne révèle pas la structure interne du système.
- Chaque cas d'utilisation est un flot complet et faisant du sens du point de vue d'un acteur particulier.

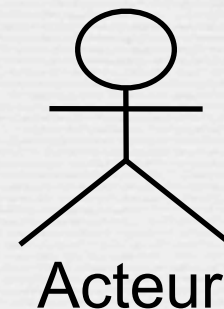
# Trouver les use-cases



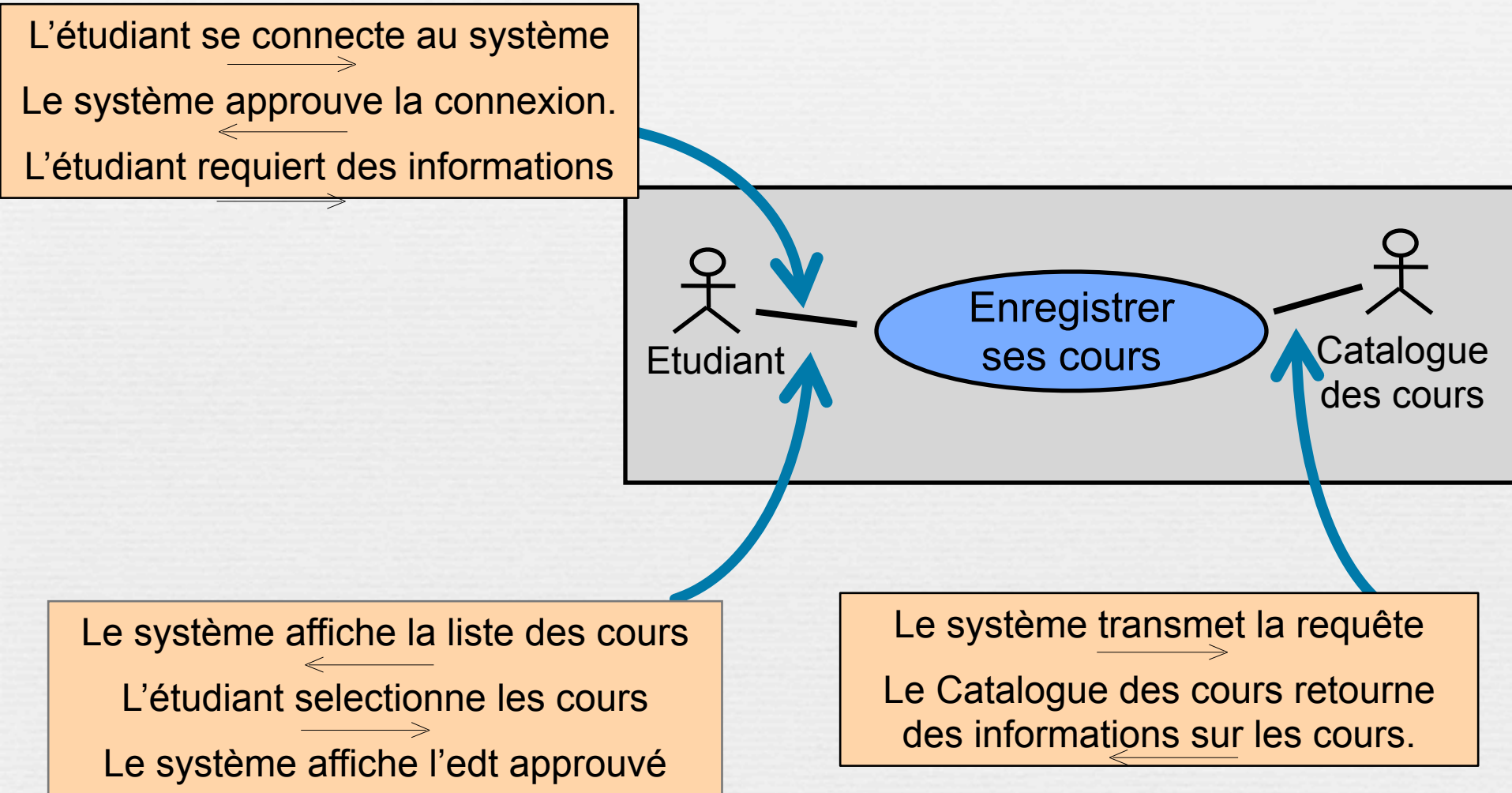
Quels sont les objectifs de chaque acteur?

- Pourquoi l'acteur utiliserait-il le système?
- Est-ce que l'acteur créera, stockera, modifiera, supprimera ou lira des données dans le système? Si oui, pourquoi?
- Est-ce que l'acteur nécessite d'informer le système sur des événements externes ou des changements?
- Est-ce que l'acteur doit être informé de certains événements dans le système?

Quels buts dois-je atteindre en utilisant le système?



# Communication : un dialogue





# Nommer un use-case

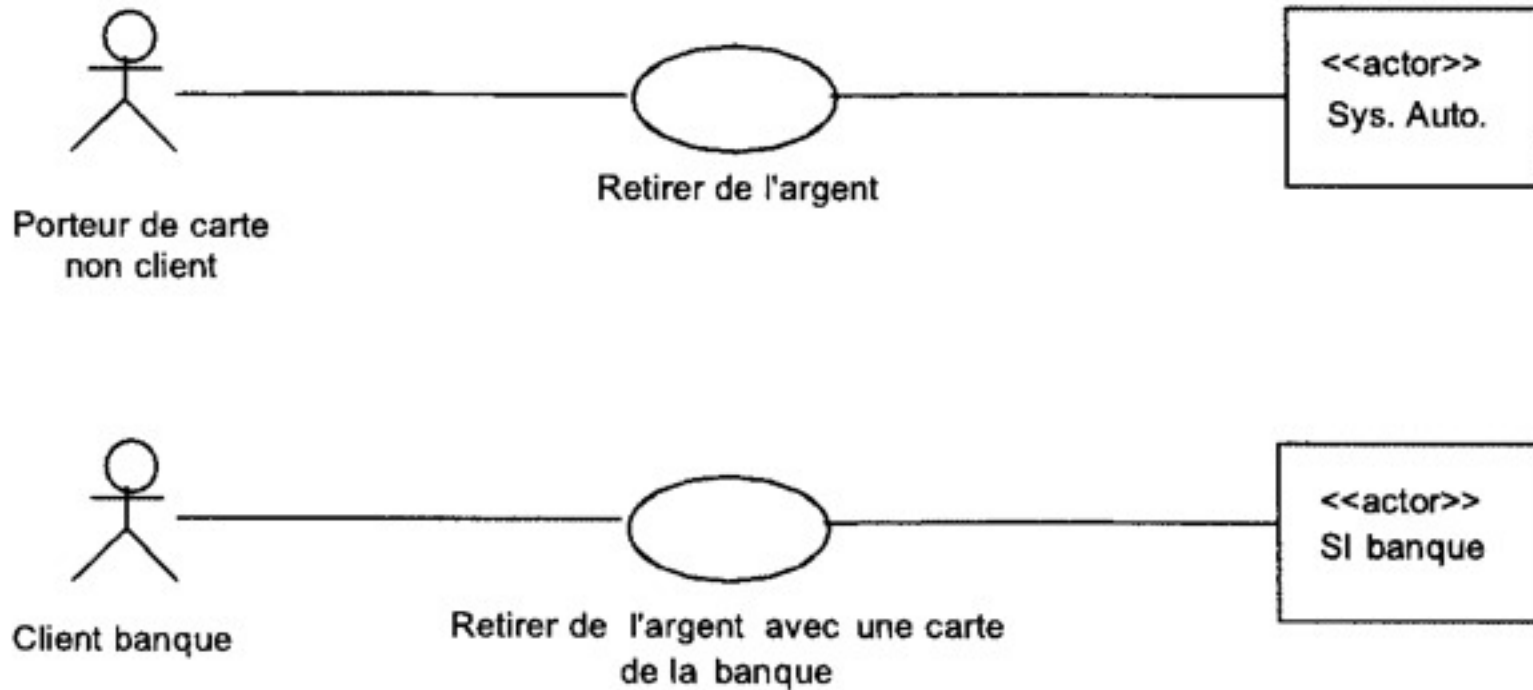
- Le nom doit être unique, intuitif et auto-explicatif
- Il doit commencer par un **verbe** et utiliser une simple combinaison **verbe-nom**
- Définir clairement et sans ambiguïté le gain des résultats observables
- Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation
- Décrire le comportement fournit par le cas d'utilisation



?

- Enregistrer des Cours
- Enregistrement de cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

# Diagramme des cas d'utilisation amélioration



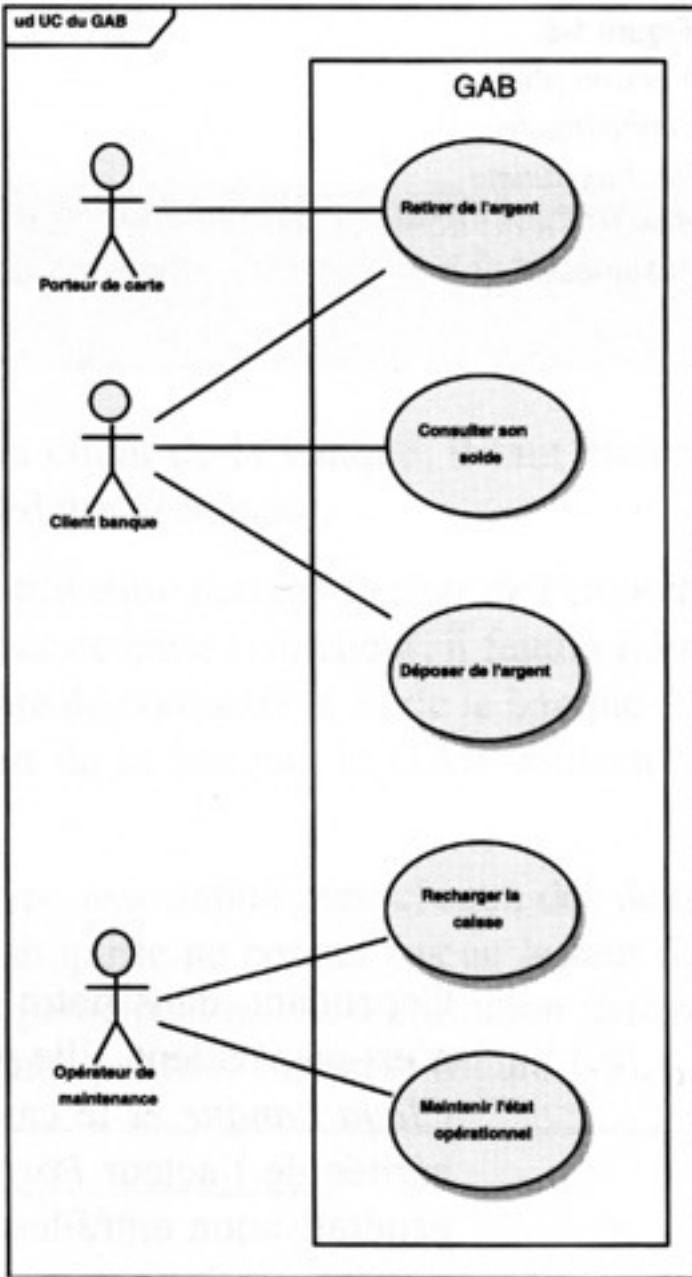
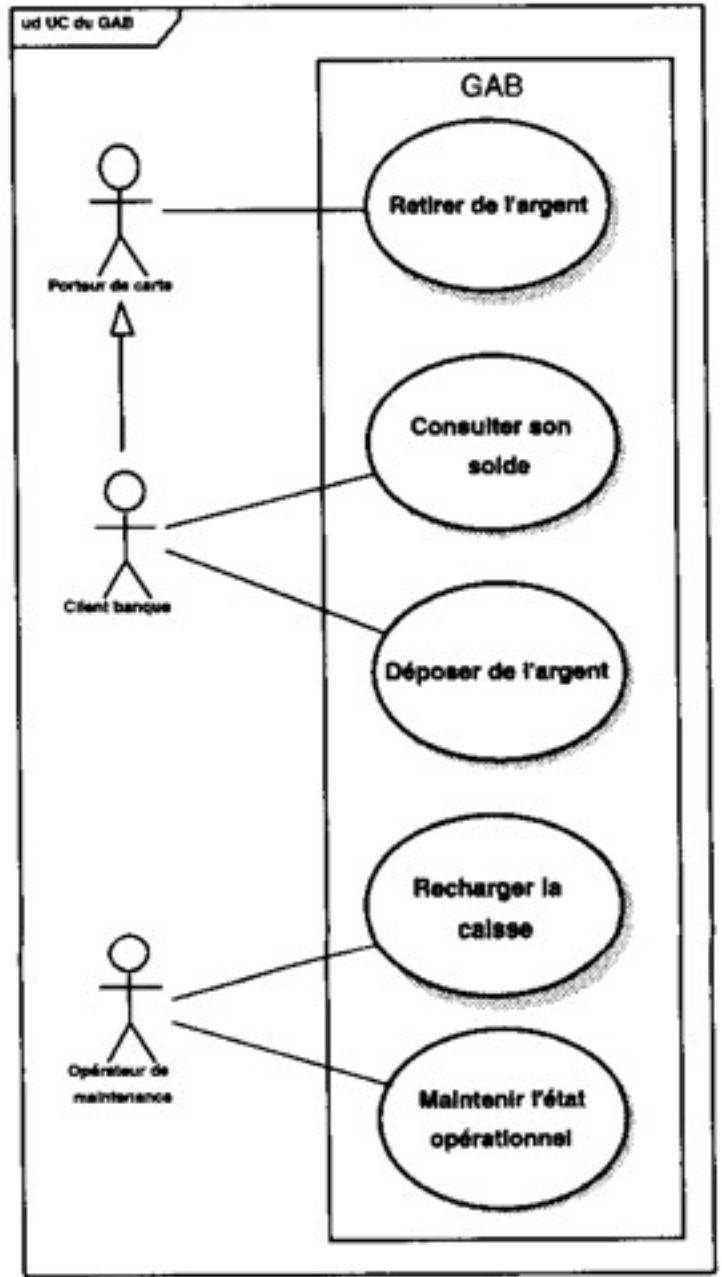


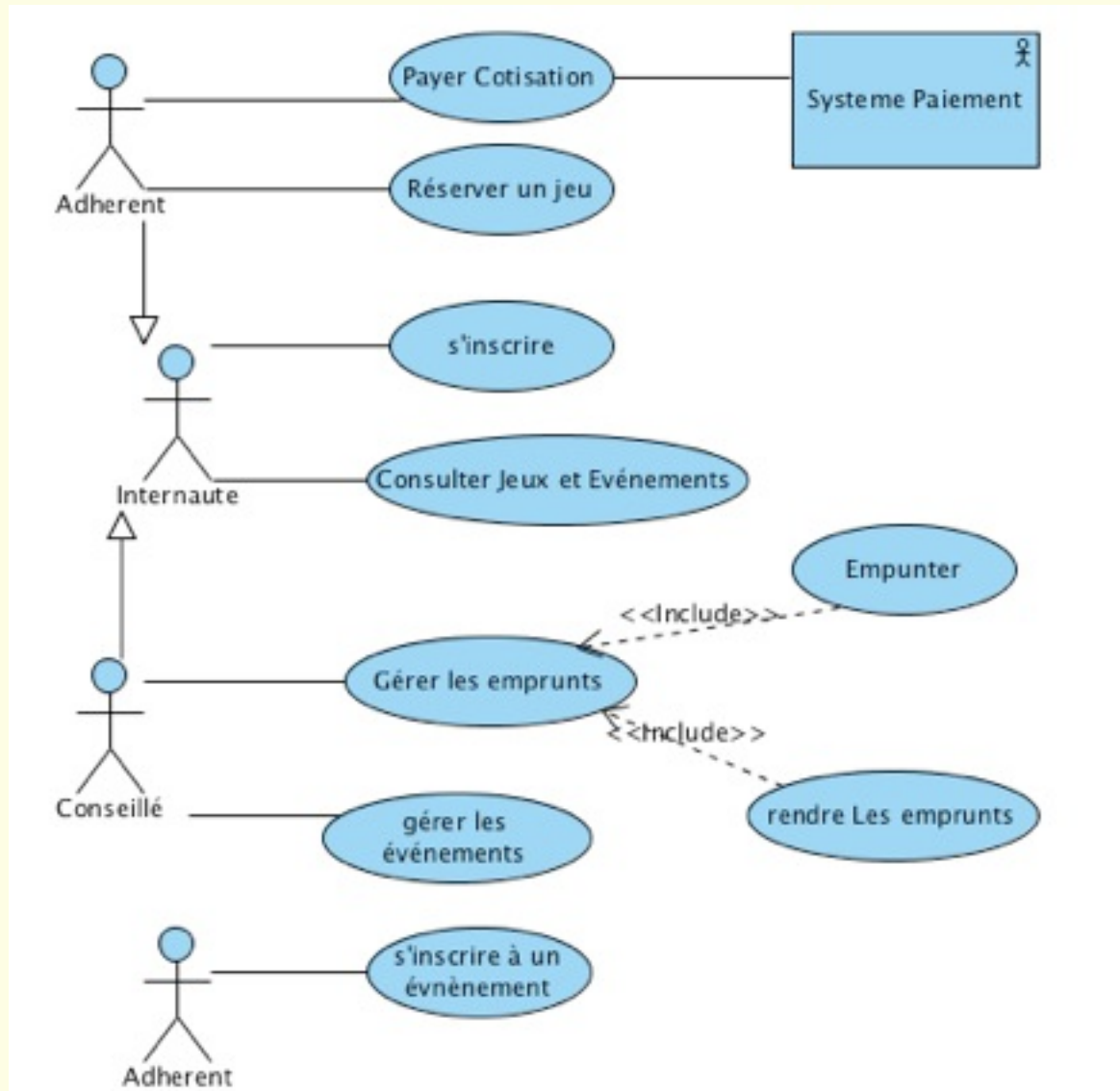
Diagramme des cas d'utilisation sans les acteurs secondaires



# UML au travail : Une ludothèque

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
- (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- (3) Les jeux empruntés sont rendus à un conseiller....
- (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe

# UML au travail : Une ludothèque



# Description succincte d'un UC

Sommaire d'identification :

- Titre
- Résumé
- Acteurs
- Date de création
- Date de mise à jour
- Version
- Responsable

# UML au travail : Une ludothèque

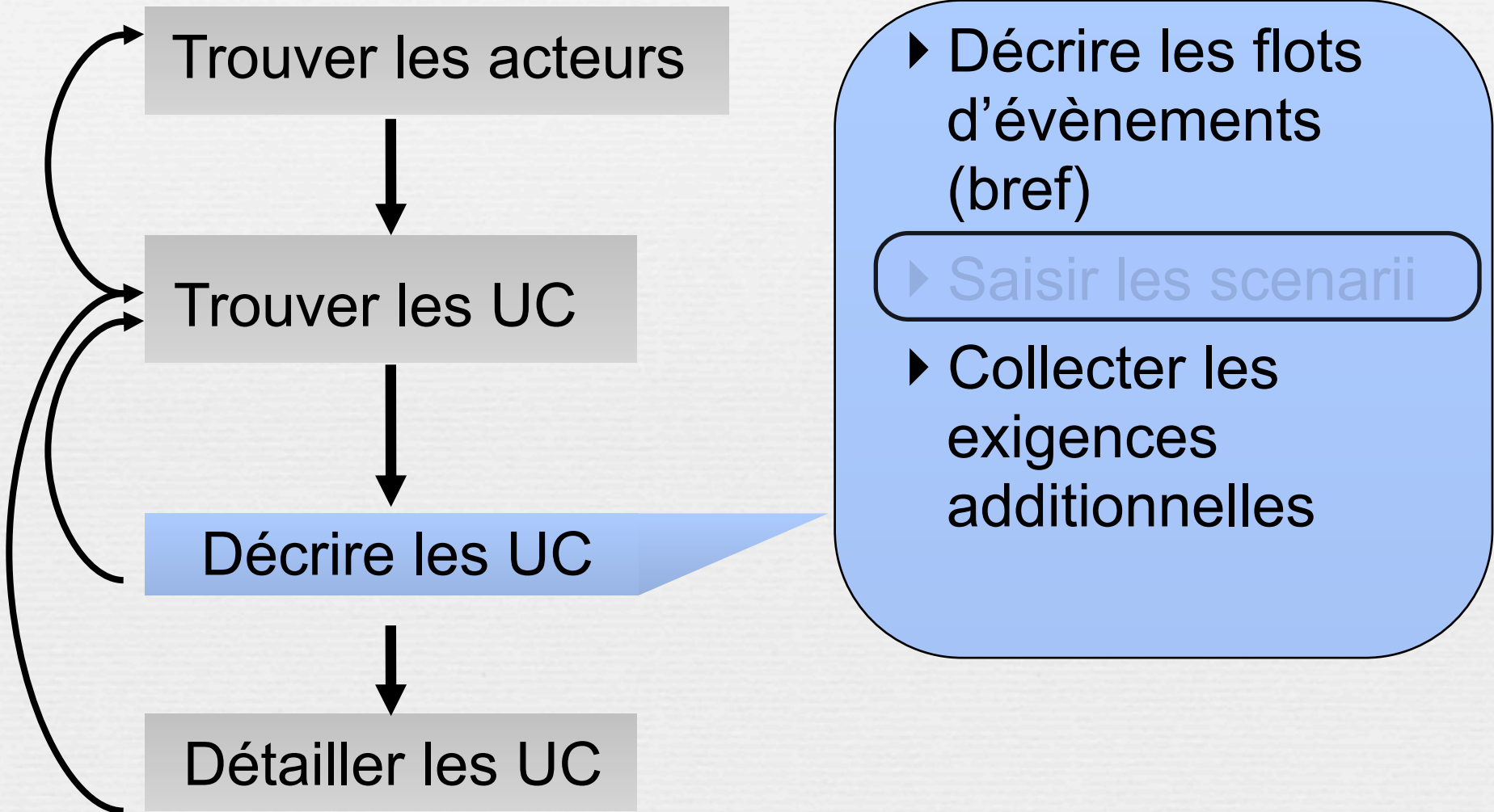
## Emprunter un jeu

**Acteurs** : Conseiller

**Objectif** : Décrire l'emprunt d'un jeu par un **Conseiller**

**Aperçu** : Un **Conseiller** passe le lecteur de code barre sur le jeu; Le **Conseiller** saisit l'identifiant de l'adhérent. Le système vérifie la droit d'emprunter de l'adhérent. Il enregistre l'emprunt.

# Processus d'écriture des UC





# Décrire une UC



Décrire chaque étape du UC par des phrases courtes, organisées séquentiellement.

Numéroter et nommer les étapes.



## Use Case Name

**Brief Description**

**Basic Flow**

1. First step
2. Second step
3. Third step

**Alternative Flows**

1. Alternative flow 1
2. Alternative flow 2
3. Alternative flow 3



Structurer le flot de base en étapes majeures



Identifier les flots alternatifs.

# UML au travail : Une ludothèque

## Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 2) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

## Flot alternatif:

- A4a) L'adhérent n'a pas payé sa cotisation.
- A4b) adhérent sous pénalité de retard

...

# Décrire les UC

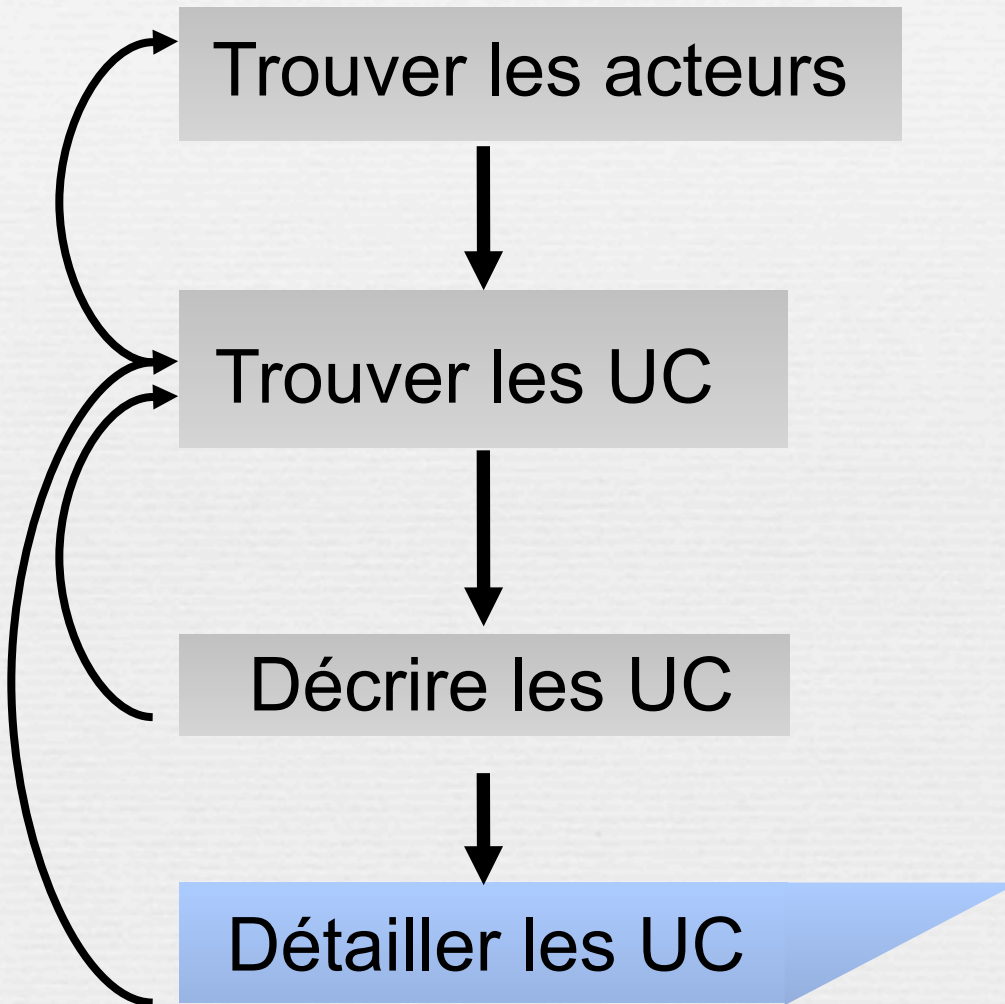
- Un processus itératif : ne pas tout détailler, pas trop tôt
- Un processus de découverte : Décrire vous aide à découvrir ce que vous ne connaissez pas. Une brève description sert de point de départ.
- Un processus d'évaluation : UC trop petit ou trop gros ? partagé?

# Exigences additionnelles



Collecter les exigences système qui ne peuvent pas être allouées à des UC spécifiques dans des documents additionnels.

# Processus d'écriture des UC



- ▶ Détailler les flots d'événements
- ▶ Structurer les flots d'événements
- ▶ Spécifier les propriétés additionnelles

# Description détaillée d'un UC

- Description des scénarios:

 Préconditions

- Scénario Nominal
- Flots alternatifs
- Flots d'erreur
- Postconditions
- Exigences non fonctionnelles
- Besoins d'IHM

# Préconditions

- Décrire l'état dans lequel doit être le système avant que le UC puisse commencer.
  - ▶ Simples déclarations qui définissent l'état du système, exprimées comme ses conditions qui doivent être remplies
  - ▶ Il ne faut jamais se référer à d'autres UC qui doivent être effectuées avant cet UC
  - ▶ Devraient être énoncées clairement et devraient être facilement vérifiables
- **Facultatif:** Utilisez uniquement si nécessaire pour clarifier
- **Exemple**
  - ▶ UC «Inscription à des cours»
  - ▶ *Préconditions*
    - ▶ La liste des offres de cours pour le semestre a été créée et est disponible au service d'inscription.
    - ▶ L'élève a ouvert une session d'inscription dans le système

# UML au travail : Une ludothèque

## Description:

L'adhérent emprunte un/deux jeu(x).

## Pré-condition:

Le client est un adhérent.

## Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 2) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

## Flot alternatif:

4a) L'adhérent n'a pas payé sa cotisation.

....



# Description détaillée d'un UC

- Description des scénarios:
  - Préconditions
  - Scénario Nominal
  - Flots alternatifs
  - Flots d'erreur
  - Postconditions
  - Exigences non fonctionnelles
- Besoins d'IHM

# Formulation



- Utilisez la voix active
  - ▶ Dire: “Le Professeur attribue des notes à chaque étudiant”
  - ▶ Au lieu de : “Quand le Professeur a attribué les notes”
- Dire ce qui déclenche l'étape
  - ▶ Dire: “Le UC commence quand le Prof. **choisit** de donner une note.”
  - ▶ Au lieu de : “Le UC commence quand le Prof. **décide** de ....”
- Dire qui fait quoi (utiliser le nom d'acteur)
  - ▶ Dire: “L'étudiant choisit ... ..”
  - ▶ Au lieu de: "L'utilisateur choisit ....."
  - ▶ Dire: “Le Système valide ...”
  - ▶ Au lieu de: "Le choix est validé ...”

<Acteur>  
 <fait>

<Système>  
 <fait>

# Description détaillée d'un UC

- Description des scénarios:
  - Préconditions
  - Scénario Nominal
  - Flots alternatifs
  - Flots d'erreur
  - Postconditions
  - Exigences non fonctionnelles
- Besoins d'IHM

# UML au travail : Une ludothèque

## Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

## Flot alternatif:

- 4a) L'adhérent n'a pas payé sa cotisation.
  - => 1. Le système invite l'adhérent à payer
  - 2 Paiement de la cotisation
  - 3. puis retour à l'étape 5
- 3a) Un jeu n'est pas disponible
  - => 1. Le système avertit l'adhérent
  - 2. retour à l'étape 2

# UML au travail : Une ludothèque

## Flot basique:

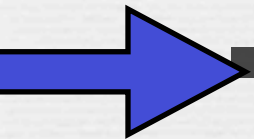
- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

## Flot d'erreur:

- 3a) la connexion au système d'emprunt échoue
- => 1. une alerte vers le système de pilotage est levée
2. Le système prévient l'adhérent et lui propose de passer plus tard
3. **le cas d'utilisation se termine en échec.**

# Description détaillée d'un UC

- Description des scénarios:
  - Préconditions
  - Scénario Nominal
  - Flots alternatifs
  - Flots d'erreur
  - Postconditions
  - Exigences non fonctionnelles
- Besoins d'IHM



# Postconditions

- Décrire l'état dans lequel doit être le système à la fin du UC
  - ▶ Utiliser lorsque l'état du système est une condition préalable à un autre UC, ou lorsque les résultats possibles du UC ne sont pas évidents pour le lecteur
  - ▶ Il ne faut jamais se référer à d'autres UC qui doivent être effectués avant cet UC
  - ▶ Devraient être énoncées clairement et devraient être facilement vérifiables
- **Facultatif:** Utilisez uniquement si nécessaire pour clarifier
- **Exemple**
  - ▶ UC «Inscription à des cours»
  - ▶ *Postconditions*
    - ▶ À la fin de ce cas d'utilisation, soit l'étudiant a été inscrit à des cours, soit l'inscription a échoué et aucune modification n'a été apportée à aux cours...

on peut commencer par les postconditions avant même les flots.

# UML au travail : Une ludothèque

## Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

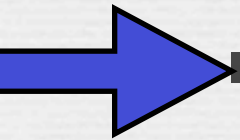
## Postcondition

L'emprunt est enregistré : l'adhérent a à présent la liste des jeux qu'il avait emprunté précédemment + les emprunts courants ; ...



# Description détaillée d'un UC

- Description des scénarios:
  - Préconditions
  - Scénario Nominal
  - Flots alternatifs
  - Flots d'erreur
  - Postconditions
  - Exigences non fonctionnelles
  - Besoins d'IHM



# Description textuelle du cas d'utilisation:

« RETIRER DE L'ARGENT »

informations optionnelles

## Exigences non fonctionnelles

| Contraintes      | Descriptif   |
|------------------|--|
| Temps de réponse | L'interface du GAB doit réagir en l'espace de 2 secondes au maximum. Une transaction nominale de retrait doit durer moins de 2 minutes |
| Concurrence      | Non applicable (mono-utilisateur)  |
| Disponibilité    | Le GAB est accessible 7j/7, 24h/24 . L'absence de papier pour les tickets ne doit pas empêcher les retraits.                           |
| Intégrité        | Les interface du GAB doivent être très robustes pour prévenir le vandalisme  |
| Confidentialité  | La vérification du code saisi doit être fiable à $10^{-6}$   |

# UML au travail : Une ludothèque

## Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

Temps de réponse : L'interface du système doit réagir en l'espace de 5 secondes au maximum.

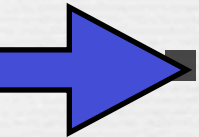
Robustesse : L'enregistrement de l'emprunt est soumis à une transaction.

Sécurité : Authentification,

...

# Description détaillée d'un UC

- Description des scénarios:
  - Préconditions
  - Scénario Nominal
  - Flots alternatifs
  - Flots d'erreur
  - Postconditions
  - Exigences non fonctionnelles
- Besoins d'IHM



# Description textuelle du cas d'utilisation:

« RETIRER DE L'ARGENT »

**informations optionnelles**

## Besoins d'IHM

Les dispositifs d'entrée/sortie à la disposition du porteur de carte doivent être :

- Un lecteur de carte bancaire.
- Un clavier numérique (pour saisir son code), avec des touches «validation », « correction » et « annulation ».
- Un écran pour l'affichage des messages du GAB.
- Des touches autour de l'écran pour sélectionner un montant de retrait parmi ceux qui sont proposés.
- Un distributeur de billets.
- Un distributeur de tickets.

# Faire le point sur les UC

- Les interactions entre le système et les acteurs sont claires
- Les séquences de communication sont conformes aux attentes de l'utilisateur
- Quand/comment les UC commencent/se terminent est clair
- Le flot de base de base donne un résultat observable pour un ou plusieurs acteurs

# Quel niveau de détail?

✓ Saisir toutes les exigences pour tous les demandeurs

X Pas de détail des interfaces utilisateurs

✓ Informations et événements

X Pas de détail des processus internes non liés à une exigence

X Pas les formats de données, ni les contrôles

How much detail in a use case?

Enough to satisfy all stakeholders that their interests (requirements) will be satisfied in the delivered system.

# Ecrire des UC : Défis

1. Comment garder les UC précis et concis?
2. Comment traiter les interfaces utilisateur?
3. Comment traiter un flux quand
  - a. Un acteur doit choisir parmi différentes options?
  - b. Un acteur peut répéter des actions avant de passer à la suite?
  - c. Les étapes ne sont pas nécessairement séquentielles?
4. Comment gérer le comportement conditionnelle dans les cas d'utilisation?



# 1. Comment garder les UC précis et concis? (1)

## ✓ Capturer le vocabulaire commun dans un glossaire

- ➔ Définir les termes utilisés dans le projet dans le glossaire, pas dans les flux
- ➔ Aide à éviter les malentendus

**Use Case**

**5. Enter Customer Information**

The system prompts the Customer to enter their *Customer Details*.

The Customer enters the *Customer Details*.

The Customer creates the account.

**Glossary**

**Customer Details** Information that uniquely identifies and provides contact information for a customer located in the U.S.A. The information consists of Name, two address lines, city, state, ZIP code, and daytime phone number.

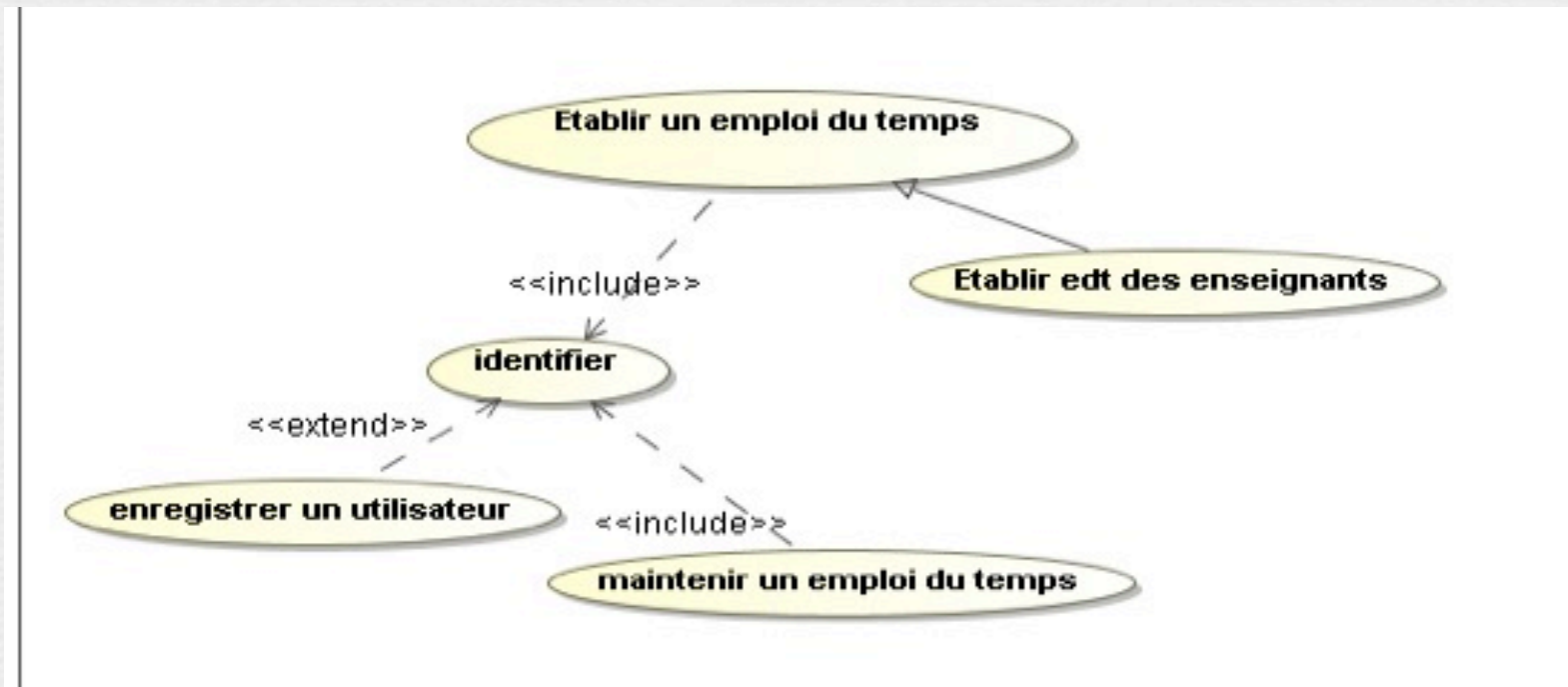
Implementation

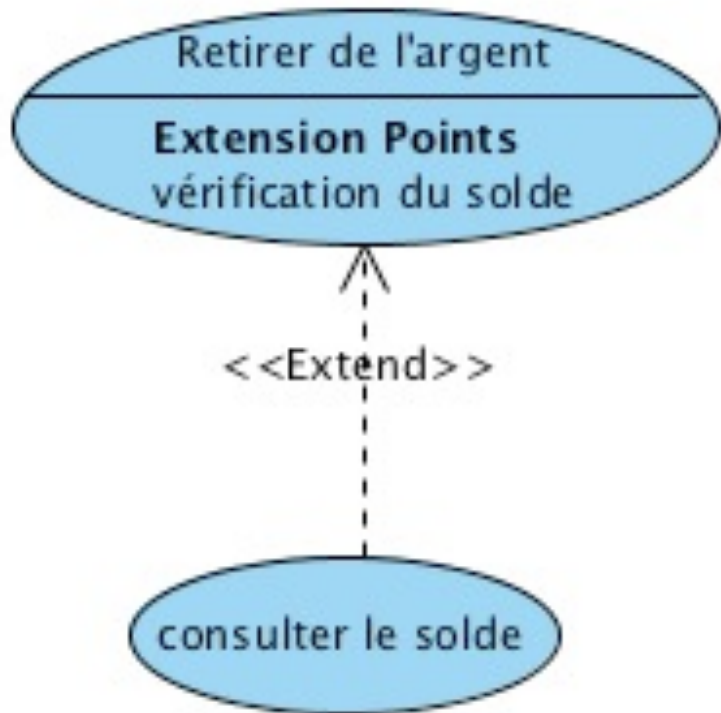
# Relations entre UC :

## Include, Extend, Specialize

Au fur et à mesure que les cas d'utilisation sont documentés, des relations peuvent apparaître

- Une relation **include** : utilisation systématique
- Une relation **extend** dénote un comportement optionnel





# Extends

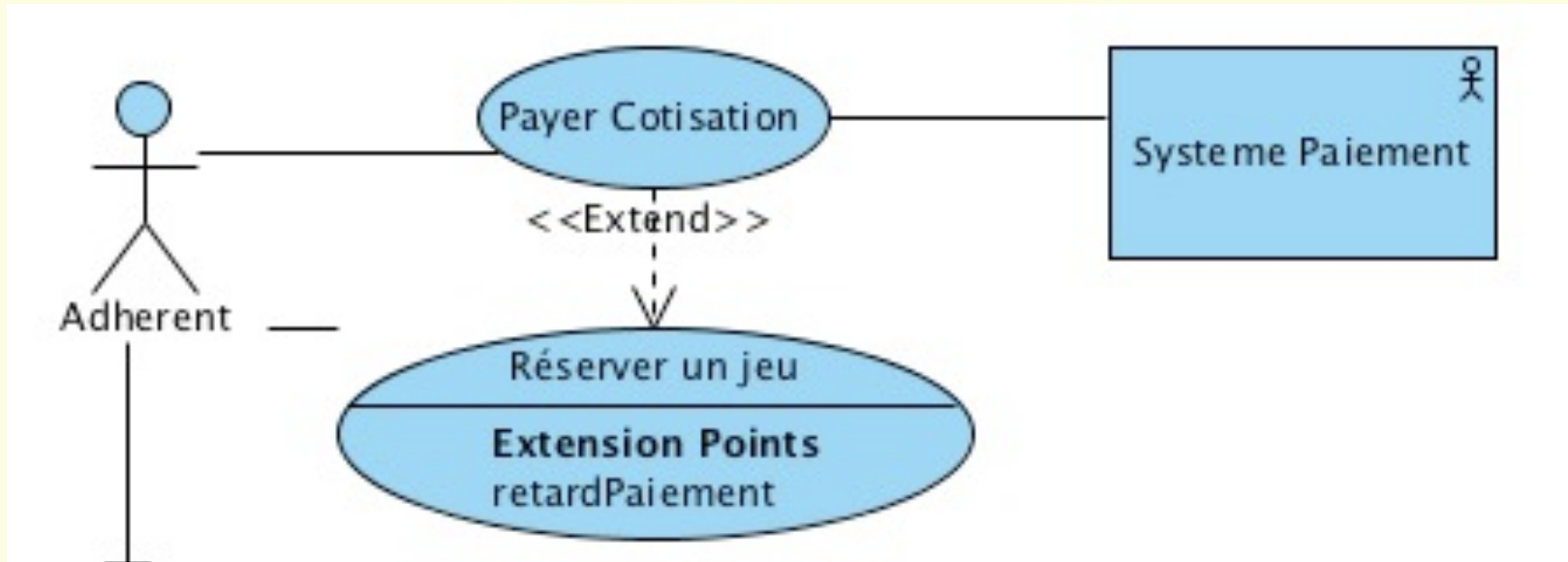
Les deux UC sont indépendants.

Permettre au client de consulter son solde avant de saisir le montant : extension optionnelle.

## *Flot enrichi*

8. Le GAB demande au client de saisir le montant  
*Point d'extension : Vérification du solde*
9. Le client saisit le montant désiré de retrait

# Ludothèque



## La relation <<include>>

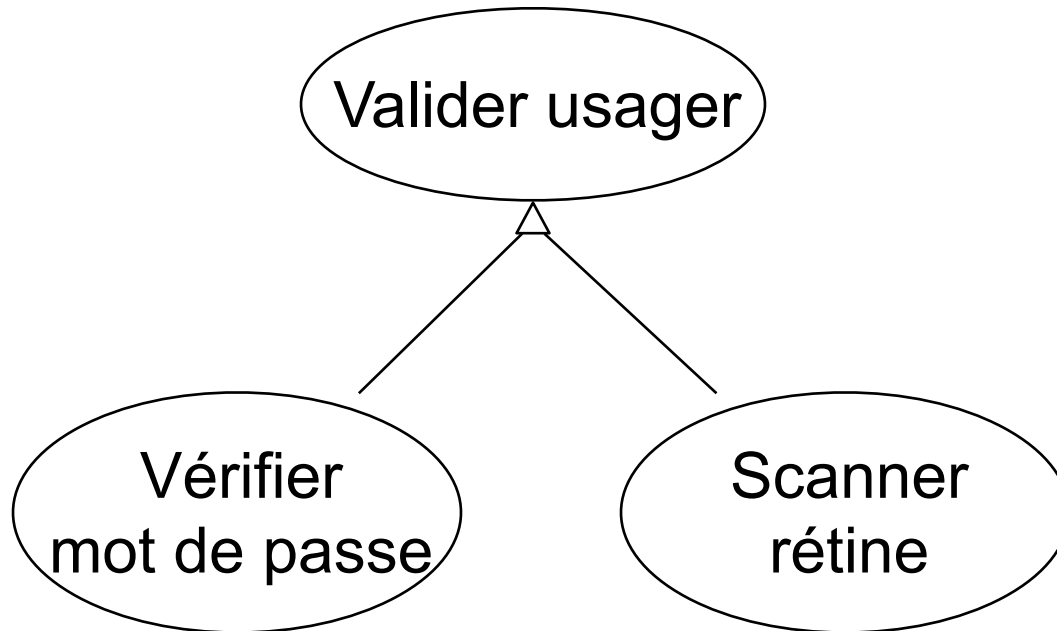
La relation *d'include* a pour seul objectif de **factoriser une partie de la description** d'un cas d'utilisation qui est commune à d'autres cas d'utilisation.

Le cas d'utilisation inclus dans les autres cas d'utilisation **n'est pas à proprement parler un vrai cas d'utilisation** car il n'a pas d'acteur déclencheur ou receveur d'évènement.

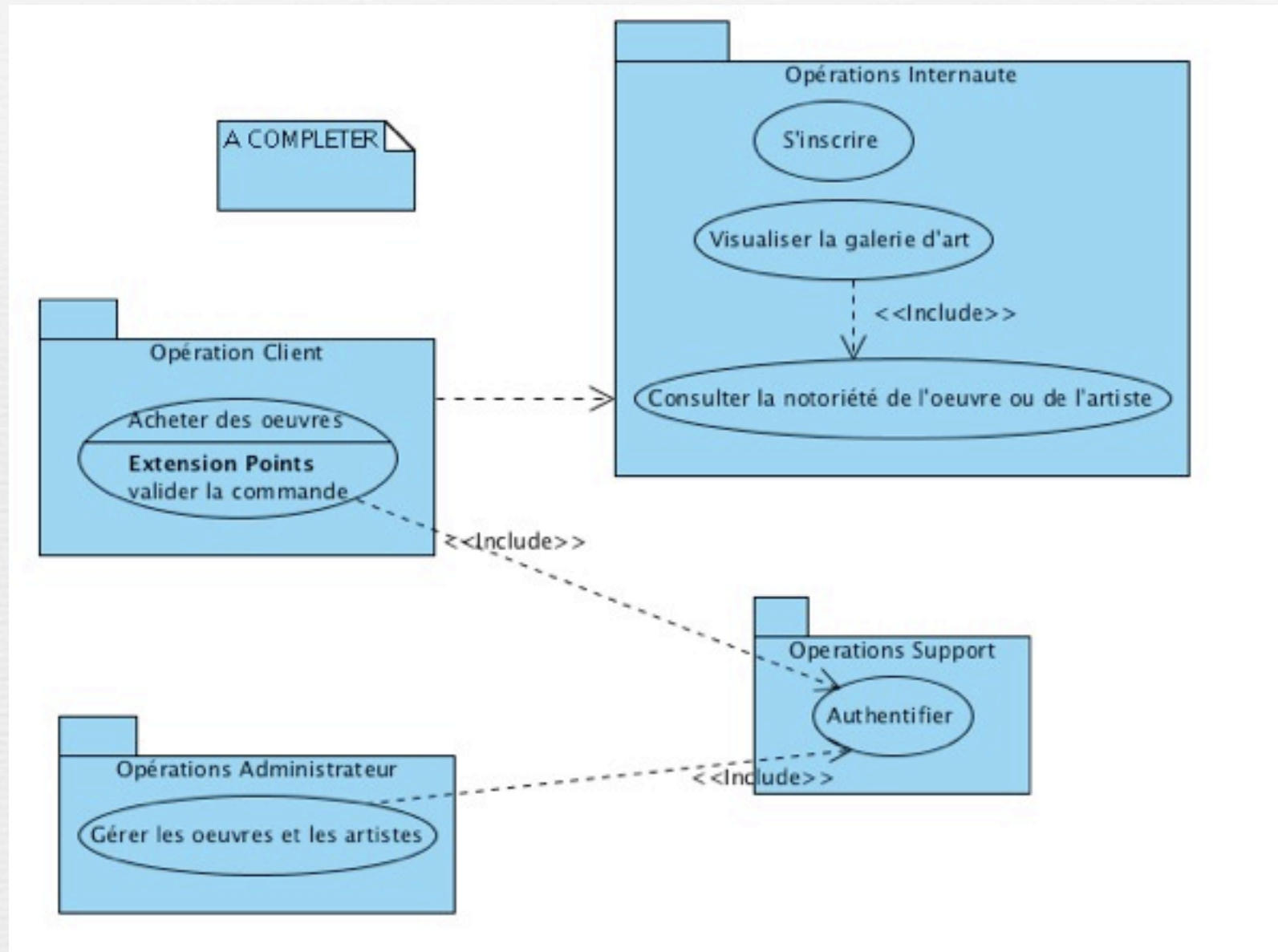
Il est juste un artifice pour faire de la réutilisation d'une portion de texte.

# Généralisation

- L'association de généralisation entre cas d'utilisation a la même sémantique que pour les classes



# Structuration des UC en packages



## 2. Comment traiter les interfaces utilisateur?

### ✓ Laissez l'interface utilisateur hors des UC

- ➔ Les UC sont indépendants de l'interface utilisateur
- ➔ Décrire les interfaces utilisateur avec des modèles dédiés ou des prototypes

#### Words to Avoid

|        |        |           |
|--------|--------|-----------|
| Click  | Drag   | Form      |
| Open   | Close  | Drop      |
| Button | Field  | Drop-down |
| Pop-up | Scroll | Browse    |
| Record | Window |           |

#### Words to Use

|           |          |
|-----------|----------|
| Prompts   | Chooses  |
| Initiates |          |
| Specifies |          |
| Submits   | Selects  |
| Starts    | Displays |
| Informs   |          |

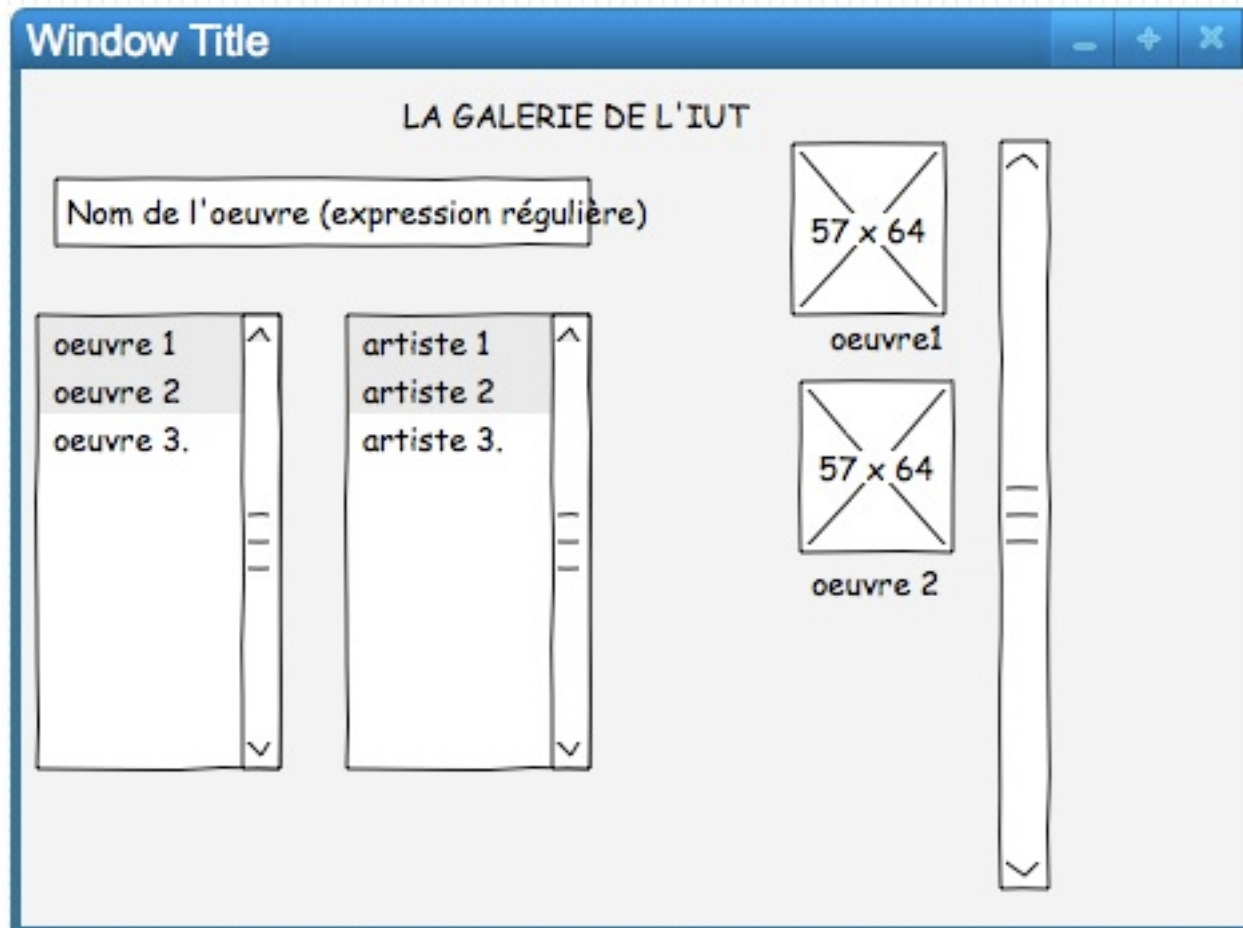
“Le système fait ~~sonner~~ le téléphone.”  
notifie





# Si c'est nécessaire...Après...




- avec l'addon Firefox :
- <http://pencil.evolus.vn/en-US/Downloads/Application.aspx>



# Résumé

# Cas d'utilisation : résumé

- Se servir des Cas d'Utilisation UML pour identifier les exigences fonctionnelles.

| Construct              | Description  | Syntax  |
|------------------------|--|---|
| <b>use case</b>        | A sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system. |    |
| <b>actor</b>           | A coherent set of roles that users of use cases play when interacting with these use cases.                                    |   |
| <b>system boundary</b> | Represents the boundary between the physical system and the actors who interact with the physical system.                      |  |

# Cas d'utilisation : résumé

| Construct             | Description   | Syntax                |
|-----------------------|---|-----------------------|
| <b>association</b>    | The participation of an actor in a use case. i.e., instance of an actor and instances of a use case communicate with each other.  | —————                 |
| <b>extend</b>         | A relationship from an <i>extension</i> use case to a <i>base</i> use case, specifying how the behavior for the extension use case can be inserted into the behavior defined for the base use case. | <<extend>><br>----->  |
| <b>generalization</b> | A taxonomic relationship between a more general use case and a more specific use case.  | —————>                |
| Construct             | Description   | Syntax                |
| <b>include</b>        | An relationship from a <i>base</i> use case to an <i>inclusion</i> use case, specifying how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.    | <<include>><br>-----> |